An Improved CNN Approach for Network Intrusion Detection System

Jianwei Hu, Chenshuo Liu, and Yanpeng Cui
(Corresponding author: Chenshuo Liu)

School of Cyber Engineering, Xidian University
No. 2, Taibai Road, Xi'an 710071, China
(Email: coulon_x@163.com)

(Received Mar. 14, 2020; Revised and Accepted Aug. 23, 2020; First Online May 31, 2021)

Abstract

To solve the low average recognition rate of a multi-class intrusion detection system based on Convolutional Neural Network (CNN), a CNN-based intrusion detection method optimized by Fruit Fly Optimization Algorithm (FOA) is presented in this article. The proposed approach first designs the model of a multi-class intrusion detection system based on CNN. To solve the class imbalance problem, FOA is applied to the pre-training process. During the training process, each batch is obtained by the resampling method according to the resampling weights, which are the output of the pre-training process. Finally, the NSL-KDD dataset is used to test the model. Compared with the CNN method without data equalization, the proposed method is more accurate.

Keywords: CNN; FOA; IDS; NSL-KDD

1 Introduction

In recent years, with the advancement of cloud computing technology, the threshold for becoming a webmaster has been lowed a lot. Anyone can get a cloud server from a cloud service provider (CSP) at a very low cost. Therefore, cyber security has become a major concern due to more and more people without network security technology have become webmasters. A large number of cloud hosts are threatened by network attacks include but not limited to Denial-of-Service (DoS), Injection attacks, Brute Force [17]. A large number of cloud hosts have been hacked and become members of botnets or Bitcoin trojan virus vectors.

Providing unified protection for the tenants without knowledge of cyber security is undoubtedly one of the best ways to reduce the security risk. Intrusion Detection System (IDS) is the defense system which can alert users or block attacks when detecting network intrusions. More and more CSPs are using IDS to protect their tenants' cloud servers [11, 19]. IDS are mainly classified into two types: Host-based IDS (HIDS) and Network-based IDS (NIDS). The main difference between the two is their way to obtain the detected information. HIDS mainly obtains information such as logs and system calls by deploying software in the monitored hosts. NIDS is used to monitor and analyze network traffic to protect a system [25]. In contrast, NIDS is more convenient to be deployed in the cloud computing environment. In hence, Network-based Intrusion Detection Systems has attracted our attention [6].

IDS has been developed for many years since its inception. Over the past few years, machine learning has achieved remarkable success in the field of intrusion detection. However, the research on Convolutional Neural Network (CNN) in intrusion detection technology is still lacking [9, 10]. CNN is a well-known deep learning model which has been proved to be a very good classifier in many fields. Actually, IDS is also a classification system, which means IDS based on CNN may perform well.

Today, network attacks happen all the time. A large amount of normal network traffic is mixed with various types of network attacks. The proportion of these types of attacks is often very uneven. However, machine learning methods often have requirements for sample equalization. Unbalanced training data will have a great negative impact on training results [3]. Therefore, this paper proposes a Convolutional Neural Network based intrusion detection system using Fruit Fly Optimization Algorithm (FOA) as an optimizer. The proposed system realizes the application of Convolutional Neural Network in intrusion detection system and solves the influence of data imbalance on the system.

The remainder of this paper is organized as follows: In Section 2, related research on Convolutional Neural Network based intrusion detection systems are discussed. In Section 3, a data balancing method based on Fruit Fly Optimization Algorithm is proposed and an intrusion detection system based on a Convolutional Neural Network is presented. In Section 4, the evaluation methods of the experimental are introduced and the experimental results are compared with those of other researchers. And in the

final section, the main work of this paper is summarized and the possible future research directions are proposed.

2 Related Work

The concept of intrusion detection systems has been in place for decades. Many researchers have proposed diverse detection methods. To compare with other methods, researchers generally use public datasets as experimental objects in experiments. The datasets commonly used to test intrusion detection methods are discussed in [24]. Among these datasets, NSL-KDD is the most widely used dataset in intrusion detection related researches. Each item in this dataset contains 41 features and a classification label. The classification label indicates whether this item is normal access or one type of attack. These attacks are divided into 39 sub-categories. And according to the attacks' impact, the researchers divided these 39 kinds of attacks into 5 categories, including Normal, DOS, R2L, U2R, and PROBING [7]. To compare with existing methods, this article will also use the NSL-KDD dataset as the experimental object.

Machine learning is a popular research direction in the field of computer today. In fact, as early as a decade ago, many researchers have begun to apply machine learning algorithms to intrusion detection systems [14]. In recent years, with the improvement of computers' computing power, deep learning technology received more and more attention. In [10], the author introduced the application of deep learning technology in intrusion detection methods. And in conclusion, it is proposed that the Convolutional Neural Network as a good classifier has not been exploited in the field of intrusion detection.

Nowadays, intrusion detection technology based on Convolutional Neural Network has attracted researchers' attention. In [13], Li *et al.* proposed an effective way to transfer the dataset into an image. They use the NSL-KDD dataset as the experimental dataset and use ResNet50 and GoogleNet to build the intrusion detection system. In the end, experiments proved that Convolutional Neural Networks also have good performance in intrusion detection. But their method can only judge whether there is an attack.

In fact, we hope that the intrusion detection systems can not only detect whether there is an attack but also identify the type of attack. Therefore, some researchers have also used CNN to construct a multi-class intrusion detection system [23]. Potluri *et al.* proposed a new CNN model. However, in their experiments, it was found that, just like the distribution of attacks in reality, the amount of data between different classifications in the dataset varies widely. Therefore, the recognition rate is not satisfactory for those attack categories with less training data.

So, the method proposed in this paper attempts to solve the drawbacks in the above studies.

3 Proposed Method

3.1 Data Equalization Based on Fruit Fly Optimization Algorithm

After manual classification, the data of NSL-KDD is divided into five categories, which are Normal, DOS, R2L, U2R, and PROBING [7]. The distribution of these five types of data in the training dataset is shown in Figure 1. From the figure, we can see that the type with the most data is Normal data. The number of Normal data is 67343. The type with the least data is U2R which has only 52 data. The amount of data's ratio between the two is close to 1000 to 1. The class imbalance problem will lead to a large impact on the Convolutional Neural Network. Even if U2R attacks cannot be identified, the system's highest accuracy rate can reach 99.9%. During the training process, the optimizer's pursuit of overall accuracy is likely to result in that the U2R attacks' identification rate is close to 0%, which is indeed happened.

However, U2R attacks are very harmful which means we cannot ignore the identification ability of those attacks that have few training data. If an intrusion detection system can only identify partial attacks, then the system is not reliable. The solution to data imbalance is generally oversampling and undersampling, as opposed to some classical machine learning models, oversampling will not cause overfitting in CNN [3]. But it is difficult to choose sampling weights to maximize the recognition rate of each attack type.

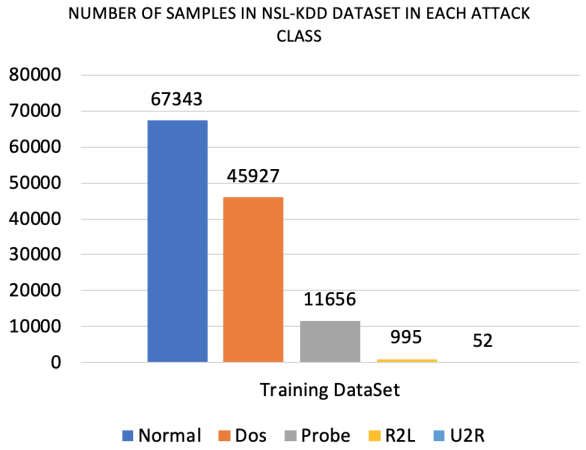

Figure 1: Number of samples in NSL-KDD dataset in each attack class

In recent years, intelligent swarm algorithm has gradually become a research hotspot in the computer field. Intelligent swarm algorithm is a method for solving optimization problems by simulating natural processes. Many intelligent swarm algorithms have been proposed such as ant colony optimization, particle swarm optimization, bee-inspired algorithms. Many people have proved that intelligent swarm algorithms can achieve very good results in optimization problems, and are faster than traditional

algorithms [5, 18].

Fruit Fly Optimization Algorithm (FOA) is a method for finding global optimization based on the food finding behavior of the fruit fly. The core idea of the algorithm is to simulate the foraging process of the fruit fly in which the fruit fly using their keen sense of smell and vision to fly to the food. The flow of FOA is as follows [21]:

1) Randomly assign a initial position to the fruit fly population.

$$\begin{aligned} X_{axis} &= RandomValue, \\ Y_{axis} &= RandomValue. \end{aligned}$$

2) Give each fruit fly a random flight direction and distance.

$$\begin{aligned} X_i &= X_{axis} + RandomValue, \\ Y_i &= Y_{axis} + RandomValue. \end{aligned}$$

3) Since the location of the food is not known, the distance to the origin ($Dist$) is first estimated, and then the smell concentration judgment value (S) is calculated.

$$\begin{aligned} Dist_i &= \sqrt{X_i^2 + Y_i^2} \\ S_i &= \frac{1}{Dist_i} \end{aligned}$$

4) Then the fitness function is used to find the smell concentration ($Smell$) of the individual location of the fruit fly.

$$Smell_i = FitnessFunction(S_i).$$

5) Find out the fruit fly with maximal smell concentration among the fruit fly swarm.

$$\begin{aligned} bestSmell &= max(Smell), \\ bestIndex &= i \rightarrow (Smell_i = bestSmell). \end{aligned}$$

6) Store the best smell concentration value and the fly's location if bestSmell is greater than the previous value. Then fruit fly swarm will fly towards that location.

$$\begin{aligned} Smellbest &= bestSmell, \\ X_{axis} &= X_{bestIndex}, \\ Y_{axis} &= Y_{bestIndex}. \end{aligned}$$

7) Repeat the implementation of steps 2-6 N (the number of iterations) times.

FOA is very simple and fast. Thus, it is employed for data equalization, aiming to find the best resampling weights. We assume that the sampling weights of five types of data in each batch are: w_0, w_1, w_2, w_3, w_4

So smell concentration judgment value S_i can be expressed as follow:

$$S_i = (w_0^i, w_1^i, w_2^i, w_3^i, w_4^i).$$

It is known that the range of the weight is [0, 1], so the value of X_i and Y_i is limited. Fruit Fly Optimization algorithm is applied to the pre-training process to achieve data equalization. The specific details are shown in Algorithm 1. As shown in line 8 and line 9, in each round, the training dataset is resampled according to smell concentration judgment value (S_i). And training will be performed using this training dataset. Then the average accuracy of the five types of data is used as the *fitness function* to obtain the smell concentration $Smell_i$. In the end, the program returns $BestW$, which is the data sampling weight can make the model get the best average accuracy.

Algorithm 1 Fruit fly optimization algorithm based data sampling

Input: dataset: trainning dataset, popsize: number of fly, maxgen: number of iterations
Output: Sampling ratio: w_0, w_1, w_2, w_3, w_4

1: $(X_{axis}, Y_{axis}) = initRandXY()$
2: **for** $i = 0, 1, ..., popsize - 1$ **do**
3: $(X_i, Y_i, Dist_i, S_i, Smell_i) = initFlyState()$
4: **end for**
5: **for** $i = 0, 1, ..., maxgen - 1$ **do**
6: **for** $j = 0, 1, ..., popsize - 1$ **do**
7: $(X_j^{i+1}, Y_j^{i+1}, Dist_j^{i+1}, S_j^{i+1}) = updateState()$
8: $TrainDataSet = Resample(S_j^{i+1})$
9: $Smell_j^{i+1} = TrainModel(TrainDataSet)$
10: **end for**
11: Smellbest = getLocalBest()
12: BestW = updataBest(Smellbest)
13: **end for**
14: Return BestW;

3.2 Intrusion Detection Model based on Convolutional Neural Network

Convolutional Neural Network is a feedforward neural network with a deep structure that includes convolutional calculations. Different from traditional fully-connected neural networks, Convolutional Neural Networks share weights in the convolutional layer, which greatly reduces the number of weights and improves efficiency. Nowadays Convolutional Neural Networks have been used in various fields such as image classification, object detection, object tracking, pose estimation, text detection, visual saliency detection, action recognition, scene labeling, speech and natural language processing [15, 16]. In particular, Convolutional Neural Networks have very good capabilities in image classification [12]. A typical Convolutional Neural Network mainly includes input layer, convolutional layer, pooling layer, and fully-connected layer. The details of each part are as follows [20]:

Input layer: The input layer is mainly used to provide input data to the Convolutional Neural Network. We first convert the original training data into $8 * 8$ grayscale images as input [13]. We refer to the method of image recognition and use the grayscale images as the input data. In the training phase, the training data we used are the resampling training data according to the weights obtained by FOA.

Convolutional layer: The major difference between Convolutional Neural Networks and BP Neural Networks is that the Convolutional Neural Networks have the convolutional layers. The convolution kernel in the convolutional layer acts like the filters in image processing, which can help us extract higher-dimensional features and reduce the amount of calculation [1]. In this paper, 3 convolutional layers are contained in our Convolutional Neural Networks. Each convolutional layer uses a wide convolution with 0 paddings for convolution calculations and uses ReLU Function as the activation function.

Pooling layer: A pooling layer is allocated between two continuous convolutional layers, which is used to reduce the number of parameters and to prevent overfitting. The most commonly used pooling method is max-pooling. So this article also uses the max-pooling method. There are two pooling layers inserted in the gaps of the three convolutional layers.

Fully connected layer: In a fully connected layer, all neurons have connections with the neurons in former layers. Usually, the fully connected layer is at the tail of the Convolutional Neural Network. The Convolutional Neural Network designed in this paper has two fully connected layers. In the first fully connected layer, a random dropout is added to prevent overfitting. The second fully connected layer uses Softmax Function as the activation function for classification.

In summary, the Convolutional Neural Network finally proposed in this paper is shown in Figure 2.

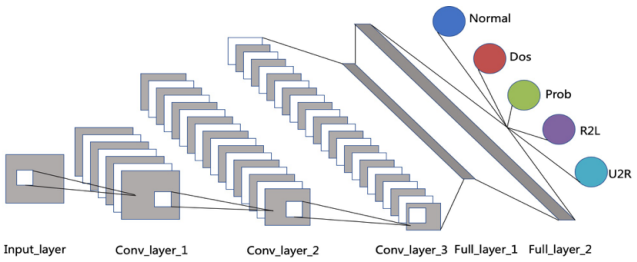

Figure 2: Convolutional neural network structure designed in this paper

4 Experimental Setup & Results

4.1 Implementation

The Convolutional Neural Network in this paper is implemented with the Tensorflow framework. The computer used is a Lenovo Workstation ThinkStation P520 with 2080Ti. In NSL-KDD dataset, the training data set consists of 21 types of attacks, and 17 new attack types are in the test set [8]. In experiments, we found that this method cannot classify unknown types of attacks, so we filtered the unknown types of attacks in the test dataset. The details of the experimental steps are as follows:

Data processing process: In the data processing part, all discrete features are first converted into a binary vector by the one-hot encoding method. Then all continuous features will be normalized. The normalized result is discretized in units of 0.1. Then these data will be converted to a binary vector with the one-hot encoding method. After the previous processing, the original training dataset is converted into a binary vector with 464 dimensions which can be converted to a grayscale vector with 58 dimensions. Then we add 8 zero paddings at each of the grayscale vectors. So it can be converted into an $8*8$ grayscale image as shown in Figure 3 [13].

Data equalization process: To reduce the impact of the class imbalance problem in the Convolutional Neural Network, in the second step the Fruit Fly Optimization algorithm is used for pre-training which can help us get resampling weight of each class.

Training process: The resampling weights obtained in the second step are used to resample to obtain the training batch for each epoch. In the training process, cross-entropy is used as a loss function, and stochastic gradient descent is used for optimization. When the accuracy rate does not increase significantly in 30 iterations, training will be stopped.

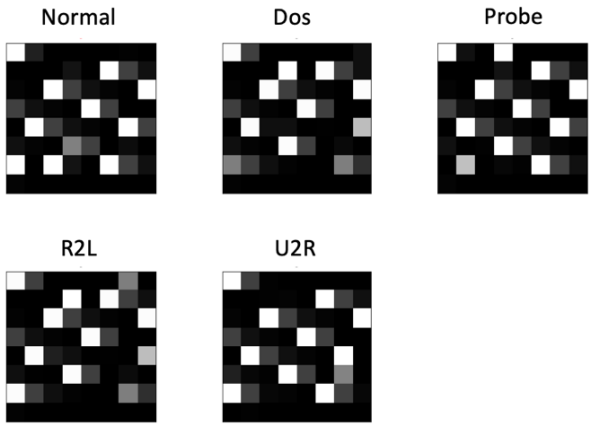

Figure 3: Grayscale images of different attack types

4.2 Experimental Evaluation

We use the NSL-KDD dataset as the experimental object. The recall score, precision score, and F1 score are used to make a simple assessment of the experimental results. Their calculation formulas are as follows:

$$RecallScore = \frac{TP}{TP+FN} \quad (1)$$

$$PrecisonScore = \frac{TP}{TP+FP} \quad (2)$$

$$F1Score = \frac{2TP}{2TP+FP+FN} \quad (3)$$

TP: The number of data classified as one type which belongs to that type.

FP: The number of data classified as one type which doesn't belong to that type.

FN: The number of data classified as not that type which belongs to that type.

The recall score, precision score, and F1 scores of each type of attack are listed in Figure 4. And we compared the results with those without data equalization experiments in Tables 1, 2, 3. From the experimental results, the average recall rate of the five classifications in the system even reached 87.1%. And we can find that the recall rate of the two types of attack, R2L and U2R, has improved significantly, which means that when such attacks occur, we are likely to correctly identify such attacks. Although these two types of attacks occur not very frequently, they appear to be more harmful.

There is no significant improvement in Precision Score and F1 Score. The main reason is that the amount of each type of data differs too much in the test data. For the Normal type, although less than 0.1% of the data is misclassified as the U2R type, it will cause the FP value of the U2R type to increase significantly. From Equation 2, we can see that the increase in FP will result in a decrease in Precision Score. The most important thing for intrusion detection system is its ability to detect attacks. In the method proposed in this paper, the intrusion detection system's ability to identify attacks has been significantly improved.

It can be seen that the optimization of the FOA alleviates the class imbalance problem in Convolutional Neural Networks. From the overall results of multiple evaluation indicators, our method has significantly improved the accuracy of the system.

Table 1: Recall score comparison

	Normal	Dos	Prob	R2L	U2R
FOA Optimized	91.75%	95.26%	90.77%	94.67%	63.46%
No Data Equalization	93.74%	86.35%	88.19%	0%	0%

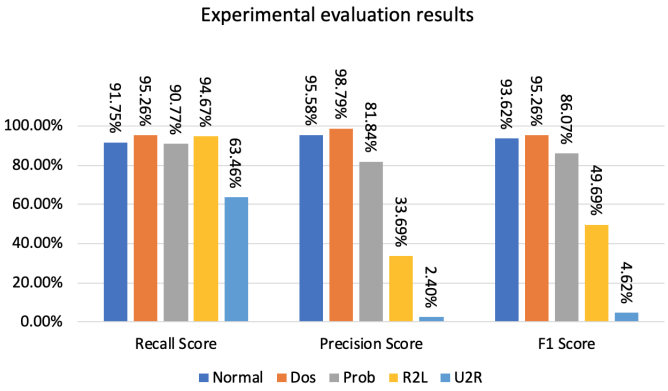

Figure 4: Experimental evaluation results

Table 2: Precision score comparison

	Normal	Dos	Prob	R2L	U2R
FOA Optimized	95.58%	98.79%	81.84%	33.69%	2.4%
No Data Equalization	97.82%	84.23%	85.46%	0%	0%

Table 3: F1 score comparison

	Normal	Dos	Prob	R2L	U2R
FOA Optimized	93.62%	95.26%	86.07%	49.69%	4.62%
No Data Equalization	93.06%	98.01%	87.52%	0%	0%

5 Conclusions & Future Work

In this paper, an intrusion detection system based on Convolutional Neural Networks is proposed. The NSL-KDD dataset is used for training and testing. In order to solve the class imbalance problem, this paper uses Fruit Fly Optimization Algorithm to perform a pre-training process to obtain the best possible training data resampling weights. Finally, the problem that some attack types cannot be identified due to the small number of training samples is solved.

Feature selection method is commonly used to find the optimal feature subset to improve the system performance by eliminating redundant and irrelevant features from dataset [2]. The method of Convolutional Neural Network avoids the optimal feature selection. In the future, directly converting network packets into pictures to avoid the problem of artificial feature selection may be a new research direction. Moreover, the Convolutional Neural Network develops rapidly, and the structure proposed in this paper can also be improved. For example, some scholars have proposed dropout function which can reduce overfitting and improve accuracy [4, 22]. In future research, these methods can be considered to optimize the structure of Convolutional Neural Networks. At last, in our method, when the intruders know the data set and resampling weights, they may analyze the undetected at-

tack package to construct an attack that can bypass our detection. This detection and bypass process is similar to Generative Adversarial Network's (GAN) training process. In the future, it may be considered to combine the GAN and CNN to increase the difficulty to bypass intrusion detection.

References

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *International Conference on Engineering and Technology (ICET'17)*, pp. 1–6, 2017.

[2] K. K. R. Amrita, "A hybrid intrusion detection system: Integrating hybrid feature selection approach with heterogeneous ensemble of intelligent classifiers," *International Journal of Network Security*, vol. 20, no. 1, pp. 41–55, 2018.

[3] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[4] S. Cai, J. Gao, M. Zhang, W. Wang, G. Chen, and B. C. Ooi, "Effective and efficient dropout for deep convolutional neural networks," *Machine Learning*, 2019. (arXiv:1904.03392)

[5] A. Chakraborty and A. K. Kar, "Swarm intelligence: A review of algorithms," in *Nature-Inspired Computing and Optimization*, pp. 475–494, 2017.

[6] A. Dewanje and K. A. Kumar, "A new malware detection model using emerging machine learning algorithms," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 24–32, 2021.

[7] L. Dhanabal and S. P. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[8] R. H. Dong, H. H. Yan, and Q. Y. Zhang, "An intrusion detection model for wireless sensor network based on information gain ratio and bagging algorithm," *International Journal of Network Security*, vol. 22, pp. 231–241, 2020.

[9] T. T. Gao, H. Li, and S. L. Yin, "Adaptive convolutional neural network-based information fusion for facial expression recognition," *International Journal of Electronics and Information Engineering*, vol. 13, no. 1, pp. 17-23, 2021.

[10] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system- a taxonomy and survey," *Journal of Cryptography and Security*, 2017. (arXiv:1701.02145)

[11] S. Iqbal, M. L. M. Kiah, B. Dhaghighi, M. Hussain, S. Khan, M. K. Khan, and K. K. R. Choo, "On cloud security attacks: A taxonomy and intrusion detection and prevention as a service," *Journal of Network and Computer Applications*, vol. 74, pp. 98–120, 2016.

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

[13] Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *International Conference on Neural Information Processing*, pp. 858–866, 2017.

[14] H. J. Liao, C. H. R. Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[15] Y. Lou, Y. He, L. Wang, and G. Chen, "Predicting network controllability robustness: A convolutional neural network approach," *Systems and Control*, 2019. (arXiv:1908.09471)

[16] Y. Lou, Y. He, L. Wang, K. F. Tsang, and G. Chen, "Knowledge-based prediction of network controllability robustness," *Physics and Society*, 2020. (arXiv:2003.08563)

[17] I. Müller M. Almorsy, J. Grundy, "An analysis of the cloud computing security problem," in *Proceedings of the 30th APSEC Cloud Workshop*, 2010. (https://www.researchgate.net/publication/255708329_An_analysis_of_the_cloud_computing_security_problem)

[18] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm and Evolutionary Computation*, vol. 33, pp. 1–17, 2017.

[19] A. Khalid N. H. Hussein, "A survey of cloud computing security challenges and solutions," *Journal of Computer Science and Information Security*, vol. 14, 2016.

[20] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *Journal of Neural and Evolutionary Computing*, 2015. (arXiv:1511.08458)

[21] W. T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.

[22] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in *Asian Conference on Computer Vision*, pp. 189–204, 2016.

[23] S. Potluri, S. Ahmed, and C. Diedrich, "Convolutional neural networks for multi-class intrusion detection system," in *International Conference on Mining Intelligence and Knowledge Exploration*, pp. 225–238, 2018.

[24] S. K. Sahu, S. Sarangi, and S. K. Jena, "A detail analysis on intrusion detection datasets," in *IEEE International Advance Computing Conference (IACC'14)*, pp. 1348–1353, 2014.

[25] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman,

"Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

Biography

Jianwei Hu has gained the doctorate degree. He is a graduate supervisor of Xidian University. He majors in network security and network confrontation, communication reconnaissance and communication confrontation.

Chenshuo Liu born in Xi'an, a post-graduate student majoring in cyberspace security. He is now studing at Xidian University. He is interested in information security technology and intrusion detection technology.

YanPen Cui has gained the doctorate degree. She is a graduate supervisor of Xidian University. She majors in network security and cloud computing security.