Mitigating Model Replacement Attacks in Federated Learning-based Intrusion Detection Systems

Pavlos S. Bouzinis*, Panagiotis Radoglou-Grammatikis†‡, Nikos A. Mitsiou‖, Thrassos K. Oikonomou‖, Vasiliki Koutsioumpa‖, Georgios Th. Papadopoulos**, Vasiliki Kotsirou†,George K. Karagiannidis‖, and Panagiotis Sarigiannidis†

***Abstract*—Federated learning (FL) is a decentralized approach that enables devices to collaboratively train a shared deep learning (DL) model while keeping their raw data private and not disclosing it to a third party. Thanks to its privacy-preserving principles, FL has gained significant interest for designing intrusion detection systems (IDS) in the field of cybersecurity. However, FL is susceptible to malicious participants who may attempt manipulate or undermine the global model. One such example is the model replacement attack, where a malicious participant aims to replace the global model with a malformed one. This altered model may contain a backdoor that misclassifies certain instances while behaving normally for others. In this work, we design an FL-based IDS that is robust against model replacement attacks, whose core defense mechanism is based on weight clipping during the server's aggregation step. The evaluation results on the CIC-IoT-2023 dataset demonstrate the effectiveness of the proposed approach in suppressing the attacker's backdoor, while maintaining the detection accuracy of the FL-based IDS.**

***Index Terms*—federated learning, intrusion detection systems, model replacement attacks, backdoor attacks**

I. Introduction

Although the Internet of Things (IoT) offers numerous benefits to modern life, it also faces significant and unconventional security challenges. The heterogeneity of IoT devices, their dependence on a wide range of technologies, such as fog and cloud computing, and the use of various communication protocols, combined with the inconsistent resource capabilities of these devices, all contribute to the IoT's vulnerability to security threats. In this challenging landscape, recent advancements in artificial intelligence (AI) and machine learning (ML) have demonstrated promising potential for enhancing intrusion detection systems (IDSs) through the integration of these technologies. However, these centralized AI solutions typically require collecting sensitive network data from multiple devices, creating privacy concerns.

One particularly promising privacy-preserving approach is federated learning (FL) [1], a decentralized training paradigm that enables multiple devices to collaboratively develop AI-based IDS models without sharing their raw training data and keeping them localized. Although FL has gained significant attention for the design of privacy-preserving IDSs, it also faces security threats in the training process itself due to participation of multiple, potentially untrusted devices with malicious motives.

Common attacks in FL-based IDS include poisoning attacks, where adversaries manipulate local training data to corrupt the global model, and backdoor attacks, where malicious participants introduce hidden triggers during local training that cause the model to misclassify specific inputs without affecting overall performance. Moreover, another type of attack in FL settings is the *model replacement attack* [2], in which malicious clients attempt to directly overwrite the global model with a compromised version, often aiming to insert backdoor tasks that bypass detection during inference or subtly corrupt the intended model behavior. However, to the best of our knowledge, this attack and the design of the corresponding defense mechanism in the context of IDS remains understudied.

A. Contribution

In this work, we address the critical gap in defending against model replacement attacks in FL-based IDS, where malicious actors aim to substitute the global model with a compromised version during FL training. Unlike prior work focused on general data poisoning or label-flipping attacks [3]–[6], we tackle the more aggressive model replacement scenario, where

‡‡This work has received funding from the European Union's Horizon Europe research innovation action programme under grant agreement No. 101135423 – ENACT and the the National Recovery and Resilience Plan 'Greece 2.0' under the European Union's NextGenerationEU framework (ΑΣΚΟΣ - Grant Agreement no. ΥΠ3ΤΑ-0560864).

* Pavlos S. Bouzinis is with MetaMind Innovations, Kila Kozani, 50100, Kozani, Greece - E-Mail: pbouzinis@metamind.gr

† P. Radoglou-Grammatikis, V. Kotsirou and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Campus ZEP Kozani, 50100, Kozani, Greece - E-Mail: pradoglou@uowm.gr; vaso-kots@hotmail.com; psarigiannidis@uowm.gr

‡P. Radoglou-Grammatikis is also with K3Y Ltd, Studentski District, Vitosha Quarter, Bl. 9, 1700 Sofia, Bulgaria - E-Mail: pradoglou@k3y.bg

‖N. A. Mitsiou, T. K. Oikonomou, V. Koutsioumpa, G. K. Karagiannidis is with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54124, Thessaloniki, Greece - E-Mail: nmitsiou@ece.auth.gr; toikonom@ece.auth.gr; vasioakou@ece.auth.gr; geokarag@auth.gr

‡G. Th. Papadopoulos is Department of Informatics and Telematics Harokopio University of Athens, Omirou 9, Tavros, Athens, Greece, GR17778 Greece - E-Mail: g.th.papadopoulos@hua.gr

adversaries send crafted model updates to instantly replace the global model with a backdoored one. Our key contributions are:

- We formalize the model replacement attack in the context of FL-based IDS, where an adversary manipulates local model weights to inject a backdoor (e.g., misclassifying SYN-flood attacks as normal traffic) while preserving performance on non-targeted tasks.
- We utilize a weight clipping-based defense during server-side aggregation, dynamically clipping updates to suppress malicious weight magnitudes without relying on methods that exclude clients using similarity metrics, which may bias benign clients.
- We validate our approach on the CIC-IoT-2023 dataset, demonstrating that the defense suppresses backdoor accuracy, while maintaining main-task detection performance, even under repeated attacks.

B. Related Work & Motivation

In this subsection, we review studies that have investigated adversarial attacks in FL-based IDS, mainly targeting data poisoning and backdoor attacks, using them as a key motivation for our work.

First, in [3], authors investigate data poisoning attacks in FL-based IDS, where the attacker flip certain labels of the benign dataset, with the aim to introduce backdoor tasks. This backdoor aims to corrupt the global FL model to falsely predict targeted inputs as normal. Evaluation results show that the attack bypasses defense mechanisms that rely on differential privacy and norm-based malicious update detection. Additionally, [4] investigates data poisoning attacks based on both label flipping and manipulation of feature values associated with the network traffic. Thus, it is considered that malicious clients send model updates to the server, locally trained on the poisoned data. Defense mechanisms against the poisoning attacks are considered at both the model and data level. The former checks for model updates that are highly likely to be malicious and rejects them, while the latter aims to filter out poisoned data present on the local dataset. The evaluation results validate the efficiency of the aforementioned defense mechanisms. Following that, [5] also proposed a two-phase defense mechanism against poisoning or backdoor attacks. In the initial phase, clients are categorized as benign, malicious, or uncertain based on their local weight updates. In the subsequent phase, the server aggregates the weights of clients within each group according to their predicted identity, ultimately making a decision regarding the uncertain clients and determining whether they are benign or not, relying on the accuracy of the aggregated models. Moreover, [6] again proposes a defense mechanism for label-flipping attacks, that is based on a similarity metric considering the loss function of the local model updates. In addition, [7] investigates the generation of poisoned data samples, in the feature space, by using generative adversarial networks. To counter this attack, authors propose a defense mechanism that detects poisoned samples, relying on the important neurons of local models, followed by an unsupervised anomaly detection on these important activations. Finally, [8] proposes a personalized FL-based IDS framework, while using a detector at the server side which identifies poisoned clients and limits their participation, by relying on weight similarity metrics.

Although previous studies have made progress in addressing various aspects of poisoning and backdoor attacks in FL-based IDS, several challenges remain open and require further exploration. A specific type of attack, the model replacement attack [9], as well as the appropriate defense mechanisms to counter it, has not received attention in the FL-based IDS literature. In the model replacement attack, the attacker aims to substitute the global FL model with a malicious one, by modifying the local model weights accordingly to achieve this. Previous works [3]–[6], [8] have focused on demonstrating the effects of a poisoned dataset or a backdoor task, implicitly introduced through standard FL training. In these cases, the attacker simply trains the local model on a malicious dataset in each FL round. However, such approaches are effective only when the attacker possesses a relatively large proportion of the training dataset, which may be uncommon. On the contrary, a model replacement attack aims to directly replace the global model with a malicious one during the targeted round of the attack. It is also noteworthy that existing studies adopt defense mechanisms that primarily rely on excluding clients that submit "anomalous" model updates to the server. However, such methods contradict the core motivation of FL, which assumes participation of heterogeneous clients with different data distributions, leading to dissimilar model weights. Therefore, defense mechanisms that rely on weight similarity metrics [4]–[6], [8], may introduce bias against certain benign clients. Lastly, while [3] investigated defense mechanisms based on weight norm-clipping, their study did not account for attacks involving weight scaling. As a result, the analysis did not fully explore the effectiveness of such defenses against this type of manipulation, that is related with model replacement attack.

II. Model Replacement Attack

First, we consider a standard FL setup with a central server acting as coordinator. Let $\mathcal{N} = \{1, 2, ..., N\}$ be the set of FL clients indexed by i. Each client possesses a local dataset $\mathcal{D}_i$, with size $D_i = |\mathcal{D}_i|$, while the size of the overall dataset is expressed as $D = \sum_{i \in \mathcal{N}} D_i$. Denote the local model parameters of client i at the t-th FL round, as $\boldsymbol{w}_i^t$, and the global model as $\boldsymbol{w}^t$. According to the standard FL process, in each round t, the clients train their local model $\boldsymbol{w}_i^t$ on their own dataset and send the difference $\Delta \boldsymbol{w}_i^t = \boldsymbol{w}_i^t - \boldsymbol{w}^t$ to the server, which updates the global model as follows:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \sum_{i \in \mathcal{N}} p_i \Delta \boldsymbol{w}_i^t, \tag{1}$$

where $p_i = \frac{D_i}{D}$ is the proportion of data samples owned by client i relative to the entire dataset. The whole procedure is repeated for a predefined number of rounds or until a convergence criterion is met. At the end of the FL training process, the objective is to generate a global model that will serve as an IDS, designed to detect cyberattacks and anomalies.

A. Attacker Assumptions and Backdoor Objective

We assume the presence of a single attacker throughout the overall FL process. Given that the attacker has compromised client i and has complete control over it, it has the following capabilities:

i) The attacker has full access to the local dataset $\mathcal{D}_i$. Therefore, it can modify $\mathcal{D}_i$ by any means (e.g., change labels, add new samples, modify feature values, etc.) and generate a malicious dataset $\mathcal{D}_{\text{mal}}$.

ii) The attacker has complete control over the local training process of the impersonated client. This allows them to adjust training hyperparameters, alter local model weights, and transmit any modified weights to the server.

The objective of the attacker is to create a backdoor task, ensuring that the global FL model fails on specific targeted tasks while still performing successfully on others. In the context of this work, the attacker can manipulate certain target labels of $\mathcal{D}_i$. For instance, considering a dataset $\mathcal{D}_i$ suitable for training a network IDS, network traffic samples representing SYN-flood DoS attacks may be label-flipped to appear as normal in $\mathcal{D}_{\text{mal}}$. Consequently, the malicious client's backdoor task is to ensure that SYN-flood DoS attacks go undetected. However, for instance, UDP flooding attacks may still be detected since the attacker has no intention of interfering with their detection. Therefore, through this scheme, the attacker seeks to bypass SYN-flood DoS attacks without detection by the IDS, weakening its effectiveness and making it vulnerable to this type of attack. To achieve its goals, the attacker should train an optimal malicious backdoor model $\boldsymbol{w}^*_{\text{mal}}$ by using the altered dataset $\mathcal{D}_{\text{mal}}$. This backdoor model should perform well at the backdoor task, while minimizing disruption to other tasks that do not conflict with it. It is noteworthy that the attacker has no control of the aggregation process, which is solely coordinated by the server. Hence, the only possible way to affect the FL training process is by sending a malicious update to the server, as described in the following subsection.

B. Description of the Attack

We assume model update poisoning attacks that rely on the model replacement method [2]. Specifically, given that one attacker is present in a round t (assuming that it is client $i = 1$), the attacker aims to replace the global model with its malicious backdoored model $\boldsymbol{w}^*_{\text{mal}}$, by sending to the server the following update

$$\Delta \boldsymbol{w}_1^t = \beta \left(\boldsymbol{w}^*_{\text{mal}} - \boldsymbol{w}^t \right). \tag{2}$$

Here, $\beta = \frac{1}{p_1}$ is a boost factor that facilitates the malicious model replacement. This choice of β adjusts the weight magnitude of the backdoored malicious model, ensuring that it persists through the server's global averaging and ultimately replaces the global model [2]. To see this more clearly, the aggregation update of the server is given as

$$\begin{aligned} \boldsymbol{w}^{t+1} &= \boldsymbol{w}^t + p_1 \Delta \boldsymbol{w}_1^t + \sum_{i=2}^{N} p_i \Delta \boldsymbol{w}_i^t \\ &= \boldsymbol{w}^*_{\text{mal}} + \sum_{i=2}^{N} p_i \Delta \boldsymbol{w}_i^t \approx \boldsymbol{w}^*_{\text{mal}}, \end{aligned} \tag{3}$$

where the last approximation arises from the fact that, in later rounds, the global model has nearly converged, implying that $\Delta \boldsymbol{w}_i^t \approx \mathbf{0}$, $\forall i > 1$. Therefore, the attacker can theoretically replace the global model with its malicious backdoored one, with the attack being more effective at the late stage of the FL process, when the global model reaches convergence. Moreover, it's important to note that the boost factor β is influenced by the number of users and their contribution to the FL training, particularly in terms of data size. While the attacker may not have direct access to this information, it can iteratively increase β across attack attempts during multiple rounds and assess which value of β leads to higher backdoor task accuracy, by testing the global model obtained on the respective round. It is also noted that if multiple attackers emerge in the same round, they can coordinate and properly adjust their β's to distribute the update evenly among themselves, while the intended goal of model replacement can be still satisfied. Finally, the training of $\boldsymbol{w}^*_{\text{mal}}$ can be performed either offline, before the actual FL training process, or online, i.e., the attacker can train progressively $\boldsymbol{w}^*_{\text{mal}}$ along with the evolution of the FL rounds. In both cases, the objective remains the same, that is, to replace the global FL model with $\boldsymbol{w}^*_{\text{mal}}$.

Finally, the attacker can define an attack policy $\mathcal{P}$ that determines when to launch the attack, as well as any additional conditions that may apply. For example, the attacker may choose to launch the attack every K rounds, in which case the policy can be defined as: $\mathcal{P}^t = \mathbb{1}_{\{t \mod K = 0\}}$, where $\mathbb{1}$ is the indicator function, returning the value 1 if the condition in the brackets is satisfied and 0 otherwise. Although this example shows that $\mathcal{P}^t$ is a function of t, the attacker may choose any policy $\mathcal{P}$ of interest, while this choice is left arbitrary.

C. Defense Mechanism

Since the weight re-scaling operation of the attacker introduces noticeable differences in magnitude between backdoored and benign updates, [2] proposes the norm clipping defense, which is integrated on the servers' aggregation process. This method restricts the norm of local updates before the aggrega-

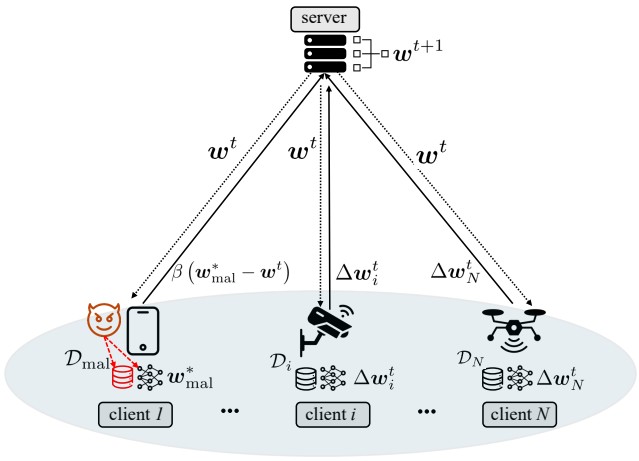

Fig. 1. Illustrative example of model replacement attack in FL.

tion by clipping them to ensure their l_2 norm does not exceed a predefined threshold, M, as shown below.

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t + \sum_{i\in\mathcal{N}} p_i \frac{\Delta\boldsymbol{w}^t_i}{\max\left\{1, \left\|\Delta\boldsymbol{w}^t_i\right\|_2 / M\right\}} + \boldsymbol{n}. \quad (4)$$

Note that, being inspired by differential privacy techniques, [2] also suggests adding Gaussian noise after clipping the updates, as an additional measure of defense. The noise is denoted as $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I})$, with zero mean and variance σ^2, where $\boldsymbol{I}$ is the identity matrix. Regarding the clipping threshold M, it is usually fixed throughout the training, while its choice may be arbitrary and relies on trial-and-error methods. Intuitively, M should be selected in a way such that the magnitude of the malicious updates are restricted. However, by setting a too small value, it is likely that the benign updates will be clipped as well, which potentially hinders convergence or leads to sub-optimal solutions. Empirically, [10] showed that a reasonable value for M is in the order of the average l_2 norm of clients' updates, i.e., $\mathbb{E}_{i\in\mathcal{N}}[\|\Delta\boldsymbol{w}^t_i\|_2]$. Therefore, M can be defined as a function of time

$$M^t \triangleq \alpha \cdot \mathbb{E}_{i\in\mathcal{N}}\left[\left\|\Delta\boldsymbol{w}^t_i\right\|_2\right], \quad (5)$$

where α is a hyperparameter that adjusts the magnitude of M^t. This strategy implies that M^t varies over rounds, since the average l_2 norm of the updates may fluctuate as well. The impact of α on the effectiveness of the defense mechanism will be evaluated in the experimental results section.

Algorithm 1 summarizes the model replacement attack with the integration of the defense mechanism by the server, while Fig. 1 provides an illustrative representation. Notice that from step 8, it is evident that the attacker launches the attack following policy $\mathcal{P}$. If the policy conditions are not met, we assume that the attacker acts as a legitimate client. In this case, the attacker trains the model on the benign dataset and submits a valid update to the server, as demonstrated in steps 11 and 12. Finally, observe that in Algorithm 1, $\boldsymbol{w}^*_{\text{mal}}$ is trained offline, before the actual FL process. As stated previously, an alternative option for the attacker is to progressively train $\boldsymbol{w}^*_{\text{mal}}$ in parallel with the evolution of the FL training. In such a case, the attacker should include a training update before submitting the malicious update in step 9. However, this approach may not be very effective for a model replacement attack, as the malicious model will lack sufficient effectiveness and will be more difficult and slow to train, considering it is being influenced by benign updates in every round.

Algorithm 1 Model replacement attack and server's defense

1: server **initializes** global model $\boldsymbol{w}^0$
2: attacker **defines** $\mathcal{P}$, $\mathcal{D}_{\text{mal}}$, β, and **trains** $\boldsymbol{w}^*_{\text{mal}}$ on $\mathcal{D}_{\text{mal}}$
3: **for** $t = 0, 1, 2, \ldots.$ **do**
4: server **sends** $\boldsymbol{w}^t$ to all clients
5: **for** each client $i \in \mathcal{N} \setminus \{1\}$ **do**
6: **train** local model $\boldsymbol{w}^t_i$ on $\mathcal{D}_i$
7: **send** $\Delta\boldsymbol{w}^t_i = \boldsymbol{w}^t_i - \boldsymbol{w}^t$ to the server
8: **if** $\mathcal{P}$ **then**
9: attacker **sends** $\beta\left(\boldsymbol{w}^*_{\text{mal}} - \boldsymbol{w}^t\right)$ to the server
10: **else**
11: attacker **trains** local model $\boldsymbol{w}^t_1$ on $\mathcal{D}_1$
12: attacker **sends** $\Delta\boldsymbol{w}^t_1 = \boldsymbol{w}^t_1 - \boldsymbol{w}^t$ to the server
13: server **computes** $\boldsymbol{w}^{t+1}$ according to (4)
14: **output**: global model $\boldsymbol{w}$

III. Experiments

A. Dataset

The experiments are conducted in the CIC-IoT-2023 Dataset [11], which is a realistic IoT attack dataset stemming from a detailed topology with multiple IoT devices classified as either attackers or targets. It includes 48 features, which are defined by metrics such as packet flow statistics, used application layer protocols, TCP flags, and more. For the sake of brevity, we do not provide an exhaustive list of all the features; additional details can be found in [11]. The dataset also classifies attacks into the following 8 types: "Brute force", "dDoS", "DoS", "Mirai", "Recon", "Spoofing", "Web-based", and "Normal", where the latter indicates benign traffic. Therefore, our main classification task is to identify the aforementioned type of attacks. In the original dataset, each attack type is further subdivided in additional categories. For example, "DoS" attack can occur as "TCP Flood", "HTTP Flood", "SYN Flood", or "UDP Flood". While we do not use this subcategorization in the classification task, the attacker leverages this information to manipulate the compromised client's dataset, as we will explain later.

B. Backdoor Task

The attackers backdoor task is to replace the global model with its malicious one, with the aim to classify "SYN Flood"

labeled instances as "Normal", instead of "Dos". As such, the attacker modifies the compromised client's dataset $\mathcal{D}_1$, by flipping the "DoS" label corresponding to a "SYN Flood" attack, to "Normal". This modification leads to the generation of the malicious dataset $\mathcal{D}_{\text{mal}}$. The motive of such behavior is to force the trained FL-based IDS to bypass "SYN Flood" attacks, while still functioning as intended for all other attacks. This approach increases the likelihood that the model remains vulnerable to future attacks that exploit SYN flooding.

C. Experiment Settings

1) General: The number of clients has been set as $N = 10$, while all clients participate in each FL round. The number of rounds is set to 40. Each client receives an equal proportion, $p_i = 0.1$, of the original dataset. Because of the significant class imbalance in the dataset, each client creates synthetic instances for the minority classes in the training set using SMOTE [12]. The clients normalize their local datasets using standard scaling, relying on a single scaler fitted on the overall dataset $\mathcal{D}$. Although this approach may not be entirely practical, we use the same scaler across all clients to avoid issues related with non-identical and independent distributions (non-IID) of data among clients, as it lies outside the scope of this work. However, techniques like StatAvg [13], can be seamlessly integrated into our approach, as it accounts for individual scaler per client and tackles potential non-IID data challenges. As regards model design and training, the local model of each client is a neural network consisting of 3 fully connected hidden layers with 128 neurons, followed by ReLU activation. Batch size was set to 512 and each client performed one local epoch per FL round. We used the Adam optimization algorithm with a learning rate of 0.008 and exponential decay rates for the moment estimates were set to 0.99 and 0.999, respectively.

2) Attack Simulation: We consider the presence of a single attacker that has compromised the 1-st client. We assume that, the attacker launches the malicious model with the aim to replace the global model every 5 rounds, meaning that its policy is described as $\mathcal{P}^t = \mathbb{1}_{\{t \mod 5=0\}}$. The server sets the hypermeter $\alpha = 3$ for the clipping threshold, unless specified otherwise. Finally, the standard deviation of the Gaussian noise added to the server's aggregation update is $\sigma = 0.01$.

D. Evaluation Results

In what follows, all the demonstrated results have been averaged over 3 individual runs. Moreover, for a fair comparison, the test set is balanced, meaning that all classes are equally represented. Note that the evaluation metrics include the testing accuracy of the global model both in the main task and the backdoor task, as described below:

- **Main task**: This represents the model's core intrusion detection capability, measured as the classification accuracy across all attack categories and normal traffic, where
$$\text{ACC} = \frac{\text{correctly classified samples}}{\text{total test samples}}. \tag{6}$$
This metric reflects the IDS's fundamental detection capability that must be preserved under attack scenarios.
- **Backdoor task**: It measures the attacker's objective by calculating the percentage of targeted "SYN Flood" attack samples that are misclassified as "Normal", specifically:
$$\text{ACC} = \frac{\text{"SYN Flood" samples classified as "Normal"}}{\text{total "SYN Flood" samples}}. \tag{7}$$

Firstly, Fig. 2 showcases the evolution of testing accuracy across FL rounds in the absence of an attack, which also serves as a baseline for the subsequent evaluation of the attack's impact. Moreover, the indicated curves incorporate the defense mechanism on the server side for $\alpha = 1$ and $\alpha = 3$, adjusting the clipping threshold accordingly. It is evident that for an appropriate choice of α (e.g., $\alpha = 3$), the inclusion of the defense mechanism does not decrease the testing accuracy of the main task. This is crucial to ensure that the defense mechanism does not disrupt normal operation when an attack is absent. However, it may slightly slow down the convergence compared to the vanilla model aggregation. In contrast, a smaller value of α (e.g., $\alpha = 1$) results in an accuracy gap of $\sim$2%, due to the weight clipping effect. As previously noted, setting α too low can cause benign updates to be clipped as well, potentially resulting in suboptimal solutions. Thus, α should be selected in a way that ensures that performance does not degrade under normal operation, as this is the typical mode in which the FL training functions, i.e., usually all clients are benign.

Next, Fig. 3 shows the effect of the attack when no defense mechanism is applied on the server's side. It is evident that the accuracy of the backdoor task is high during the particular rounds in which the attack was launched, which reaches $\sim$82%. This indicates that the global model effectively misclassifies "SYN Flood" DoS attacks as benign traffic. On the other hand, while the main task accuracy experiences a noticeable decline during these malicious updates, it does not deteriorate entirely and begins to recover once normal training resumes, suggesting that the attack selectively compromises the model's behavior on the target class without completely undermining its general functionality. This periodic yet substantial manipulation of model performance confirms both the feasibility and effectiveness of the backdoor attack in unsecured federated learning environments.

Finally, Fig. 4 demonstrates the main task and backdoor task accuracy of the global model, when applying the defense mechanism. It can be observed that the defense mechanism completely suppresses the attack, as the backdoor accuracy reaches 0%. Additionally, while the main task accuracy experiences a

slight decline during the attack rounds, this decrease is minimal and diminishes as the rounds progress. Furthermore, the overall performance remains nearly identical to the ideal case shown in Fig. 2, demonstrating that the defense mechanism effectively maintained the main task accuracy at desirable levels.

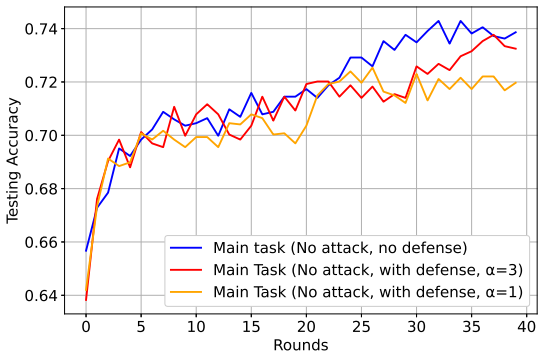

Fig. 2. Testing accuracy vs. rounds in the absence of an attack.

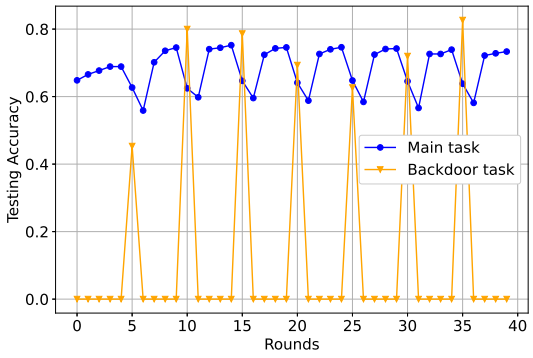

Fig. 3. Impact of the model replacement attack without a defense mechanism.

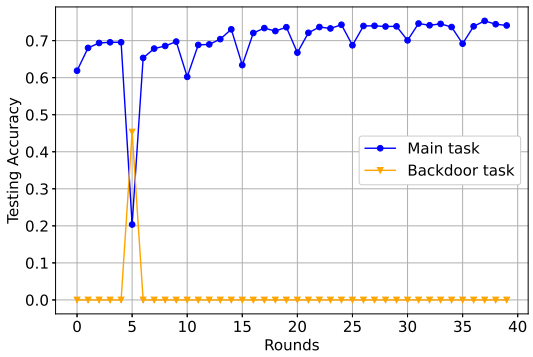

Fig. 4. Impact of the defense mechanism against the model replacement attack.

IV. Conclusion

In this work, we addressed the critical threat of model replacement attacks in FL-based IDS, where malicious participants aim to overwrite the global model with a backdoored version. By including a dynamic weight clipping defense during server-side aggregation, we effectively suppressed adversarial updates while preserving the model's primary detection accuracy. Experimental results demonstrated that the proposed approach neutralized the attacker's ability to induce misclassifications without compromising the IDS's overall performance. This defense provides a lightweight yet robust countermeasure against model replacement attacks, enhancing the security of FL-based IDS deployments in IoT environments. Future work may explore adaptive weight clipping thresholds within the defense mechanism.

References

[1] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[2] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.

[3] T. D. Nguyen, P. Rieger, M. Miettinen, and A.-R. Sadeghi, "Poisoning attacks on federated learning-based iot intrusion detection system," in *Proc. Workshop Decentralized IoT Syst. Secur.(DISS)*, vol. 79, 2020.

[4] Z. Zhang, Y. Zhang, D. Guo, L. Yao, and Z. Li, "Secfednids: Robust defense for poisoning attack against federated learning-based network intrusion detection system," *Future Generation Computer Systems*, vol. 134, pp. 154–169, 2022.

[5] Y.-C. Lai, J.-Y. Lin, Y.-D. Lin, R.-H. Hwang, P.-C. Lin, H.-K. Wu, and C.-K. Chen, "Two-phase defense against poisoning attacks on federated learning-based intrusion detection," *Computers & Security*, vol. 129, p. 103205, 2023.

[6] R. Yang, H. He, Y. Wang, Y. Qu, and W. Zhang, "Dependable federated learning for iot intrusion detection against poisoning attacks," *Computers & Security*, vol. 132, p. 103381, 2023.

[7] Y. Zhang, Y. Zhang, Z. Zhang, H. Bai, T. Zhong, and M. Song, "Evaluation of data poisoning attacks on federated learning-based network intrusion detection system," in *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, 2022, pp. 2235–2242.

[8] T. T. Thein, Y. Shiraishi, and M. Morii, "Personalized federated learning-based intrusion detection system: Poisoning attack and defense," *Future Generation Computer Systems*, vol. 153, pp. 182–192, 2024.

[9] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 2938–2948.

[10] Y. Guo, Q. Wang, T. Ji, X. Wang, and P. Li, "Resisting distributed backdoor attacks in federated learning: A dynamic norm clipping approach," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 1172–1182.

[11] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[13] P. S. Bouzinis, P. Radoglou-Grammatikis, I. Makris, T. Lagkas, V. Argyriou, G. T. Papadopoulos, P. Sarigiannidis, and G. K. Karagiannidis, "Statavg: Mitigating data heterogeneity in federated learning for intrusion detection systems," *IEEE Transactions on Network and Service Management*, 2025.