IEICE Communications Express, Vol.13, No.8, 319–322

ComEX

LETTER

Proposal of temporal feature layers for network traffic dataset generation using C-GAN

Yukito Onodera[1, a)], **Erina Takeshita**[1], **Tomoya Kosugi**[1], **and Satoshi Suzuki**[1]

Abstract As machine learning research in the networking field has become more active in recent years, the demand for network traffic datasets has increased. On the other hand, the amount and types of publicly available network traffic datasets are scarce as training datasets for machine learning. Therefore, we focus on the generative adversarial network (GAN) as a data generation model, aiming to use generated rather than publicly available training datasets. However, existing GANs have difficulty generating sufficiently diverse network traffic to improve generalization ability while representing variations across weekdays, weekends, and date. This study proposes a new layers inserted into the conditional GAN model with the functions of expanding dimensionality of time-series traffic data and embedding temporal position information. Experimental results show that the model with the proposed layers inserted generated diverse network traffic data that represents temporal features.

Keywords: network traffic generation, generative adversarial networks
Classification: Network

1. Introduction

In recent years, machine learning (ML) has been studied in the networking field, such as for network traffic prediction, packet classification, and traffic imputation [1, 2]. Network traffic shows multiple patterns of periodicity and peculiar variations from day to day and week to week. Thus, the number of data in a network traffic representative dataset needs to be increased. However, the amount and types of publicly available network traffic training datasets are scarce, and creating new dataset is highly cost-intensive and time-consuming [3]. Therefore, there is a high demand for traffic generation that captures periodicity and fluctuation patterns.

Generative Adversarial Networks (GAN) [4] have attracted attention as network traffic data generation models. GANs consist of a generator and a discriminator. The generator generates data similar to the original traffic from the noise data, and the discriminator determines whether the data is the original traffic or the data generated by the generator. On the basis of the discriminator's decision results, the generator and the discriminator update their internal parameters. Since GANs cannot indicate the condition of generated data, conditional GAN (C-GAN), which generates data in accordance with a set of conditions, was proposed. C-GAN, traditionally used in computer vision, have recently been applied to network traffic prediction and anomaly detection [5]. C-GAN trains specific conditions of the target data and generate data corresponding to those conditions, making it useful for predictions and anomaly detection tailored to the target environment.

C-GAN finds it difficult to generate a diverse training dataset while capturing the variations due to weekdays, weekends, date and months such as network traffic. As a premise in ML, the training dataset needs to be of sufficient size and diversity to enhance generalization performance and expressiveness. Here, diversity means that the data in the dataset has a wide range of different features and patterns. In context of C-GANs, diversity means the ability to genereate different data. However, due to the parallel in process of generator and discriminator. Training traditional C-GANs are GAN is prone to a problem called mode collapse, where the same data may be generated depending. When using the generated data as training data, learning from the same data can lead to overfitting, rendering it meaningless [6]. Therefore, it is necessary to generate data that avoids mode collapse. However to the best of our knowledge, there is no C-GAN model for network traffic yet.

This paper proposes a method for generating data that maintains diversity while considering temporal features by transforming time-series data through the proposed method. The propose method involves inserting new layers C-GAN conditionated on the day of the week and date. By changing the condition input part of the C-GAN and inserting the new layers with position encoding and token embedding before the discriminator, the time-series data is temporally extended, learning a distribution similar to that of the original traffic and preventing mode collapse.

Experimental results shows that the data generated by the C-GAN with the proposed inserted layers captured the features of the day of the week and the time of the day and has sufficient diversity to be used as training data. The proposed layers were implemented in regressional and conditional GAN (RCGAN) [7], and training and data generation was performed using GEANT, a public network traffic dataset. Numerous generated data are analyzed by two evaluations, comparing the GEANT dataset as the ground truth.

2. Proposed method

This chapter describes the proposed method with an overview.

[1] Access Network Service Systems Laboratories, NTT Corporation, 3-9-11 Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
[a)] onodera.yuikito@ntt.com

DOI: 10.23919/comex.2024XBL0062
Received March 25, 2024
Accepted April 23, 2024
Publicized June 11, 2024
Copyedited August 1, 2024

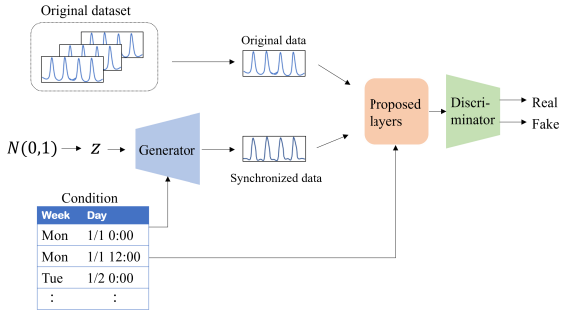


Fig. 1 Overview

2.1 Overview

This paper presents a method for adding a new layers to the middle of the standard C-GAN model between the generator and the discriminator, as illustrated in Fig. 1. The new layers increase the number of dimensions (token embedding) and temporal features (temporal embedding) of the time-series data and adds positional information (positional encoding) to the data with dimensional and temporal features. The details of each process step are described in Sec. 2.3. The data input for the proposed layer is an arbitrarily generated length t. The new layers perform spatiotemporal processing on the data input for the detection unit to generate time-series data. For each extended dimension, the data is given information that considers the respective time axis. This can convert data into specific temporal features, emphasizing temporal features and preventing mode collapse while allowing for training.

2.2 Impact of proposed method

In the standard C-GAN model, Generator G takes random noise $z \in \mathbb{R}^r$ as input and attempts to generate synthetic data similar to the training data distribution. Discriminator D aims to accurately determine whether the input data is original or generated, while Generator G aims to maximize the false-positive rate of Discriminator. Both models are trained on the basis of the value function V(G, D).

$$\min_G \max_D \mathrm{V(D,G)} = \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log \mathrm{D}(x|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-\mathrm{D}(\mathrm{G}(z|y)))], \quad (1)$$

where x is the original time-series data, z is the data generated by Generator, and y is the condition. If the loss function of each model is defined as L_D, L_g, the loss function of the GAN is expressed as follows:

$$\mathrm{L_D} = -\frac{1}{2}\mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log \mathrm{D}(x)] - \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[\log(1-\mathrm{D}(\mathrm{G}(z)))], \quad (2)$$

$$\mathrm{L_G} = -\frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[\log \mathrm{D}(\mathrm{G}(z))]. \quad (3)$$

2.3 Proposed layers

The process of the proposed layers is shown in Fig. 2. The proposed layers first perform token embedding on the input data to increase the dimensionality of the data. Next, temporal embedding is performed to add the respective temporal features, and location encoding is performed on them.

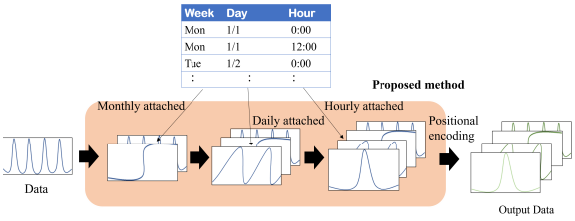


Fig. 2 Detail of proposed method

Each step of this process is explained below. Token embedding is initially performed on the input data to increase the dimensions using an embedding layer. In this process, a one-dimensional convolutional layer is used to transform each token of the input data into a high-dimensional feature space. The mathematical expression for the convolution layer is described as follows:

$$x_{\mathrm{conv}} = \mathrm{Conv1d}(x), \quad (4)$$

where x is the input data, and Conv1d is the convolution layer, and this convolution operation maps the original data x into a higher dimensional feature space with dimension number d, yielding d-dimensional data x_{conv}.

Next, temporal features (e.g., hour, day of the week, [day], and month) of the time-series data are embedded to give temporal significance to the data. This works by using different embedding layers for each time unit, mapping each time element to a specific dimensional space. On the basis of a specific time unit, an embedding operation, $Embed$, is performed on the input data. The following equation can express the value x_{embedd} after embedding each time unit.

$$x_{\mathrm{embedd}} = x_{\mathrm{conv}} + \mathrm{Em_{hour}}(x_{\mathrm{hour}}) + \mathrm{Em_{week}}(x_{\mathrm{week}}) + \mathrm{Em_{day}}(x_{\mathrm{day}}) + \mathrm{Em_{month}}(x_{\mathrm{month}}) \quad (5)$$

where $\mathrm{Em_{hour}}, \mathrm{Em_{week}}, \mathrm{Em_{day}}, \mathrm{Em_{month}}$ represent the embedding functions for the hour, day, [day], and month, respectively. $x_{\mathrm{hour}}, x_{\mathrm{week}}, x_{\mathrm{day}}, x_{\mathrm{month}}$ denotes the corresponding time unit of the input data. Finally, the positional encoding PE incorporates the relative position information of each element in x_{embedd}. The model can incorporate the relative position information of each input data element, enabling it to generate x_{encoded} that embeds the order and pattern of the time-series data. The following equation represents positional encoding PE:

$$x_{\mathrm{encoded}} = \mathrm{PE}(x_{\mathrm{embedd}}), \quad (6)$$

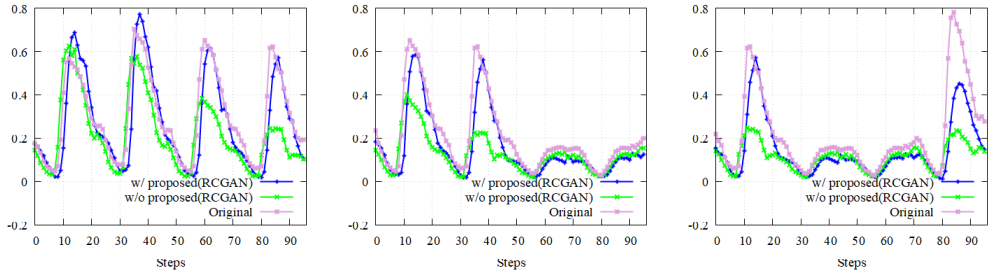
$$\mathrm{PE}(\mathrm{pos}, 2i) = \sin\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right), \quad (7)$$

$$\mathrm{PE}(\mathrm{pos}, 2i+1) = \cos\left(\frac{\mathrm{pos}}{10000^{2i/d_{\mathrm{model}}}}\right). \quad (8)$$

Where pos is the position, i is the index of the dimension and d_{model} is the number of dimensions in the model. This allows the model to capture the relative positional information of each element of the input data and to better emphasise the order and patterns of the time-series data.

3. Experiment

This chapter describes the experiment.



(a) Monday to saturday (b) Saturday to sunday (c) Friday to monday

Fig. 3 Experimental results

3.1 Experimental setup

The experiment used GEANT, a representative network dataset that exhibits the features of network traffic. The dataset shows a daily cycle and different amplitude peak differences between weekdays and weekends. The data acquisition period was from April 2022 to August 2023, with hourly intervals. Each dataset was separated by 96 generated length. The starting position of each dataset was shifted by one to generate the dataset [8].

The proposed layers were implemented in RCGAN, and two types of evaluations were conducted to analyze the data generated. (i) First, we evaluated the generated network traffic to determine whether it captured the features of each hour of the day. This was done to check whether the data was generated in such a way that the daily cycle, which is the features of dataset, and the peaks on weekdays were high and the peaks on weekends were low. At this time, we also determined whether the data can enable multiple patterns of days of the week to be generated. (ii) Next, two statistical methods were used to evaluate the traffic data generated in evaluation (i) and determine whether the traffic data generated in experiment (ii) could be generated in a way that maintained diversity. The Kolmogorov-Smirnov (KS) test and principal component analysis (PCA) were used as evaluation methods. The KS test is used to compare the generated traffic with the original traffic to determine if the generator can learn similar features. PCA plots the data generated in the experiment and checks for sparsity in the distribution plots to ensure that data with different features has been generated.

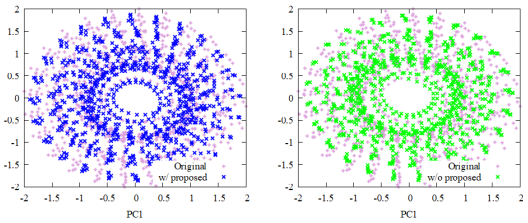
The experiment details are described below. The RCGAN model without the proposed layers were used for the conventional method. The proposed method (i.e., the RCGAN model with the proposed layers inserted) used generated length 96 and training epoch 10000. The model parameters $d = 20$ were employed. We used the temporal of year, month, date, and day of week.

3.2 Experiment result

First, the results of evaluation (i) are shown in Fig. 3. Figure 3 illustrates one of the 300 datasets generated by the proposed method (w/ Proposed) and the conventional method (w/o Proposed), together with the original traffic under the same conditions, where the vertical axis is the amount of normalized traffic, and the horizontal axis is steps. In Fig. 3, three conditions with different start dates are chosen. Also, each of the 24 steps indicates the time from 0 : 00 to 24 : 00 each day. Figure 3(a) shows an example of traffic variation from Monday to Thursday, showing Steps 0 – 23 for Monday, Steps 24 – 47 for Tuesday, Steps 48 – 71 for Wednesday, and Steps 72 – 95 for Thursday. Peaks of the original traffic on weekdays (e.g., Steps 11 – 15 on Monday) show roughly the same amplitude. The falling part (Steps 17 – 22) also shows a similar decrease to the original traffic. The proposed method shows amplitudes similar to the original, while the amplitude of the conventional method appears to be gradually decreasing. Figure 3(b) shows an example of Thursday to Sunday traffic variation. The amplitude of the original traffic is reduced when comparing the part of the original traffic corresponding to Step 12 on Thursday and Step 60 on Saturday. The proposed method shows similar amplitudes to the original on weekdays and weekends. On the other hand, the amplitude of the conventional method is smaller than that of the proposed method on weekends. However, it can be seen that the amplitude is gradually decreasing. Figure 3(c) shows an example of traffic variation from Friday to Monday. The original traffic shows decreasing amplitude when comparing Step 12 on Friday to Step 36 on Saturday and increasing amplitude when comparing Step 60 on Sunday to Step 84 on Monday. The proposed method shows similar amplitudes to the original traffic. However, the amplitude of the conventional method is smaller than that of the proposed method at Step 84 on Monday, although the amplitude gradually decreases from Friday to Saturday.

Figure 3 shows that the proposed method can reproduce the features of variation over time. In the original data, the peak values are higher on weekdays than on weekends. The proposed method reproduces this variation, where as the conventional RCGAN method cannot be said to reproduce it because the peak values gradually decrease from Monday to Sunday and do not increase significantly from Sunday to Monday. This may be because the proposed method adds temporal features by temporal embedding to the C-GAN.

Next, the results of the KS test and PCA analysis for the traffic generated in evaluation (i) are described. First, we explain the KS test results. The distance between the original data and the data generated by the proposed method is 0.156, and the p-value is 0.144. The KS test is used to determine the statistical significance of the p-value, and the distance indicates the maximum difference between the two distributions. Suppose the p-value is more significant than 0.05. There is no statistically significant difference in that case, and the null hypothesis that the two sample populations come from the same distribution cannot be rejected.



(a) w/ proposed (b) w/o proposed

Fig. 4 Experimental results of PCA

In other words, the original data and data generated by the proposed method show no statistically significant difference and are similar in distribution. On the other hand, the distance between the original data and the data generated by the conventional method is 0.216, and the p-value is 0.0143, indicating a statistically significant difference between the original and generated data and that the data are distributionally different.

Figure 4 shows a point-by-point plot of the 300 traffic data generated during inference using PCA. The original traffic under the same conditions used during inference is also plotted. Figure 4(a) shows that the plots are sparse within a particular cluster for the proposed method, indicating that the conditions generate the data diversity. Figure 4(b) shows that the plots are concentrated in the case of the conventional method, and the same data is highly likely to be generated in different conditions.

The KS test results show that the probability distribution is made close to the original traffic by using the proposed method. Based on the KS test results, the null hypothesis is not rejected. The proposed method will likely consist of a sample with a distribution close to that of the original traffic. The PCA results show that the generated data has diversity. Different plots are illustrated for each generated data, so data generation under different conditions is unlikely to produce identical data. The present evaluation reduces the 96 generated growth data to two dimension of information using PCA. Therefore, points plotted close together are likely to have the same traffic data generated. This means that the graph should be as sparse as possible, and the data of the proposed method is sparser than that of the conventional method.

The evaluation (ii) results suggest that token embedding and temporal embedding enable learning with a distribution close to that of the original traffic but with diversity. This is likely because the proposed method expands the data on the basis of temporal features and adds location information. The convolutional layer spatially extends specific data and increases the dimensionality. As the data is extended in accordance with the parameters of the convolution layer, it is likely to have features similar to the original traffic. By adding the respective temporal information, the variance of the pre-transformed data may be tolerated.

The results of evaluations (i) and (ii) show that by including the proposed layers, traffic generation solves the problems of conventional C-GANs. Specifically, the results show that the RCGAN with the proposed layers can generate traffic that captures temporal features such as peaks and the periodicity of weekdays and weekends that appear in the datasets used in the experiments. The data analysis also showed that the distribution of traffic is close to that of the original traffic, yet the multiple generated data have different features. In other words, the generation can capture a wide range of features and patterns in the dataset.

4. Conclusion

This paper aims to generate a network traffic dataset while maintaining diversity of temporal features because we suppose that existing generative adversarial networks (GANs) need help generating datasets that learn the essential temporal features of network traffic and exhibit the diversity necessary for effective machine learning (ML). Therefore, we develope a new layers of GAN model designed to represent the temporal dynamics of network traffic accurately. Our model enhances the conventional conditional GAN (C-GAN) framework by inserting a novel layers tailor for expanding the dimensionality of time-series data (through token embedding), embedding temporal features (via temporal embedding), and incorporating temporal positional information (with positional encoding). Experimental results show that the data generated with our model can catch temporal features of network traffic. The visual evaluation found that the temporal features, such as peaks and the periodicity of weekdays and weekends, appearing in the dataset are captured. The statistical evaluation showed that a diverse dataset was generated for the given conditions by PCA analysis and was distributionally close to the KS test. Our future work will focus on conducting comparative studies with other generative models and evaluating the performance of ML models trained on datasets generated by our approach.

References

[1] G.O. Ferreira, C. Ravazzi, F. Dabbene, G.C. Calafiore, and M. Fiore, "Forecasting network traffic: A survey and tutorial with open-source comparative evaluation," *IEEE Access*, vol. 11, pp. 6018–6044, 2023. DOI: 10.1109/access.2023.3236261

[2] A. Azab, M. Khasawneh, S. Alrabaee, K.-K.R. Choo, and M. Sarsour, "Network traffic classification: Techniques, datasets, and challenges," *Digital Communications and Networks*, 2022. DOI: 10.1016/j.dcan.2022.09.009

[3] A. Roy, H. Zeng, J. Bagga, G. Porter, and A.C. Snoeren, "Inside the social network's (datacenter) network," ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp. 123–137, Aug. 2015. DOI: 10.1145/2829988.2787472

[4] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," vol. 63, no. 11, pp. 139–144, 2014. DOI: 10.1145/3422622

[5] M. Usama, M. Asim, S. Latif, J. Qadir, and Ala-Al-Fuqaha, "Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems," 2019 15th IWCMC, pp. 78–83, 2019. DOI: 10.1109/iwcmc.2019.8766353

[6] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative adversarial networks in time series: A survey and taxonomy," 2021.

[7] C. Esteban, S.L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional gans," 2017.

[8] T.G. Dietterich, "Machine learning for sequential data: A review," 2002 Proc. SSPR 2002 and SPR 2002 Windsor, Ontario, Canada, August 6–9, pp. 15–30, 2002. DOI: 10.1007/3-540-70659-3_2