Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 8424398, 6 pages
https://doi.org/10.1155/2022/8424398

Research Article

Network Traffic Prediction: Apply the Transformer to Time Series Forecasting

Qian Kong,[1] **Xu Zhang**,[1] **Chongfu Zhang**,[1,2] **Limengnan Zhou**,[1,2] **Miao Yu**,[1] **Yutong He**,[1] **Youxian Chen**,[1] **Yu Miao**,[1] **and Haijun Yuan**[1]

[1] *School of Electronic and Information Engineering, University of Electronic Science and Technology of China, Zhongshan Institute, Zhongshan 528402, China*
[2] *School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Qian Kong; kongqian01@yeah.net

Received 16 June 2022; Revised 21 September 2022; Accepted 3 October 2022; Published 30 October 2022

Academic Editor: Rosa M. Benito

Along with the development of technology and social progress, the Internet is increasingly widely used in life. Mobile communication, fiber optic broadband, and other essential Internet networks have gradually become indispensable in everyday life. The task of further improving and optimizing the quality of Internet network links and improving the efficiency of Internet networks has been on the agenda. This paper proposed a deep learning-based network traffic prediction model, which can capture the characteristics of network traffic information changes by inputting past network traffic data to achieve the effect of future network traffic prediction. The model structure is flexible and variable, which improves the problems of other methods that cannot capture long time series prediction features and cannot parallelize the output. It also has apparent advantages in time complexity and model convergence speed without the evident disadvantage of time lag. Based on this network traffic prediction model, it can help Internet service providers optimize network resource allocation, improve network performance, and allow Internet data centers to provide abnormal network warnings and improve user service level agreements.

1. Introduction

With the increasing number of Internet users, the penetration rate of mobile and fixed-line users is high. The extensive coverage and structure of the network make it easier to collect and diversify network traffic data [1]. To provide customers with better broadband network quality and enterprises with better network optimization equipment, a comprehensive network traffic forecasting model is on the agenda and aims to implement the following features:

(1) Optimization of network resource allocation: Accurate traffic forecasting models can detect long-term future traffic demand, providing guidance for early planning of resources, freeing up network resource consumption, and unlocking the potential for network traffic growth.

(2) Improved utilization of network resource: Operators and network service providers can use a predictive network traffic model to improve the mobile experience of their subscribers. By analyzing network traffic resources, the number of base stations in high-load areas can be improved. It also can reduce the energy consumption of base stations in low-load regions to improve network resource utilization.

(3) Optimization of network resource service levels: The network prediction model can provide a more advanced understanding of the network attack traffic's size and guide the server to carry out traffic refinement cleaning. Similarly, it ensures the regular operation of user services through load balancing and fault migration of network resources to improve the user QoS level.

Machine learning models have become increasingly popular in academic and industrial applications over the last decade as GPU, and edge accelerator processing speeds have increased, providing a new avenue to assist traditional industries. Network traffic prediction refers to extracting feature information from past traffic information to predict future network data traffic. Several network traffic prediction models have been proposed recently. ARIMA (autoregressive integrated moving average) [2] is the traditional time-domain forecasting method which is widely used in the financial direction. However, in predicting long time series or complex data, these models do not perform very well. The ARIMA model is simple and easy to apply but requires stable time series data and, by its nature, cannot capture nonlinear relationships, and the RNN family of models [3–5] (recurrent neural network (RNN) and their derivative models, long short term memory (LSTM) networks, and gated recurrent units (GRU), etc.) can better discover features in the time domain through its unique shared memory mechanism. Still, its time complexity is high and cannot parallelize the output. In 2017, Google proposed the transformer model [6], which does not use popular processing models such as convolutional neural networks (CNN) and RNNs but instead uses the attention mechanism entirely to achieve more accurate results in the direction of natural language processing (NLP) and computer vision (CV) combined with the original self-attention (Attention) mechanism is suitable for network traffic prediction on the ground, due to its solid fitting ability, low time complexity, and parallelized output.

The rest of the paper is organized as follows: Section 2 describes the attention mechanism and model architecture, while Section 3 focuses on some basic details and model training parameters. In the next two sections of the paper, the results of the visual network traffic prediction model are visualized, summarized, and extended in the future.

2. Model Structure

With time, transformer and BERT (bidirectional encoder representations from transformers) models [7] came into prominence in NLP. Attention mechanisms were already migrated and applied to various aspects of deep learning models. Even papers state that [8] the self-attentive tool is a generalized version of CNN. In the direction of deep learning methods, most time series prediction models use RNNs and related models. Still, due to their unique shared memory mechanism, which leads to high time complexity in the number of parameters and the inability to parallelize the output, the self-attentive mechanism differs from traditional neural networks such as DNNs and RNNs in its high fitting capability. It has the advantages of parallelized creation and low time complexity, which are ideal for it and is suitable for network traffic prediction.

2.1. Self-Attention. As shown in Figure 1, in the overall mechanism of the self-attentive mechanism (input and output), each result is related to each input, and each predicted traffic is obtained from all output traffic information before prediction.

In the model calculation, there are three independent variables and one dependent variable:

Independent variables such as Q (query), K (key), and V (value), which are obtained from the initialization of the model need to be optimized and iterated afterward. The dependent variable which is α self-attentive coefficient is obtained linearly by the independent variables as an intermediate state, which is not displayed in the figure. The internal structure of the model calculation can be shown in Figure 2:

The computational mechanism is divided into three stages:

(1) Parameter initialization:The model is initialized by the parameters with their respective parametric quantities (independent variables) Q, K, and V.

(2) Parameter calculation:

  (a) The self-attentive scores are calculated from Q corresponding to the predicted values and K of all common inputs.

  (b) Matrix multiplication of the self-attentive scores with V to obtain the initial output state corresponding to each input value.

(3) Parameter summation: All output values from the second stage are summed to obtain the corresponding.

2.2. Attention Score. The previous section mentioned that Q computes the attention fraction and also K. Generally, the point multiplication or summation method is commonly used. Point multiplication and summation are pointed multiplication and summation operations between tensors in linear algebra, and the structure of the calculation is shown in Figure 3.

The input network traffic data are initialized to produce the corresponding Q and K. When Q and K are operated; they are directly dotted in the dotted multiplication method or summed in the summation method and then passed through the Tanh function to obtain the corresponding attention fraction.

2.3. Multihead Attention. The transformer proposes the multiheaded attention mechanism [9]. Its main idea is to map the input vectors to different subspaces by increasing the number of parameters Q, K, and V of the self-attention mechanism, allowing the model to understand the input sequence from different perspectives. The comparison of the multiheaded attention mechanism and self-attention mechanism is shown in Figure 4 as machine translation.

By observing the overall model structure in Figure 5, the model input to the encoder is a time series T with multidimensional features, which flows through the numerical embedding layer and the positional embedding layer and then enters multiple multihead self-attentive mechanisms

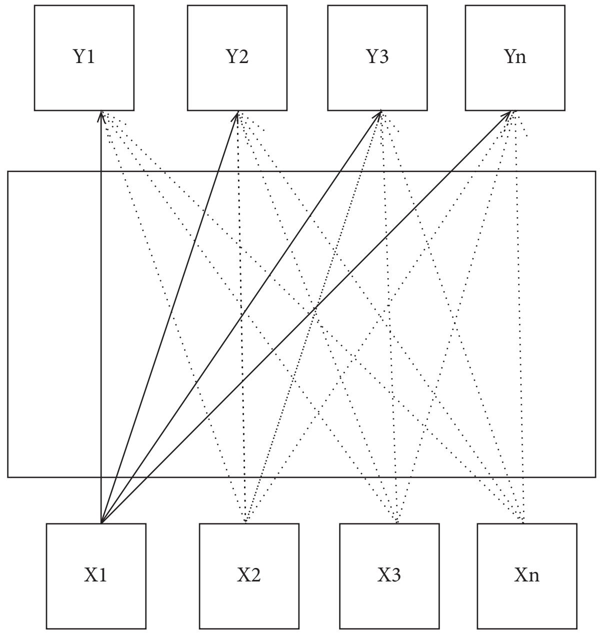

Figure 1: Structure of the self-attention mechanism.

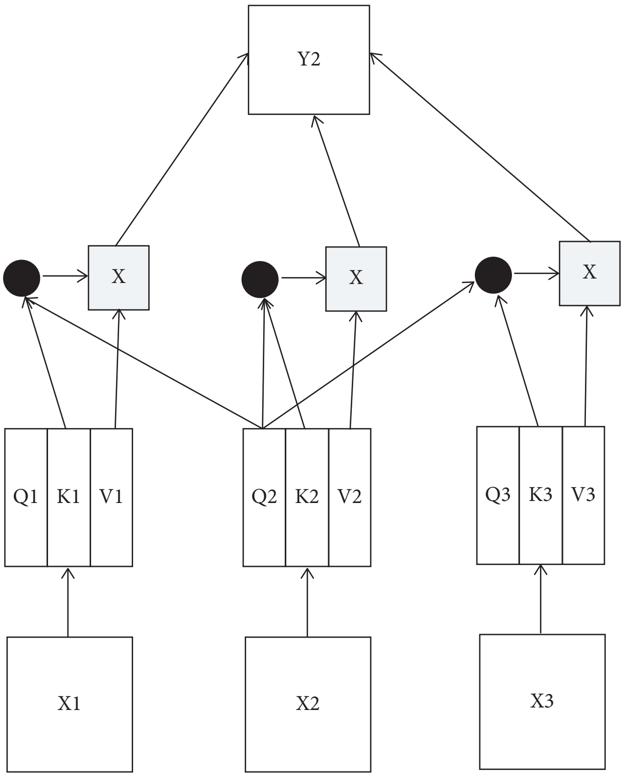

Figure 2: Self-attention mechanism computational structure.

and the numerical regularization layer and then extracts the input K and V from the last layer to the decoder.

The input of the decoder has the same structure as the encoder's input—the input length is smaller than that of the encoder. The Q of the result of the multiheaded attention layer through the mask is combined with the output of the encoder into multiple multiheaded attention layers and the data value regularization layer. Finally, the prediction result is output.

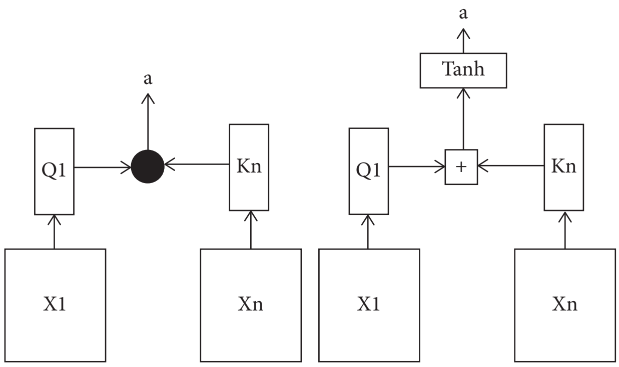

Figure 3: Attention score.

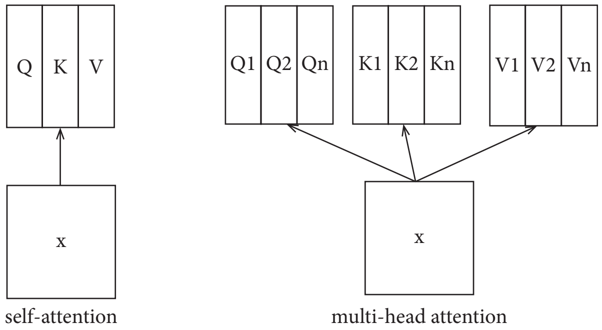

Figure 4: Multihead attention.

3. Training

3.1. Datasets. In this experiment, Vietnam's two years of 4G base station data are used as the training dataset using the random sliding window method, as shown in Figure 6. We used the base station traffic for the past 168 days to predict the data traffic for the next 32 days.

3.2. Loss Function. The model is back-propagated and optimized using mean absolute error (MAE) in the training dataset. Both mean square error (MSE) and MAE are used in the validation dataset to determine the model's merit. Their functional expressions are as follows:

$$\begin{aligned}\mathrm{MAE} &= \frac{\sum_{i=1}^{n}\left|y_i - x_i\right|}{n}\\ &= \frac{\sum_{i=1}^{n}\left|e_i\right|}{n},\end{aligned} \tag{1}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2. \tag{2}$$

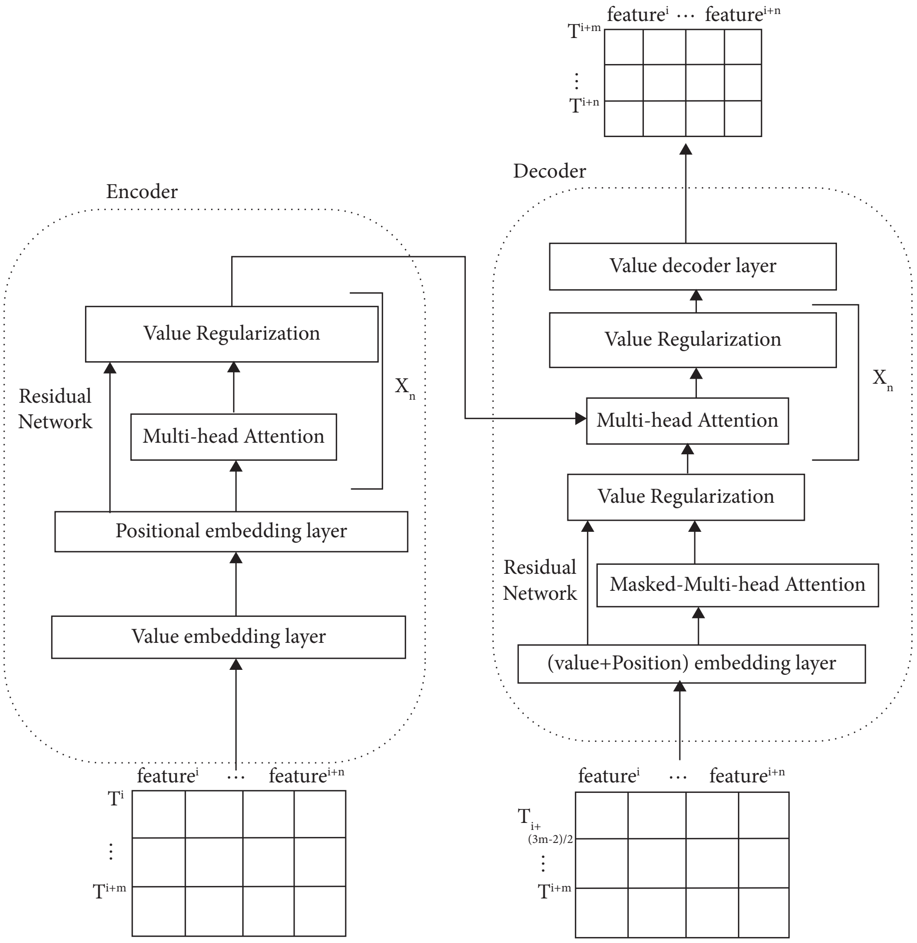

Figure 5: Model structure.

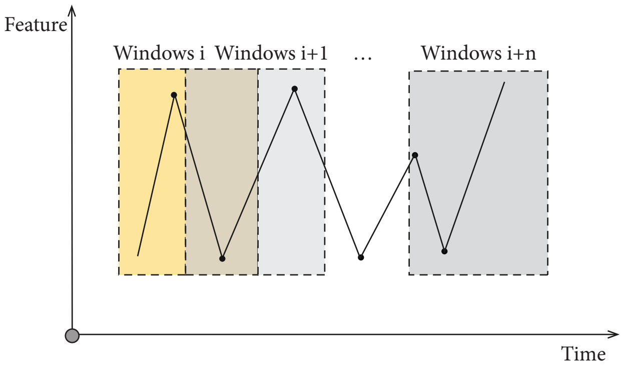

Figure 6: Slide window.

3.3. Optimizer. We used the R-Adam optimizer [10] with learning rate = 1e−3, betas = (0.9, 0.999), eps = 1e−8, weight-decay = 0, compatible with the traditional Adam [11], and SGD optimizers control the variance of the adaptive rate to achieve faster convergence and robustness.

3.4. Metrics. We use a fault-tolerant accuracy algorithm to keep the predicted data at the actual data's Tr (tolerate-rate) edge:

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\left(1 \text{ if } Tr * Y \leq \widehat{Y} \leq Tr * Y \text{ else } 0\right). \tag{3}$$

3.5. Other Details

3.5.1. Hardware. We train our model on an Nvidia GPU (RTX 5000), using an Intel CPU under Linux, Ubuntu, with all datasets on "cuda" and all raw parameters in the model initialized to a zero matrix.

3.5.2. Activation Function. Unlike the transformer model, we replace the ReLU [12] (rectified linear unit) activation function with the GELU (Gaussian error linear units) activation function for high-performance neural networks, which incorporates the idea of randomness regularization, combining nonlinearity with stochastic regularization.

4. Result

This section explains the time series prediction results of using the past 168 hours of data traffic to predict the next 32 hours of data and shows the advantages of the transformer in

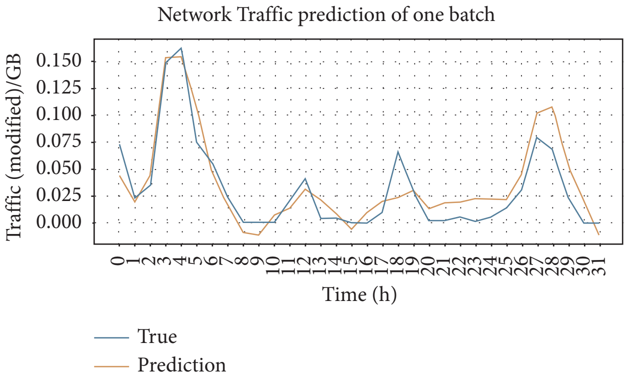

Figure 7: Traffic prediction results 1.

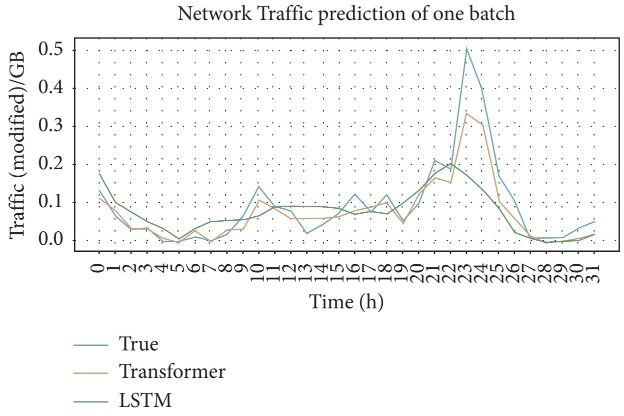

Figure 8: Traffic prediction results 2.

Table 1: Performance of the proposed model vs. other deep learning models.

Model	Mae	MSE	Fault tolerant rate (%)
RNN	0.05117	0.05476	10.675
GRU	0.05175	0.05484	11.144
LSTM	0.05277	0.05438	10.900
Proposed model	0.03993	0.03653	13.109

the network traffic prediction model more visually through visual images.

4.1. Network Traffic Prediction Model Results. Figure 7 shows the prediction results of a batch, with the horizontal axis being the time series and the vertical axis being the network traffic size (normalized).

The blue curve is the accurate data, and the orange line is the transformer prediction data. As can be seen from the graph, the overall flow magnitude and trend predicted by the short time series are accurate. The overall time lag of the model is almost nonexistent, and the prediction curve grows in parallel with the natural curve, which can be applied to entire network prediction systems.

4.2. Comparison with the Results of LSTM. Figure 8 illustrates that the network traffic prediction model fits the actual data better than the LSTM model and has better prediction results at the abnormal time points 22–27. From points 10, 19, 21, etc., we can see that the transformer model is less affected by time lag than the LSTM model and is more suitable for practical use.

By performing gradient operation on the output in order to make the model converge, so as to achieve the smallest value of the loss function, the gap between the predicted value of the model and the actual value becomes smaller, and the predicted value of the model will be reduced. In light of this, we calculated the MAE, MSE, and fault tolerant rate of these models base on equations (1) and (2). As shown in Table 1, compared with other deep learning models, the MAE and MSE of proposed model is smaller, and the fault tolerance accuracy is (prediction accuracy) higher, by about 30%.

5. Conclusion

In this paper, the transformer deep learning model is used to predict network traffic, which is the theoretical foundation and basis for resource preallocation. Practical comparison proves that the training model adopted has a faster convergence speed, higher accuracy, and is easier to handle multidimensional feature data. We apply it to time series prediction based on real-life network traffic data using an attention mechanism with high fitting capability and parallel output. The proposed network prediction model can better understand the size of network traffic and provide a theoretical basis for the refined allocation of server resources. Through the analysis of network traffic resources, the number of base stations in high-load areas can be optimized and the energy consumption of base stations in low-load areas can be reduced, and thus improve the utilization of network resources.

Data Availability

In this experiment, Vietnam's two years of 4G base station data are used as the training datathat are set using the random sliding window method. https://www.kaggle.com/naebolo/predict-traffic-of-lte-network.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (No. 61901096), National Key R&D Program of China No. 2018YFB1801302, Project for Innovation Team of Guangdong University No. 2018KCXTD033, Social Welfare Science and Technology Research Project of Zhongshan City (Nos. 2020B2018 and 2021B2026), and Construction Project of Professional Quality Engineering in 2022 (No. YLZY202201).

References

[1] K. C. Chang, K. C. Chu, H. C. Wang, Y. C. Lin, and J. S Pan, "Energy saving technology of 5G base station based on Internet of things collaborative control," *IEEE Access*, vol. 8, pp. 32935–32946, 2020.

[2] J. A. Bastos, "Forecasting the capacity of mobile networks," *Telecommunication Systems*, vol. 72, no. 2, pp. 231–242, 2019.

[3] A. Yamamoto, H. Osanai, and A. Nakao, "Prediction of traffic congestion on wired and wireless networks using RNN," in *Proceedings of the International Conference on Ubiquitous Information Management and Communication*, pp. 315–328, Springer, Taichung, Taiwan, January 2019.

[4] S. Wang, Q. Zhuo, and H. Yan, "A network traffic prediction method based on LSTM," *ZTE Communications*, vol. 17, no. 2, pp. 19–25, 2019.

[5] N. Ramakrishnan and T. Soni, "Network traffic prediction using recurrent neural networks," in *Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 187–193, IEEE, FL, USA, December 2018.

[6] A. Vaswani, N. Shazeer, and N. Parmar, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[7] J. D. M. W. C. Kenton and T. L. K. Bert, "Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the naacL-HLT*, vol. 1, pp. 4171–4186, 2019.

[8] J. B. Cordonnier, A. Loukas, and M. Jaggi, "On the relationship between self-attention and convolutional layers," *Proceedings of ICLR*, vol. 2021, pp. 1–18, 2020.

[9] I. Sutskever, Oriol Vinyals, and V. Quoc, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 27, no. 3, 2014.

[10] L. Liu, H. Jiang, and P. He, "On the variance of the adaptive learning rate and beyond," *Proceedings of ICLR*, vol. 2021, pp. 1–14, 2020.

[11] A. KingaD, "A method for stochastic optimization," in *Proceedings of the ICLRAnon International Conference on Learning Representations*, SanDego, CL, USA, April 2015.

[12] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks, Proceedings of the fourteenth international conference on artificial intelligence and statistics," *JMLR Workshop and Conference Proceedings*, vol. 15, pp. 315–323, 2011.