2023 3rd International Conference on Electronic Information Engineering and Computer Communication (EIECC) | 979-8-3503-5996-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/EIECC60864.2023.10456702

Network traffic prediction study based on the adaptive attention mechanism

Zhifang Zhang
School of Computer and Information, Fujian Agriculture and Forestry University, Fuzhou, Fujian, China
zfzhang@fafu.edu.cn

Zhaogang Shu*
School of Computer and Information, Fujian Agriculture and Forestry University, Fuzhou, Fujian, China
*Corresponding author: zgshu@fafu.edu.cn

Sheng Yang
School of Mathematics and Computer Science, Wuyi University Forestry University, Wuyishang, Fujian, China
270721675@qq.com

Shuwu Chen
School of Computer and Information, Fujian Agriculture and Forestry University, Fuzhou, Fujian, China
chenshuwu@fafu.edu.cn

Tingjiang Guo
School of Computer and Information, Fujian Agriculture and Forestry University, Fuzhou, Fujian, China
1425821304@qq.com

***Abstract*—Computer network traffic refers to the total amount of data passing through the network in a certain period of time, and is an important parameter to measure the load and running status of the network. In the background of cloud service and network slicing, predicting network traffic demand is beneficial for network operators to adjust network resources and meet user demand. However, the actual network traffic demand is characterized by real-time and sudden, the traditional network traffic prediction model has the problems of less prediction time points and low prediction accuracy, and the prediction model based on neural network has the problems such as gradient and error accumulation. Therefore, improving the network prediction model and avoiding the above problems are the focus of the current research. In this study, based on the time series theory of network traffic data, we propose a network traffic prediction algorithm using a Self-Adaptive Attention Mechanism (AAM). This algorithm dynamically adapts to process input data in an adaptive manner, reducing computational complexity while minimizing information loss. The Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics are employed to evaluate the prediction model's performance. Experimental results indicate that compared to classical models like the Auto-regressive Integrated Moving Average Model (ARIMA) and Long Short-Term Memory networks (LSTM), AAM exhibits higher accuracy and better predictive performance on datasets with significant fluctuations. For instance, when predicting 6 time points, AAM reduces RMSE by approximately 16.50% and 61.40% compared to ARIMA on two datasets, respectively. When predicting 36 time points, the reduction is about 13.12% and 70.51%, and for the maximum of 144 time points, reductions are approximately 51.77% and 73.40%.**

Keywords-Network traffic prediction; Network slice; Time series; Self-Adaptive attention mechanism

I. INTRODUCTION

With the development of 5G, edge computing, software-defined network (SDN) [1], network function virtualization (NVF) [2][3] and [4] technologies, SDN and NFV can use virtualization technology to separate specific network functions from dedicated devices to general hardware devices. At the same time, the future network to achieve refined, automatic, intelligent operation and peacekeeping management will become a new challenge [5]. Network traffic prediction can cope with these challenges, and accurately perceive application-level network traffic such as edge network, wireless network, and provide fine-grained traffic measurement.

Network traffic refers to the total traffic of the network link per unit time. In the process of network traffic collection, a fixed time interval is usually selected to obtain a definite time series data [6].

When it comes to network traffic prediction, it is challenging to address all issues with a single unified model. Traditional models struggle to capture the characteristic differences in network traffic, failing to reflect the complex variations of non-stationary network traffic, which impacts the design, training, and performance improvement of prediction models. Currently, some deep learning prediction methods have made significant progress in handling prediction problems. However, they also have substantial limitations. Discovering time dependencies from long-term time series may not be reliable, as these dependencies might be masked by temporal patterns. Additionally, determining parameters for some neural network models can be difficult, leading to issues such as gradient vanishing and exploding.

To address the aforementioned challenges, this paper proposes a network traffic prediction model based on time series theory, incorporating an adaptive attention mechanism. The model takes time series data as input into an encoder and decoder. Leveraging the adaptive attention mechanism, it utilizes Q, K, V matrix operations to capture relationships between data, ultimately yielding the prediction results.

The organization structure of this article is as follows: the first section is the introduction, the second section introduces

Industry-University-Research Innovation Fund for Future Network Technology by Education Department of China : No. 2021FNA05003;

Natural Science Fund project by Technology Department of Fujian Province, China : No. 2020J01574

the relevant work, the third section introduces the model and algorithm, the fourth section introduces the data sets, experimental results and measurement indicators. Finally, Section 5 reaches the conclusions and the future prospects.

II. Related work

Currently, network traffic prediction models are primarily classified into two categories: linear models and nonlinear models [7]. T. Anderson introduced the Auto-Regressive Integrated Moving Average algorithm [8]. Li Y utilized discrete wavelet transform to decompose network traffic into high-frequency and low-frequency components in a time series. They then employed Prophet and Gaussian process regression to predict the two components separately, adding them to obtain the final prediction result [9]. Lazaris A, Prasanna V K, and others utilized Long Short-Term Memory and improved neural networks for prediction tasks[10] [11]. Xu F, Lin Y, and collaborators proposed a network traffic prediction method based on Simulated Annealing Arithmetic (SAA)-optimized Auto-Regressive Integrated Moving Average model combined with Back Propagation Neural Network [12]. This method integrates the linear model ARIMA, the nonlinear model BPNN, and the optimization algorithm SA to achieve accurate prediction tasks. LAN X and Li Jiacheng used BP neural networks [13], wavelet neural networks [14], and other neural networks for network traffic prediction. Compared to shallow neural networks, deep neural networks can extract high-dimensional and abstract features from training samples. TANG F, MAO B, and others proposed an intelligent traffic control algorithm (ST-DeLTA) based on convolutional neural networks, assisting deep learning for traffic prediction tasks by handling network traffic with spatiotemporal characteristics [15].Vinchoff C, Chung N, and colleagues used a nonlinear GCN-GAN model to predict burst-traffic in optical networks, employing graph convolutional generative adversarial networks for optical network traffic prediction [16]. Lohrasbinasab I introduced a statistical learning and machine learning (ML)-based approach [17], expanding on existing network traffic prediction techniques. Zhang L, Zhang H, and others proposed an end-to-end online prediction model for network traffic [18], consisting of wavelet transformation and LSTM components. With the rise of attention mechanisms proposed by Vaswani A, Shazeer N, Li M, Wang Y, and others introduced a wireless network traffic prediction deep learning method based on attention mechanisms [20]. Zeng A, Chen M, and others began exploring time series-related problems [21].

However, short-term network traffic generally experiences severe fluctuations. Most of the above researchers place it within a suitable short time scale, extracting features between data before making predictions. Linear models typically require the manual setting of various parameters based on experience to fit data and are suitable for short-term traffic prediction. Nonlinear models like LSTM struggle to overcome gradient-related issues, limiting their applicability. Recent applications of the transformer, proposed by Vaswani, in computer vision, natural language processing, and time series [19], as well as the research on informer by Zhou H and others, offer new avenues for network traffic prediction, especially in long sequence time-series forecasting (LSTF) [22].

Differing from existing research, this paper, based on the time series theory of network traffic data, utilizes traffic data that directly reflects bandwidth resource demands. The proposed network traffic prediction model employs an Adaptive Attention Mechanism (AAM). The study compares predictions made with ARIMA, LSTM, and AAM on two datasets to provide a comprehensive analysis.

Unlike existing time series prediction models, this paper makes the following main contributions: a) Introducing an Adaptive Attention Mechanism algorithm to address network traffic prediction, providing more accurate predictions at multiple time points. b) Compared to traditional linear models and nonlinear models in neural networks, the AAM model can take into account 'long-standing' traffic features in network traffic that existing methods may overlook. It achieves better results in multi-target and multi-step prediction scenarios in traffic prediction problems.

III. Methods

A. Model introduction

The Adaptive Attention Mechanism (AAM) model aims to address the challenge of long sequence prediction in network traffic. It employs an adaptive approach to process input data, analyzing relationships, effectively overcoming limitations of traditional models. In particular, the Adaptive Attention Mechanism differs significantly in methodology from classical models like ARIMA and Recurrent Neural Networks, despite being rooted in time series theory. Traditional models struggle to adequately capture the differences in network traffic characteristics, impacting the design, training, and performance improvement of prediction models. Additionally, some neural network models, such as LSTM, face challenges in determining parameters and are prone to issues like gradient vanishing and exploding. The (AAM) effectively addresses these issues.

Its structure is illustrated in Figure 1. Its key feature lies in dynamically and adaptively selecting input data feature vectors at each stage. It is characterized by dynamic adaptive selection of the input data at each stage. Below is the input data sequence. The data input part of the decoder should be filled with 0 to prevent paying attention to the future part in advance, that is, shading (Mask attention) operation, which is the form of the adaptive attention mechanism to shading. The red dashed line divides the entire model into an encoder and a decoder. The left encoder operation is the calculation of multi-head adaptive attention, the small blue trapezoid represents the distillation operation, the purpose is to reduce the computational amount. After the encoder, the data is processed by the fully connected feature graph and input to the right decoder.

The right decoder calculates the results of the left encoder and the masked data entered by the decoder. Then, through a fully connected layer, the predicted output of the green grid part is obtained. The selection of attention mechanism is a key problem. The improper choice may cause information loss, while the fixation of the down-sampling rate may lose important information on the one hand, and may contain redundant features on the other hand. Therefore, the method of adaptive dynamic selection of attention matrix is adopted to reduce the computational burden and reducing the information

loss, so as to better achieve the prediction goal. Where, adaptive attention specific operation first uses KL divergence to measure similarity between and the matrix, ranked according to scores. Then it adaptively selects a subset of the data to reduce the network size. Finally, the generative decoder only needs a fully connected layer in the inference stage to avoid error accumulation in the inference process.

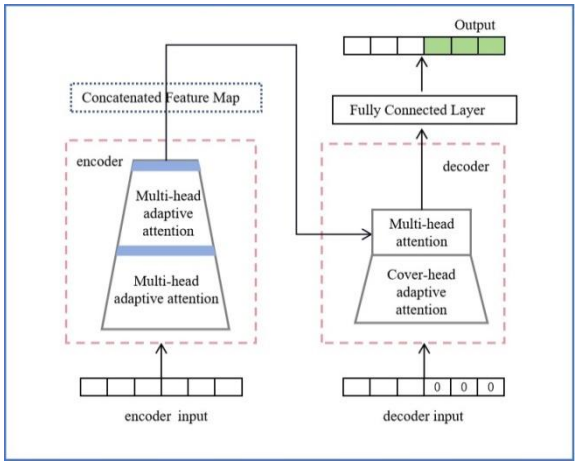

Figure 1. Schematic representation of the self-adaptive attention mechanism

The model of the adaptive attention mechanism shown in Figure 1 consists of the left encoder and the right decoder. Among them, the encoder consists of multiple coding layers, which is a robust remote dependence to extract the input data of long sequences.

The core of the encoder part is the computational adaptive attention. First, the data goes through the initial position encoding and time encoding, and the output obtained by the encoder operation is used as the input to the partial attention matrix Q of the decoder. The decoder is composed of multiple decoding layers. The decoder data input adopts the mode of "known + unknown", and a part of the known time series is used to form the input part. The time series data to be predicted is filled with “0” coverage to prevent the current position from paying attention to future information and avoid auto-regressive. Finally, the prediction results are obtained by fully connected convolution. The following are the specific details of each part of the model:

Encoder part: The encoder on the left side of the model consists of N (N=6, 4, 2). Each layer has two sub-layers. One is the multi-head adaptive attention mechanism, and the other is a fully connected feed-forward network. The two sub-layers are connected with residual differences for data normalization. To facilitate the calculation, all the sub-layers and the embedded layers in the model will produce the output with 512 dimensions.

Decoder part: Similar to the encoder, residual connections are used in each sub-layer and normalized later. The input uses the output of the encoder and the masked data sequence to prevent the current position from paying attention to the subsequent position.

Input data location coding: Input data embedding consists of three independent parts: a scalar projection, local timestamp (location) and global timestamp embedding (minutes, hours, days, weeks, etc.). The model does not contain recursion, in order for the model to utilize the order information of the sequence, injecting information about the relative or absolute position of the sequence as the sequence marker. Add "positional coding" to the input data sequence embedding at the bottom of the encoder and decoder stack. Location coding has the same dimension d , and sums between the different frequency sine and cosine functions, as shown in Equations (2) and (3).

A one-dimensional convolutional filter was used to project the scalar context f_i^t , represented by an input vector, as shown in the formula (1), $i \in \{1, 2, 3, \ldots, L_f\}$, α is the factor of the magnitude between the balanced scalar projection and the local and global embedding. The sequence input is normalized and set to 1, and the global timestamp is a learnable embedding vector, as shown in equation (4), and the embedding schematic is shown in Figure 2.

$$\mathrm{SE}_{(\mathrm{L}_f\times(\mathrm{t}-1)+\mathrm{i}} \tag{1}$$

$$\mathrm{PE}_{(\mathrm{pos},2\mathrm{i}+1)} = \sin\left(\mathrm{pos}/10000^{2\mathrm{i}/\mathrm{model}}\right) \tag{2}$$

$$\mathrm{PE}_{(\mathrm{pos},2\mathrm{i}+1)} = \cos\left(\mathrm{pos}/10000^{2\mathrm{i}/\mathrm{model}}\right) \tag{3}$$

$$\mathrm{F}_{feed[i]}^t = \alpha u_i^t + \mathrm{PE}_{(\mathrm{L}_f\times(\mathrm{t}-1)+\mathrm{i}} + \sum_p (\mathrm{SE}_{(\mathrm{L}_f\times(\mathrm{t}-1)+\mathrm{i}})_p \tag{4}$$

Adaptive attention mechanism: The attention function [19] in the original self-attention mechanism consists of the query Q matrix, the key K matrix and the value V matrix finally mapped to the output, in which the query, key value and output are expressed by vectors. The output is counted as a weighted sum of the values, where the weights assigned to each value are calculated by the compatibility function of the query and the corresponding keys. Adaptive attention model attention is "dot product attention". The input consists of the dimension as the query vector q , the key vector k , and the value vector v . The query point product of all key values is calculated, dividing each result by $\sqrt{\mathrm{d_k}}$, and the function $soft\ \max$ is applied to obtain value weights. Zhou H et al. [22] proposed the self-attention mechanism as formula (5).

$$\mathrm{Attrention}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\mathrm{V} \tag{5}$$

$Q \quad R^{L_q\times d}$, $K \quad R^{L_k\ d}$, $V \quad R^{L_v\ d}$, L represents the length of the time-series data.

The adaptive attention mechanism mentioned in this paper, different from Vaswani and Zhou H et al. [19][20], makes the number of feature vectors of input data in each stage of the network into dynamically adaptive, without discarding important information to process redundant information without wasting computing resources. In the adaptive attention module, the score is calculated based on the KL divergence metric q and k similarity, mainly following the formula (6). As shown in Figure 3, multiple q vectors constitute the weight matrix W^q , and many k components form the weight matrix W^k .After calculation $q^1 \cdot k^1, q^1 \cdot k^2, \ldots,$ the scores

are sorted, and a subset of data is automatically selected according to the sorted scores. The specific selection process is shown in Algorithm 1.

$$\mathrm{M}(q_i,\mathrm{K}) = \ln \sum_{j=1}^{L_K} e^{\frac{q_i K_j^T}{\sqrt{d}}} - \frac{1}{L_K}\sum_{j=1}^{L_K} e^{\frac{q_i K_j^T}{\sqrt{d}}} \quad (6)$$

The i th q is defined as probability for all concerns k , is the attention probability distribution. $\frac{1}{L_K} = q(k_j \mid q_i)$ is a uniform distribution, L_k is the query vector length. The first term in the formula is the operation q_i in the matrix K , and the second term is the average of them.

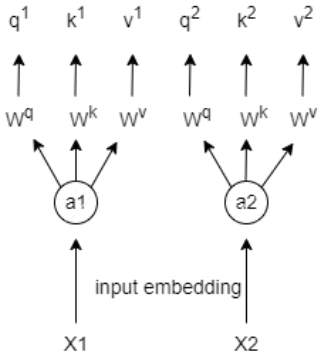

Figure 2. Schematic representation of the input data embedding

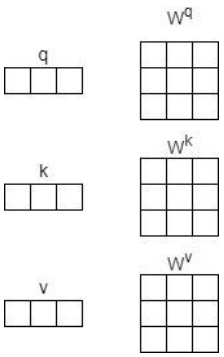

Figure 3. A Schematic representation of the weight matrix

The adaptive attention mechanism selection process is shown in Table I and can be expressed as Equation (7).

$$\mathrm{Attrention}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{softmax}(\frac{Q^{\gamma}K^T}{\sqrt{d_k}})\mathrm{V} \quad (7)$$

Where Q^{γ} are the adaptively sampled and a matrix of the same size q .

Algorithm 1 introduces the selection process of the adaptive attention mechanism, and sets the feed-forward network layer (the size of the inner layer is 2048) and the $dropout$ layer. The data with 10% of retention is the validation set. Thus, all experiments were performed under eight time-order random training and validation, resulting in a mean of eight runs. All data were normalized so that the variable mean was 0 and the standard deviation was 1.

Algorithm 1. Adaptive attention selection process
Preconditions and inputs: $Q \quad R^{m\ d}, K \quad R^{n\ d}, V \quad R^{n\ d}$, data set
1 Set the experimental hyper-parameters, along with a constant sampling step size: $length$
2 randomly select $\overset{\gamma}{K}$ from K
3 Set standard points: $\overset{\gamma}{S} = Q(\overset{\gamma}{K})^T$
4 calculate the q measured values $M = \max(\overset{\gamma}{S}) - mean(\overset{\gamma}{S})$, ranking the measurements M from small to large
5 Q^{γ} matrix was selected from Q , ranked according to the q score of the fourth step , with $length$ as the length
6 calculate Adaptive attention: shown in Equation (7)
7 outputs: Adaptive mechanism of attention

IV. Results and Discussion

A. Experimental dataset

Data set 1 was obtained from the WIDE project, starting from 2020 / 8 / 1 0:00 to 2022 / 8 / 31 23:50. Each data was measured in 10 minutes, totaling 109585 data records. The data format includes the amount of data from 6 IP ports and a total flow data, while each record includes the date, start time and end time, ID number and other labels. Data set 2 is from ISP Internet data, which is the UK Academic Network (https://github.com/rankinjl/internet) backbone traffic, starting from 2005 / 6 / 7 07:00 to 2005 / 7 / 28 13:55. Data is collected every five minutes, and the data is kept in two decimal places after processing. The dataset tracks each IP port and its total traffic consumption for a month. The data format of part of the data set is shown in Table II.

TABLE I. Data set 1 Basic format of traffic data

Data	IP1	IP2	IP3	IP4	IP5	IP6 Total(GB)
2020/8/1 0:00	6.12	6.39	2.31	3.87	1.39	2.76 22.85
2020/8/1 0:10	4.61	1.91	2.92	3.14	1.74	2.57 16.90
2020/8/1 0:20	3.72	1.68	1.96	6.40	2.13	1.28 17.16
2020/8/1 0:30	5.89	1.49	5.51	1.52	1.87	3.64 19.92
2020/8/1 0:40	4.50	2.28	4.90	3.56	1.95	3.81 21.01
2020/8/1 0:50	4.64	4.21	1.93	13.30	2.31	2.76 29.13

B. Data handling

In the basic time series model, the network traffic prediction problem essentially is the time series problem. Bandwidth resource size affects the amount of bytes transmitted by a network slice over time. Therefore, network traffic prediction is defined as a multi-step and multi-target byte prediction problem.

The network traffic prediction results reflect the bandwidth resource requirements to some extent. The traffic time series data of the network data over a period of time can be defined as formula (8).

$$x_i^t = \{x_i^0, x_i^1, ..., x_i^t\} \quad (8)$$

Where X_i^t represents the amount of bytes through which the slice i passes in the time slot $[t, t+\tau]$ and, in the prediction task, τ is the time interval. According to the time series data definition, the prediction task is to predict individual network traffic data based on its history of time series data, which n represents the first n step.

C. Evaluating indicator

To evaluate the model performance, the mean absolute error (MAE) and the root mean square error (RMSE) indicators were used. The smaller values of MAE and RMSE indicate a better prediction. These two indicators are defined as shown in Equations (9) and (10).

$$\mathrm{MAE} = \frac{1}{n}\sum_{n=1}^{n} |y_i' - y_i| \quad (9)$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{n=1}^{n} (|y_i' - y_i|)^2} \quad (10)$$

y_i is the true value, y_i' is the predicted value.

D. Experimental result

The AAM method is compared with the traditional time series prediction method ARIMA and the deep learning LSTM algorithm. Data set 1 use length starts from 0:00 on August 1, 2020 to 0:0:00 on October 10, 2020, with 70% for training data, 10% for validation data, and the rest as test set. The purpose is to predict the results for 6, 36, 72, 144 time points, with the test time point starting at 02:30 on 26 September 2020. The length used in data set 2 started from 7:00 on June 7, 2005 to 13:45 on July 28, 2005, and the test time point started from 14:20 on July 20, 2005.

The network traffic prediction results reflect the bandwidth resource requirements to some extent. The traffic time series data of the network data over a period of time can be defined as formula (8). TABLE III and IV show 6, 36, 72, 144 points, for ARIMM, LSTM and AAM models, respectively. It is difficult to predict user traffic in a short period of time. Compared with the ARIMA model and the LSTM model, the results at 10 min and 5min timescales show that the adaptive attention mechanism model MAPE and RMSE are lower. At the same time, when the data set predicts 144 points, MAE and RMSE are actually lower and more accurate than predicting 72 time points. At the 72 points predicted on dataset 2, MAE and RMSE are instead lower and more accurate than predicting the 36 time points. This shows that the proposed AAM model can take into account older data information.

TABLE II. Data set 1 Experimental results

Method / Forecast points	ARIMA MAE	ARIMA RMSE	LSTM MAE	LSTM RMSE	AAM MAE	AAM RMSE
6	3.45	4.12	3.08	3.84	2.90	3.44
36	3.88	4.65	2.92	3.14	3.42	4.04
72	5.93	7.81	5.89	7.19	4.87	6.10
144	8.27	10.13	5.92	7.57	3.78	4.89

TABLE III. Data set 2 Experimental results

Method / Forecast points	ARIMA MAE	ARIMA RMSE	LSTM MAE	LSTM RMSE	AAM MAE	AAM RMSE
6	0.54	0.57	0.37	0.39	0.20	0.22
36	0.76	0.78	0.38	0.41	0.20	0.23
72	0.78	0.81	0.39	0.42	0.18	0.21
144	0.82	0.94	0.42	0.48	0.2	0.25

As shown in Figure 4, is the resulting graph of 144 time points predicted by the AAM method on Dataset 1, and Figure 7 shows the predicted results on Dataset 2. As shown in Figure 5, the LSTM method predicts 144 time points on Dataset 1, and Figure 8 shows the LSTM method predicts the results on Dataset 2. As shown in Figure 6, the ARIMA method predicts the 144 time point results on Dataset 1, and Figure 9 predicts the results on Dataset 2. By comparing the results of the two models on the two data sets, it can be seen that the AAM model has a good prediction effect, the LSTM model has reduced performance due to error back-propagation, and the ARIMA model has the worst effect, which is not suitable for the long sequence prediction task.

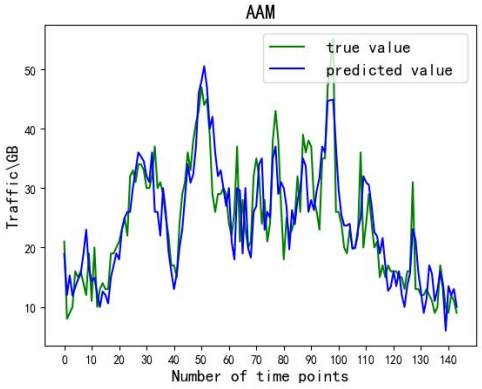

Figure 4. AAM in the data set 1 prediction outcome

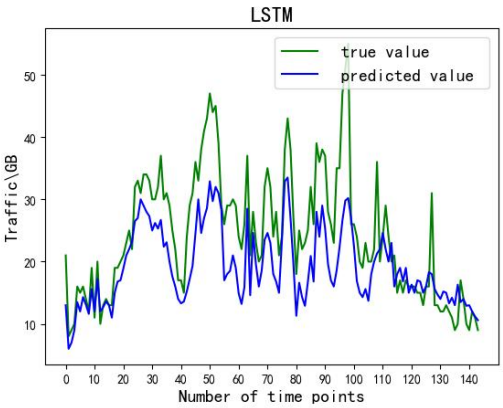

Figure 5. LSTM in the data set 1 prediction outcome

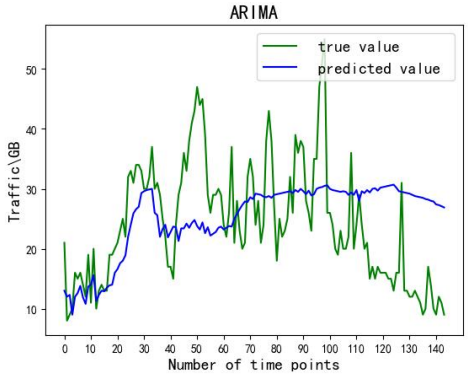

Figure 6. ARIMA in the data set 1 prediction outcome

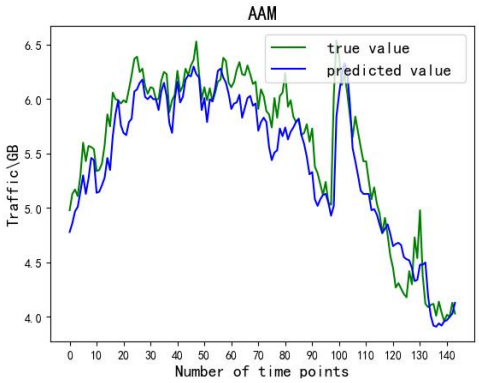

Figure 7. AAM in the data set 2 prediction outcome

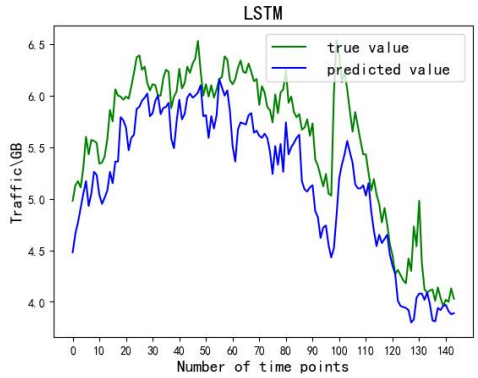

Figure 8. LSTM in the data set 2 prediction outcome

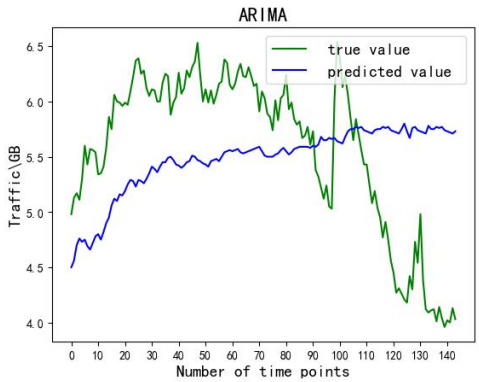

Figure 9. ARIMA in the data set 2 prediction outcome

V. CONCLUSIONS

In this paper, we analyze network traffic data based on time series theory and compare the performance of existing network traffic prediction models. In addition, a new flow prediction model of the adaptive attention mechanism is proposed. Different from the previous deep learning models such as LSTM, the adaptive attention mechanism replaces the loop layer commonly used in the codec structure, which improves the prediction accuracy. In the future, model prediction performance at different time scales and at different time resolutions will be investigated, while comparing the universality of models in different data contexts.

REFERENCES

[1] MCKEOWN N, ANDERSON T, BALAKRISHNA H, et al. Open Flow: enabling innovation in campus networks [J].ACM SIGCOMM computer communication review, 2008, 38(2):69-74.

[2] Han B, GOPALAKRISHNAN V, JI L, et al. Network function virtualization: Challenges and opportunities for innovations [J].IEEE communications magazine, 2015, 53(2): 90-97.

[3] SUN J, ZHANG Y, LIU F, et al. A survey on the placement of virtual network functions [J]. Journal of Network and Computer Applications, 2022: 103361.

[4] ALLIANCE N. Description of network slicing concept [J]. NGMN 5G P, 2016, 1 (1).

[5] KAZMI S M A, KHAN L U, TRAN N H, et al. Network slicing for 5G and beyond networks [M].Berlin: Springer, 2019.

[6] G. ZHANG, D.B.T. HUANG, International Conference on Intelligent Networking and Collaborative Systems, Short-term network traffic prediction with ACD and particle filter, 2013, pp. 189–191.

[7] KANG MENGXUAN, SONG JUNPING, FAN PENGFEI, GAO BOWEN, ZHOU XU, LI ZHUO. Review of research on deep learning-based network traffic prediction [J]. Computer Engineering and Application, 2021, Vol. 57 (10):1-9

[8] T. ANDERSON , The Statistical Analysis of Time Series. HOBOKEN, NJ, USA: Wiley, 1971

[9] LI Y , MA Z, PAN Z, et al. Prophet model and Gaussian process regression based user traffic prediction in wireless networks [J]. Science China Information Sciences, 2020, 63(4): 1-8.

[10] LAZARIS A, PRASANNA V K. An LSTM framework for modeling network traffic[C]//2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). IEEE, 2019:19-24.

[11] LAZARIS A, PRASANNA V K. Deep Flow: a deep learning framework for software-defined measurement [C]//Proceedings of the 2nd Workshop on Cloud-Assisted Networking.2017:43-48.

[12] XU F, LIN Y, HUANG J, et al. Big data driven mobile traffic understanding and forecasting: A time series approach [J]. IEEE transactions on services computing, 2016, 9(5): 796-805.

[13] LAN X. Analysis and research of several network traffic prediction models[C]/2013 Chinese Automation Congress, Changsha, 2013.

[14] Li JIACHENG. Research on campus network traffic prediction based on wavelet neural network [D] . Nanchang: Nanchang University. 2019

[15] TANG F, MAO B, FADLULLAH Z M, et al. ST- DeLTA: an novel spatial-temporal value network aided deep learning based intelligent network traffic control system [J].IEEE Transactions on Sustainable Computing, 2020 ,5 (4):568-580.

[16] VINCHOFFV C, CHUNG N, Gordon T, et al. Traffic prediction in optical networks using graph convolutional generative adversarial networks[C]//2020 22nd International Conference on Transparent Optical Networks (ICTON). IEEE, 2020: 1-4.

[17] LOHRASBINASAB I, SHAHRAKI A, TAHERKORDI A, et al. From statistical-to machine learning-based network traffic prediction[J]. Transactions on Emerging Telecommunications Technologies, 2022, 33(4): e4394.

[18] ZHANG L, ZHANGH, TANG Q, et al. LNTP: An end-to-end online prediction model for network traffic [J]. IEEE Network, 2020, 35(1): 226-233.

[19] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [J]. Advances in neural information processing systems, 2017, 30.

[20] LI M, WANG Y, WANG Z, et al. A deep learning method based on an attention mechanism for wireless network traffic prediction [J]. Ad Hoc Networks, 2020, 107: 102258.

[21] ZENG A, CHEN M, ZHANG L, et al. Are transformers effective for time series forecasting? [J]. arXiv preprint arXiv: 2205.13504, 2022.

[22] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(12): 11106-11115.