A Framework for Moving Target Defense based on Federated Semi-Supervised Learning

Wanying Lu
Wenbin Zhai*
Feng Wang
Yu Fan
wanyinglu@nuaa.edu.cn
wenbinzhai@nuaa.edu.cn
fengwang06@nuaa.edu.cn
fanyu2017@nuaa.edu.cn
Nanjing University of Aeronautics and Astronautics
Nanjing, Jiangsu Province, China

ABSTRACT

In recent years, with the rapid development of the Internet, it has penetrated into all areas of daily lives. However, due to the complexity of the Internet architecture, there are inevitably some inherent security threats, which could be exploited by adversaries to cause great damage. Moving Target Defense (MTD) has been proposed to solve this problem by building a dynamic, heterogeneous and redundant system architecture. Unfortunately, most of the existing data arbitration algorithms for MTD are based on the majority consensus voting algorithm, which cannot cope with common mode escape. Therefore, in this paper, we propose a framework for moving target defense based on Federated Semi-Supervised Learning (FSSL), called FedDA. In addition to the output data of heterogeneous executives, FedDA leverages their behavior data to assist in data arbitration. Meanwhile, we consider a more realistic assumption that the behavior data of heterogeneous executives is not annotated with ground-truth lables, and FSSL is used for model training. Finally, a data arbitration algorithm combined with historical confidence is proposed to identify malicious executives. Extensive experimental results show that our method can well resist common mode escape and outperform the state-of-the-art data arbitration algorithms.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**.

KEYWORDS

Moving Target Defense, Data Arbitration, Federated Semi-Supervised Learning.

CECCT 2023, November 17–19, 2023, Guilin, China

ACM ISBN 979-8-4007-1630-0/23/11…$15.00
https://doi.org/10.1145/3637494.3638748

ACM Reference Format:
Wanying Lu, Wenbin Zhai, Feng Wang, and Yu Fan. 2023. A Framework for Moving Target Defense based on Federated Semi-Supervised Learning. In *2023 International Conference on Electronics, Computers and Communication Technology (CECCT 2023), November 17–19, 2023, Guilin, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3637494.3638748

1 INTRODUCTION

The rapid development of the Internet profoundly changes the social structure and relationship, and greatly promotes the development and progress of human society. However, simultaneously with the massive deployment of the Internet, it also suffers from numerous malicious attacks and damages, in which it will be compromised and exploited by people with ulterior motives due to software or hardware loopholes. According to incomplete statistics, in April 2022 alone, there were more than 5.07 million terminals infected with Trojans or botnet malicious programs in China. Therefore, effective countermeasures are needed to detect malicious terminals to enhance the security of the Internet.

There are already some defensive approaches to detect intrusions on the Internet, however, due to the complexity of the Internet architecture, there are inevitably some inherent security threats, which may be due to design flaws or human omissions. In addition, driven by interests or many other reasons, attacks also develop and evolve rapidly, which may make existing detection schemes ineffective. In order to deal with the above severe challenges, Moving Target Defense (MTD) [14] is proposed, whose core idea is to build a dynamic, heterogeneous and redundant system architecture, thereby constructing an uncertain, heterogeneous, and non-persistent mimic environment through the dynamic switching of equivalent functional bodies (e.g., middleware, web server, operating system, etc.) to detect and destroy the attack chain [6].

Data arbitration [7] is the core of MTD and has an important impact on the efficiency and security of MTD. There have been many researches [11] on data arbitration schemes, most of which are based on the majority consensus voting algorithm. The main idea is that multiple equivalent functional bodies perform the same task, and then their respective outputs are compared by the data arbiter, based on which the data with the largest number of approvals is selected as the correct output. It is natural and has been widely

used, but it has the disadvantage that it can not effectively deal with the problem of "common-mode escape" [12], where there are multiple malicious executives, especially when the malicious executives are in the majority. For example, there are N executives selected and scheduled, while more than $\lceil N/2 \rceil$ executives are malicious.

To solve the above problem, it is not enough and feasible to rely solely on the outputs of heterogeneous executives. Fortunately, various behaviors of heterogeneous executives, such as network communication, resource consumption, software event, and user interaction can be exploited to assist the data arbitration [3]. At the same time, machine learning (ML), especially deep learning (DL), is very good at learning key features and behavior patterns from high-dimensional spaces. However, the model training in ML relies on a large amount of labeled data [2], which is unrealistic in MTD, because the data of heterogeneous executives is often not annotated with ground-truth labels. In addition, since the heterogeneous executives contain private data of different users, the raw data can not be transmitted to the data arbiter for model training due to the consideration of privacy preservation [9].

Therefore, in this paper, we propose a framework for Moving Target Defense based on Federated Semi-Supervised Learning (FSSL) [5], called FedDA. First, we utilized the FSSL for model training based on the status and behavior data of equivalent functional bodies (e.g., network traffic). Then, based on the trained model, a data arbitration algorithm combined with historical confidence is proposed to identify malicious executives in response to common mode escape.

In particular, the main contributions of this paper are summarized as follows:

- A framework for moving target defense based on Federated Semi-Supervised Learning (FSSL) called FedDA is proposed. First, we consider a more realistic assumption that the behavior data of heterogeneous executives is all unlabeled, based on which the FSSL is adopted for model training. Then the behavior data of each heterogeneous executive is utilized along with their output data for data arbitration, and a data arbitration algorithm combined with the historical confidence is designed to address the common mode escape.
- Extensive experiments are conducted based on the CIC-IDS 2017 dataset [10] through the Keras library and TensorFlow framework. The results show that FedDA can well resist common mode escape and outperform the state-of-the-art data arbitration algorithm [11].

2 RELATED WORK

2.1 Data Arbitration

Data arbitration algorithms are a key part of the MTD system, which can be utilized to distinguish normal and correct data. Most of the existing methods are based on the traditional majority consensus voting algorithm. Shen *et al.* [11] propose a reliable multi-rule data arbitration method with adaptive thresholds for random and covert data injection attacks. By cutting off future communication with malicious neighbors, data injection attacks can be effectively defended. Wei *et al.* [4] design a confidence calculation method based on Logistic function. Based on the classification of the impact of different historical periods, it can effectively remove various abnormal noises, thereby improving the security of the data arbitration mechanism. Aiming at the virtual machine escape scenario in the honeypot system, Lu [8] propose a dual mimic mechanism and a honeypot architecture under this mechanism, which can resist the escape vulnerability of the virtualization platform through the heterogeneity of the underlying virtualization platform.

2.2 Federated Semi-Supervised Learning

FSSL is a distributed learning model that uses unlabeled data of multiple clients and limited labeled data of the server for model training without revealing the original private data, in which a small amount of labeled data is helpful for training guidance, while massive unlabeled data is beneficial to fully learn the diverse knowledge of the data and prevent the model from overfitting. Zhao *et al.* [13] use unlabeled data and Long Short-Term Memory model for automatic encoding on the client side, and use Softmax classifier at the server for learning, thus achieving higher accuracy and better performance than FSSL based on data augmentation. Jeong *et al.* [5] propose a method based on the consistency verification of predictions between clients to improve the performance of FSSL, and design two experimental scenarios for evaluation based on the location of the labeled data (i.e., labels at server and labels at clients).

3 SYSTEM MODEL

The MTD system presented in this paper consists of users, middlewares, the scheduler, monitor, and data arbiter. The system architecture is depicted in Fig. 1. Users consist of two communication parties. For ease of presentation, we assume that user A is the data sender and user B is the data receiver. The middleware may be compromised by adversaries to perform attacks on data, such as tampering, or dropping. Therefore, to verify the correctness of the data, data is transmitted through heterogeneous and redundant middlewares. In other words, the data sender A selects a certain number of middlewares from the middleware resource pool through the scheduler.

Multiple heterogeneous middlewares selected by the scheduler perform the same task, e.g. transmit the same data from user A, and then their respective output data will be transmitted to user B. At the same time, the monitor collects the status and various behaviors of each middleware, such as network communication, resource consumption, software event, and user interaction, and then these monitored data will be transmitted to the data arbiter for further analysis and detection. After receiving the output data from each middleware, the data arbiter is responsible for discriminating the correct data and identifying malicious middlewares.

4 METHOD

4.1 Basic Idea

The basic idea is to use the behavior and status data of each middleware along with the output data for the data arbitration in order to deal with the common mode escape. Specifically, the behavior and status data BD_i of each middleware M_i is combined with its output data OD_i to perform data arbitration. First, the behavior and status data of all users on different middlewares are exploited

to train the detection model in the form of FSSL, due to privacy concerns and limitations of real-world scenarios. Then, based on the trained detection model, the behavior and status BD_i of each middleware M_i in this data transmission can be judged, and the trust value P_i of each middleware, that is, the probability that its behavior belongs to normal can be obtained. Next, data arbitration is implemented based on the output OD_i of each middleware, however, unlike the traditional majority consensus voting algorithm, the weight W_i of each middleware is no longer fixed at 1. Instead, a data arbitration algorithm combined with the historical confidence is proposed, in which the weight W_i of each middleware M_i consists of the multiplication of two parts, one is the historical confidence of the middleware H_i, which is determined by its behavior in the past period (i.e., a window size T), and the other is the trust value of the middleware P_i, which can be obtained by the detection model. Finally, we weighted average the weights of all middlewares based on the output data instead of accumulation to get the approval scores of different output data. The output data with the highest approval score is considered as normal and wins the arbitration.

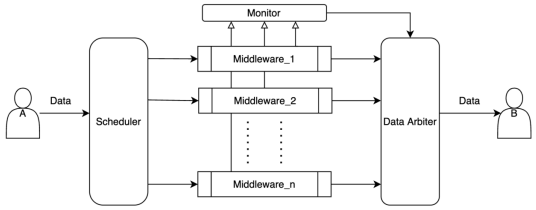

Figure 1: The architecture of the MTD system.

We further illustrate our method with the example shown in Fig. 2. There are five middlewares $M_1 \sim M_5$. At the first round, based on the detection model and the behavior data BD of each middleware, the data arbiter gets the trust value of each middleware, namely: $P_1 = 0.21, P_2 = 0.96, P_3 = 0.90, P_4 = 0.86, P_5 = 0.92$. Meanwhile, the initial historical confidence of each middleware is 1, namely $H_i = 1, 1 \leq i \leq 5$. Therefore, the weight of each middleware is $W_1 = P_1 * H_1 = 0.21, W_2 = P_2 * H_2 = 0.96, W_3 = 0.90, W_4 = 0.86, W_5 = 0.92$. Then, we average the weights of middlewares based on the difference of the output data to obtain the approval score of each output data, namely the approval score S_1 of the output data D_1 is $S_1 = (W_2 + W_3 + W_4 + W_5)\ /\ 4 = 0.91$, and $S_2 = W_1\ /\ 1 = 0.21$. Since $S_1 > S_2$, the output data D_1 is arbitrated as normal data and wins the data arbitration. Furthermore, the historical confidence H_1 of the middleware M_1 need to be correspondingly reduced due to the wrong output data. For convenience, we set the window size to 5, namely $T = 5$. Therefore, H_1 is reduced to $4/5 = 0.8$.

What's more, we next explain how our method further efficiently deals with common mode escape. At the forth round r_4, the output data $OD_1 \sim OD_4$ of the middlewares $M_1 \sim M_4$ are all D_2. The malicious data D_2 will win the data arbitration if the traditional majority consensus algorithm is adopted, i.e. it is not effective against common mode escape. However, according to our method, the weight of each middleware is $W_1 = 0.15 * 0.4 = 0.06, W_2 = 0.12 * 0.6 = 0.072, W_3 = 0.45 * 0.8 = 0.36, W_4 = 0.27 * 1 = 0.27, W_5 = 0.93 * 1 = 0.93$. Therefore, the approval score of each data is $S_1 = 0.93, S_2 = (W_1 + W_2 + W_3 + W_4)\ /\ 4 = 0.1905$. Since $S_1 > S_2$, the output data D_1 will win the data arbitration. Therefore, it is clear that our data arbitration algorithm can efficiently solve the problem of common mode escape.

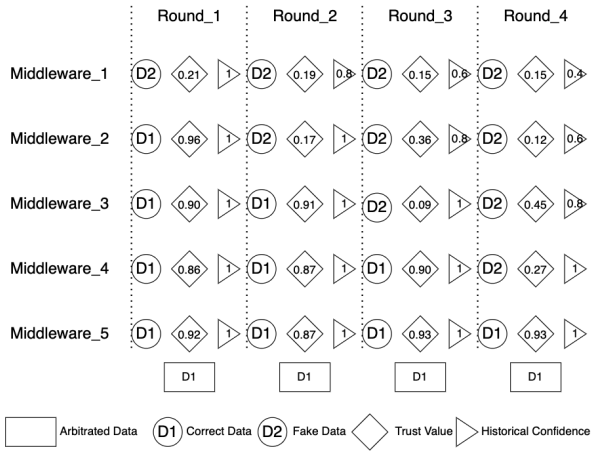

Figure 2: The illustrative example of FedDA.

4.2 Model Training based on FSSL

Most existing data arbitration methods are based on the majority consensus voting algorithm, that is, the output data of each middleware is compared, and then the data with the most approvals is selected as the correct output. This scheme is easy to fail in the scenario of common mode escape, as depicted at the round r_2 and r_3 in Fig. 2.

In order to address the above challenges, in addition to the output data of middlewares, the status and various behaviors of each middleware are collected for further analysis. However, how to make full use of the limited labeled data on the data arbiter side and the massive unlabeled data on the middleware side for model training and performance optimization is a significant challenge. Motivated by this, in this paper, we design a model training method based on FSSL, which consists of a server and multiple clients. In our MTD system, the data arbiter is the server and has a certain amount of labeled data, while each middleware acts as a client with a large amount of unlabeled data, as shown in Fig. 3. The server and clients implement local training based on their own private data, and cooperate to train a global model under the coordination and command of the server in order to profit from data of peer entities without sharing local raw data. The details and procedures of the model training are described as follows.

4.2.1 Model Initialization. At the beginning of each training round r, the server S distributes the global model parameters to each client $M_i, 1 \leq i \leq N$, as shown in **Step 1** in Fig. 3. At the first round r_0, the global model parameter ω_g^0 is specified or initialized randomly, otherwise it ω_g^r is aggregated from the local model parameters of clients ω_i^r.

4.2.2 Model Training. In **Step 2**, model training consists of two parts, namely supervised learning based on labeled data on the

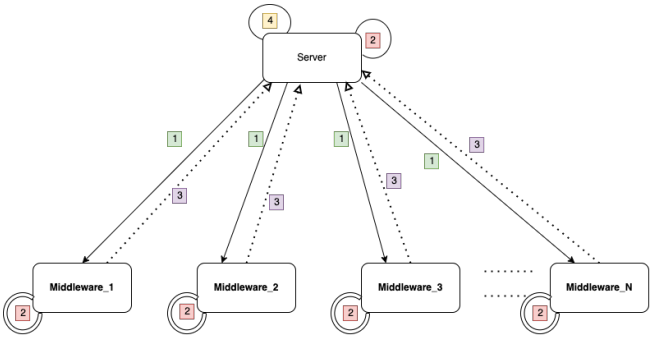

Figure 3: The overview of federated semi-supervised learning procedure.

server side and unsupervised training based on unlabeled data on the client side.

Unsupervised training at clients: The data of the client is unlabeled, and we utilize the pseudo-label technique [1] for unsupervised training of the client. The training objective for each client M_i is to minimize the following loss function $F_i(\omega_i^r) = \frac{1}{|D_i|}\sum_{j=1}^{|D_i|} \text{sgn}(\max(\bar{y}) \geq \alpha)\, l(\arg\max(\bar{y}), \bar{y})$, where ω_i^r is the model parameters of the client M_i at round r; D_i is the local private unlabeled training samples $D_i = \{(x_j) \mid 1 \leq j \leq |D_i|\}$ in which x_i is the feature vector; sgn( ) is the indicator function, $\bar{y} = p(\omega_i^r, x_j)$, which means the prediction of the current model parameter ω_i^r on the sample x_j; α is the threshold hyperparameter, which is used to determine which samples with high confidence can be attached with pseudo-labels, and $l(\,)$ is the cross-entropy loss function. Each client performs E epochs of local training and stochastic gradient descent (SGD) or Adam are used to find the optimal solution. The model parameter ω_i^r is updated by $\omega_i^{r+1} = \omega_i^r - \eta \nabla F_i(\omega_i^r)$, where η is the learning rate and $\nabla F_i(\omega_i^r)$ is the gradient of the loss function with respect to ω_i^r. After completing the local training, each client uploads the updated parameters ω_i^{r+1} to the server, as present in **Step 3**.

Supervised learning at the server: In this paper, unsupervised training on the client side is carried out concurrently with supervised learning on the server side. In other words, after the server distributes the global model parameter ω_g^r to clients, it uses the parameter for supervised learning based on its local labeled data samples $D_s = \{(x_j, y_j) \mid 1 \leq j \leq |D_s|\}$ in parallel. The loss function can be represented as $F_s(\omega_s^r) = \frac{1}{|D_s|}\sum_{j=1}^{|D_s|} l(y_j, p(\omega_s^r, x_j))$, where ω_s^r is the model parameter of the server S at round r.

4.2.3 Model Update. After receiving the local model parameters uploaded by all clients, the server performs global aggregation in **Step 4** to obtain new global model parameter for the next round of training. FedAvg [9] is a mainstream federated learning aggregation function and has been proven to be convergent. In this paper, since we consider a novel FSSL scenario, a new aggregation function is proposed on this basis of FedAvg: $\omega_g^{r+1} = \frac{|D_s|}{|D_c|+|D_s|}\omega_s^{r+1} + \sum_{i=1}^{N} \frac{|D_i|}{|D_c|+|D_s|}\omega_i^{r_i+1}$, where $|D_c| = \sum_{i=1}^{N} |D_i|$. It retains the weighted average idea of FedAvg, while taking the supervised learning at the server into consideration.

Repeat the above steps until the global model converges or a certain specified number of training rounds R reaches.

4.3 Data Arbitration

In the previous section, we obtain the detection model based on the FSSL, and then in this section, we use the trained model for data arbitration. Therefore, a data arbitration algorithm combined with the historical conference is proposed. Each middleware M_i is associated with a historical confidence H_i based on its behavior over a period of time in the past (i.e., a recent window W). Initially, the confidence H_i of each middleware is 1, indicating that our initial trust in the middlewares, because generally speaking, the middleware will not be maliciously invaded when the system is just running. Then, when the data is transmitted through multiple middlewares, the monitor transmits the behavior and status data BD_i of each middleware to the data arbiter for detection, and the trust value of each middleware P_i can be obtained. This trust value P_i of each middleware will be multiplied by its historical confidence H_i as the weight W_i of this middleware in the data arbitration. Next, the weight of middlewares with the same output data D_j are weighted averaged, and then it is used as the approval score S_j for this output data. Finally, the data with the highest approval score is considered as normal data, and the behavior of the corresponding middlewares are normal while others are malicious. At the same time, we update the historical confidence of each middleware, that is, the proportion of normal behavior in the recent W data arbitrations.

5 EXPERIMENTAL RESULTS

5.1 Evaluation Setup

In this paper, we use CIC-IDS 2017 [10] dataset. Since the distribution of different types of samples is extremely unbalanced, we clean and process the dataset, and select 8 attack types, and about 285,000 data from the dataset for experiments. The labeled data at the server accounts for about 10% of the total training samples, and the ratio of training data and test-validation data is 8:2. The MTD system for experiments consists of one data arbiter (i.e., the server) and 5 heterogeneous middlewares (i.e., 5 clients), each of which deploys a Convolutional Neural Network (CNN) model for anomaly detection and data arbitration. Furthermore, in order to evaluate the performance under different data distributions, we design two experimental scenarios, namely the basic and balanced scenarios. The base scenario represents a real MTD system, where some middlewares may perform normal operations while the others may be under various attacks. In the balanced scenario, the data of each middleware is independent and identically distributed, and only the data size of different middlewares varies. Furthermore, to evaluate the detection accuracy of our scheme for common-mode escape, the number of malicious middlewares varies from 1 to 4 at different time periods.

To evaluate the performance of our method more concretely, we first explore the performance of FSSL through the following five metrics: Accuracy, Precision, Recall, F1-Score, and FPR. We then further explore the accuracy of data arbitration, using the Data Arbitration Accuracy (DAA) = the number of correctly arbitrated data / the total number of arbitrated data.

5.2 System Performance

5.2.1 The influence of the data size at the server. As shown in Table 1, with the increase of labeled data at the server, the performance of FedDA improves gradually, this is because the increase of labeled data on the server side can better guide the unsupervised training on the client side. Meanwhile, when the proportion of labeled data is only 1%, even in basic scenarios, FedDA can still achieve 75% detection accuracy, and when the percentage of labeled data is boosted to 12.5%, FedDA can achieve higher than 97% accuracy in both scenarios. At the same time, we also conduct local semi-supervised learning experiments. The experimental results show that the performance of FedDA is close to that of the local semi-supervised learning.

Table 1: Results Summary of Different Data Sizes of the Server

		Accuracy	Precision	Recall/TPR	F1-Score	FPR
Basic	1.75%	0.7487	0.7881	0.5090	0.6063	0.0631
	3.5%	0.7849	0.8090	0.6354	0.7058	0.0175
	6.5%	0.8016	0.8163	0.5356	0.6314	0.0493
	10%	0.9225	0.9225	0.7423	0.8074	0.0250
	12.5%	0.9707	0.9707	0.8548	0.9027	0.0051
Balanced	1.75%	0.8225	0.8350	0.5938	0.6734	0.0553
	3.5%	0.8363	0.8496	0.6147	0.6921	0.0478
	6.5%	0.9543	0.9543	0.8095	0.8654	0.0109
	10%	0.9711	0.9711	0.8827	0.9143	0.0132
	12.5%	0.9752	0.9752	0.8904	0.9219	0.0106
LocalSSL	1.75%	0.8521	0.8521	0.6183	0.6979	0.0499
	3.5%	0.8775	0.8775	0.6654	0.7381	0.0412
	6.5%	0.9663	0.9663	0.8468	0.8941	0.0086
	10%	0.9833	0.9833	0.9103	0.9409	0.0041
	12.5%	0.9839	0.9839	0.9163	0.9439	0.0050

5.2.2 The impact of the number of malicious middlewares. In this section, the data arbitration algorithm in [11] is used as a comparison algorithm, and the results are summarized in Table 2. First, when the number of malicious middleware is in the minority (that is, the number is 1 or 2), both methods can achieve almost perfect data arbitration accuracy. However, when the number of malicious middleware accounts for the majority, the accuracy of the comparison algorithm drops off a cliff. When the number of malicious middleware is 4, its accuracy rate becomes only 3%! In contrast, FedDA still maintains a higher than 90% accuracy. The reason is that the comparison algorithm only uses the output data of each middleware, and it is based on the traditional majority consensus voting algorithm, while FedDA combines the behavior data of each middleware to make arbitration. In addition, the weight of each middleware is averaged based on the output data to get the approval score of each output data, instead of accumulating, which can well resist the adverse impact of common mode escape.

6 CONCLUSION

Due to the complexity of the Internet architecture, there are inevitably security threats, and Moving Target Defense is proposed to solve the above problem. However, we find that most existing studies are mainly based on the majority consensus voting algorithm, which can not cope with the problem of common mode escape, namely there are multiple malicious executives, especially when the malicious executives are in the majority. Therefore, in this paper, we propose a framework for Moving Target Defense based on Federated Semi-Supervised Learning (FSSL), called FedDA. First, we consider a more realistic assumption that the behavior data of heterogeneous executives is all unlabeled, based on which the FSSL is adopted for model training. Then the behavior data of each heterogeneous executive is utilized along with their output data for data arbitration, and a data arbitration algorithm combined with the historical confidence is designed to address the common mode escape. Through Extensive experiments, we demonstrate that FedDA can well resist common mode escape and outperform the state-of-the-art data arbitration algorithms.

Table 2: Data Arbitration Accuracy of Different Number of Malicious Middlewares

		1	2	3	4	Mixed
FedDA	Basic	1.0000	0.9990	0.9950	0.9425	0.9923
	Balanced	1.0000	0.9785	0.9308	0.8263	0.8848
Shen et al. [11]	-	1.0000	1.0000	0.1330	0.0318	0.1980

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under No. U20B2050 and 82004499, and Public Service Platform for Basic Software and Hardware Supply Chain Guarantee under No. TC210804A.

REFERENCES

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[2] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. 2019. Anomaly detection using autoencoders in high performance computing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9428–9433.

[3] Enrique Mármol Campos, Pablo Fernández Saura, Aurora González-Vidal, José L Hernández-Ramos, Jorge Bernal Bernabe, Gianmarco Baldini, and Antonio Skarmeta. 2021. Evaluating Federated Learning for intrusion detection in Internet of Things: Review and challenges. *Computer Networks* (2021), 108661.

[4] Wei GUO, Fan ZHANG, Zhaoqi WU, Jin WEI, Jiangxing WU, and Wenle ZHOU. 2020. Confidence Skewing Problem and Its Correction Method in Mimic Arbitration Mechanism. *Chinese Journal of Electronics* 29, 3 (2020), 547–553.

[5] Wonyong Jeong, Jaehong Yoon, Eunho Yang, and Sung Ju Hwang. 2021. Federated Semi-Supervised Learning with Inter-Client Consistency & Disjoint Learning. In *International Conference on Learning Representations (ICLR) 2021*. International Conference on Learning Representations (ICLR).

[6] Hai Jin, Zhi Li, Deqing Zou, and Bin Yuan. 2019. Dseom: A framework for dynamic security evaluation and optimization of mtd in container-based cloud. *IEEE Transactions on Dependable and Secure Computing* 18, 3 (2019), 1125–1136.

[7] W Li, Z Zhang, L Wang, and J Wu. 2018. The modeling and risk assessment on redundancy adjudication of mimic defense. *Journal of Cyber Security* 3, 5 (2018), 64–74.

[8] Xiangyu Lu, Peng Yi, Youjun Bu, and Bo Chen. 2022. Mimic honeypot based on dual mimicry mechanism. In *Third International Conference on Electronics and Communication; Network and Computer Technology (ECNCT 2021)*, Vol. 12167. SPIE, 83–89.

[9] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[10] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* 1 (2018), 108–116.

[11] Congqi Shen, Shuang-Xi Chen, and Chun-Ming Wu. 2019. A Decentralized Multi-ruling Arbiter for Cyberspace Mimicry Defense. In *2019 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 1–6.

[12] JiangXing Wu. 2022. Problems and solutions regarding generalized functional safety in cyberspace. *Security and Safety* 1 (2022), 2022001.

[13] Yuchen Zhao, Hanyang Liu, Honglin Li, Payam Barnaghi, and Hamed Haddadi. 2020. Semi-supervised federated learning for activity recognition. *arXiv preprint arXiv:2011.00851* (2020).

[14] Jianjun Zheng and Akbar Siami Namin. 2019. A survey on the moving target defense strategies: An architectural perspective. *Journal of Computer Science and Technology* 34, 1 (2019), 207–233.