Generation of Similar Traffic Using GAN for Resolving Data Imbalance

Woo Ho Lee(✉), Chae Sang Lim, and Bong Nam Noh

Interdisciplinary Program of Information Security,
Chonnam National University, Gwangju, South Korea
leeouho@naver.com

Abstract. Recently, as the practical application of deep learning has become possible, research on the problems pertaining to intrusion detection has increased.

However, it is difficult to detect a small number of attack traffic when the real network is connected to produce an imbalance between the attack traffic class data and the normal traffic data necessary for learning. In this study, we propose a method to improve the accuracy of attack traffic data detection by creating similar attack traffic, using a Generative Adversarial Network (GAN) algorithm of deep learning. The proposed method generates similar attack traffic for NSL–KDD, ISCX 2012, and USTC_TFC 2016 datasets, which are well-known intrusion detection learning data sets. Experiments have shown that the data imbalance in each data set can improve classification accuracy by 10–12%, owing to the degradation problem.

Keywords: Deep learning · Intrusion detection · Security · Generative Adversarial Network

1 Introduction

Recently, studies on deep learning have been conducted for various purposes in the fields of big data, cloud computing, malware detection, and traffic management. Among them, intrusion detection is one of the fields that is most actively researched by using deep learning. Currently, studies on intrusion detection are being conducted using network traffic datasets that are available to the public. In particular, NSL–KDD [1], ISCX2012 [2], and USTC-TFC2016 [3] datasets are often used as training data. However, datasets used for deep learning have insufficient attack traffic information compared to normal traffic. This can cause an imbalanced data problem in the learning processes that use deep learning. The imbalanced data problem refers to imbalanced ratios of data classes where the accuracy of the class with a large amount of data is high, and the accuracy of the class with a small amount of data is low, thus increasing the difficulty of determining performance. To solve the imbalanced data problem, this study proposes an oversampling method that uses the GAN algorithm to create similar attack traffic in a dataset for learning purposes.

J. J. Park et al. (Eds.): CUTE 2018/CSA 2018, LNEE 536, pp. 1–7, 2020.
https://doi.org/10.1007/978-981-13-9341-9_1

The structure of this paper is as follows:

- Section 2 analyzes various datasets and outlines studies on intrusion detection, using deep learning with datasets.
- Section 3 describes the structure and data preprocessing of the proposed method.
- Section 4 describes the details of the experiment.
- Section 5 presents the conclusion and future research subjects.

2 Related Work

2.1 Analysis of Datasets

The datasets generally used on intrusion detection studies using deep learning were analyzed. As shown in Table 1, the NSL–KDD and USTC-TFC 2016 datasets have insufficient attack traffic compared to normal traffic. Furthermore, the DDOS attack traffic accounts for more than 50% of all attack traffic, causing the imbalanced data problem. In the case of the ISCX 2012 dataset, the ratio of the attack traffic that can be used for training to the normal traffic ranges from 1:515 at the minimum to 1:1220 at the maximum, causing the imbalanced data problem.

Table 1. Analysis of NSL–KDD, ISCX 2012, and USTC-TFC 2016 datasets

<table>
<tr><td colspan="9" align="right">Whole dataset</td></tr>
<tr><td colspan="3">ISCX 2012 Dataset</td><td colspan="3">NSL–KDD Dataset</td><td colspan="3">USTC-TFC 2016</td></tr>
<tr><td colspan="2">Traffic</td><td>Count</td><td colspan="2">Traffic</td><td>Count</td><td colspan="2">Traffic</td><td>Count</td></tr>
<tr><td colspan="2">Normal</td><td>41,480</td><td colspan="2">Normal</td><td>80,767</td><td colspan="2">Normal</td><td>345,407</td></tr>
<tr><td rowspan="8">Bot</td><td>Neris</td><td>8,039</td><td rowspan="8">Ab-normal</td><td>Probe</td><td>5,356</td><td rowspan="8">Ab-normal</td><td>Cridex</td><td rowspan="8">406,633</td></tr>
<tr><td>Rbot</td><td>6,073</td><td>DoS</td><td>223,488</td><td>Geodo</td></tr>
<tr><td>Virut</td><td>18,914</td><td>U2R</td><td>228</td><td>Htbot</td></tr>
<tr><td>Menti</td><td>217</td><td>R2L</td><td>1,376</td><td>Miuref</td></tr>
<tr><td>Sogou</td><td>34</td><td></td><td></td><td>Neris</td></tr>
<tr><td>Mutio</td><td>2,013</td><td></td><td></td><td>Nsis-ay</td></tr>
<tr><td>Nsis-ay</td><td>4,395</td><td></td><td></td><td>virut</td></tr>
<tr><td>Total</td><td>39,685</td><td>Total</td><td>230,448</td><td>Zeus</td></tr>
<tr><td colspan="2">Total</td><td>81,165</td><td colspan="2">Total</td><td>311,215</td><td colspan="2">Total</td><td>752,040</td></tr>
</table>

2.2 Deep Learning-Based IDS-Related Research

Recently, studies on intrusion detection using deep learning have diversified and increased rapidly. Yao et al. [4] proposed a classification method using k-means, KNN, decision tree, and random forest algorithms based on the NSL-KDD dataset and a classification method for attack traffic. In this study, the attack detection rate was 95.4% for normal traffic, 93.9% for DOS attacks, 56.1% for probe attacks, and 77.2% for U2R and R2L attacks. The low detection rate of the probe attacks at 56.1% confirmed that the smaller the number of attack data is, the lower the detection rate becomes. Rathore et al. [5] evaluated datasets for real-time processing. They applied random forest tree, conjunctive rule, SVM, and naive Bayes procedures to DARPA, KDD99, and NSL–KDD datasets. This study had a limitation because the accuracy was 99.9% in the experimental environment; however, it was not applicable to an actual environment. Tang et al. [6] used a deep neural network with the NSL–KDD dataset to apply an intrusion detection system (IDS) in a software-defined networking environment. They selected six features among 41. The experiment results showed a loss rate of 7.4%, and an accuracy of 91.7%. They also described a classification method using all 41 features.

Mylavarapu et al. [7] conducted a study on real-time detection using the ISCX2012 dataset. For real-time processing, they applied the multilayer perceptron neural networks based on Storm, and classified traffic at 89% accuracy. Yu et al. [8] proposed a session-based network intrusion detection method using CTU013 and ISCX 2012. This method used the stacked denoising autoencoder and showed an accuracy of 98.11% in the experimental environment. However, studies on datasets in an actual environment are insufficient. Wang et al. [3] suggested a method of applying the convolutional neural network (CNN) model with only several hundred bytes of session traffic, which detects malware traffic in the early stages. This method is characterized by independence from protocols because it classifies images using the CNN algorithm. Research on intrusion detection using deep learning is being conducted to improve the accuracy of intrusion detection by extracting the characteristics of attack data, using undersampling and few-shot methods [9]. Most studies on intrusion detection use the undersampling method to increase the layers of the neural network because it is effective for detecting a few classes [10]. However, the undersampling method shows a low accuracy in an actual environment, although its accuracy in an experimental environment is high. In an actual environment, the imbalanced data problem becomes worse compared to the learning data, and the detection rate decreases [11].

Therefore, in this study, in order to apply the oversampling method as a solution to the imbalanced data problem, a similar traffic generation method is proposed to increase the training data of the attack class traffic using a GAN.

3 Proposed Similar Traffic Creation System Using GAN

3.1 Network Feature Extraction

In this thesis, 15 data set characteristics, such as Duration, Header Length, IPversion, Protocol, Flag, and Session were extracted and correlated to detect malware using network information. Through correlation analysis, six Features of Duration, TCP Header, Port, Session Data, and Flag were selected and applied to a GAN Algorithm. To adjust the size of the data, the hash value is applied and quantified. This allowed the creation of images of the same size as when they were created. The figure below (Fig. 1) shows the CSV file and the correlation analysis for each Feature.

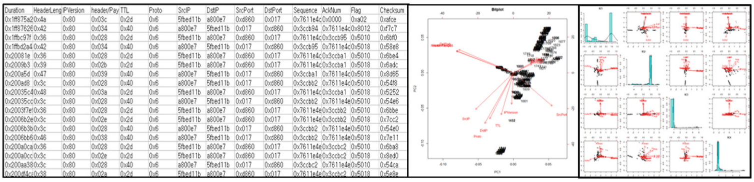

Fig. 1. System architecture for similar traffic generation

3.2 GAN Using Similar Traffic Algorithm

DCGAN is applied to generate network traffic efficiently [12]. Discriminators used in CNN use the usual stride convolution. The generator will produce an image that is as closely as possible similar to the input value. In the case of network traffic, we use Fractionally Stride convolution to increase the size of the Feature-map in case of small images. Discriminator is a Stride Convolution that determines the step size of the kernel when learning an image. It reduces the number of unnecessary parameters and selects only important features when using a normal pooling layer, but it has the disadvantage of losing image position information. In the case of network traffic, we used Stride Convolution because the area is divided and location information is important (Fig. 2).

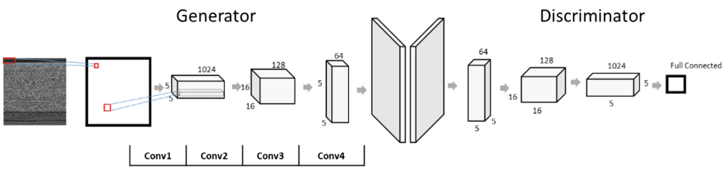

Fig. 2. Illustration of similar traffic created based on a GAN

Algorithm 1 creates images using GAN. From the attack traffic generated through the data-preprocessing step, the GAN algorithm is used to create similar traffic for each class. The criteria for generating similar traffic were between 0.98 and 1 for the loss function of the Generator and between 0 and 1 for the loss function of the Discriminator. The generated similar traffic that meets these criteria was updated to the training data by class. A similar traffic-create criterion generates similar traffic for class loss of 40% of total attack traffic. The created data were only used as training data. If the created similar data would be used for validation data, an overfitting problem can be generated. Thus, it was only used as training data for the test (Fig. 3).

Algorithm 1 GAN Create Algorithm

```
D_A= Dataset (Attack Traffic)
D_S = Similar Traffic Class
D_C = Similar Traffic Create
D_T =  D_A + D_C
g = Generate loss
d = Discriminator loss
Function-GAN(D_A)
FOR counter i = 0 to Ds length DO
  IF (D_A * 40 /100) > (D_A * D_S[counter i] /
100) THEN
    FOR counter j = 0 to file length DO
      IF d>0.98 && d < 1 THEN
        IF g >0 && g<0.1 THEN
          D_T = D_A+D_C
        ENDIF
      ENDIF
    ENDFOR
  ENDIF
ENDFOR
RETURN with D_T
```

Fig. 3. GAN algorithm used for image creation

4 Details of Experiment

4.1 Detection Experiment Using Similar Traffic Dataset

An experiment was conducted to examine the effect of normal/attack traffic created with the GAN Generator on the classification. In this experiment, mirai, virut, smoke_bot, menti, and zeus, which have small amounts of attack traffic, were used.

In the first experiment, the accuracy was analyzed depending on the learning volume using the datasets. The accuracy was measured up to 5,000 epochs under the same conditions for attack traffic of the learning data as before the experiment using GAN.

When the validation test was performed at more than 4,000 epochs, the accuracy was 90–92%, thus improving by approximately 10–12% compared to before the creation of the similar traffic. Furthermore, the learning accuracy for the test data also increased by approximately 7–12%. Therefore, the accuracy improved as the learning volume increased with the similar traffic creation method of the GAN algorithm, and the detection performance was improved by resolving the imbalanced data problem of the datasets (Fig. 4).

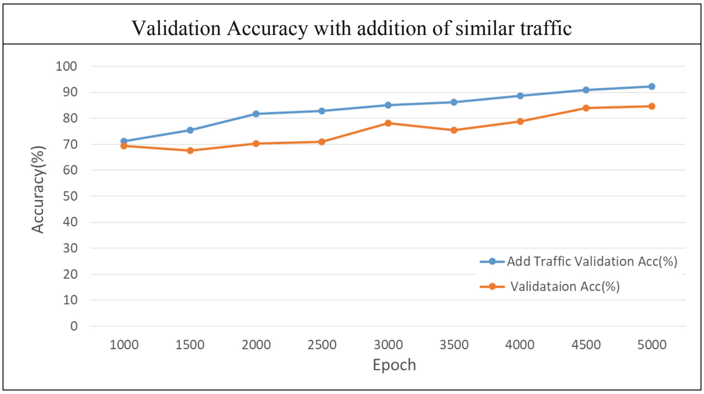

Fig. 4. Measuring and comparing epoch reference accuracies.

In the second experiment, experiments were performed for each data set using the CNN algorithm in a real-time environment.

Table 2 shows the experimental results for classification of attack traffic for NSL-KDD, USTC-TFC 2016, and ISCX 2012 datasets using CNN after adding the data created using GAN to the existing datasets. The classification accuracy of attack traffic in the datasets generally improved. In particular, the dataset validation accuracy increased by 12% and the test accuracy increased by 10%. These results prove that the method of continuously increasing the attack traffic using GAN can be an effective detection method for increasing accuracy in research environment cases with a small amount of attack traffic.

Table 2. Comparisons of attack traffic classification experiments using CNN

Performance	Before Accuracy	After Accuracy	10%		20%		30%	
Dataset			Precision	Recall	Precision	Recall	Precision	Recall
USTC-TFC	83–87	91–95	83.4	83.8	86.2	86.7	91.1	91.6
ISCX-2012	76–81	89–91	82.1	83	85.6	85.3	88.2	89.4
NSL–KDD	75–80	85–93	80.3	80.1	82.1	82.5	84.1	92.1

5 Conclusion

This study investigated a method to improve detection accuracy by creating similar traffic of GAN and increasing the ratio of attack traffic. The method presented in this paper contributed to solving the problem of performance degradation due to data imbalance in the data set using GAN. The existing datasets have an imbalanced data problem, which impedes the detection of attacks or increased learning rate. As a solution to this problem, we added learning data to similar traffic generated by the GAN algorithm and improved the accuracy in attack detection experiments. In order to create similar traffic, between 0.98 and 1 more images of the Generator was added to the training data. The experiment results showed that the increase of the created similar traffic improved the learning rate by 9–12%, and the detection accuracy for the attack traffic by approximately 13%. Furthermore, we were able to improve classification accuracy according to attack characteristics. In future, classification to determine the type of attack traffic and transformation of the generated images into text form using GAN will be researched.

References

1. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.: A detailed analysis of the KDD CUP 99 data set. Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) (2009)
2. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. **31**(3), 357–374 (2012). ISSN 0167-4048. https://doi.org/10.1016/j.cose.2011.12.012
3. Wang, W., et al.: Malware traffic classification using convolutional neural network for representation learning. In: 2017 International Conference on Information Networking (ICOIN). IEEE (2017)
4. Yao, H.P., Liu, Y.Q., Fang, C.: An abnormal network traffic detection algorithm based on big data analysis. Int. J. Comput. Commun. Control **11**(4), 720–733 (2016)
5. Rathore, M.M., Ahmad, A., Paul, A.: Real time intrusion detection system for ultra-high-speed big data environments. J. Supercomput. **72**(9), 3489–3510 (2016)
6. Tang, T.A., et al.: Deep learning approach for network intrusion detection in software defined networking. In: 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM). IEEE (2016)
7. Mylavarapu, G., Thomas, J., Ashwin Kumar, T.K.: Real-time hybrid intrusion detection system using apache storm. In: High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conference on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on. IEEE (2015)
8. Yu, Y., Long, J., Cai, Z.: Session-based network intrusion detection using a deep learning architecture. In: Modeling Decisions for Artificial Intelligence. Springer, Cham (2017)
9. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)
10. Tavallaee, M., et al.: A detailed analysis of the KDD CUP 99 data set. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009. IEEE (2009)
11. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: Proceedings of IEEE International Conference on Computer Vision (ICCV), Venice, Italy (2017)
12. DCGAN: Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)