Detecting Zero-day Attack with Federated Learning using Autonomously Extracted Anomalies in IoT

Takahiro Ohtani, Ryo Yamamoto, and Satoshi Ohzahata
The University of Electro-Communications, Tokyo, Japan
t.ohtani@net.lab.uec.ac.jp, ryo-yamamoto@uec.ac.jp, ohzahata@uec.ac.jp

2024 IEEE 21st Consumer Communications & Networking Conference (CCNC) | 979-8-3503-0457-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCNC51664.2024.10454669

***Abstract*—In recent years, Internet of Things (IoT) has become an essential element of our daily lives. However, IoT devices used in IoT environments have limited available resources due to power and cost constraints, and this fact makes it difficult to implement advanced security measures on them. In fact, zero-day attacks targeting vulnerable IoT devices have occurred, and introducing an anomaly-based intrusion detection system (IDS) that can detect zero-day attacks is one of the countermeasures against the attacks. However, existing methods still suffer from limited detection ability due to a lack of training data. To solve this problem, this paper proposes an intrusion detection method that aggregates zero-day and false positive (FP) attack candidates extracted by an unsupervised anomaly detection algorithm using a one-class classification algorithm and FL. The detection performance evaluation confirms that the proposed method can share the autonomously detected zero-day attacks among IoT networks while suppressing FPs generated during the candidate extraction process.**

***Index Terms*—IoT, Network, Security, Intrusion detection, Zero-day attacks, Federated learning, Machine learning**

I. Introduction

In recent years, the use of Internet of Things (IoT) has rapidly expanded in medical, industrial, and smart home applications. However, IoT devices generally have limited computing and communication resources compared to non-IoT devices due to their power and cost constraints, and this fact makes it difficult to apply robust security measures. In fact, there is malware that forms botnets through attacks including zero-day attacks [1], [2]. A zero-day attack is an attack that exploits a software vulnerability before vendors or other parties take countermeasures. One countermeasure against zero-day attacks on IoT devices is to install a network intrusion detection system (NIDS) that monitors network traffic and alerts network administrators to threats when it detects signs of an intrusion. One of the advantages of NIDS is that it is not required to perform intrusion detection on resource-limited IoT devices since it is attached to networks.

IDS detection methods can be classified into signature-based IDS and anomaly-based IDS. The former detects intrusion based on communication patterns of known attacks, and the latter detects intrusion based on traffic deviations from the normal state of the network. Anomaly-based IDS can detect unknown attacks without updating patterns, whereas signature-based IDS cannot. However, anomaly-based IDS has a drawback that it may increase the false positive rate (FPR) when observations that are originally classified as normal may exceed the predefined normal range [3]. IoT networks are generally composed of heterogeneous devices and the diversity of attacks is higher than that of non-IoT networks [4].

In addition, the distributed learning-enabled anomaly-based IDS allows us to collect attack samples from networks and train intrusion detection models that can detect a wide variety of attacks even when the number of samples per network is small. Therefore, distributed learning is employed to build improved intrusion detection models for the anomaly-based IDS. The combination of distributed learning and anomaly-based IDS enables to collection of attack samples from a large number of networks and training models that detect a wide variety of attacks even when the number of samples per network is small. However, distributed learning has privacy issues and communication overhead due to the direct exchange of training data during model training.

Federated Learning (FL) [5], which builds a global model by aggregating the updates from clients, is one of the key technologies that can secure training data inside the network of clients to address the privacy and overhead issues mentioned above. Thus, anomaly-based IDS and FL in IoT networks have high affinity in terms of resource limitation, number of devices, device diversity, zero-day attack countermeasures, and privacy protection, and some IDSs have been proposed [6], [7], especially for zero-day attacks. However, these IDSs suffer from the fact that attack information cannot be shared between different types of devices, and the phase of extracting and labeling zero-day attacks has not been discussed.

In this paper, we propose a novel FL-enabled anomaly-based IDS for IoT that can eliminate the drawbacks of existing IDS. The unique features of the proposed IDS are the following: 1) autonomous candidate extraction that accepts a certain level of False Positive(FP)s from captured traffic, 2) Applying Online One-Class Classification Support Vector Machine (Online OC-SVM) algorithm [8] to build an intrusion detection model based on the candidates and share the model with FL, 3) Traffic classification based on the model. The proposed IDS realizes an autonomous zero-day attack detection regardless of device type without pre-processing by the features.

II. Related Work

Anomaly-based IDSs face a high FP rate and that becomes even more severe in IoT networks [6]. It is also difficult to build an anomaly-based detection model that can be applied to all behaviors for IoT networks with heterogeneous devices due to the large number of FPs. To address these issues, intrusion

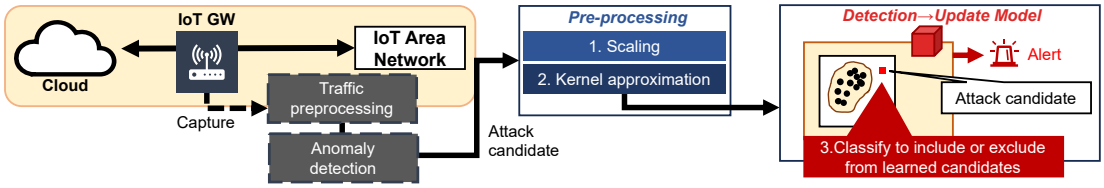


Fig. 1: Overview of the proposed method

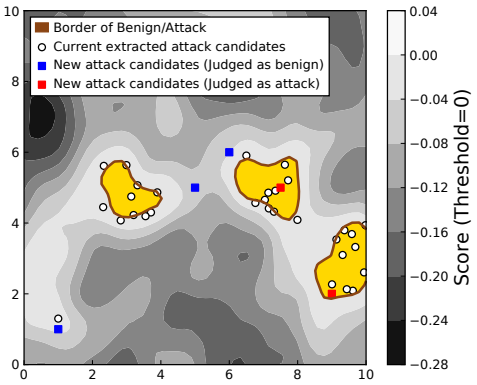


Fig. 2: Example of classification using OC-SVM

detection models for individual device types using Gated Recurrent Units (GRUs) have been proposed. In addition, FL is used to build intrusion detection models to efficiently use the small amount of data collected from each IoT network. The detection performance evaluation shows that the method could achieve a high detection performance under the environment with Mirai-infected IoT devices. However, the method requires data from multiple devices for building a model for a single device type to improve detection performance, and the attack information cannot be shared among different device types. Although zero-day attacks target devices running specific vulnerable software, there are cases where common software is running on different device types.

The fact that the estimated total amount of data generated by IoT networks in the entire world is about 79.4 ZB makes it difficult to collect training data and build an intrusion detection model on a central server after installing IDSs on each IoT network due to network resource constraints [7]. Therefore, a method that learns zero-day attacks detected by each IoT edge device using a Deep Neural Network (DNN) and shares the learned models among devices using FL has been proposed. the detection performance evaluation demonstrated that an attack unknown to one device could be detected by other devices that learn similar attacks by sharing various types of attacks through FL. However, there is no discussion about traffic data labeling within devices.

FL can share zero-day attack information efficiently and enables attack detection on each network.However, autonomous detection and sharing of attack information among devices in a shared environment with FL is impossible. Therefore, this paper proposes a method to build an intrusion detection model by aggregating attack candidates (TPs and FPs) using Online OC-SVM and sharing the intrusion detection model by FL to realize autonomous detection of zero-day attacks in IoT networks regardless of device type.

III. Proposed Method

A. Overview

As shown in Fig. 1, this paper assumes that the proposed method extracts FP-included attack candidates from the captured traffic by IoT gateways (GWs) or other devices, and intrusion detection is performed against the attack candidates. The intrusion detection process consists of a pre-processing phase, a detection and model update phase, and a model aggregation phase. The building of an intrusion detection model based on the attack candidates and the detection process are performed using Online OC-SVM.

OC-SVM is an anomaly detection algorithm that learns past normal conditions and determines whether new data is normal based on them. OC-SVM can also determine whether a new attack candidate is already included in the past attack candidates. Fig.2 shows a graphical example of the determination procedure of OC-SVM. The IoT networks in which the proposed method is installed communicate with each other using FL to aggregate the intrusion detection models built in each network. The aggregated models facilitate the clustering for candidates that are true attacks, that is, prevent the clustering from including FP attack candidates.

B. Pre-processing phase

The pre-processing is applied to extracted attack candidates that include FPs by existing methods such as anomaly detection by unsupervised learning, and the candidates after pre-processing are used in the next phase. In this phase, scaling and kernel approximation are applied to the attack candidates.

The scaling aims to unify the scales among the features and improve the learning performance of OC-SVM. In scaling, Min-Max normalization is applied to each feature of the attack candidates in the range between 0 and 1. However, it is unable to define the scale used for normalization in real-time detection. Consequently, we define the scale based on the typical values of each feature.

Kernel approximation aims to make nonlinear data classifiable on a linear SVM. By applying kernel approximation to training and inference data, it becomes possible to solve linearly inseparable problems with a linear SVM, which requires less training cost than a nonlinear SVM. A linear SVM is a parametric model, and using linear SVM in intrusion detection models facilitates the exchange of weights by FL. Furthermore, a linear SVM enables online learning which allows it to update existing models only with new training data. This capability is crucial for real-time intrusion detection. In the proposed method, the Random Fourier Features (RFF) [9] algorithm is applied to attack candidates to which scaling has been applied.

TABLE I: Dataset (**Evaluation 1**)

Device	Device Name	Attack Type	Train data points (Attacks)
D_1	Danmini	mirai-scan	157,233 (107,685)
D_2	Ennio	gafgyt-tcp	108,134 (95,021)
D_3	Philips B120N/10	mirai-ack	266,363 (91,123)
D_4	Provision PT-737E	mirai-udpplain	118,835 (56,681)
D_5	Provision PT-838	mirai-scan	195,610 (97,096)
D_6	SimpleHome XCS7-1002-WHT	gafgyt-junk	80,455 (28,305)

TABLE II: **Evaluation 1** Result

Device	AUC	TPR	TNR	F1
D_1	0.999	0.999	0.997	0.999
D_2	0.999	0.999	0.998	0.999
D_3	0.999	0.966	0.999	0.982
D_4	0.998	0.979	0.998	0.989
D_5	1.000	0.995	0.998	0.971
D_6	0.015	0.000	0.998	0.001

C. Detection and update model phase

In this phase, the intrusion detection model determines whether the input attack candidates pre-processed in the last phase are included in the past attack candidates. Then, the new candidates are used as the input of online learning for the intrusion detection model training and update regardless of the authenticity of the attack. The proposed method uses online OC-SVM, which supports online learning, to build the intrusion detection model since it can determine whether new input data belongs to the learned class.

D. Model aggregation phase

The previous phase enables intrusion detection model building and intrusion detection using the extracted attack candidates within a single IoT network. In addition, the effect of detection performance degradation due to the inclusion of candidates that are FP is reduced by continuously updating the model based on the attack candidates, and improving the model autonomously. However, as mentioned above, there are concerns about FP detection of normal traffic in IoT networks due to the diversity of attacks across networks and the small amount of traffic in each network. Thus, the proposed method shares the intrusion detection model built in the previous phase among IoT networks using FL.

IV. Experimental Evaluation

A. Evaluation Environment

The proposed method was implemented using Python with scikit-learn and Flower [10] libraries.

For the detection performance evaluation, we assume that the existing method extracts attack candidates including FP, and evaluate the intrusion detection performance of the proposed method by processing these attack candidates according to each phase of the proposed method. We use the N-BaIoT [11] dataset for the evaluation. N-BaIoT is based on actual attacks against nine types of real devices with the Mirai botnet and the Gafgyt botnet in an IoT network environment.

For the evaluations, the attack and normal data in N-BaIoT are first combined, and the order of normal and attack data is randomly shuffled. The dataset is then scanned from the head and we regard the attacked traffic rows as attack candidates. However, as previously mentioned, FPs are expected in unsupervised anomaly detection, and thus, rows that do not contain attacks are added to the pool of attack candidates with a certain probability. The proportion of FPs in the pool is changed to evaluate the impact on the detection performance.

The dataset processed in the above procedure is divided into training data (75%) and test data (25%) for extracting attack candidates with the correct label. The training data is then used to extract attack candidates and build intrusion detection models with the proposed method. The intrusion detection is performed using the models built for the test data to evaluate the detection performance.

B. Evaluation Metrics

The area under the ROC curve (AUC) and F_1 are used as evaluation metrics. The AUC can explain the performance of the classifier without the influence of threshold values. The ROC curve is a curve with the False Positive Rate (FPR) on the X-axis and the True Positive Rate (TPR) on the Y-axis. In the performance evaluation, ROC curves are generated from the output values of the decision function in OC-SVM. The F_1 is the harmonic mean of the precision and the recall, and a value closer to 1 represents higher prediction performance.

C. ***Evaluation 1:*** *Detection performance of zero-day attacks*

Evaluation 1 aims to evaluate whether the proposed method can provide an effective detection model for zero-day attacks, and assumes that there are six IoT networks and one IoT device for each network. Table I shows the device in each network, the total data points, and the attack types in the dataset. We assume that D_1–D_4 are subjected to different types of attacks and that the proposed outlier detection method can extract attack candidates. Each client builds an intrusion detection model based on the candidates and sends the model weights to the central server D_{C}. D_{C} distributes the global model weights to D_1–D_6. In this evaluation, D_1–D_6 shares the same intrusion detection model without sharing the training data, and D_5–D_6 does not contribute to learning.

Table II shows the intrusion detection performance of each device after sharing the global model. The table shows that D_5 has high detection performance and this indicates the proposed method can detect attacks that are new to the network by sharing the model. It also shows the TPR is 0 and the True Negative Rate (TNR) is close to 1 for D_6. This is an expected result since the attack on D_6 is not included in the model.

D. ***Evaluation 2:*** *Detection performance when FP is included*

Evaluation 2 aims to confirm that the proposed method can suppress the detection performance degradation caused by FPs, and assume four IoT networks and one IoT device for each network. D_1–D_4 first learns attack candidates detected by the one-class classification algorithm to build detection models, and then they communicate with the central server for FL. The parameters are the same as in **Evaluation 1**.

Fig. 3–6 show the AUC score and F1 score, and the scores for each client and the average of 20 measurements are shown as dots as well as the standard deviation shown in error bars.

Fig.3 and 5 show that the AUC for most attacks in Mirai and Gafgyt improves by about 10% when FPs are included. Fig.4 and 6 show that the proposed method improves the detection performance when FPs are less than 10%. On the other hand, the intrusion detection model with FL performs better when FPs exceed 10% in some cases due to the degradation of precision. This is because FP candidates with similar characteristics tend to be aggregated by FL and this leads FL to misclassify truly normal features as attack. To address the issue, excluding attack candidates similar to the ones once identified as attack candidates from subsequent anomaly detection to prevent FL from learning them as novel candidates can be an effective strategy.

V. Conclusion

In this paper, we proposed a method for intrusion detection by aggregating attack candidates using a one-class classification algorithm and federated learning in order to solve existing problems in zero-day attack detection using federated learning in IoT networks. In the future, we plan to investigate a method for measuring the detection performance including the step of extracting the attack candidate, measuring the detection performance for various attack scenarios, and measuring the processing performance using practical IoT devices.

References

[1] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110.

[2] "NJCCIC threat profile Satori," New Jersey Cybersecurity Communications Integration Cell, Jan. 2018, "Accessed: July 10, 2023". [Online]. Available: https://www.cyber.nj.gov/threat-center/threat-profiles/botnet-variants/satori

[3] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1–11, Aug. 2018.

[4] D. Swessi and H. Idoudi, "A survey on Internet-of-Things security: Threats and emerging countermeasures," *Wireless Personal Communications*, vol. 124, no. 2, p. 1557–1592, May 2022.

[5] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, Feb. 2017, pp. 1273–1282.

[6] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Jul. 2019, pp. 756–767.

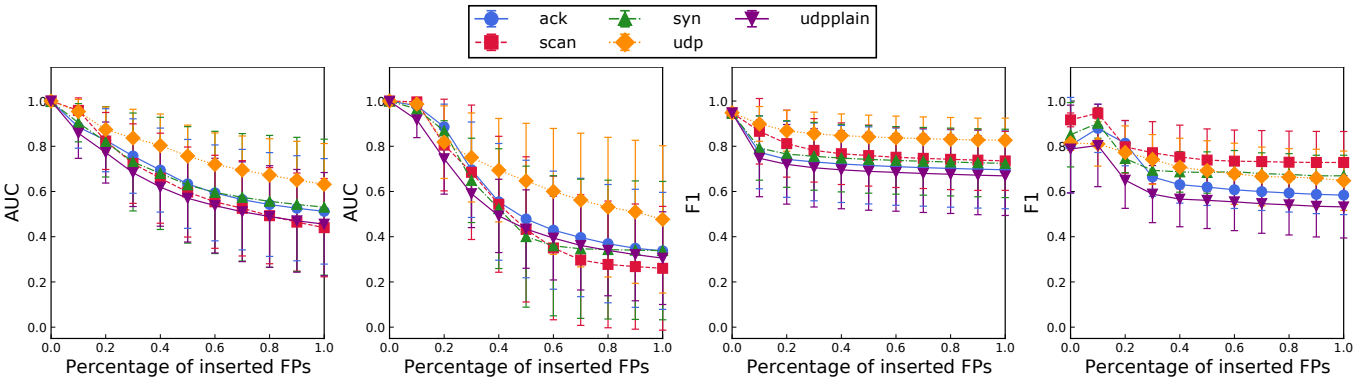
[7] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3930–3944, Jul. 2022.

[8] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[9] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," vol. 20, Jan. 2007, pp. 1177–1184.

[10] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," *CoRR*, vol. abs/2007.14390, Jul. 2020. [Online]. Available: https://arxiv.org/abs/2007.14390

[11] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, Oct. 2018.

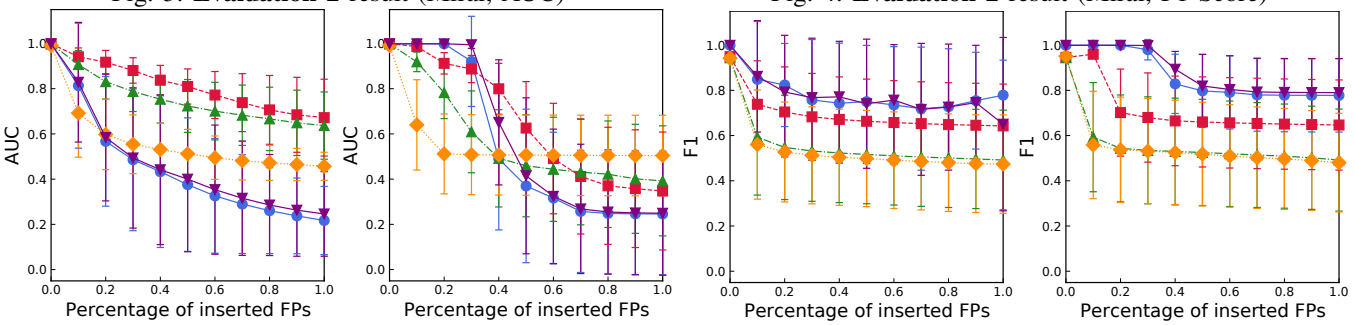


(a) without FL (b) with FL

Fig. 3: **Evaluation 2** result (Mirai, AUC)

(a) without FL (b) with FL

Fig. 4: **Evaluation 2** result (Mirai, F1 Score)



(a) without FL (b) with FL

Fig. 5: **Evaluation 2** result (Gafgyt, AUC)

(a) without FL (b) with FL

Fig. 6: **Evaluation 2** result (Gafgyt, F1 Score)