Proactive Network Traffic Prediction using Generative Adversarial Network

Gyurin Byun[1], Van-Vi Vo [2], Syed M. Raza [3], Duc-Tai Le[4], Huigyu Yang[5], and Hyunseung Choo [1,2,3,4,5,*]

[1]Dept. of AI Systems Engineering, Sungkyunkwan University, Suwon, South Korea
[2]Convergence Research Institute, Sungkyunkwan University, Suwon, South Korea
[3]Dept. of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, South Korea
[4]College of Computing and Informatics, Sungkyunkwan University, Suwon, South Korea
[5]Dept. of Superintelligence Engineering, Sungkyunkwan University, Suwon, South Korea
[*]Corresponding author (choo@skku.edu)

***Abstract*—Proactive traffic management in 5G networks is crucial for optimizing network efficiency, ensuring quality of service, adapting to dynamic conditions, and enhancing the user experience. In this paper, we propose a Generative Adversarial Network (GAN) architecture that leverages spatiotemporal features in network traffic data to predict future traffic. Our approach incorporates a Convolutional Long Short-Term Memory (ConvLSTM) model within the generator of the GAN, which collaborates with the discriminator, utilizing a Convolutional Neural Network (CNN) model, to provide essential feedback for training the generator. This integration ensures that our model not only predicts future traffic with improved accuracy but also adapts to dynamic network conditions. Based on experimental results using network traces, our model significantly outperforms the baseline, reducing prediction error by 12% while forecasting network traffic for the next 1 minute. These findings represent a significant advancement in proactive network management, particularly in addressing the challenges posed by real-time streaming and other latency-sensitive applications in 5G networks.**

***Index Terms*—Network traffic prediction, Generative Adversarial Network, Convolutional Long Short-Term Memory, Spatiotemporal features.**

I. Introduction

The recent rapid growth in computing capability and wireless technology has triggered massive adoption of mobile devices, an increase in mobile content and services, and the rise of new technologies like the Internet of Things (IoT). As a result, monthly global internet traffic is estimated to reach 607 exabytes (EB) in 2025 and 5,016 EB in 2030 [1]. The increase in network traffic, combined with heightened user quality of service (QoS) requirements, presents new challenges for operators in managing networks. In response to these challenges, there has been a notable increase in research focusing on developing innovative traffic management approaches in the literature.

Traditional traffic management approaches [2], [3], which rely on threshold-based algorithms, often struggle to keep up with the rapid changes in traffic conditions. As a result, these approaches fall short of effectively optimizing network resource utilization and ensuring consistent quality of service. To overcome these limitations, a proactive approach is necessary. By accurately forecasting upcoming traffic volumes, proactive traffic management enables the network to adequately prepare for incoming traffic. Deep learning (DL)-based approaches present a compelling solution for network traffic prediction, leveraging their advanced capability to analyze intricate patterns and swiftly adapt to dynamic traffic conditions.

Convolutional LSTM (ConvLSTM) [4], [5], combine the features of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) to handle both spatial and temporal aspects of data. Traditional feed-forward networks struggle with sequential data due to their independent processing approach. Recurrent Neural Network (RNN) improved this by linking sequence data, but faced gradient vanishing issues with longer sequences. LSTM addresses this by managing information flow with a cell state and gates, enabling better learning of long-term dependencies. ConvLSTM leverages this to efficiently process time-series data transformed into images, integrating spatial feature extraction of CNN with LSTM sequential data handling.

Generative Adversarial Network (GAN) [6] is a model primarily used for data generation, characterized by the structure in which two neural networks compete and evolve together. A GAN model consists of a generator that creates fake data that resembles real data and a discriminator that differentiates between real and fake data. When using GAN for prediction tasks, the generator produces fake prediction results similar to real data and the discriminator distinguishes between the two sets of results. The generator is trained to deceive the discriminator and create data more akin to the real data. In contrast, the discriminator is trained to become better at distinguishing predicted data. Through this process, the performance of the generator is enhanced, making it useful as a prediction model [7].

The core of network traffic prediction lies in time-series forecasting. While many scholars have delved into this area [8], [9], existing forecasting methods are primarily tailored for regularly sampled sequences, posing challenges in handling

This work was supported by IITP grant funded by the Korea government(MSIT) under Artificial Intelligence Graduate School (No.2019-0-00421), and Artificial Intelligence Innovation Hub (No.2021-0-02068) and the ICT Creative Consilience Program(IITP-2023-2020-0-01821).

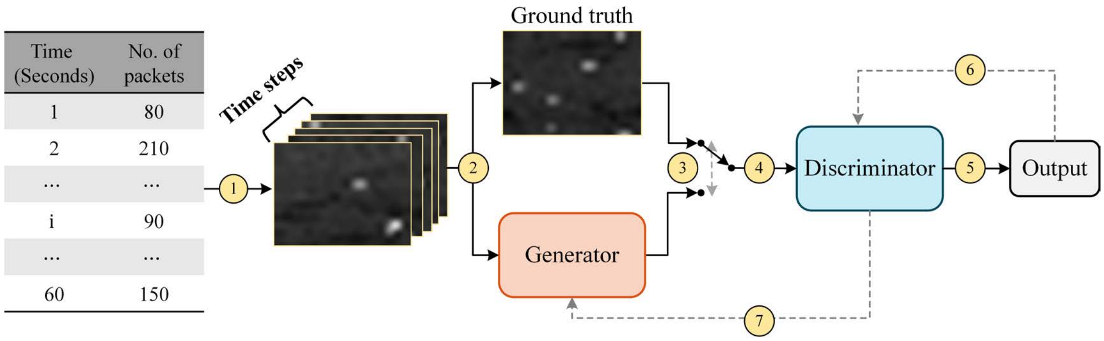

Fig. 1. Overall structure of proposed model

the uncertainty and long-term dependencies inherent in time-varying network traffic data. To address these issues, our paper introduces a pioneering approach. It involves converting raw packet trace data into images for training, facilitating the efficient extraction of spatiotemporal characteristics. Additionally, we propose an innovative model that integrates ConvLSTM with a GAN framework, intending to significantly improve the accuracy and reliability of network traffic predictions.

The rest of the paper is presented as follows: we describe the GAN-based network traffic prediction model in Section II. Section III presents the experiment results. Finally, we conclude our work and discuss our future research in Section IV.

II. GAN-Based Network Traffic Prediction Model

The dataset used in the paper, called CAIDA2019, consists of *.pcap* files containing payload-based anonymous traffic. The data were captured by the Applied Internet Data Analysis (CAIDA) center for one hour on January 17, 2019 collected traces on several backbone OC192 links to facilitate data sharing. The CAIDA located on the UC San Diego campus in La Jolla, CA is an applied Internet data analysis center founded in 1997 and has a research infrastructure to conduct network research and support large-scale data collection, queueing, and data distribution to scientific research communities. One *.pcap* file contains an hour of capture and has a very large size (117GB) [10].

$$\text{pixel} = \frac{T_{cur} - T_{min}}{T_{max} - T_{min}} \times 255 \tag{1}$$

We present our proposed GAN-based network traffic prediction model in Fig. 1. We first pre-process the raw data at step 1 by calculating the number of packets in 1 second to a pixel value using Equation 1 before converting them into images by time steps. In Equation 1, T_{cur} is the number of packets in 1 second, T_{min} is the minimum number of packets in 1 second period in the whole dataset, and T_{max} is the opposite of T_{min}. This process is repeated till the end of traffic data in traces to create pixels that are placed row-wise in an image of 20×15 representing traffic in a minute. Moreover, for better performance, the last part of an image is overlapped in the next image, and the size of the overlapping is determined through a sliding window. The smaller the window, the smaller the overlapping, and the more extended the prediction of future traffic, and vice versa.

The procedure of the proposed model is presented from step 2 to step 7 in Fig. 1. This model consists of a generator and a discriminator. The generator takes a sequence of five traffic images as input 2. A Convolutional Long Short-Term Memory (ConvLSTM) is used in the generator that learns from the input sequence and predicts the next image in the sequence 3. The discriminator takes predicted and ground truth images as inputs 4, and distinguishes between them using CNN 5. Outputs of the discriminator are utilized to calculate discriminator loss, respectively, for updating the weights in the discriminator 6. The intermediate weight vectors of the discriminator produced by inputs, respectively, are used to calculate featuring matching loss for updating the generator weights 7.

The ConvLSTM in the generator consists of four fully connected layers of LSTM, and 2D-convolution operations are done in each LSTM cell. The results from each layer are passed to the next layer after batch normalization. The final layer outputs the predicted image, and the weights of the whole ConvLSTM are updated using discriminator loss. The discriminator consists of a CNN that has three convolution and three max-pooling layers with each layer having 64 filters of 2×2 dimensions, and this is followed by a flattening layer. A single fully connected layer at the end of the CNN performs the binary classification and its discriminator loss is to update the weights of CNN.

As the purpose of the proposed model is to predict the next image in the sequence, the feedback provided by the discriminator to the generator based on the binary classification loss is not meaningful. Instead, the difference between the predicted image and the ground truth allows the ConvLSTM in the generator to learn to predict images with higher accuracy. Therefore, weight vectors from the last convolution layers of CNN for the inputs of predicted and ground truth images are extracted, and their L2-norm is used.

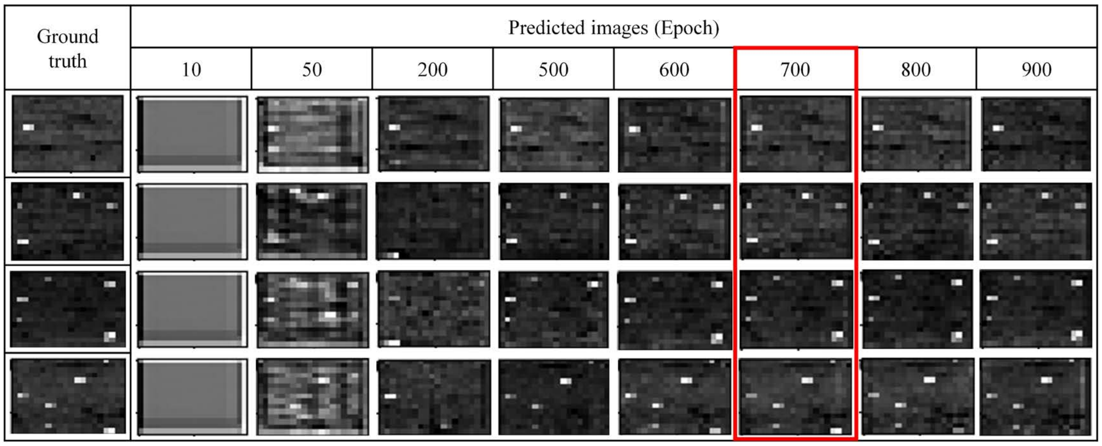

Fig. 2. Comparison of predicted images by epoch.

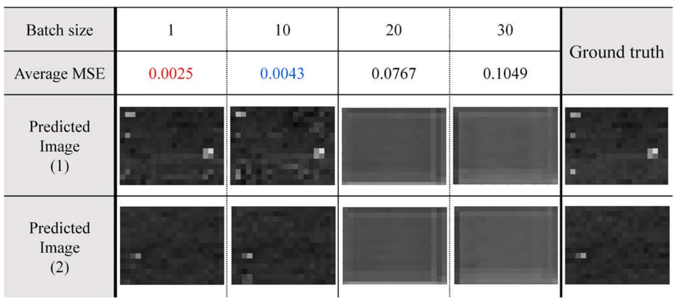

Fig. 3. Comparison of predicted images by batch size.

III. EXPERIMENT RESULTS

The proposed model is implemented in Python using Keras libraries. The implementation and all the experiments are carried out on a hardware environment consisting of Intel i7 CPU, 32GB RAM, and GeForce RTX 3080 Ti GPU. Moreover, each experiment is repeated over 15 times to establish statistical relevance, and the average results are presented in this paper. Converting time-series data into images generated 58 datasets. The evaluation of the results is carried out using Mean Square Error (MSE) for n image sequences in the test dataset. If we have y_i representing a pixel in the predicted image, and in the ground truth image t_i represent the corresponding pixel, then the MSE of the proposed model can be computed as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2 \tag{2}$$

We must set an appropriate epoch value when building a model to prevent underfitting and overfitting. Fig. 2 compares the prediction images for four ground truths across different epochs. Results show underfitting for epochs below 50, while from 50 up to 600, the model learns the finer details of the images. As epochs increase, both training and testing errors decrease. However, overfitting begins after 700 epochs, so in this study, the best fit is determined to be at 700 epochs.

Next, we compare the impact of different batch sizes on model training, a key hyperparameter in addressing poor learning performance. Larger batch sizes accelerate learning and convergence, reducing the risk of local optima, but can lead to overfitting due to less variance in loss values. Conversely, smaller batch sizes increase iterations and steps, offering regularization benefits but taking longer to train and possibly leading to local minima. Fig. 3 shows experimental results at 700 epochs for batch sizes 1, 10, 20, and 30. Larger batch sizes (20, 30) show errors stabilizing at local optima, indicating a halt in learning. While batch sizes 1 and 10 yield good results, a middle value of 5 is chosen as the optimal batch size for the model, considering training time.

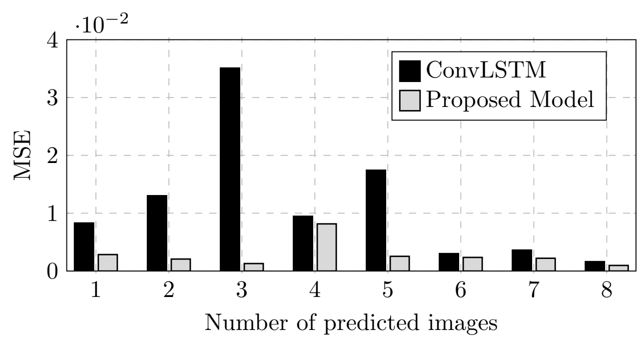

Fig. 4. The proposed model comparison with ConvLSTM.

The results of the proposed model are generated for increasing traffic prediction durations by reducing the sliding window size, and they are compared with ConvLSTM in Fig. 4. Intuitively, the average MSE of the proposed model increases with the increase in prediction duration as the overlapping part of the two consecutive images reduces. This shows that the high degree of overlapping allows model to better learn the correlation among traffic images. Moreover, the proposed model outperforms the ConvLSTM by showing 12% lower errors for 1 min traffic prediction durations, respectively. This better performance can be attributed to adversarial learning and feature matching in GAN architecture that allows the model to learn better with a high convergence rate.

IV. Conclusion and discussion

This preliminary study proposes a method for extended prediction of future traffic. In particular, it uses image representations of network traffic for achieving extended prediction and proposes a GAN-based prediction model for attaining better performance. The training of the proposed model exploits feature matching for stabilizing the GAN learning process with a higher convergence rate. The initial results show better results compared to ConvLSTM; however, a thorough evaluation is required to confirm the superiority of the proposed model over the state-of-the-art. A comprehensive performance analysis for diverse datasets with the improved proposed model is currently ongoing and will be addressed in our future study.

References

[1] F. Tariq, M. R. A. Khandaker, K.-K. Wong, M. A. Imran, M. Bennis, and M. Debbah, "A speculative study on 6G," IEEE Wireless Communications, vol. 27, no. 4, pp. 118-125, 2020.

[2] K. Zhang, R. Gençay, and M. E. Yazgan, "Application of wavelet decomposition in time-series forecasting," Econ. Lett., vol. 158, pp. 41–46, 2017.

[3] P. Cortez, M. Rocha, and J. Neves, "Evolving time series forecasting arma models," J. Heuristics, vol. 10, no. 4, pp. 415–429, 2004.

[4] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015.

[5] W. Jiang, "Internet traffic matrix prediction with convolutional LSTM neural network." Internet Technology Letters 5, no.2, 2022.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems, vol. 27, 2014.

[7] K. Zhang, G. Zhong, J. Dong, S. Wang, Y. Wang, "Stock market prediction based on generative adversarial network," Procedia Comput., pp. 400–406, 2019.42sa1420.

[8] K. Wu, J. Liu, P. Liu, and S. Yang, "Time series prediction using sparse autoencoder and high-order fuzzy cognitive maps," IEEE Trans. Fuzzy Syst., vol. 28, no. 12, pp. 3110–3121, Dec. 2020.

[9] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," IEEE Commun. Mag., vol. 57, no. 6, pp. 114–119, Jun. 2019.

[10] Datasets for CAIDA UCSD Anonymized Internet Traces https://www.caida.org/data/passive/dataset.xml, retrieved on 11, 2020.