Hindawi
Security and Communication Networks
Volume 2022, Article ID 2993963, 10 pages
https://doi.org/10.1155/2022/2993963

*Research Article*

# A Network Data Reinforcement Method Based on the Multiclass Variational Autoencoder

**Yanze Qu**,[1] **Hailong Ma**,[2] **Yiming Jiang**,[2] **Liang Wang**,[1] **and Jing Yu**[2]

[1]*Information Engineering University, Zhengzhou 450003, China*
[2]*National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450001, China*

Correspondence should be addressed to Yanze Qu; quyanze@foxmail.com

Received 14 December 2021; Revised 26 May 2022; Accepted 14 June 2022; Published 12 July 2022

Academic Editor: Tao Zhang



Anomaly detection models based on deep learning come up against difficulties on the deployment in real scenarios such as generalization problem. The performance of the model based on specific dataset is not as good as expected in other scenarios. In order to avoid this problem, it is a feasible solution to collect network data from the target environment to train the model. This paper proposes a network data reinforcement method based on the multiclass variational autoencoder to complete training tasks with little amount data. In this paper, anomaly detection models based on MLP and CNN are designed, respectively, and validation experiments are carried out on the CICIDS-2018 dataset. Compared with unreinforced models, models based on this method get faster convergence speed during training. During evaluation, models based on this method achieve an average accuracy of 93.69%, while unreinforced models only get an average accuracy of 55.63%. In addition, this method provides competitive results on insufficient data compared with those existing models on sufficient data.

## 1. Introduction

In order to deal with the increasing network threats, a variety of network security technologies have emerged, such as vulnerability detection technology and anomaly detection technology [1]. The network anomaly detection system (NADS) is one of the most popular network security measures in the field. It reduces the impact of network security events by monitoring abnormal behaviour in the network and linking with other security systems [2]. In recent years, the relevant technologies of the system have been deeply integrated with deep learning (DL), and a variety of network anomaly detection methods based on deep learning have been derived [3, 4].

However, in most of related papers, the implementation of DL solution is not performed in real network scenarios, and they normally show a proof of concept [5]. Generalization performance of models based on DL is the leading reason, which means that the model trained on dataset A cannot run as expected on dataset B. This problem gets worse in NADS based on DL for the following reasons.

(1) The lack of unified representation. The models are deeply coupled with data. For image, the N-D matrix has become a recognized choice. For text, word vector is widely adopted. For network, there is no widely accepted representation. In reference [6], the whole packet is regarded as the input of the model. But in reference [7], the statistics of network flow becomes the input. Even among public network datasets, there is a gap in the selection of features, such as CICIDS [8] and NSL-KDD [9].

(2) Environmental fingerprint effect. The detection model based on DL depends on data distribution. In different network environments, due to factors such as country, network scope, and network equipment, the data distribution will be quite different, thus contributing worse generalization performance.

In order to make the detection model achieve ideal performance in real network scenarios, there are some further studies in the field, such as transfer learning [10]. There is no doubt that the model trained with the data collected in the target environment gets closer to the ideal performance. However, due to the need to simulate a lot of network attacks and the technical barriers of network equipment, the collection of large-scale and high-quality network security dataset is difficult and costly.

In this paper, we propose a network data reinforcement method based on the multiclass variational autoencoder (MCVAE). The reinforcement method can be divided into two dimensions: quantity and quality. In terms of quantity, the data volume is expanded through the deep generation model, similar but not the same [11]. In terms of quality, the MCVAE can control the proportion of different classes by adjusting parameters, thus solving the unbalance problem in network data [12]. This method lowers data requirement of the network anomaly detection model based on DL and facilitates the training, making the rapid iteration of relevant work possible. Meanwhile, it is a feasible practice for few-shot learning.

The rest of the paper is organized as follows. Section 2 depicts the theory, involved technology, and implementation process of this method. This part focuses on the design and construction of MCVAE and detection models. Section 3 provides the dataset, experimental environment, and experimental results. Finally, the paper is concluded in Section 4.

## 2. Materials and Methods

The network anomaly detection system based on DL has gradually become the mainstream to sense cyberspace threats. Nowadays, there have been a variety of excellent implementations. In reference [13], HongYu Yang proposed a network intrusion detection model IBIDM based on improved convolutional neural network (IDNN), reaching 92.94% on precision and 0.76% on FPR in five-classification on NSL-KDD. In reference [14], Shone N et al. constructed a very successful network intrusion detection system using a stacked network based on the nonsymmetric deep autoencoder (NDAE) as a feature extractor, and achieved 87.37% F-score, 85.42% recall rate, and 100% precision rate in five-classification tasks on NSL-KDD. Based on autoencoder, Bovenzi G et al. proposed a hierarchical hybrid intrusion detection approach in IoT scenarios, getting advantage on $F1$-score than the traditional method [15]. To avoid ambiguity, the above metrics' expression is shown in equation (1)–(4) and the meaning of important parameters is provided in Table 1.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{1}$$

$$\text{FP Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3}$$

Table 1: The meaning of TP, FP, TN, and FN in this paper.

| Name | Meaning |
|---|---|
| True positive (TP) | The number of malicious samples classified as malicious |
| False positive (FP) | The number of benign samples classified as malicious |
| True negative (TN) | The number of benign samples classified as benign |
| False negative (FN) | The number of malicious samples classified as benign |

$$F - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{4}$$

Although a series of research and experiment shows that this technology is mature enough, its deployment in real scenarios grapples with problems [16]. Different from image data and text data, network security data gets some characteristics. For image, the RGB value of pixels is usually regarded as features, and the $N$–$D$ matrix is a recognized choice. For text, word vector encoded by various embedding algorithms is widely adopted. However, network data gets more various, and there is no suitable feature engineering for most tasks. The complexity is getting worse with the development of novel network-related applications. In addition, the environmental fingerprint effect of network security data greatly affects the migration and application of model in the network field and limits the practical deployment of related technologies.

In the case that traditional methods are used to enhance the generalization performance such as dropout [17] and transfer learning [18] which do not work well in the network field, in order to make full use of the advantages of deep learning, it is a feasible scheme to collect the training data in the target environment where the system is to be deployed. This paper utilizes the deep generative model to reduce the workload of data collection and complete data reinforcement and organically combines the generative model with the recognition model to complete the construction of an anomaly detection model. The construction process is shown in Figure 1.

*2.1. Data Collection.* In order to facilitate reproduction, CICIDS-2018 [8] is selected for experiments. We are aware of the limitations of these public dataset, but CICIDS-2018 remains widely used benchmarks amongst these works, enabling us to draw credible conclusion. CICIDS-2018 gets traffic diversity and sufficient volumes, providing both feature set and metadata. Meanwhile, they have identified eleven criteria [19] that are necessary for building a reliable benchmark dataset. The details of experiment dataset will be shown in the section Results and Discussion.

*2.2. Data Preprocessing.* In order to avoid the influence of different value ranges among features, a dimensionless method is necessary. Min-Max scaling, shown as equation (5), is widely adopted because of the outstanding performance in terms of image data [20]. All pixels of image have a unified value range [0–255]. Through linear scaling, Min-Max scaling can efficiently fulfil data preprocessing. However, Z-score

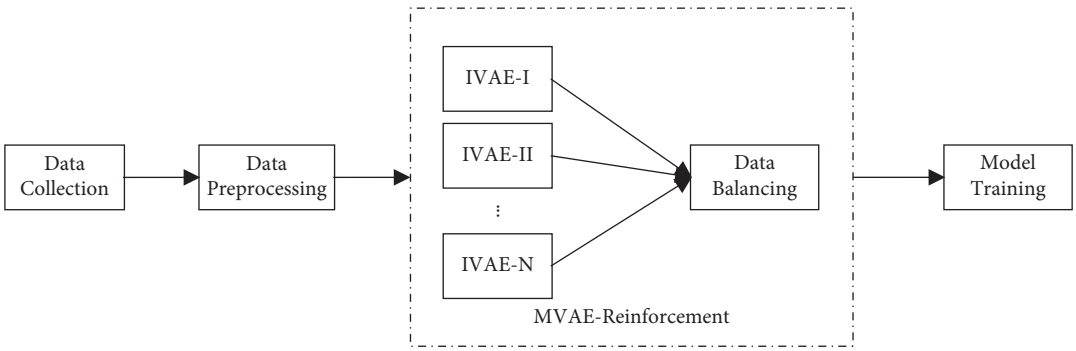


Figure 1: The construction process. IVAE-N means improved variational autoencoder for class-N.

```
def pre_processing (phase):
    if (phase = = 'before MCVAE'):
        data = get_data() # get the dataset to be processed
        data = features_selected (data, features) # select features uses for training
        data = z_score (data) # scale data with z_score algorithm
    elif (phase = = 'before Model Training'):
        data = label_encode (data) # map labels to 0 (benign) and 1 (nonbenign)
    else:
        print ('error') # if the function is used in other phases, an
        return 0 error will return
    return data # output the processed dataset
```

Algorithm 1: Data Preprocessing.

scaling, shown in equations (6)–(8), may become a more suitable algorithm in network for the following reasons.

(1) Z-score scaling is nonlinear, reducing the loss of information carried by raw data during the operation.

(2) Z-score scaling is compatible with outliers. For network data, the value range among features varies greatly. Under the circumstance, the impact of outliers on Min-Max scaling is destructive.

$$x_{\text{new}} = \frac{x_{\text{old}} - x_{\min}}{x_{\max} - x_{\min}}, \tag{5}$$

$$x_{\text{new}} = \frac{x_{\text{old}} - \text{mean}}{s}, \tag{6}$$

$$\text{mean} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{7}$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \text{mean})}. \tag{8}$$

In addition, the training phase in Figure 1 focuses on binary classification, that is, to judge whether the network behaviour is benign or not. The labels on CICIDS-2018 are various. Therefore, label encoding is a necessary step before the training phase, mapping all nonbenign labels to malicious ones. The overall pseudocode of data preprocessing is shown in Algorithm 1.

*2.3. MVAE-Reinforcement.* In 2012, Krizhevsky achieved great success in the field of approximating functions using deep learning and made a great performance breakthrough in ImageNet dataset, namely, Alexnet [21]. Then, deep learning has ushered in a new upsurge of development.

The generative model is a novel area of deep learning which learns to approximate distribution, defined data in high-dimensional space [22]. Variational autoencoder (VAE) is one of the most recognized generative models, which can perform efficient approximate inference and learning for intractable distributions based on Variational Bayes (VB) [23]. In order to effectively reinforce data, more examples that are like those already in training datasets is in demand but not the same ones. For example, if several birch trees are the input of the model, a birch forest with diversity is expected to output.

According to latent variable theory, network data can be defined in a high-dimensional latent space **Z**, as the following equations

$$\begin{aligned} &\text{Data field: } X = \{x_i\}_{i=1}^{N}, \\ &\text{Latent field: } Z = \{z_i\}_{i=1}^{M}. \end{aligned} \tag{9}$$

$$x_i = f(z_1, \ldots, z_M), x_i \sim P_\vartheta(x|z), z_j \sim P_\theta(z). \tag{10}$$

VAE utilizes deep neural network to infer the intractable distribution of latent variable $z$ and generates target data by randomly sampling on the distribution. Its basic structure is shown in Figure 2. The model adopts encoder–decoder architecture, in which the encoder layer approximates the

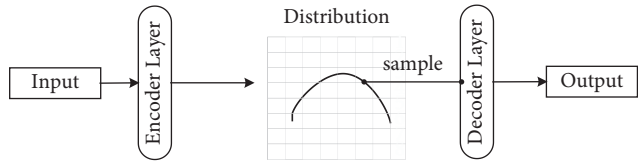


Figure 2: Basic structure of VAE.

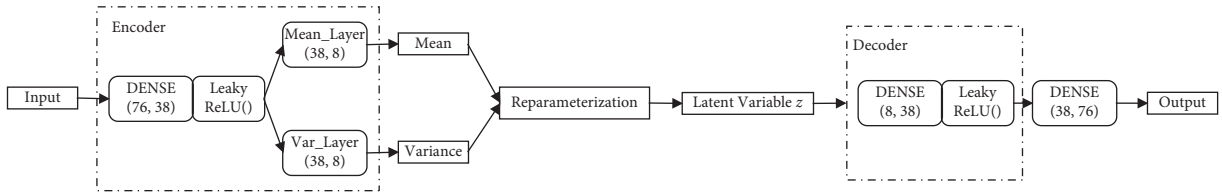


Figure 3: Basic structure of the IVAE.

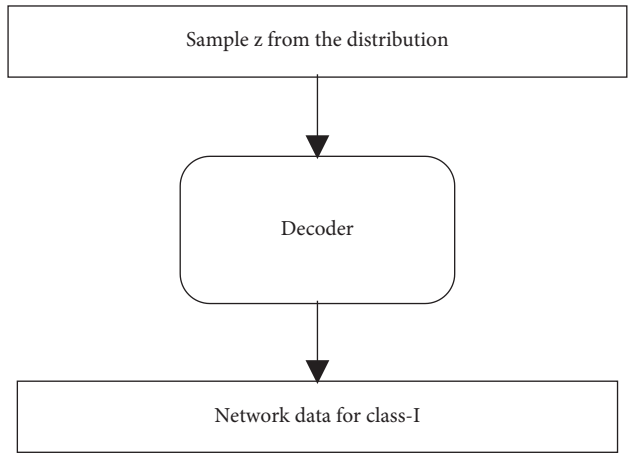


Figure 4: Data generator.

distribution of latent variable and the decoder layer is responsible for inferring the data defined by latent variable.

At present, VAE has been successfully applied in some fields, and its mathematical principle and generation effect are recognized [24]. Based on the theory of VAE, this paper improves traditional VAE and proposes improved variational autoencoder (IVAE) for network data. The difference lies in the selection and connection of layers. Meanwhile, a dense layer is connected after the decoder layer to realize the data generation without constraints such as sigmoid [25]. The details of IVAE are shown in Figure 3.

Among them, the function of encoder is to reduce feature dimensions and approximate the distribution of latent variable $z$ via inferring statistical mean and variance. A decoder is responsible to generate target data according to the randomly sampled $z$. Reparameterization is a trick to realize sampling from distribution, as shown in equation (10).

$$z = \text{mean} + \sigma * \sqrt{\text{variance}}, \quad \sigma \varepsilon N(0, I). \tag{11}$$

The goal of the model is to obtain approximate data in the same space as original data. In order to achieve this goal, the loss function, as shown in equation (10), is composed of reconstruction loss (equation (11)) and KL loss (Equation (12)). The reconstruction loss, mean square error function (MSE), is to calculate the difference between the input and output. The KL loss is to measure the distance between the distribution of the latent variable $z$ fitted by the encoder and the standard normal distribution. There is a hyperparameter before the KL loss term to get a balance between the two loss functions and prevents the occurrence of unilateral convergence. In this paper, the hyperparameter is 1, referring the research [23]

$$\text{Loss} = \text{Reconstruction Loss} + \text{weight} * \text{KL Loss}, \tag{12}$$

$$\text{Reconstruction Loss} = E(\text{output} - \text{input})^2. \tag{13}$$

$$\text{KL Loss} = -\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log\left(\left(\sigma_j^{(i)}\right)^2\right) - \left(\mu^{(i)}\right)^2 - \left(\sigma_k^{(i)}\right)^2\right). \tag{14}$$

After the training, the whole architecture is truncated. The encoder is discarded, and the generation of network data

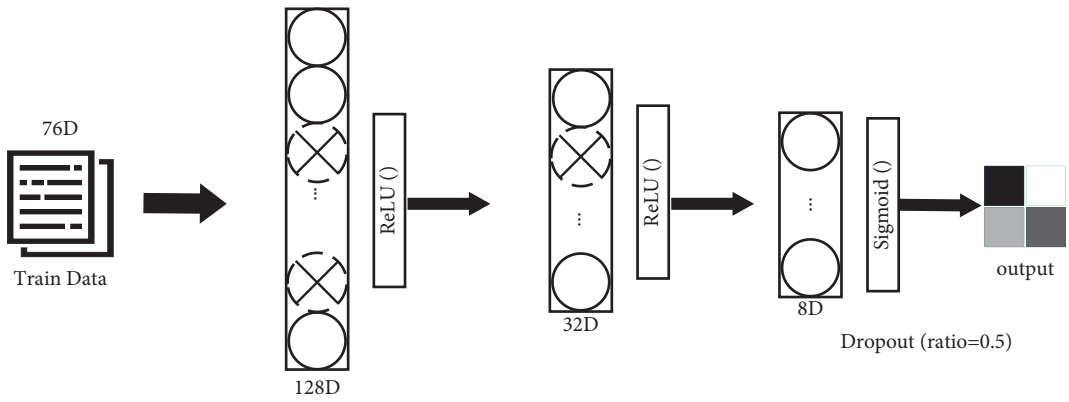


Figure 5: The design of neural network based on MLP.

is realized by randomly sampling on distribution and decoding, as shown in Figure 4. The generation amount of various data can be adjusted as needed, thus balancing the proportions of different classes and completing data reinforcement.

According to different purposes and attack methods, network behaviour can be classified into different classes. MVAE-reinforcement constructs specific IVAE for various network behaviour, thus gaining excellent perception to all the kinds of behaviour. Our method effectively prevents the confusion of network security data, avoids the problem of data unbalance [26], and provides a feasible solution for few-shot learning.

*2.4. Model Training.* In order to enhance the persuasion of the experiment, this paper designs the corresponding neural networks based on two widely used algorithms, multilayer perception (MLP) and convolutional neural network (CNN). In this paper, network data is trained in the form of feature set, which belong to tabular data. MLP gets a good learning ability for tabular data, which is an important reason for choosing it as the benchmark model. As for CNN, it stands out in dimension compression and feature extraction. Meanwhile, it realizes the sharing of learning parameters through convolution kernel, greatly reduces the number of parameters of the model, and improves the generalization performance of the model [27]. The special channel mechanism makes this network structure have the ability to learn multiple potential patterns from the same local information.

The design of MLP in this paper is designed as Figure 5, and the structure of CNN-based neural network is shown in Figure 6. The other hyperparameters used in these two models are the same, as shown in Table 2.

## 3. Results and Discussion

In order to verify the effect of MCVAE-reinforcement, we design validation experiments as shown in Figure 7.

Based on raw data and reinforced data, the same DL algorithm is used to train anomaly detection models. Then, these models are evaluated by the same dataset to quantify their performance. Finally, the effectiveness of the reinforcement method is proved by comparative analysis. At the same time, in order to improve the reliability of the experiment, this paper conducts experiments, respectively, based on two algorithms, MLP and CNN, to eliminate the specificity brought by the selection of algorithms and draws a universal conclusion.

All experiments were conducted in the following environments, TensorFlow 2.4.0, keras-applications 1.0.8, keras-preprocessing 1.1.2, and scikit-learn 1.0.1.

*3.1. Experiment Data.* The experiment in this paper is based on CICIDS-2018 [8], which is developed by the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC), including a variety of attack scenarios, which has high authenticity and reference value. In the CSV-format data, the research team divides network traffic data by Src IP, Dst IP, Src Port, Dst Port, Protocol.

This paper mainly focuses on Friday-16-02-2018.csv. In the dataset, there are three kinds of network behaviours: normal behaviour (benign), DoS attacks based on slow HTTP test (DoS attacks-SHT), and Hulk-DoS attacks (DoS attacks-H).

There are four kinds of datasets needed during experiments. One hundred thousand items are randomly selected from Friday-16-02–2018.csv as A-Data and another one hundred thousand items as evaluation data. After that, a thousand items of each category in A-Data are randomly selected as raw data. Reinforced data are a combination of raw data and the data generated by MCVAE.

There are 3000 rows data in raw data, including 1000 rows data labelled as benign, 1000 rows data labelled as DoS attacks-SHT, and 1000 rows labelled as DoS attacks-H. The dataset aims at simulating the insufficient dataset collected in real scenarios due to insurmountable difficulties. Based on the dataset, IVAE-Benign, IVAE-DoS attacks-SHT, and IVAE-DoS attacks-H are trained. Through MVAE-reinforcement, 30000 rows data are generated, respectively, for the three kinds of network behaviour. After mixing with raw data, reinforced data is ready. The detail of these datasets is shown in Table 3. After MCVAE-reinforcement, Algorithm 1

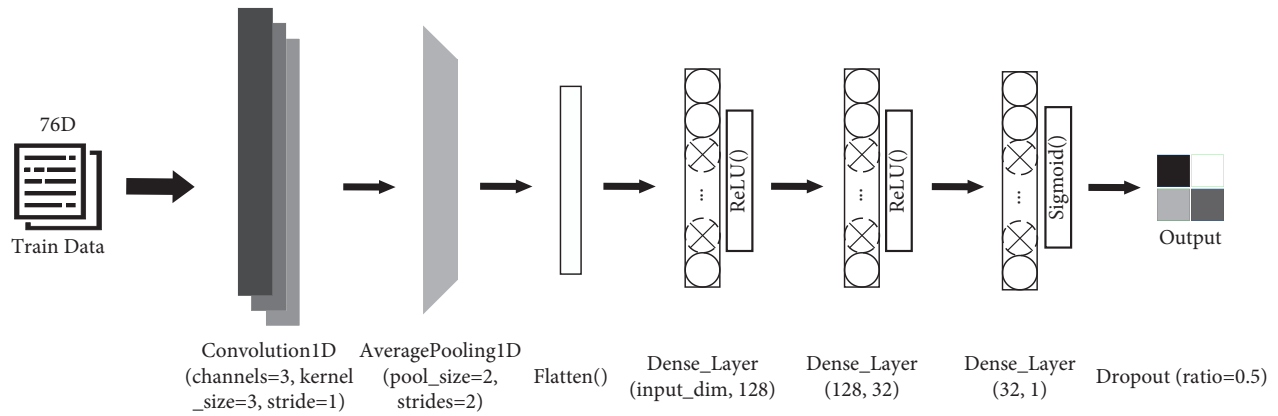


Figure 6: The design of neural network based on 1D-CNN.

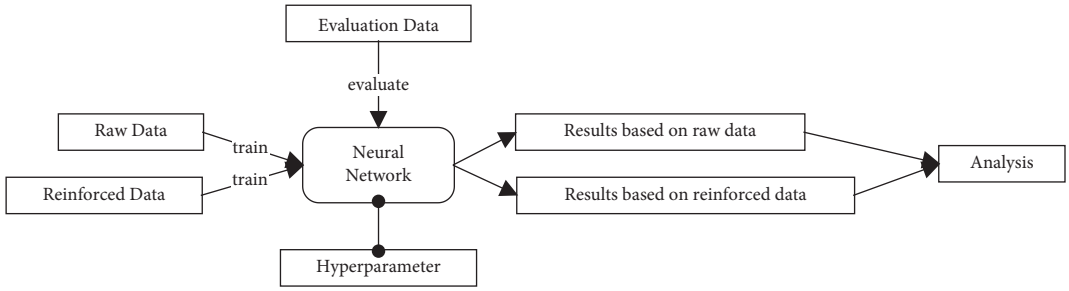


Figure 7: The design of experiments.

Table 2: Some hyperparameters used in model training.

| Hyper parameter | Value |
|---|---|
| Loss function | Binary cross entropy (BCE) |
| Optimizer | Adam [28] |
| Validation mothed | 5-Fold cross validation |
| Validation ratio | 20% |
| Batch size | 64 |

Table 3: The details of experiment datasets.

| Name | Num of benign | Num of DoS attacks-SHT | Num of DoS attacks-H | Total num |
|---|---|---|---|---|
| A-data | 42,814 | 13,300 | 43,886 | 100,000 |
| Evaluation data | 42,553 | 13,355 | 44,092 | 100,000 |
| Raw data | 1,000 | 1,000 | 1,000 | 3,000 |
| Reinforced data | 31,000 | 31,000 | 31,000 | 93,000 |

is needed to complete label encoding before model training, mapping benign to "0", and mapping the others to "1."

*3.2. Experiment Result.* In order to verify the effectiveness of the method proposed in this paper, two kinds of anomaly detection models (ADM) are trained, respectively, on raw data and reinforced data, with the same algorithm, loss function, and optimizer. The training environment of the models is consistent except dataset. After two epochs, the ADM on reinforced data has reached convergence. The metrics at this time are shown in Table 4. After five epochs, the ADM on raw data has reached convergence. The results are shown in Table 5. From these results, it can be seen that MCVAE-reinforcement can greatly shorten training time. In this experiment, the ADM based on MCVAE-reinforcement

Table 4: The metrics of ADMs in the end of epoch 2.

| Dataset | Model | Accuracy (%) | Loss | Val_Accuracy (%) | Val_Loss |
|---|---|---|---|---|---|
| Raw data | ADM (MLP) | 95.54 | 0.1054 | 94.67 | 0.3213 |
| | ADM (CNN) | 90.38 | 0.3417 | 95.01 | 0.1551 |
| Reinforced data | ADM (MLP) | 97.93 | 0.0422 | 97.11 | 0.0345 |
| | ADM (CNN) | 99.99 | 0.0017 | 91.17 | 0.8283 |

Table 5: The metrics of ADMs in the end of epoch 5.

| Dataset | Model | Accuracy (%) | Loss | Val_Accuracy (%) | Val_Loss |
|---|---|---|---|---|---|
| Raw data | ADM (MLP) | 96.83 | 0.0700 | 99.67 | 1.1841e-04 |
| | ADM (CNN) | 99.67 | 0.0310 | 99.67 | 0.0203 |
| Reinforced data | ADM (MLP) | 98.98 | 0.0327 | 97.16 | 0.7763 |
| | ADM (CNN) | 99.99 | 0.0010 | 95.38 | 0.9995 |

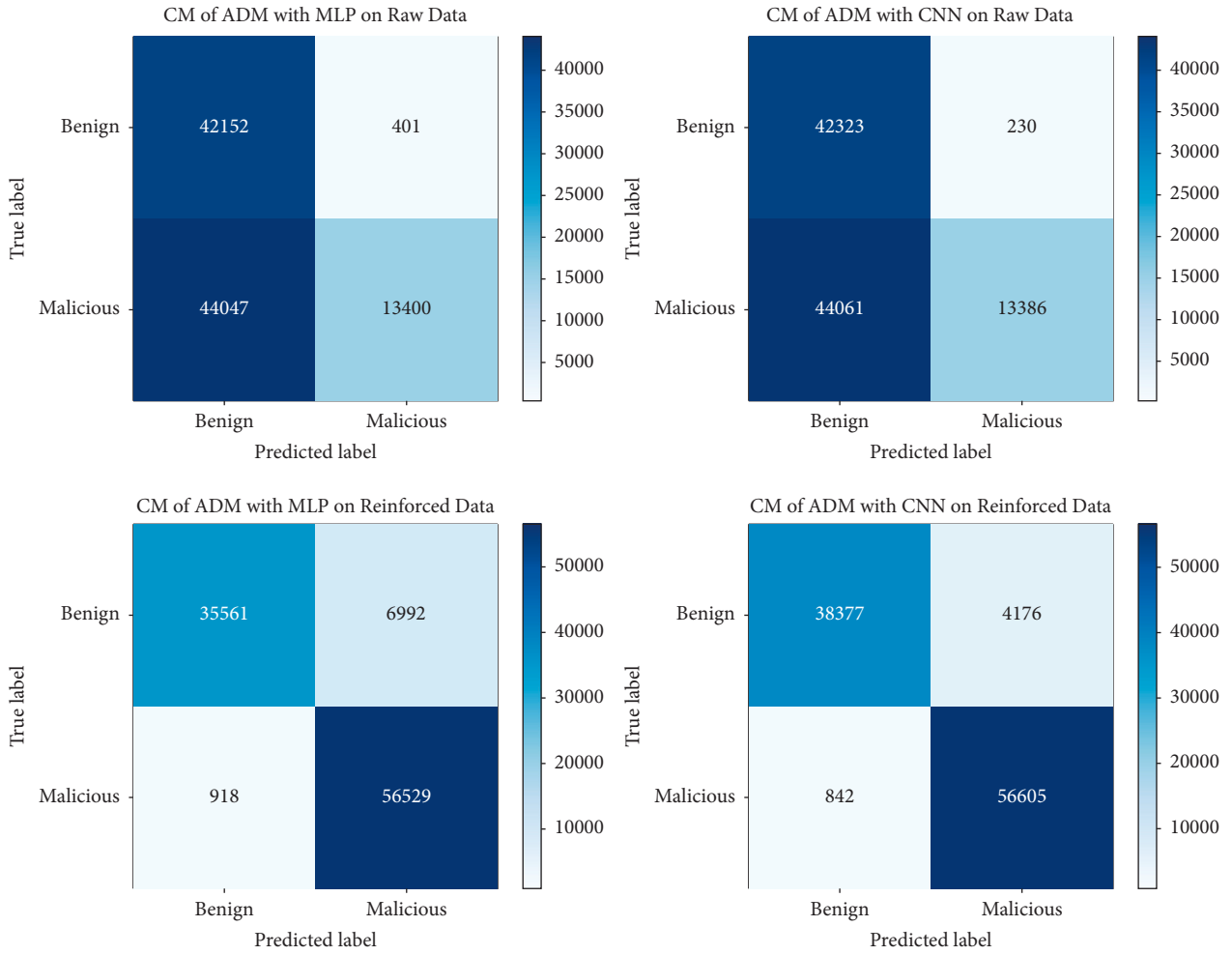


Figure 8: Confusion matrixes during the evaluation stage.

Table 6: The metrics of ADMs during the evaluation stage.

| Dataset | Model | Accuracy (%) | Precision (%) | Recall (%) |
|---|---|---|---|---|
| Raw data | ADM (MLP) | 55.55 | 48.90 | 99.06 |
| | ADM (CNN) | 55.71 | 48.99 | 99.46 |
| Reinforced data | ADM (MLP) | 92.09 | 97.48 | 83.57 |
| | ADM (CNN) | 95.29 | 97.86 | 90.90 |

Table 7: The metrics of ADMs on A-Data and raw data.

| Model | ADM on reinforced data | ADM on A-Data | SVM on A-Data |
|---|---|---|---|
| Loss convergence during training | 0.0010 | 0.0162 | — |
| Accuracy convergence during training | 99.99% | 99.77% | — |
| Accuracy during evaluation | 95.29% | 99.75% | 99.62% |
| Precision during evaluation | 97.86% | 99.84% | 99.78% |
| Recall during evaluation | 90.90% | 99.59% | 99.59% |
| Training time | 9.82 s | 9.43 s | 184.90 s |
| Evaluation time | 2.25 s | 2.28 s | 6.11 s |

takes half the time of ADM without it to reach convergence. And in Table 4, ADMs on reinforced data get better performance than ADMs on raw data. Finally, when all ADMs reach convergence, it can be seen that the performance of ADMs on raw data is getting closer with those on reinforced data. This phenomenon may be due to the amount of raw data. Under few training items, models find a shortcut after a large number of iterations, and then get a deceptive performance.

For further evaluating the performance of ADMs with MCVAE-reinforcement and without it, these ADMs are evaluated on evaluation data. The dataset is collected at the same time period as raw data. And there are no duplicate items with raw data to evaluate the performance and generalization ability of these models. The metrics during evaluation are shown in Table 6 and the confusion matrixes are listed in Figure 8.

As can be seen from the results, ADMs based on raw data are incompetent for anomaly detection, with an average accuracy of 55.63%. Meanwhile, ADMs based on reinforced data have achieved an average accuracy of 93.69%, proving the effectiveness of MCVAE-reinforcement.

In addition, there is a comparative experiment among the ADMs on sufficient training data and the ADM with MCVAE-reinforcement on insufficient training data. In this experiment, A-Data is selected as the sufficient training data, whose detail is shown in Table 3. And raw data is selected as the insufficient training data. 1D-CNN is assigned as the baseline model for ADMs based on deep learning, with the same parameters as before. And the support vector machine (SVM) is adopted as the representative of the traditional algorithm. The results are shown in Table 7.

As can be seen from the table, with MCVAE-reinforcement, the ADM on insufficient data gets similar performance to ADMs on sufficient data. Meanwhile, it can continue the advantages of the training algorithm, getting great improvement on running time than traditional methods.

## 4. Conclusions

As an excellent tool technology, due to its strong learning ability and excellent portability, deep learning has been widely and successfully applied in many fields, such as Faster R-CNN [29] for image and Bert [30] for natural language processing. How to effectively apply it to solve network security difficulties is a research hotspot.

Meanwhile, deep learning usually has high requirements for training datasets, which is mainly reflected in two aspects. The first is the requirement for the quantity of data. If the amount of training data is insufficient, it will lead to poor performance. The second is the requirements for the quality of the data. The training dataset is expected to be well-constructed. In order to solve this problem, this paper utilizes MCVAE-reinforcement to assist the training of network intrusion detection model and organically combines the deep generation model and the deep recognition model, which provides a reference for the application of deep learning in network security production practice. Experiments show that the model training method proposed in this paper can effectively improve the convergence speed, detection performance, and generalization performance of the model. MCVAE-reinforcement can complete the construction of high-precision detection model with a little amount of training samples.

There are some shortcomings in this paper, that is, the ability of VAE depends on the proper data distribution in sampled dataset. If the data in the sampled dataset cannot effectively represent the common characteristics of the corresponding network behaviour, it will affect the quality of the generated data and lead to the decline of the performance of ADMs. As for future direction, applying theoretical achievements related with generative adversarial nets [31] may further improve the model. In addition, there are some studies aimed at enhancing the interpretability and reliability of deep learning, such as explainable artificial intelligence (XAI) techniques [32],

which can be beneficial to optimize performance of VAE-based network.

## Data Availability

All data used to support the findings of this study are available from reference [8].

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This study was supported by the National Key R&D Program of China (2020YFB1806402).

## References

[1] J. P. Anderson, "Computer security threat monitoring and surveillance," Technical report, Washington, DC, USA, 1980.

[2] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, no. 3, pp. 447–489, 2019.

[3] M. Rabbani, Y. Wang, R. Khoshkangini et al., "A review on machine learning approaches for network malicious behavior detection in emerging technologies," *Entropy*, vol. 23, no. 5, p. 529, 2021.

[4] Y. Mirsky, T. Doitshman, Y. Elovici, and S. Asaf, "Kitsune: an ensemble of autoencoders for online network intrusion detection," 2018, https://arxiv.org/abs/1802.09089.

[5] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and A. Jose, "Towards the deployment of machine learning solutions in network traffic classification: a systematic survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2018.

[6] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 43–48, Beijing, China, July 2017.

[7] J. Kim, N. Shin, S. Y. Jo, and H. Sang, "Method of intrusion detection using deep neural network," in *Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 313–316, Jeju Island, South Korea, February 2017.

[8] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, 2018.

[9] M. Tavallaee, E. Bagheri, W. Lu, and A. Ali, "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the 2009 IEEE symposium on computational intelligence for security and defense applications*, pp. 1–6, IEEE, Ottawa, Canada, July 2009.

[10] X. Hu, C. Gu, Y. Chen, and F. Wei, "CLD-net: a transfer learning internet encrypted traffic classification scheme based on convolution neural network and long short-term memory network," in *Proceedings of the 2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pp. 1–5, IEEE, Beijing, China, October 2021.

[11] R. Salakhutdinov, "Learning deep generative models," *Annual Review of Statistics and Its Application*, vol. 2, no. 1, pp. 361–385, 2015.

[12] Y. L. Murphey, H. Guo, and L. A. Feldkamp, "Neural learning from unbalanced data," *Applied Intelligence*, vol. 21, no. 2, pp. 117–128, 2004.

[13] H.Yu Yang and F.Y. Wang, "Network intrusion detection model based on improved convolutional neural network," *Journal of Computer Applications*, vol. 39, no. 9, pp. 2604–2610, 2019.

[14] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[15] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, "A hierarchical hybrid intrusion detection approach in IoT scenarios," in *Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–7, Taipei, Taiwan, October 2020.

[16] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1563–1572, Las Vegas, NV, USA, June 2016.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, and S. Ilya, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] R. Chen, S. Zhang, D. Li, Y. Zhang, F. Guo, and W. Meng, "Logtransfer: cross-system log anomaly detection for software systems with transfer learning," in *Proceedings of the 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, pp. 37–47, IEEE, Coimbra, Portugal, October 2020.

[19] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. Ali, "An evaluation framework for intrusion detection dataset," in *Proceedings of the 2016 International Conference on Information Science and Security (ICISS)*, pp. 1–6, IEEE, Pattaya, December 2016.

[20] A. Singh, K. Gaurav, A. K. Rai, and Z. Beg, "Machine learning to estimate surface roughness from satellite images," *Remote Sensing*, vol. 13, no. 19, p. 3794, 2021.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[22] C. Doersch, "Tutorial on variational autoencoders," 2016, https://arxiv.org/abs/1606.05908.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, https://arxiv.org/abs/1312.6114.

[24] T. Salimans, D. Kingma, and M. Welling, "Markov chain Monte Carlo and variational inference: bridging the gap," 2015, https://arxiv.org/abs/1410.6460.

[25] R.-J. Zhang, W. Chen, H. Meng-xin, and L. F. Wu, "Detection of abnormal flow of imbalanced samples based on variational auto-encoder," *Computer Science*, vol. 48, no. 7, pp. 1–13, 2021.

[26] S. J. Jian, Z. G. Lu, D. Du, B. Jiang, and B. X. Liu, "Overview of network intrusion detection technology," *Journal of Cyber Security*, vol. 5, no. 4, pp. 96–122, 2020.

[27] K. O'shea and R. Nash, "An introduction to convolutional neural networks," 2015, https://arxiv.org/abs/1511.08458.

[28] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[30] J. Devlin, M.-W. Chang, K. Lee, and T. Kristina, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, and B. Xu, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[32] A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, "XAI meets mobile traffic classification: understanding and improving multimodal deep learning architectures," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4225–4246, 2021.