Hindawi
Security and Communication Networks
Volume 2022, Article ID 9509417, 9 pages
https://doi.org/10.1155/2022/9509417

Research Article

CRND: An Unsupervised Learning Method to Detect Network Anomaly

YanZe Qu,[1] **HaiLong Ma**,[2] **and YiMing Jiang**[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 HaiLong Ma; longmanclear@163.com

Received 4 September 2022; Revised 11 October 2022; Accepted 17 October 2022; Published 28 October 2022

Academic Editor: Wenjuan Li

Network anomaly detection system (NADS) is one of the most important methods to maintain network system security. At present, network anomaly detection models based on deep learning have become a research hotspot in the area because of their advantage in processing high-dimensional data and excellent performance on detecting anomaly. However, most of the related research studies are based on supervised learning, which has strict requirements for dataset such as labels with high accuracy. However, there are some difficulties in obtaining a large amount of data with complete label message, thus seriously hindering the development and deployment of NADS based on DL. In this paper, we propose an unsupervised learning method to detect network anomaly, contrastive representation for network data (CRND). Based on contrastive learning, without label message, a qualified model is trained, providing more possibilities for the field. On CICIDS2018, the evaluation experiment proves that CRND can achieve 96.13% accuracy with only 200 items, and its $F1$-score reaches 0.96, which is far higher than that of other existing unsupervised learning methods. As fine-tuning is carried out, $F1$-score can reach a convergence level of 0.99, and the detection performance is the same as that of the detection model based on supervised learning.

1. Introduction

According to the 2021 China Internet Security Report [1], the number of network attacks monitored in 2021 has increased significantly compared with that in 2020, including 60% year-on-year increase in DDoS attacks and 241% year-on-year increase in web attacks, three times as many APT attacks as in 2020, and other attacks have also increased to different degrees. These statistics show that network attacks are increasing day by day. How to detect network anomaly efficiently and accurately is an important issue to ensure network security.

With the popularity of network applications and the continuous increase of network users, the current network environment gets large amount of data and is evolving at high speed. In order to cope with network characteristics of the new era, network anomaly detection systems based on deep learning have become a hotspot in the field [2]. However, most of the existing methods in the field are based on supervised learning, which has requirements for the structural characteristics of datasets, that is, the datasets need to be annotated manually. At the same time, deep learning also has a requirement for the data volume. Insufficient data lead to the models with poor performance. Building a well-labelled dataset with sufficient data capacity is costly and error prone, requiring a lot of human labour and time, which greatly hinders the development and iteration of network anomaly detection models based on deep learning [3]. Meanwhile, the network behaviour is increasingly diverse, and new malicious behaviours emerge endlessly. The development process of network anomaly detection model based on supervised learning is too lengthy to adapt to the evolution speed of the current network environment.

Unsupervised learning/self-supervised learning can avoid the cost of building large-scale and well-structured datasets, reduce the workload to get deep learning-based models, and shorten the development cycle. Self-supervised learning is a subclass of unsupervised learning, which completes model training by using a self-defined pseudo-label as a supervised signal. Unsupervised learning focuses on representation learning, which aims at learning an efficient, accurate, and universal potential representation. It is often used to construct pretrained models, which is the key support for the convenience and industrialization of models based on deep learning. At present, unsupervised learning has achieved great success in the area of natural language processing and image recognition, such as GPT [4, 5], BERT [6], and the contrastive learning framework MoCo [7]. These achievements have brought revolutionary changes to their respective areas.

Comparative learning, one of the mainstream methods in the area of unsupervised learning [8], gets the potential representation of data by learning the differences before the samples in the form of dynamic dictionary query. Based on the idea of comparative learning, this paper proposes an unsupervised representation learning method of network data, contrastive representation for network data (CRND). In the evaluation on CICIDS2018, our model gets 96.13% on accuracy with only 200 rows of data, and the detection performance is equal to or even better than that of the traditional supervised learning method. The contributions of this paper can be summarized as follows:

(1) We apply comparative learning method to the area of network security, and an unsupervised learning method for network data representation is proposed. On this basis, a network anomaly detection model is constructed, which avoids the huge cost of constructing a large-scale training dataset annotated by humans.

(2) We design a data augmentation algorithm based on autoencoder (AE), named as sparse augmentation network (SAN), to serve as the transformer in contrastive learning framework, which can provide a referable choice for data augmentation methods in nonimage data.

(3) We propose the model CRND, which is trained on data without label message. In the target environment, with CRND, only a small amount of data is needed for fine-tuning, and an excellent detection model can be obtained, providing a method to construct pretrained model for network researchers.

2. Related Works

The work of this paper mainly involves network anomaly detection and contrastive learning. Next, related works in the two areas are reviewed.

2.1. Network Anomaly Detection. Network anomaly detection is an important topic in network security. In recent years, influenced by the success of deep learning [9] in the area of image [10] and natural language processing [11], the network anomaly detection models based on deep learning have become a research hotspot.

In [12], a model based on long short-term memory (LSTM) is proposed, which is trained on the normal network traffic data. The model judges whether the network environment is abnormal by comparing the predicted value of the model for the next state with the true value. In [13], Yang and Wang proposed an intrusion detection model IBIDM based on improved convolutional neural network (ICNN), reaching 92.94% on accuracy in the five classification tasks of NSL-KDD. In [14], Shone et al. constructed a deep anomaly detection model based on nonsymmetric deep autoencoder (NDAE), which achieved 87.37% $F1$-score and 100% accuracy on the five classification tasks of NSL-KDD dataset. Kim et al. [15] built an anomaly detection model based on multilayer perceptron (MLP) with statistical network security data. On 10% KDD CUP99, it has achieved 99.3% on accuracy and 0.12% on false-alarm rate. To avoid ambiguity, the calculation method of the abovementioned metrics is shown in equations (1)–(3), and the meaning of variables in the equations is shown in Table 1.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{1}$$

$$\text{F1} - \text{score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \tag{2}$$

$$\text{false} - \text{alarm} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{3}$$

It can be seen that network anomaly detection models based on deep learning have reached a satisfactory level in detection performance. It is worth noting that most of the current mainstream methods are based on supervised learning, and these methods rely heavily on large-scale well-labelled data. In [3], the authors reviewed more than thirty papers related to machine learning published in top-level conferences or journals in the area of network security in the past decade and summarized the pitfalls of the existing network anomaly detection models based on machine learning. Among them, the problem of data collection and labelling is the one that exists in most of the studies and gets potentially devastating effects. In addition to the workload and cost, data acquisition and labelling are also faced with difficulties such as sampling bias and label inaccuracy. Meanwhile, supervised learning itself also faces potential dangers such as generalization error, spurious correlations, and adversarial attacks [16]. Therefore, building a deep learning model based on unsupervised learning method to avoid data problems is a research direction with prospect, innovation, and application value.

2.2. Contrastive Learning. The goal of contrastive learning is to gather samples sharing the same type and discrete those with different types. The difference between contrasting learning and other algorithms mainly lies in pretext task and loss function, and there is a correlation between them. From the two perspectives, we will review the previous research.

Pretext task is usually not the original target of models, but it can help models better complete the target task. Their proposal can enable models to learn a data representation that

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

can efficiently serve the final purpose, such as denoising autoencoder (DAE) [17], instance discrimination [18], and so on. The contrastive learning method proposed in this paper is based on instance discrimination, as shown in Figure 1. The data after data augmentation are regarded as a positive example of the original sample, and other samples in the same batch are regarded as negative examples. After that, the model will learn to distinguish the difference between positive and negative examples. In order to achieve this effect, it is necessary to define the corresponding loss function.

Different pretext tasks need to define corresponding loss functions. The loss functions of contrastive learning can be divided into three categories: predictive loss function, contrastive loss function, and adversarial loss function. The predictive loss function is a relatively common loss function, which aims at measuring the distance between the output and the established facts, such as the reconstruction error in [17] and the cross entropy in [19]. The contrastive loss function [20] is mainly used to measure the similarity between pairs of samples in the feature space. Taking instance discrimination [18] as an example, the loss function needs to have the ability to comprehensively consider the distance between the original sample and the positive sample (O-P distance) and the distance between the original sample and negative sample (O-N distance). With O-P distance decreasing or O-N distance increasing, the loss should be smaller, such as NCE [21]. The adversarial loss function [22] mainly calculates the difference between probability distributions, and the literature [23] summarizes and reviews its related research.

3. Method

3.1. Overall Architecture. The goal of this research is to avoid the cost of collecting data with complete labels, thus obtaining a reliable network anomaly detection model conveniently. In order to achieve this goal, this paper completes the representation learning of network security data on the unlabelled data $X = \{x_0, x_1, \ldots, x_n\}$ by using the contrastive learning method with instance discrimination as pretext task, converts the network security data into low-dimensional feature vectors $f_\theta(x_i) \in \mathbb{R}^d$, and then constructs CRND. Finally, in the target network environment, a reliable model can be obtained by fine-tuning. The overall flow is shown in Figure 2.

Based on contrastive learning, the architecture of CRND is shown in Figure 3. The design of each part will be described in detail in the following.

According to [7], the contrastive learning problem can be modelled as a dictionary look-up task. A certain feature vector k_i in Q is regarded as a query vector, and other vectors $\{k_0, \ldots, k_{i-1}, k_{i+1}, \ldots, k_{\text{batchsize}-1}\}$ in the set and the feature vectors $\{d_0, \ldots, d_{\text{batchsize}-1}\}$ in D are regarded as the keys of a dictionary. Consider k_i as the feature vector of data i and d_i as the feature vector of data i', which is the version of data i after augmentation. Then, the model can be regarded as a query for d_i that matches with k_i in the dictionary. The goal is to make k_i and d_i as similar as possible and make k_i as different as possible from other vectors, and then the model gets the ability to distinguish these samples and completes the task of representation learning.

3.2. Loss Function Based on Softmax. Under the above problem modelling, the loss function should meet the following requirements. The closer k_i is to d_i, the less the loss should be. Meanwhile, the more different k_i is from other vectors, the less the loss should be. In order to meet this requirement, with similarity measured by dot product, on the basis of NCE [21], the calculation formula of the loss of k_i is designed as follows:

$$L_{k_i} = -\log \frac{\exp\left(k_i \cdot d_i/\tau\right)}{\sum_{j=0, j\neq i}^{\text{batchsize}-1} \exp\left(k_i \cdot k_j/\tau\right) + \sum_{j=0}^{\text{batchsize}-1} \exp\left(k_i \cdot d_j/\tau\right)}, \quad (4)$$

where τ is a temperature parameter, which is consistent with previous work [18]. In the equation, the denominator is the sum of the similarity of k_i with all vectors in the dictionary, and the numerator is the similarity of k_i with d_i. Actually, the loss function is a log loss based on the softmax classifier with the number of categories of $2 * \text{batchsize} - 1$.

On this basis, the average loss of each calculated batch is shown in the following equation:

$$\text{Loss} = \frac{\sum_{i=0}^{\text{batchsize}-1} L_{k_i}}{\text{batchsize}}. \quad (5)$$

3.3. Data Augmentation Algorithm Based on Dimensionality Reduction and Sparsity Constraint. Data augmentation is an indispensable part in contrastive learning framework. Through appropriate information processing, it is ensured that the converted data version retains the essential characteristics of the original data and has the value of identification and matching. The vector after data augment is often regarded as a positive example of the original vector, and the two can be viewed as different perspectives of the same object. A reasonable data augment algorithm can lead to a model with outstanding performance.

Currently, those augmentation algorithms used in contrastive learning are proposed for image data, such as color conversion [24] and spatial transformation [25]. However, there are many forms of network security data, such as the

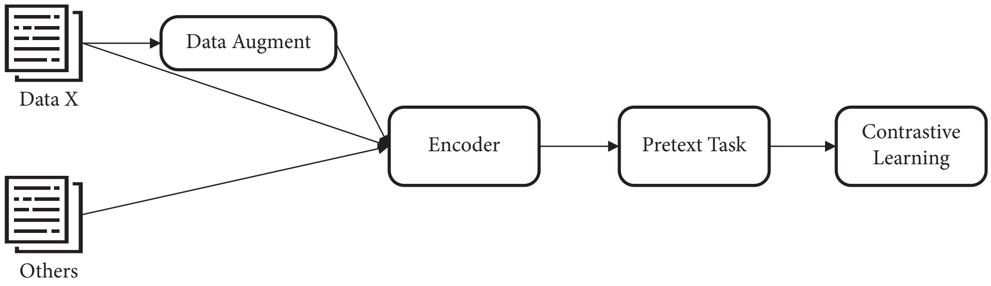

Figure 1: The process of instance discrimination.

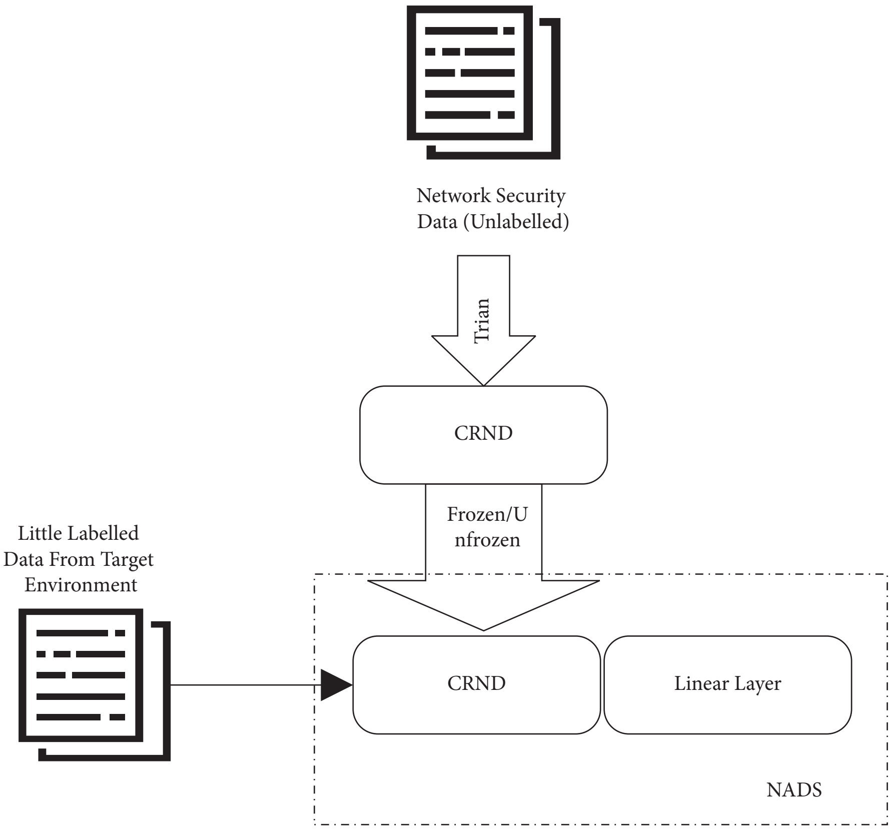
Network Security Data (Unlabelled)
Trian
CRND
Frozen/U nfrozen
Little Labelled Data From Target Environment
CRND
Linear Layer
NADS

Figure 2: Overall process. Some tasks need to freeze parameters in CRND, and some tasks will get better results when unfreezing the parameters.

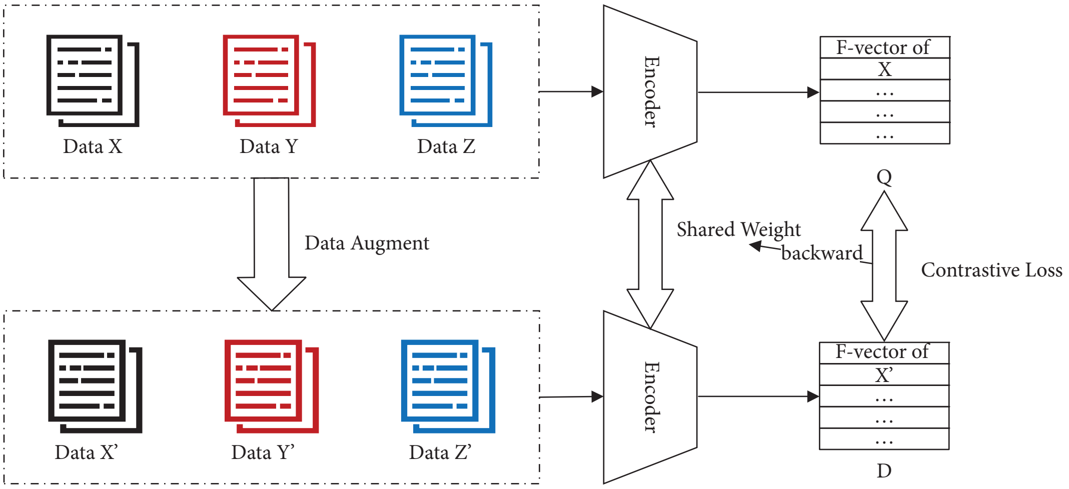

Figure 3: The architecture of CRND.

original packets and traffic statistical characteristics. The above data augmentation algorithms cannot handle network security data well. In the paper, a data augmentation algorithm based on dimensionality reduction and sparsity is designed for network security data.

Autoencoder (AE) is a common generative model, which aims at making the output as close as possible to the input [26]. In this process, the relevant algorithms need additional constraints to avoid the emergence of shortcut, so that the model can get meaningful result. Without these constraints, the identity operation is the most likely outcome. Dimension constraint [27] forms a bottleneck structure by compressing the dimensions of the middle layer, so that the model can learn effective data representation. The sparsity constraint [28] can force the model to learn meaningful output by setting the threshold for the sparsity of the intermediate expression without changing dimensions.

In order to obtain the augmented version of samples, it is necessary to ensure the dimensional consistency and content difference after the operation. Based on dimensionality reduction and sparsity constraint, sparse augmentation network (SAN) is designed as shown in Figure 4.

In Figure 4, 76D and 38D refer to the feature dimensions of the data samples during the experiment. In this paper, the relevant experiments are based on CICIDS2018 [29], in which the data samples are the statistical characteristics of a certain network flow.

The loss function of SAN is composed of reconstruction error and sparsity item. The reconstruction error is used to measure the distance between the output and the input. In this case, it indicates that the intermediate expression still has the ability to represent the original data. The sparsity item is used to restrict the sparsity of the middle layer. In this paper, the reconstruction error is calculated by mean square error (MSE), and the sparsity item is calculated by the Kullback–Leibler divergence (KLD). The loss function is shown in equations (6)–(8).

$$\text{Loss} = \text{MSE} + \text{KLD}, \tag{6}$$

$$\text{MSE} = E\left(x' - x\right)^2, \tag{7}$$

$$\text{KLD} = -\sum_x p(x)\log\frac{q(x)}{p(x)}. \tag{8}$$

In equations (7) and (8), x represents the input, x' represents the output of SAN, $p(x)$ represents the distribution of the intermediate expression, and $q(x)$ represents the expected distribution of the intermediate expression, which is determined by the threshold in SAN.

4. CRND

Based on the loss function and data augmentation algorithm proposed above, according to the construction process shown in Figure 3, CRND is trained with CICIDS2018 as the benchmark dataset.

The selection of encoder can be adjusted according to the target data. For example, in [7], the research team selected ResNet-50 [30] as the encoder network corresponding to image data with higher dimensions. In subsequent studies, ViT [31] with better performance became mainstream. But the statistics on network flow, such as CICIDS2018, gets fewer dimensions and has no temporal relationship, making multilayer perceptron (MLP) with faster training speed and less parameters become a good choice . In order to prevent overfitting, dropout [32] is attached to the network as shown in Figure 5.

5. Experiment

5.1. Experimental Setup. The experimental part of this study is mainly carried out on CICIDS2018 [29]. It is developed by the Communications Security Establishment (CSE) in cooperation with the Canadian Institute for Cybersecurity (CIC), containing benign behaviours and most of the existing abnormal behaviours, which are highly similar to the real environment, thus making experiments prove the validity of our method to a certain extent. In addition, CICIDS2018 meets the 11 indicators for constructing a benchmark dataset proposed in [33], having advantages over other datasets.

We extract 521,969 items from CICIDS2018, in which each one represents the statistics for a network flow defined by five-tuple. It should be noted that in order to be compatible with SAN, nominal features should be deleted such as Dst Port and Protocol. Then, these data are divided into training set and test set at a ratio of 4 : 1. The final shapes of these datasets are shown in Table 2, in which the first number represents the number of items and the second number represents the number of features contained in an item.

In the follow-up experiments, all experiments are carried out on binary-class task. In addition, only the training set was involved in training process to ensure the effectiveness of these experiments. The other settings are shown in Table 3, in which temperature parameter remains consistent with previous work [5, 18].

5.2. Linear Classification Protocol. Linear classification protocol refers to training a linear layer to complete the classification task with the weights in CRND frozen, which is often used to evaluate the effectiveness of models in the area of unsupervised learning. During this process, little labelled data are needed to complete training on the linear layer. The amount of labelled data and the evaluation performance of the corresponding models in the test set are shown in Table 4.

From the results, it can be seen that little data are needed to get a reliable model based on CRND. The experiment based on linear classification protocol can prove the effectiveness of CRND and the possibility of CRND as a pretrained model.

Meanwhile, it can be found from the results that with the amount of data used for fine-tuning gradually increasing, the improvement of model detection performance is not as expected. Under the constraint of linear protocol classification, its backbone network parameters are fixed, which

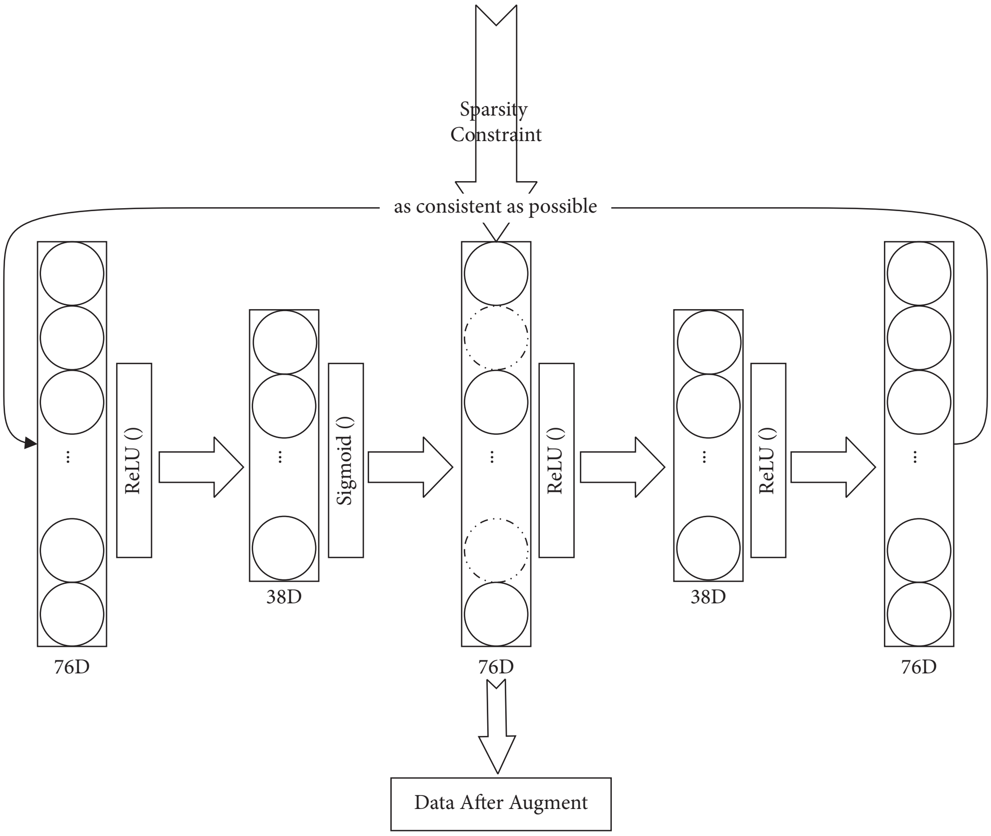

Figure 4: Sparse augmentation network.

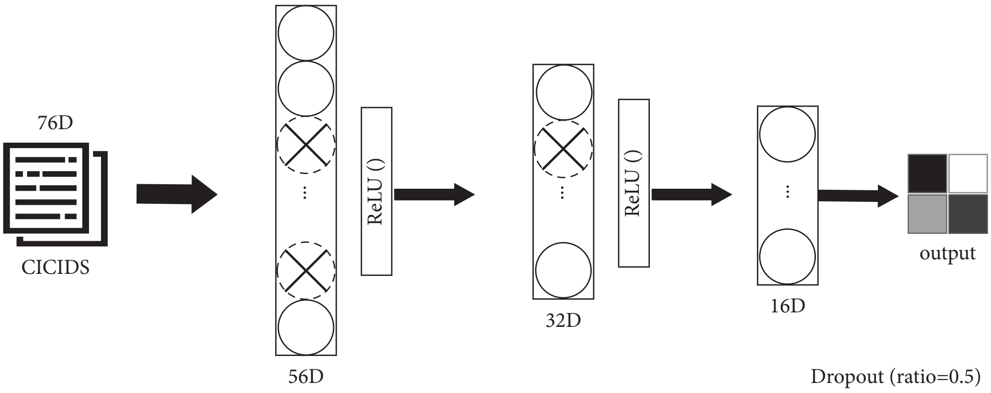

Figure 5: Encoder network.

Table 2: The shape of datasets during experiments.

Dataset	Shape
Training set	[417576, 77]
Test set	[104393, 77]

Table 3: The settings during experiments.

Parameters	Value
Optimizer	Adam
Learning rate in SAN	0.001
Learning rate in CRND	0.001
Batch size	4096
Epochs	200
Temperature parameter	0.7

limits the learning ability of the whole model, making the detection performance of the model have a superior limit, and there is resistance to subsequent performance improvement.

5.3. CRND as a Pretrained Model. On the basis of the network in linear classification protocol, unfreeze the backbone network of CRND so that its network parameters are updated along with the classifier and evaluate the detection

Table 4: The evaluation results during linear classification protocol.

Number of five-tuple	Accuracy (%)	Precision (%)	Recall (%)	False-alarm (%)	$F1$-score
10	82.35	89.67	68.84	6.53	0.78
50	**90.49**	98.17	**90.46**	1.24	0.88
100	88.85	98.73	76.29	0.81	0.86
10,000	**91.94**	**98.93**	83.03	**0.72**	**0.91**

Table 5: The evaluation results with CRND as a pretrained model.

Number of five-tuple	Accuracy (%)	Precision (%)	Recall (%)	False-alarm (%)	$F1$-score
10	90.07	98.76	79.01	0.81	0.88
50	92.06	99.24	83.06	0.53	0.90
100	95.49	97.73	92.14	1.76	0.95
200	96.13	99.45	91.94	0.42	0.96
500	96.49	99.37	92.83	0.49	0.96
10,000	97.18	99.66	94.08	0.26	0.98
100,000	99.59	99.83	99.26	0.14	**0.99**

Table 6: The evaluation results of CRND-200 and supervised method.

Model	Accuracy (%)	Precision (%)	Recall (%)	False-alarm (%)	$F1$-score
CRND-200	**96.13**	**99.45**	**91.94**	**0.42**	**0.96**
MLP-based	99.78	99.99	99.49	0.01	0.99
CRND-100000	99.59	99.83	99.26	0.14	0.99

Table 7: The evaluation results of CRND-200 and K-means method.

Model	Accuracy (%)	Precision (%)	Recall (%)	False-alarm	$F1$-score	Time (s)
CRND-200	**96.13**	**99.45**	**91.94**	**0.42%**	**0.96**	**13.10**
PCA [37]	86.37	87.52	94.81	—	0.91	13.83
Isolation forest [37]	87.90	92.23	91.07	—	0.88	79
Autoencoder [37]	87.66	91.44	91.64	—	0.92	18.62

performance of CRND as a pretrained model. The amount of data used in fine-tuning and the corresponding model performance are shown in Table 5.

From the results, it can be seen that with CRND as a pretrained model, a detection model with excellent detection performance can be obtained based on little samples. Among them, when 200 rows of data are used for fine-tuning, the model is acceptable in terms of cost, and the performance is relatively excellent. In subsequent experiments, this model is used as a baseline to compare with other methods, which is called CRND-200. In addition, we found that with the increase of the amount of data for fine-tuning, the improvement of the model performance is persistent without saturation phenomenon. To prove this point, CRND-10000 and CRND-100000 are constructed, and the results are shown in Table 5. Such a large amount of fine-tuning data may not be practical, but it is enough to prove that CRND gets the same continuous learning ability as deep unsupervised models in other fields.

When the amount of fine-tuning data reaches 10000, the $F1$-score index representing the comprehensive performance of the model reaches 0.98, and there is room for further optimization. When the amount of fine-tuning data reaches 100000, the $F1$-score of the model reaches 0.9954, which indicates that the CRND method has one of the most important characteristics of deep unsupervised learning. With the increase of the amount of fine-tuning data, the model can evolve continuously without premature convergence.

Based on the model in [15], the whole training set is used to construct a network anomaly detection model based on supervised learning. The evaluation results are shown in Table 6.

It can be seen that when only 200 five-tuples are used, CRND's $F1$-score has approached that of the supervised model trained by more than 400,000 five-tuples. In addition, with the increase of the data used for fine-tuning, the CRND method has the ability to match or even exceed the performance of the supervised learning method.

6. CRND and Other Unsupervised Learning Methods

In previous studies, there are few network anomaly detection models based on deep unsupervised methods. Most of the existing unsupervised methods applied in the area of network anomaly detection are based on machine learning [34–36]. Among them, the anomaly detection algorithms based on PCA and so on are mainstream. Now, a network anomaly detection model is built based on other unsupervised methods [37]. The results are shown in Table 7. These papers do not get false-alarm involved, so the values of the method in Table 7 are denoted as —.

It can be seen from the results that the performance of the K-means method on high-dimensional datasets is poor, and it has no detection ability. Meanwhile, CRND handles high-dimensional data well based on deep learning, thus getting a more extensive practical value.

7. Conclusions

In this paper, an unsupervised learning method is proposed to assist in the construction of network anomaly detection model. Based on contrastive learning, CRND is designed. Meanwhile, SAN that can be widely used in various data is proposed. Experiments on CICIDS2018 prove the effectiveness of CRND. The results show that CRND can fully learn the potential characteristics of network security data, and qualified performance can be achieved through fine-tuning in the target network environment. Taking it as a pretrained model can greatly accelerate the development of the network anomaly detection model. At the same time, it does not rely on annotated datasets, which contributes to the promotion and production application of network anomaly detection model greatly. At the same time, CRND can be widely used as a feature extractor, which may be better used to solve the target problem when combined with the related research of outlier detection [38, 39].

Data Availability

The data used to support the findings of this study can be found in [29].

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 (2019YFB1802505).

References

[1] China Internet Security, "China Internet Security Report in 2021," 2021, https://www.goupsec.com/report/wangluogongji/7154.html.

[2] S. Lu, L. Ying, W. Lin et al., "New Era of Deep Learning-Based Malware Intrusion Detection: The Malware Detection and Prediction Based on Deep Learning," 2019, https://arxiv.org/abs/1907.08356.

[3] D. Arp, E. Quiring, F. Pendlebury et al., "Dos and Don'ts of Machine Learning in Computer Security," in *Proceedings of the Proc of the USENIX Security Symposium*, USENIX Association, Boston, MA, USA, September 2022.

[4] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative pre-training," 2018, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[6] 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.

[7] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, WA, USA, June 2020.

[8] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[9] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] 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.

[11] U. Kamath, J. Liu, and J. Whitaker, *Deep Learning for NLP and Speech Recognition*, Springer International Publishing, Cham, Switzerland, 2019.

[12] L. Bontemps, V. L. Cao, J. McDermott, and L. K Nhien-An, "Collective Anomaly Detection Based on Long Short-Term Memory Recurrent Neural networks," in *Proceedings of the International Conference on Future Data and Security Engineering*, pp. 141–152, Springer, Cham, Switzerland, June 2016.

[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] J. Kim, N. Shin, S. Y. Jo, and H. K. 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, korea, February 2017.

[16] F. Pendlebury, F. Pierazzi, and R. Jordaney, "TESSERACT: Eliminating experimental bias in malware classification across space and time," *Proceedings of the 28th USENIX Security Symposium (USENIX Security 19)*, pp. 729–746, 2019.

[17] P. Vincent, H. Larochelle, Y. Bengio, and M. Pierre-Antoine, "Extracting and Composing Robust Features with Denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, Helsinki, Finland, June 2008.

[18] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, June 2018.

[19] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, Santiago, chile, December 2015.

[20] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, pp. 1735–1742, IEEE, New York, NY, USA, June 2006.

[21] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: a new estimation principle for unnormalized statistical

models," in *Proceedings of the 13th international conference on artificial intelligence and statistics*, pp. 297–304, Sardinia, Italy, May 2010.

[22] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[23] J. Donahue and K. Simonyan, "Large scale adversarial representation learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International conference on machine learning*, pp. 1597–1607, PMLR, Shenzhen, China, February 2020.

[25] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and B. Thomas, "Discriminative unsupervised feature learning with convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[26] Y. Qu, H. Ma, and Y. Jiang, "A Network Data Reinforcement Method Based on the Multiclass Variational Autoencoder," *Security and Communication Networks*, vol. 2022, Article ID 2993963, 2022.

[27] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Artificial Neural Networks and Machine Learning," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 52–59, Springer, Berlin, Heidelberg, June 2011.

[28] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, pp. 1–19, 2011.

[29] 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.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An Image Is worth 16x16 Words: Transformers for Image Recognition at scale," 2020, https://arxiv.org/abs/2010.11929.

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

[33] A. Gharib, I. Sharafaldin, A. H. Lashkari, A. Lashkari, and A. Ghorbani, "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, Thailand, December 2016.

[34] M. Zhao, H. J. Yan, and M. J. Zang, "Study on a network anomaly detection method based on clustering algorithm," *Computer Networks*, vol. 46, no. 10, pp. 68–71, 2020.

[35] Q. Li and C. H. Yan, "Research of network traffic anomaly detection technique based on histogram clustering," *Netinfo Security*, vol. 57, no. 01, pp. 40–42, 2012.

[36] L. Zhang, Y. Q. Wu, and P. Zhang, "Performance anomaly detection method based on unsupervised learning," *Radio and Communications Technology*, vol. 48, no. 04, pp. 758–762, 2022.

[37] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: unsupervised machine learning techniques," *Journal of Network and Systems Management*, vol. 30, no. 1, p. 12, 2021.

[38] K. Huang and K. V. Yuen, "Hierarchical outlier detection approach for online distributed structural identification," *Structural Control and Health Monitoring*, vol. 27, no. 11, p. e2623, 2020.

[39] K. V. Yuen and G. A. Ortiz, "Outlier detection and robust regression for correlated data," *Computer Methods in Applied Mechanics and Engineering*, vol. 313, pp. 632–646, 2017.