Systematic Analysis of Label-flipping Attacks against Federated Learning in Collaborative Intrusion Detection Systems

Léo Lavaur, Yann Busnel, Fabien Autrel

Systematic Analysis of Label-flipping Attacks against Federated Learning in Collaborative Intrusion Detection Systems

Léo Lavaur
leo.lavaur@imt-atlmantique.fr
IMT Atlantique
IRISA / SOTERN
Chair CyberCNI
Rennes, France

Yann Busnel
yann.busnel@imt-nord-europe.fr
IMT Nord Europe
IRISA / SOTERN
Lille, France

Fabien Autrel
fabien.autrel@imt-atlantique.fr
IMT Atlantique
IRISA / SOTERN
Rennes, France

ABSTRACT

With the emergence of federated learning (FL) and its promise of privacy-preserving knowledge sharing, the field of intrusion detection systems (IDSs) has seen a renewed interest in the development of collaborative models. However, the distributed nature of FL makes it vulnerable to malicious contributions from its participants, including data poisoning attacks. The specific case of label-flipping attacks, where the labels of a subset of the training data are flipped, has been overlooked in the context of IDSs that leverage FL primitives. This study aims to close this gap by providing a systematic and comprehensive analysis of the impact of label-flipping attacks on FL for IDSs. We show that such attacks can still have a significant impact on the performance of FL models, especially targeted ones, depending on parameters and dataset characteristics. Additionally, the provided tools and methodology can be used to extend our findings to other models and datasets, and benchmark the efficiency of existing countermeasures.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**; *Distributed systems security*; Software and application security.

KEYWORDS

federated learning, intrusion detection, data-poisoning, label-flipping, backdoors, systematic analysis, quantitative assessment

ACM Reference Format:
Léo Lavaur, Yann Busnel, and Fabien Autrel. 2024. Systematic Analysis of Label-flipping Attacks against Federated Learning in Collaborative Intrusion Detection Systems. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30-August 2, 2024, Vienna, Austria.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3664476.3670434

1 INTRODUCTION

The interconnection of heterogeneous networks and the proliferation of internet of things (IoT) devices have led to an increase in the complexity and scale of intrusion detection systems (IDSs). In this context, collaborative IDSs (CIDSs) leverage collective knowledge to detect and mitigate threats [36], but require sharing sensitive data across the network. The promise of federated learning (FL) being a privacy-preserving distributed learning paradigm has renewed the interest in CIDS, as it allows training a global model without sharing local data [20]. Since 2018, numerous works have proposed applying FL to different subdomains of IDS, such as IoT [23] or Vehicle-to-Everything (V2X) [19]. Surveys on FL for IDSs [2, 3, 8, 11, 13, 17] have also been published, highlighting the community's interest.

Because of its distributed nature, FL is highly susceptible to various types of threats, such as poisoning and privacy attacks [29]. Extensive analyses of poisoning attacks in FL have been conducted [6, 35] and have shown significant impact on performance. However, in critical applications such as IDSs, the performance of the learning algorithm is of utmost importance, as it directly impacts the security of the monitored system. Consequently, the impact of poisoning attacks on FL for IDSs is a critical concern.

While robust approaches have already been proposed [37, 40, 43], few studies focus on understanding and quantifying the impact of poisoning attacks on FL for IDSs. In particular, the effects of label-flipping attacks has been overlooked, as no systematic study has been conducted to understand their impact on FL for IDSs to the best of our knowledge.

This work aims at filling this gap by conducting a systematic and quantitative assessment of the impact of label-flipping attacks on FL for IDSs. While simple in nature, label-flipping attacks are particularly interesting as they are easy to implement, even in a *black-boxed* system, and can have a significant impact on the trained global model. Specifically, this study aims at answering the following research questions:

RQ1. Is the behavior of poisoning attacks predictable?
RQ2. Are there beneficial or harmful combinations of hyperparameter under poisoning attacks?
RQ3. Can FL heal itself from poisoning attacks?
RQ4. Are IDS backdoors realistic using label-flipping attacks?
RQ5. Is there a critical threshold where label-flipping attacks begin to impact performance?

In summary, our contributions are threefold:

- We conduct the first systematic analysis of the impact of label-flipping attacks on CIDSs leveraging FL, answering a set of well-defined research questions.
- We provide a comprehensive understanding of the impact of these attacks on the performance of the learning algorithm.
- We introduce a reusable methodology to assess the impact of poisoning attacks on FL, with experiments that can be easily

ARES 2024, July 30-August 2, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07.
https://doi.org/10.1145/3664476.3670434

replicated and extended to other datasets, attack types, and mitigation strategies.

The remainder of this paper is structured as follows. After a brief overview of poisoning attacks in FL in Section 2, Section 3 details the methodology used to conduct the experiments, with an emphasis on reproducibility. Section 4 presents the results of the experiments, answering the research questions. Section 5 presents the related work, especially the existing analyses on the impact of poisoning attacks on FL. Finally, Section 6 discusses the implications of the results and concludes the paper.

2 PRELIMINARIES

2.1 Federated Learning for Intrusion Detection

FL is a distributed learning paradigm which enables training a global model without sharing local data [20]. Model training is structured in rounds, where an orchestrating server S randomly tasks n participants $p_i, i \in [\![1, n]\!]$ from a pool of participants P to train a model w_i^r at each round r. The round ends by the aggregation of the collected models into a new global model W_r, which is redistributed to the clients as a starting point for the next round $(r + 1)$. Depending on the scale of the federation – *i.e.*, cross-silo or cross-device [15] –, the fraction C of selected participants per round can vary. The model architecture and hyperparameters are the same across the federation, but each participant owns a local dataset d_i that is not shared with the others.

In the context of IDSs, participants usually seek to classify network flows into two classes (*benign* and *malicious*), which is a binary classification task. Consequently, each dataset d_i is a set of data points representing flows, and it associates each flow $\vec{x}_j$ to its label $\vec{y}_j$. We refer to the elements of a dataset as samples. The distribution of each dataset d_i depends on the collected traffic, and therefore varies depending on the devices or services active on the network. Various degrees of similarity between clients exist, from purely independent and identically distributed (IID) partitioning to *pathological* non-IID (NIID) settings, where all clients have unique data-distributions without class overlap [14].

To train their model, the participants use a stochastic gradient descent (SGD)-based optimizer to minimize a loss function

$$\mathcal{L}(w, \vec{x}_j, y_j), j \in [\![1, |d_i|]\!], \tag{1}$$

where $\vec{x}_j$ and y_j are the sample and its label, respectively. After computing the gradients $\nabla\mathcal{L}(w, \vec{x}_j, y_j)$, they update their model as

$$w_i^{r+1} \leftarrow w - \eta\nabla\mathcal{L}(w_i, d_i), \tag{2}$$

where η is the learning rate, or upload the gradients to the server which will update the global model W_r as a function of the gradients $\{\nabla\mathcal{L}(w_i, d_i) \mid i \in [\![1, n]\!]\}$ (*e.g.*, `FedSGD`) [20]. Whether the model is updated locally or globally, the server aggregates the uploaded parameters and broadcasts the new global model to the participants.

2.2 Poisoning Attacks in Federated Learning

The attack surface of FL is broad, and includes various types of threats, such as poisoning and privacy attacks [29]. Authors often refer to poisoning attacks in FL as *Byzantine* attacks, as they are analogous to the Byzantine Generals' Problem [16] in distributed systems. Likewise, the term *Sybil attacks* [9] is frequently used to refer to the problem of *colluding attackers* [12].

Poisoning attacks can be categorized into two main categories depending on the phase in which they are perpetrated: model-poisoning [6] or data-poisoning [35]. Model-poisoning attacks aim at manipulating the model's parameters, usually during or after training, to deviate the aggregated model from the global optimum [10]. Data-poisoning attacks, on the other hand, happen before the training phase, and manipulate data samples to degrade performance, cause misclassification, or introduce backdoors [29].

Data poisoning attacks can be categorized into clean-label and label-flipping attacks. Clean-label attacks manipulate the samples to be misclassified, either by adding new samples [42] or by modifying existing ones [21]. Label-flipping attacks, on the other hand, change the labels of the samples by flipping them to a different class [35].

Additionally, most poisoning attacks can be further separated into *untargeted* and *targeted* attacks. Untargeted attacks randomly select samples to be manipulated, and are usually easier to detect as they have a higher impact on the model's performance. Targeted attacks, on the other hand, select samples based on a specific criterion, such as the class to be targeted. In a CIDS context, targeted attacks can be used to introduce backdoors – *i.e.*, making a specific attack class be misclassified as benign – or cause targeted misclassification.

Algorithmic solutions to mitigate these attacks exist in distributed learning, such as Krum [7] or Trimmed Mean [41], and are often used as comparison for works in Byzantine-robust FL. In addition to the algorithmic countermeasures, various strategies have been proposed to detect and mitigate poisoning attacks in FL specifically, ranging from clustering [24, 32] and similarity-analysis [4, 12] to client-side evaluation [44].

3 METHODOLOGY

Assessing the impact of data-poisoning over FL implies reviewing a consequent amount of parameters and configurations. To optimize our work and make it easily reproducible, the results presented in Section 4 have been generated using a purposely designed evaluation framework based on Flower [5] and Hydra [39]. We follow the ACM's guidelines and terminology [1], and take measures to ensure the *reusability* of our artifacts, the *reproducibility* of our results, and the *replicability* of our experiments. Specifically:

1. We provide the methodology and all parameters necessary to reimplement and replicate the experiments;
2. Dependencies are pinned using Poetry for Python and Nix for system, allowing the entire software pipeline to be executed in the same conditions;
3. All experiments are seeded where possible, which makes the results reproducible within a three decimal precision;
4. The results and the code to generate them are available in open access[1], as are the datasets[2].

The results presented in this paper amount to 4940 unique runs, and close to 685 cumulated computing hours on two NixOS servers with 96 cores, 768 GB of RAM and 2 Nvidia Tesla T4 each.

[1] https://github.com/phdcybersec/ares_2024
[2] https://staff.itee.uq.edu.au/marius/NIDS_datasets/

3.1 Dataset and Pre-processing

Due to the scale of the required experiments, we select a dataset that is both representative of the problem and small enough to be processed in a reasonable amount of time. Recent works on FL and IDS [30] proposed a standardized feature set (NF-V2) making cross-dataset FL setups easier, which uses nProbe [26] and its NetFlow V9 format to extract features. The authors notably provide converted versions of known datasets, including CSE-CIC-IDS2018 [31] which is the most recent generic network dataset of the list. CSE-CIC-IDS2018 is a larger scale version of the CIC-IDS2017 [31] dataset, generated using AWS. It contains 14 attacks labels grouped in 6 classes: *DDoS*, *DoS*, *Bot*, *Brute Force*, *Infiltration*, and *Injection*.

The same authors also proposed sampled versions of the same datasets [18] to reduce the computational cost of experiments. Consequently, we use the sampled NF-V2 version of CSE-CIC-IDS2018, which is composed of 1,000,000 data points. We remove the port and IP addresses for both source and destination, as they are rather a representation of the network topology and device configurations than of traffic patterns. The categorical features are then one-hot encoded[3], and we normalize the numerical features using min-max normalization. This pre-processing step produces 39 features for each sample. Finally, we evenly split the dataset for the experiments, ensuring the same class distribution in the training and testing sets. 80% of the dataset is used for training, and 20% for testing.

To assess the representativity of the datasets sampling, we compare the projections in two dimensions of the two datasets using principal component analysis (PCA). Figure 1 presents cross-projections results, depending on the datasets used to generate the projection frame. There are consequent overlaps between the classes in this projection, implying that either 2 dimensions are not enough to separate the classes, or there are features that are not relevant to the classification task. Yet, the projected patterns are identical between the two datasets, which indicates that the sampling process does not introduce significant bias in the dataset. Therefore, experiments performed over the sampled datasets should be representative of observed the behaviors in the original dataset.

3.2 Local and Federated Training

We use a simple multilayer perceptron (MLP) model with two hidden layers, as implemented by Popoola et al. [28] who use the same datasets; a summary of the model's parameters is available in Table 1. Trained centrally, this model reaches an F1-score of 0.966 and an accuracy of 0.992 on our sampled testing set. These values can be considered as baselines for the FL experiments.

We focus on the impact of data-poisoning specifically, and therefore omit other factors that could hurt the performance of the model, such as client heterogeneity or disconnections. We also specifically concentrate our efforts on a collaborative cross-silo setting, where all clients are available at each round and $C = 1$. Consequently, the dataset is partitioned into 10 IID shards of 80,000 data points, and each client is assigned with one shard. On the server, the uploaded models are aggregated using `FedAvg` – which, since the local datasets are of similar size, is equivalent to a simple average of the weights.

[3]Binary representation of categorical variables used in machine learning (ML), where each category is represented by a binary vector.

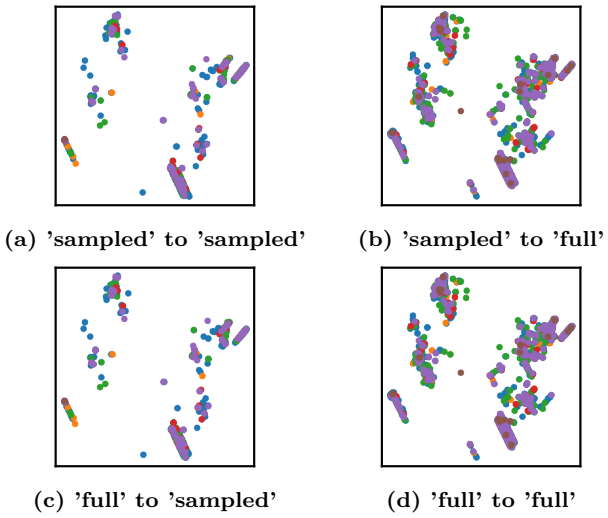


Figure 1: Cross-projections of the malicious traffic from two datasets in two dimensions using PCA. On top, the frame of reference is computed using the `sampled` dataset, and on the bottom the `full` dataset. The `sampled` dataset is then projected on the left, the `full` dataset on the right.

Table 1: Hyperparameters.

Hyperparameter	Value
Learning rate	0.0001
Hidden layers activation	ReLU
Output layer activation	Sigmoid
Input shape	49
Number of hidden layers	2
Size of the hidden layers	128
Optimizer	Adam
Loss function	Log loss
Aggregation	FedAvg

Table 2: Distribution of the two datasets.

Class	Sampled	Full
Benign	880,623	16,635,567
DDoS	73,558	1,390,270
DoS	25,574	483,999
Bot	7,595	143,097
Brute Force	6,525	123,982
Infiltration	6,108	116,361
Injection	17	432
Total	1,000,000	18,893,708

3.3 Attack Model and Implementation

We consider data-poisoning attacks where malicious participants can alter their local datasets before training. This definition covers both, participants that have been compromised and those that are deliberately modifying their data. Further, this scenario will always

be available, even with a secure and immutable FL client software. Specifically, we implement data-poisoning using label-flipping attacks, where the attacker changes the label y of a sample to a new label y_p; *i.e.*, $y_p = \neg y$ in a binary-classification problem.

3.3.1 Attacker's Objective. We consider two types of objectives for the attacker depending on the type of attack leveraged. With *targeted* attacks, the attacker aims to make a specific attack pattern undetectable. This is implemented by labeling a randomly selected fraction of a specific attack class (*e.g.*, *DDoS*) as benign. With *untargeted* attacks, on the other hand, his goal is to produce high false positives rate (FPR) and false negative rate (FNR), which can overwhelm human operators or other security systems. Here, a random fraction of the entire dataset is altered, where the label of each sample is flipped from benign to attack and vice versa. The proportion of samples that are altered is controlled by the data poisoning rate (DPR), which is the ratio of samples matching the target that are altered by each attacker on a specific round. We note the DPR, or *local poisoning rate*, as α.

3.3.2 Attacker's Knowledge and Capabilities. We consider attackers to be *gray-box* adversaries, *i.e.*, they have the same knowledge as benign clients, but are unable to modify the system's behavior, neither locally nor on the server. Further, we consider that multiple attackers can be present in the system, and that they can act in concert. This scenario is referred to as *colluding attackers*. In this case, the attackers share the same target and DPR. The proportion of attackers can vary from one single malicious client to a majority of them being malicious, and is expressed as ρ, or model poisoning rate (MPR) [21]. Note that in the context of IID partitioning, the overall poisoning rate could be regarded as $\alpha \times \rho$. This simplification is however not accurate in other partitioning strategies.

3.4 Experiments

We design a set of experiments to answer the research questions laid out in Section 1. All experiments share a common set of constants, which are complemented by a set of variable parameters. Table 3 summaries the available parameters for the experiments. Each combination is tested 10 times using a set of 10 different seeds to study the predictability of the results. Specifically, the seed impacts data-partitioning operations (both between the training and testing sets, and among clients afterward), the sample selection in poisoning attacks, and the random weights of the initial model. It also impacts all the random operations (such as data shuffling) done during model training.

The `epochs` parameter controls the aggregation frequency, *i.e.*, the number of local epochs per round ε, as well as the number of rounds R. The global number of local epochs per client is kept to 100 or 300 to preserve comparability. The `distribution` represents the number of legitimate and malicious clients in the system, and consequently the proportion of attackers. The key `scenario` represents the attackers' behavior. Scenarios defined as `continuous-`α represent a constant poisoning rate of α over the entire training process. Scenarios named `late-`r and `redemption-`r produce an attack with $\alpha = 100$ that starts or ends at round r, respectively. Parameter `target` represents the target of the attack as defined in Section 3.3; each attack class is made available as a target.

3.5 Metrics

To quantify how the experiment parameters impact the global model, we define a set of metrics to measure the attack success rate (ASR) of poisoning attacks. The definition of the ASR differs depending on the type of attack, according to the attacker's objective defined in Section 3.3. Because the ASR is based on performance and that no perfect model exists, we distinguish the absolute attack success rate (AASR) measured on the attack scenario, from the relative attack success rate (RASR) which also considers the nominal performance without attacks. Formally, the RASR is defined as:

$$\text{RASR} = \frac{\max(\text{AASR}_{benign}, \text{AASR}_{attack}) - \text{AASR}_{benign}}{1 - \text{AASR}_{benign}}, \quad (3)$$

where AASR_{benign} and AASR_{attack} are the AASR of the *benign* and *attack* scenarios respectively, under the same set of parameters. This is made possible thanks to the framework's reproducibility, which ensures two experiments started with the same seed will run under the same conditions. Following the definitions in Section 3.3, we then defined two variations of the AASR depending on the attacker's objective. Both are computed based on the confusion matrix of the model: true positives (TP), true negative (TN), false positives (FP), and TN.

Targeted attacks: Malicious participants leverage targeted attacks to make a specific attack pattern undetectable. Therefore, a successful attack forces classification of the relevant attack samples as benign. The AASR is then defined as the miss rate of the targeted attack, *i.e.*

$$\text{AASR} = \frac{\text{FN}_c}{\text{TP}_c + \text{FN}_c}, \quad (4)$$

where c is a specific attack class of the dataset.

Untargeted Attacks: Untargeted attacks aim at degrading the overall classification rate of the model. Consequently, the AASR is defined as the miss-classification rate of the model, *i.e.*

$$\text{AASR} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = 1 - accuracy. \quad (5)$$

Additionally, we use traditional binary classification metrics to observe the performance of the model under various conditions, as identified in existing surveys [8, 17] These metrics include *accuracy*, *F1-score*, and *miss rate*. Notably, we consider the main-task accuracy (MTA), defined as the accuracy of the benign scenario, to measure the impact of the attacks on the model's nominal performance. All metrics are aggregated over the 10 runs of each experiment, and the mean and standard deviation are reported for the selected metric.

4 RESULTS

The results presented in this section aim at answering the research questions defined in Section 1. Figure 2 presents the performance of the global model without malicious participants to serve as a baseline to compare with. Notably, the recall values of each of the six available attack classes indicates suboptimal performance for the "Infiltration" class, which never exceeds 0.2, and the "Injection" class, which stays around 0.86 (see Figure 2b). The feeble representation of the "Injection" class in the dataset (around 0.0017%, see Table 2)

Table 3: Experimental parameters. Default parameters are highlighted in bold and are used if not specified otherwise.

Parameter	Values	Description
batch_size	32, 128, **512**	Batch size (β)
epochs	**100_10x10**, 100_4x25, 100_1x100, 300_10x30, 300_4x75, 300_1x300	Local epochs per round (ε)
distribution	10-0, 9-1, 7-3, **5-5**, 3-7	Proportion of attackers (ρ)
scenario	continuous-{10,30,60,70,80,90,95,99}, **continuous-100**, late-3, redemption-3	Poisoning rate per round (α)
target	**untargeted**, bot, dos, ddos, bruteforce, infiltration, injection	Attack type and target
seed	1313, 1977, 327, 5555, 501, 421, 3263827, 2187, 1138, 6567	Seed for pseudo-random number generators (PRNGs)

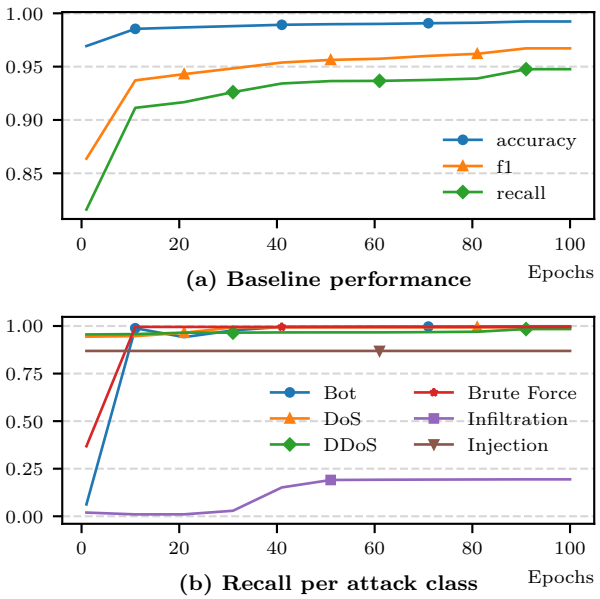


Figure 2: Performance of the global model without malicious participants. The accuracy, F1-score, and recall illustrate the performance that can be expected from the global model under the conditions selected for this study ($\varepsilon = 10$, $\beta = 512$). The recall of the six available attack classes shall serve as a reference for the RASR of targeted attacks.

prevents the model from learning from it, provoking this absence of evolution over time. The "Infiltration" class is more represented in the dataset (0.6108%, approximately the same as the "Brute Force" and "Bot" classes), but remains difficult to learn because of its apparent similarity with benign traffic.

4.1 Impact Predictability

A preliminary question to answer before quantifying the effects of label-flipping is whether the behavior of poisoning attacks is predictable. This is a requirement for generalizing our results to other datasets and models, and comparing the findings with current and future studies. Due to space constraints, we focus in this part on the parameters that have the most significant impact on the results, to assess the predictability of poisoning attacks. Specifically, the

Table 4: Experiment parameters for RQ1.

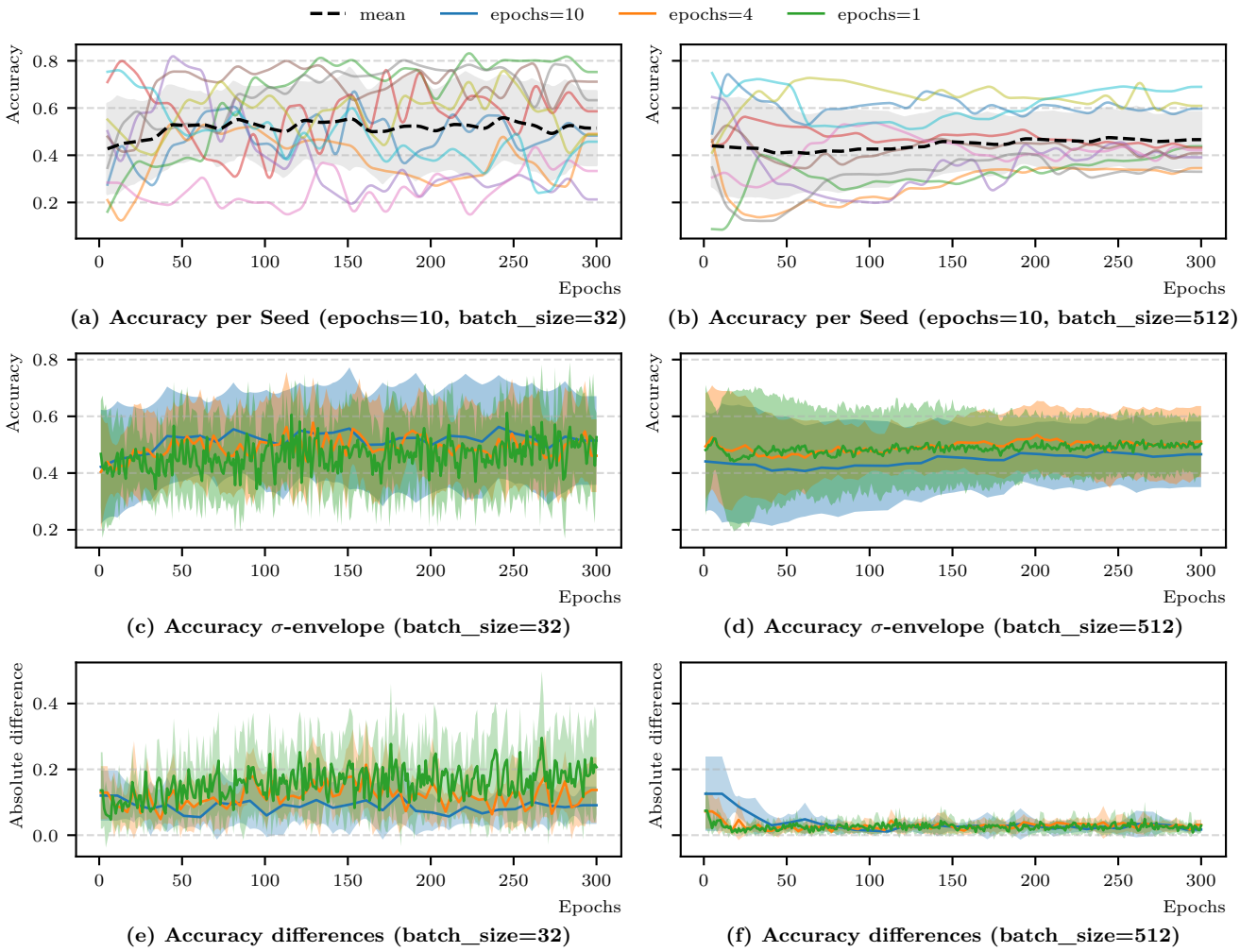
Is the behavior of poisoning attacks predictable?	
batch_size	32, 512
epochs	300_10x30, 300_4x75, 300_1x300
distribution	5-5

selected distribution contains 50% of malicious participants, which roughly equates to 50% of the training data being poisoned. The experiments are performed during 300 epochs, with three different aggregation frequencies (10, 4, and 1) and two different batch sizes (32 and 512). Table 4 summarizes the parameters used for this experiment.

Figure 3 present various metrics observing the performance of the global model over time, with different seeds. The results in Figures 3a and 3b exhibit consequent dispersion of the global accuracy between runs. Using the same parameters, the accuracy of the global model varies from 0.2 to 0.7 after 100 epochs (10 rounds under these conditions), with a standard deviation close to 0.2. This dispersion is consistent across different aggregation frequencies, as illustrated by Figures 3c and 3d.

However, the dispersion decreases over time given a big enough batch size, as shown in Figure 3d. After 100 to 120 epochs depending on the seed, the standard deviation of the accuracy stabilizes around 0.1. The absolute accuracy differences in Figure 3f indeed decrease over the first epochs, before plateauing. It can be interpreted as a consequence of the complexity of the learning tasks, which becomes harder as clients contain different labels for similar samples. Therefore, the problem probably admits a high number of local minima, which are reached depending on the seed. On the contrary, the difference between rounds using $\varepsilon = 32$ (see Figure 3e) tends to increase over time, illustrating the difficulty for each run to converge to a stable state.

Answering RQ1
The behavior of poisoning attacks is not predictable, as the dispersion between results is too important, although only the seed varies. However, the dispersion decreases over time given a big enough batch size, as the models tend to converge to a stable state. *In practice, this makes the impact difficult to predict for a specific attack instance, even though general tendencies can be extrapolated.*



(a) Accuracy per Seed (epochs=10, batch_size=32)

(b) Accuracy per Seed (epochs=10, batch_size=512)

(c) Accuracy σ-envelope (batch_size=32)

(d) Accuracy σ-envelope (batch_size=512)

(e) Accuracy differences (batch_size=32)

(f) Accuracy differences (batch_size=512)

Figure 3: Studying attack impact predictability over time, with 50% attackers. The x-axis represents the number of local epochs. Figure 3a illustrates each seed's accuracy over time using a rolling mean with a window of 5. Figures 3b and 3c display envelopes with the mean values and the standard deviation of each experiment (over the ten seeds).

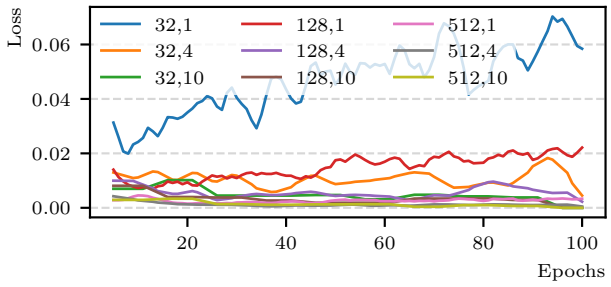


Figure 4: Mean loss over time, 50% attackers.

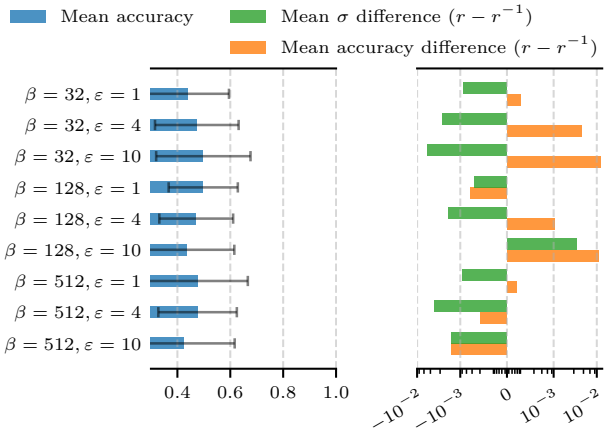
4.2 Hyperparameters Impact

To understand the impact of hyperparameters on the behavior of poisoning attacks, we study the impact of different batch sizes (β) and aggregation frequencies (ε). We reuse the conditions from Section 4.1 but limit the number of epochs to 100, as most scenarios do not show significant changes after this point (see Section 4.1). Additionally, we evaluate the hyperparameters on the *late* poisoning scenario, where the attackers only start after a 3-rounds bootstrap period. The experiment parameters are summarized in Table 5.

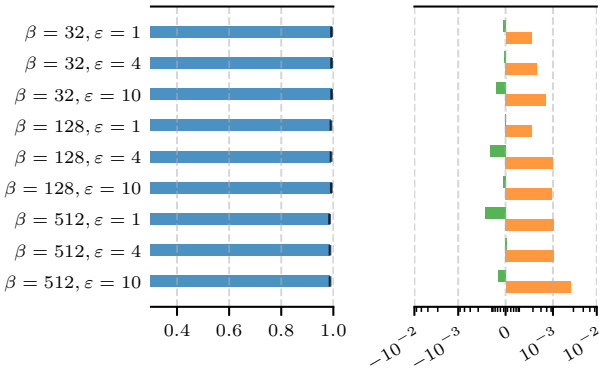
Figure 5 presents the impact of the hyperparameters on the accuracy of the global model, with and without poisoning. The differences are shown on a bi-symmetric logarithmic scale [38], defined as $x' \mapsto \text{sgn}(x) \cdot \log_{10}(1 + |\frac{x}{10^{-4}}|)$, to make up for the consequent differences in scale between combinations. Note that there is close to no dispersion in the accuracy of the benign scenario,

Table 5: Experiment parameters for RQ2.

Are there beneficial or harmful combinations of hyperparameter under poisoning attacks?	
batch_size	32, 128, 512
epochs	100_10x10, 100_4x25, 100_1x100
distribution	10-0, 5-5
scenario	continuous-100, late-3



(a) Accuracy with 50% attackers



(b) Accuracy with 0% attackers

Figure 5: Impact of hyperparameters on the accuracy of the global model with and without poisoning. The mean accuracy over 10 seeds is displayed for different combinations of batch size and aggregation frequency.

as depicted in Figure 5b, confirming that the attack is indeed responsible for the dispersion observed in Section 4.1. The different combinations present no significant impact of the selected hyperparameters on the global model's accuracy. Under poisoning, all tested parameters lead to between 0.4 and 0.5 accuracy, while they all exceed 0.95 without poisoning. This is critically low for intrusion detection: 0.5 is the score of a random classifier on a balanced binary-classification task. *Tossing a coin would yield better results.*

Still, some differences can be observed, notably in terms of dispersion. In addition to the accuracy of each parameter combination, Figure 5 also presents the average change in standard deviation, or

$$\frac{1}{R-1}\sum_{r=2}^{R}\sigma^{r}-\sigma^{r-1}, \tag{6}$$

where R is the number of rounds and σ^r is the standard deviation of the accuracy at round r between the different seeds. The average change accuracy is also displayed. For these metrics, a positive value indicates an increase in the observed metric over time.

The results indicate that their dispersion is highly dependent on the hyperparameters. Most combinations present a slight decrease in the dispersion of the results over time, which seems to be correlated with the number of local epochs ε, except for ($\beta = 128, \varepsilon = 10$). Yet, some combinations present negative accuracy differences, indicating a decrease in the accuracy over time, the most significant being the tuple ($\beta = 512, \varepsilon = 10$). Additionally, Figure 4 presents a clear correlation between the selected hyperparameters and the mean loss of the participants' datasets. In particular, runs with $\varepsilon \leq 4$ and $\beta \leq 128$ have an increasing loss, while their mean accuracy differences are close to zero, indicating greater difficulty for the participants to converge to a stable state.

However, when clients have been given the time to converge before the attack, the impact of the hyperparameters becomes more visible, particularly for the batch size as depicted in Figure 6. While the impact is instantaneous when $\beta = 32$, it takes around 20 epochs with $\beta = 512$ to reach the same accuracy. The dispersion of the results is significantly lower in the latter, as is the reached accuracy, which goes down to 0.25 after 60 epochs. A bigger batch size thus leads to a greater inertia and a lower dispersion of the results when the attack starts, but also to a lower accuracy afterward.

Answering RQ2
While the hyperparameters have an impact on the poisoning effect, no combination prevents it: on average, the performance remains the same. The results' dispersion can vary significantly depending on parameter combinations, especially when the attack occurs after the clients have converged. Then, a smaller batch size leads to a swifter effect, while a bigger batch size leads to a greater ASR. *Therefore, in performance-constrained use cases (such as the IoT), defense mechanisms might need to react faster to mitigate the attack's impact.* Round-based defenses should be less affected, but history-based defenses could be significantly impacted.

4.3 After-attack Recovery

The previous experiments allow understanding the behavior of the worst-case scenario described in Section 4.1. Notably, the attack's propagation is highly dependent on the batch size, when the attack occurs *after the clients have converged.* This experiment aims to understand the impact of label-poisoning after the attack ends. We consider a various attack distribution and a redemption scenario, where the attackers stop their operation at the third round. Table 6 summarizes the parameters used for this experiment.

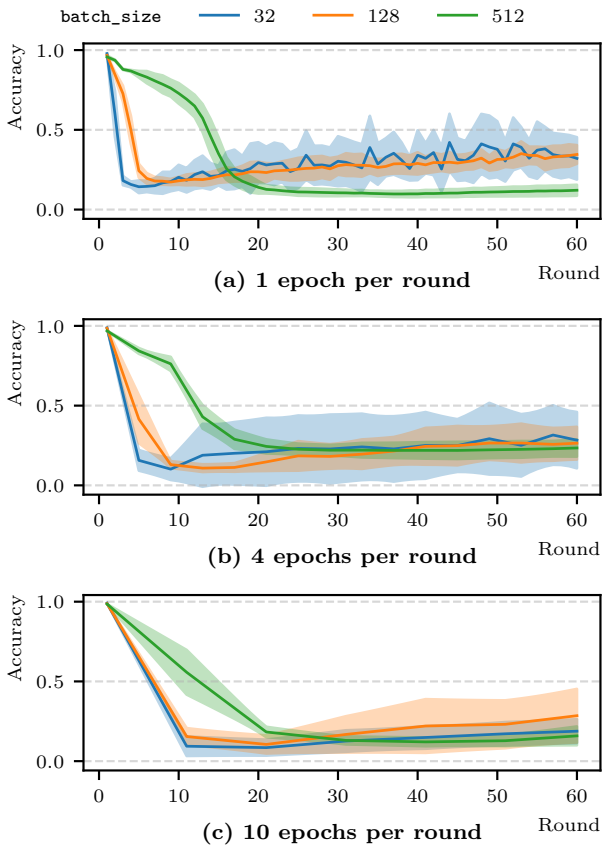


Figure 6: Impact of hyperparameters on the accuracy of the global model under late poisoning. The data is aligned to start at the last benign round before the attack, and the impact is measured over the next 60 epochs (*i.e.*, 6, 24 or 60 rounds depending on the aggregation frequency).

Table 6: Experiment parameters for RQ3.

Can FL recover from poisoning attacks?	
distribution	10-0, 9-1, 7-3, 5-5, 3-7
echo	100_10x10, 100_4x25
scenario	redemption-3

Figure 7 displays the accuracy of the global over time. Similar to Figure 6, the data is aligned: the x-axis starts at the last epochs before the attack ends, and the impact is measured over the next 40 epochs. The results show a quasi-instantaneous recovery of the global model's accuracy after the attack ends. Both values of ε display the global model reaching close to 1.0 accuracy after one round, regardless of the number of attackers and the associated accuracy during the attack. This is expected, as Figure 2a indicates that the global model's accuracy already exceeds 0.95 at the first round, in spite of the randomly initialized model parameters provided by the server before the first round. This is also consistent with the

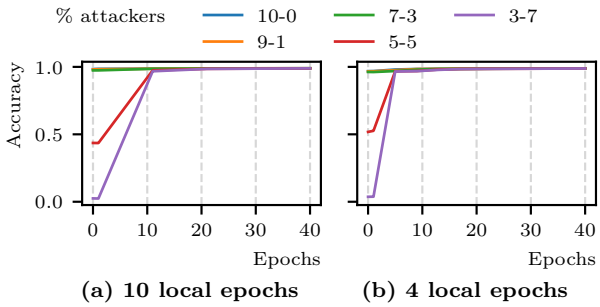


Figure 7: Accuracy of the global model after a label-flipping attack. The data is aligned to start at the last epochs before the attack ends, and the impact is measured over the next 40 epochs (*i.e.*, 4 or 10 rounds depending on ε).

Table 7: Experiment parameters for RQ4.

Are IDS backdoors realistic using label-flipping attacks?	
distribution	10-0, 7-3, 5-5, 3-7
target	dos, ddos, bot, infiltration, injection
scenario	continuous-100

results of Zhang et al. [42] on NSL-KDD [34] and UNSW-NB15 [22]. These results are consistent with the benign scenario, allowing us to infer generally that swift recovery would also be the same for other hyperparameters.

Answering RQ3
The global model recovers almost instantaneously after the attack ends. This is expected, as the global model's accuracy already reaches high values in the first round, regardless of the initial parameters.

4.4 IDS Backdoors using Label-flipping

One of the main concerns with poisoning attacks is the perspective of backdoors in the IDS, allowing attackers to bypass the system's detection capabilities afterward. To assess this risk, we study the impact of label-flipping attacks with different targets. We consider $\alpha = 100\%$ and various values of ρ to assess whether a ASR of 1.0 can be achieved. Table 7 summarizes the parameters used for this experiment.

Figure 8 presents the impact of label-flipping attacks on the accuracy of the global model for different attack targets. The AASR of the benign scenario is provided as a reference (*cf*. Figure 2b). While 30% of attackers are not enough to permanently impact the global model's accuracy, half of the targeted classes reach RASRs close to 1.0 before the end of the experiment. Additionally, the MTA of the different classes and distributions is also displayed, and generally stays close to 1.0, except for the "DDoS" class (Figure 8b), where the MTA with 70% of attackers is slightly lower. Indeed, the "DDoS" class is the most represented in the dataset, with 5.29% of

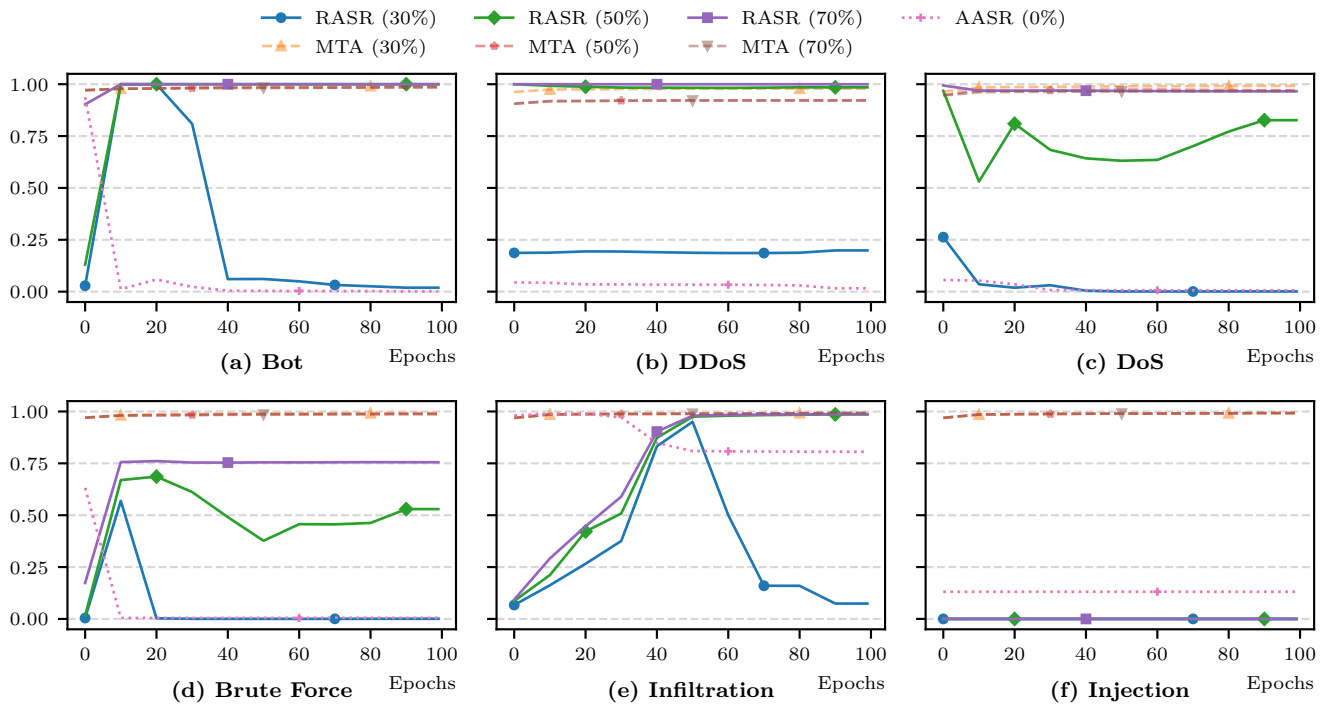


Figure 8: RASR of targeted label-flipping attacks over time, with $\beta = 512$, $\varepsilon = 10$, and $\alpha = 100\%$. The x-axis represents the number of local epochs. The AASR of the benign scenario is provided as a reference for each targeted class.

the samples. Therefore, the misclassification of roughly 70% of the samples of this class leads to a more significant impact on the global model's accuracy.

"Injection" is the only class unaffected by the attack (*cf*. Figure 8f), as attackers possess too few samples to effectively impact the global model's accuracy. Conversely, the "Infiltration" is moderately affected by the attack, even though it is already difficult to detect in the absence of malicious participants in the benign scenario. (*cf*. Figure 8e). Additionally, with the lowest proportion of attackers (*i.e.*, 30%), some targets suffer a temporary spike in RASR that can reach 1.0, before the effect of the attack fades away. This is specifically visible for "Bot" and "Infiltration" in Figures 8a and 8e respectively, as well as for "Brute Force" to a lesser extent in Figure 8d. This behavior is probably due to the similarity of traffic patterns between attack classes, as the models learns the right associations using samples from unaffected classes.

Answering RQ4
This type of attack has less impact on the global model's MTA, *i.e.*, that they are more likely to remain undetected. Although not all classes are equally impacted, *IDS backdoors are possible using label-flipping attacks, given a sufficient number of attackers and a well-represented target.* Colluding attackers can realistically create a backdoor that may later be leveraged to evade detection, raising the question of the minimum DPR and MPR necessary for such attacks to be effective.

4.5 Threshold for Effective Attacks

Previous experiments have highlighted the impact of the number of attackers on the global model's accuracy. Section 4.4 suggests that the number of attackers is a critical factor in the effectiveness of targeted attacks. This experiment aims to understand the critical threshold where label-flipping attacks begin to impact the global model's accuracy by studying both the DPR (α) and the MPR (ρ). Table 8 summarizes the parameters used for this experiment.

Figure 9 presents the RASR of considered label-flipping attacks over time, both for untargeted and targeted attacks. The entire figure emphasizes on the importance of the number of attackers in the effectiveness of the attack. With untargeted attacks especially, the RASR is insignificant (< 0.03) until the number of attackers reaches 50% (Figures 9a and 9b). With 50% attackers, clear threshold effects appear when $\alpha < 100$. $\rho = 70\%$ offers more granular results, but requires $\alpha \geq 99\%$ to maintain a RASR close to 1.0 over the entire duration of the experiment. The behavior of targeted attacks is similar, although the RASR is higher for the corresponding values of ρ and α. The RASR with 50% of attackers (Figure 9g) reaches 0.8 in average for $\alpha = 100$, and close to 0.7 for $\alpha = 99$. With $\rho = 70\%$, $\alpha \geq 90$ is enough for the RASR to reach 1.0. Below these values, the RASR quickly decreases. Further, the spikes observed in Figure 8 are visible in Figure 9f, and Figures 9g and 9h to a lesser extent.

More importantly, we can infer from $\Gamma = \rho \times \alpha$ the overall quantity of poisoned data due to our IID partitioning. For untargeted attacks, the RASR exceeds 0.5 for $\Gamma \geq 50\%$, and approach 1.0 for $\Gamma > 67\%$. For targeted attacks, the RASR exceeds 0.5 for $\Gamma > 49\%$ and

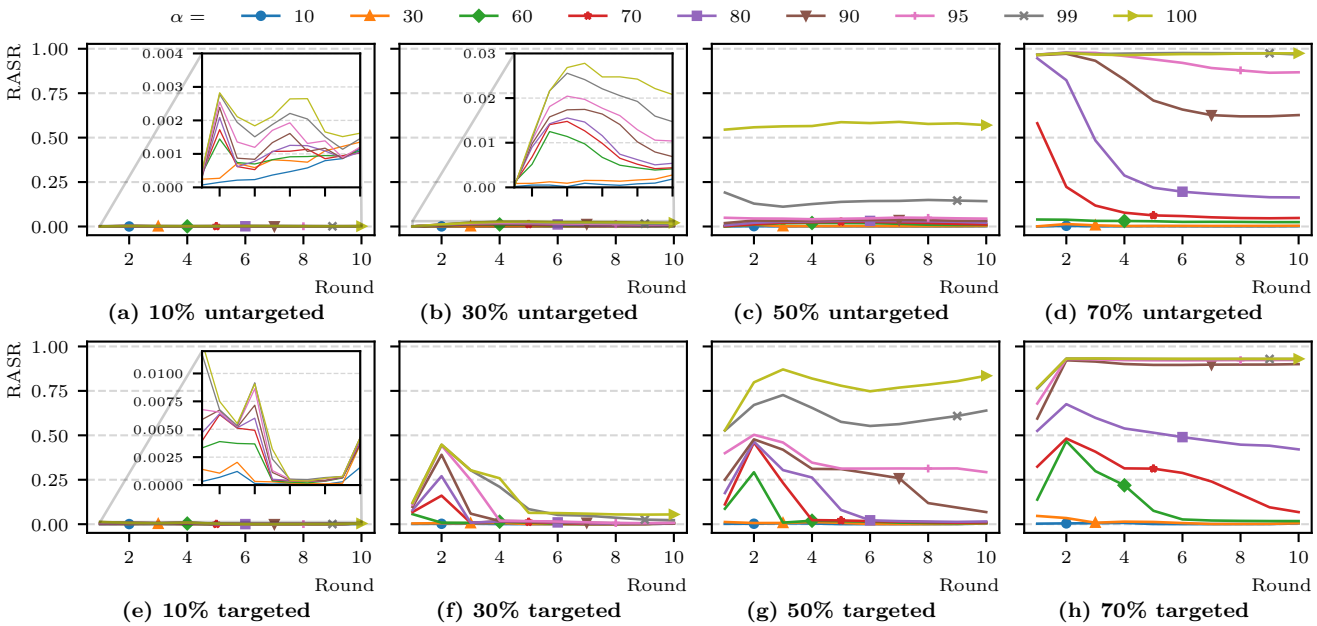


Figure 9: Evolution of the RASR of poisoning attacks over time, depending on the local poisoning rate (α), the proportion of attackers (ρ), and the type of attack. The x-axis represents the number of rounds. The value for targeted attacks is the mean of the four most effective targets: "DDoS", "DoS", "Bot", and "Brute Force" (see Figure 8). The FL round is used as the time unit.

Table 8: Experiment parameters for RQ5.

Is there a critical threshold where label-flipping attacks begin to impact performance?	
distribution	10-0, 9-1, 7-3, 5-5, 3-7
scenario	continuous-{10,30,60,70,80,90,95,99,100}
target	untargeted, dos, ddos, bot, infiltration

approaches 1.0 for $\Gamma > 56\%$. Thus, RASR and (α, ρ) exhibit fairly similar variations, albeit not linear: the higher are the DPR and MPR, the higher is the RASR. However, poisoning the entire local dataset seems more powerful than instantiating more attackers: $\alpha = 100$ and $\rho = 50$ yield higher RASR than $\alpha = 80$ and $\rho = 70$, although the latter represents more affected data overall.

Answering RQ5
The number of attackers is a critical factor in the effectiveness of label-flipping attacks, where 50% of attackers are required to effectively impact the global model's performance. The local poisoning rate is also a critical factor: the higher the local poisoning rate, the higher the RASR. However, this relation is not linear, and it exists significant threshold effects as soon as α is below 100%. *FL suffers from the same caveat as numerous other distributed systems, where the majority of participants must be honest to ensure the system's security.*

5 RELATED WORK

The literature on the impact of poisoning attacks on FL is extensive [6, 27, 33, 35], and provides insights on the behavior of poisoning attacks on generic ML tasks, such as image classification or natural language processing. Nuding and Mayer [27] focus specifically on backdoor attacks, and emphasize on the importance of the choice of the trigger pattern. Fang et al. [10], Sun et al. [33] rather study model-poisoning attacks. While often more effective than data-poisoning attacks, they are more complex to implement, as they require access to the uploaded models and knowledge of their functioning. The work of Tolpegin et al. [35] is the closest to ours, as it focuses only on label-flipping attacks. Among the most notable outcomes, the authors exhibit that targeted attacks are especially effective, having small to no impact outside the targeted class. The specificities of the IDS use case, and notably the overlap between classes, slightly contradict these conclusions.

In the context of IDSs, the literature on the impact of poisoning attacks on FL is scarcer. Zhang et al. [42] provide a systematic analysis of clean-label data-poisoning attacks, where they use generative adversarial networks (GANs) to generate poisoned samples. Other works discuss clean-label attacks to a lesser extent [25, 37]. Meanwhile, Merzouk et al. [21] provide a comprehensive analysis on data-poisoning attacks in FL for IDSs, but focus only on trigger backdoor attacks. ML backdoors work by manipulating samples to associate a specific trigger pattern with a given class so that the model misclassifies samples containing the trigger pattern. Compared with the results of Section 4.4, these attacks appear to be more effective at permanently introducing IDS backdoors. Finally,

Yang et al. [40] discuss the specific aspects of label-flipping attacks in the context of FL for IDSs, using two different datasets, NSL-KDD [34] and UNSW-NB15 [22]. However, they only implement label-flipping as a random selection of malicious samples to be flipped, which makes the results less comparable.

6 CONCLUSION

The literature on the impact of poisoning attacks on FL in the context of IDSs is scarce, and in it, label-flipping attacks have been overlooked. This study filled this gap by providing a comprehensive analysis of the impact of label-flipping attacks on FL for IDSs. We evaluated the impact of untargeted and targeted label-flipping attacks on the performance of FL models trained on CSE-CIC-IDS2018 using a standardized feature set to enable the extension of this work.

Our results highlight that (i) label-flipping attacks can have a significant impact on the performance of FL models, especially targeted ones; (ii) the ASR is closely related to the number of flipped samples overall, which can be approximated in IID settings by the product of DPR (α) and MPR (ρ); (iii) targeted label-flipping attacks strive on well-detected targets, but can be significantly mitigated by the model's generalization capabilities; and finally (iv) mitigation strategies must be adapted to the use case specificities (*e.g.*, constrained environments).

Yet, we hope that this work will inspire and fuel further research, as there are still many open questions to address. First, our results can easily be extended with more granular experiments and testing targeted attack combinations. On the other hand, while the comparison with existing works seems to corroborate our results, this study calls to be extended to other datasets. Finally, the provided evaluation framework can be used to evaluate the efficiency of existing countermeasures, or to develop new ones.

ACKNOWLEDGMENTS

This research is part of the chair CyberCNI.fr with support of the FEDER development fund of the Brittany region.

REFERENCES

[1] ACM. [n. d.]. *Artifact Review and Badging v1.1.* https://www.acm.org/publications/policies/artifact-review-and-badging-current

[2] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. [n. d.]. Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions. 195 ([n. d.]), 346–361. https://doi.org/10.1016/j.comcom.2022.09.012

[3] Mamoun Alazab, Swarna Priya R M, Parimala M, Praveen Reddy, Thippa Reddy Gadekallu, and Quoc-Viet Pham. [n. d.]. Federated Learning for Cybersecurity: Concepts, Challenges and Future Directions. ([n. d.]), 1–1. https://doi.org/10/gnm4dj

[4] Sana Awan, Bo Luo, and Fengjun Li. [n. d.]. CONTRA: Defending Against Poisoning Attacks in Federated Learning. In *Computer Security – ESORICS 2021* (Cham, 2021) *(Lecture Notes in Computer Science)*, Elisa Bertino, Haya Shulman, and Michael Waidner (Eds.). Springer International Publishing, 455–475. https://doi.org/10.1007/978-3-030-88418-5_22

[5] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Titouan Parcollet, and Nicholas D Lane. [n. d.]. Flower: A Friendly Federated Learning Research Framework. ([n. d.]). arXiv:2007.14390

[6] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. [n. d.]. Analyzing Federated Learning through an Adversarial Lens. In *Proceedings of the 36th International Conference on Machine Learning* (2019-05-24). PMLR, 634–643. https://proceedings.mlr.press/v97/bhagoji19a.html

[7] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. [n. d.]. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. 30 ([n. d.]).

[8] Enrique Mármol Campos, Pablo Fernández Saura, Aurora González-Vidal, José L. Hernández-Ramos, Jorge Bernal Bernabé, Gianmarco Baldini, and Antonio Skarmeta. [n. d.]. Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges. 203 ([n. d.]), 108661. https://doi.org/10.1016/j.comnet.2021.108661

[9] John R. Douceur. [n. d.]. The Sybil Attack. In *Peer-to-Peer Systems* (Berlin, Heidelberg, 2002) *(Lecture Notes in Computer Science)*, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Springer, 251–260. https://doi.org/10.1007/3-540-45748-8_24

[10] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. [n. d.]. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. 1605–1622. https://www.usenix.org/conference/usenixsecurity20/presentation/fang

[11] Elena Fedorchenko, Evgenia Novikova, and Anton Shulepov. [n. d.]. Comparative Review of the Intrusion Detection Systems Based on Federated Learning: Advantages and Open Challenges. 15, 7 ([n. d.]), 247. Issue 7. https://doi.org/10.3390/a15070247

[12] Clement Fung, Chris J.M. M Yoon, and Ivan Beschastnikh. [n. d.]. The Limitations of Federated Learning in Sybil Settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020)* (San Sebastian, 2020-10). {USENIX} Association, 301–316. https://www.usenix.org/conference/raid2020/presentation/fung

[13] Bimal Ghimire and Danda B. Rawat. [n. d.]. Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things. 9, 11 ([n. d.]), 8229–8249. https://doi.org/10.1109/JIOT.2022.3150363

[14] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. [n. d.]. Personalized Cross-Silo Federated Learning on Non-IID Data. 35, 9 ([n. d.]), 7865–7873. https://doi.org/10.1609/aaai.v35i9.16960

[15] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. [n. d.]. Advances and Open Problems in Federated Learning. ([n. d.]). arXiv:1912.04977 [cs, stat] http://arxiv.org/abs/1912.04977

[16] Leslie Lamport, Robert Shostak, and Marshall Pease. [n. d.]. The Byzantine Generals Problem. In *Concurrency: The Works of Leslie Lamport*. Association for Computing Machinery, 203–226. https://doi.org/10.1145/3335772.3335936

[17] Leo Lavaur, Marc-Oliver Pahl, Yann Busnel, and Fabien Autrel. [n. d.]. The Evolution of Federated Learning-based Intrusion Detection and Mitigation: A Survey. ([n. d.]).

[18] Siamak Layeghy and Marius Portmann. [n. d.]. *On Generalisability of Machine Learning-based Network Intrusion Detection Systems.* arXiv:2205.04112 [cs] http://arxiv.org/abs/2205.04112

[19] Hong Liu, Shuaipeng Zhang, Pengfei Zhang, Xinqiang Zhou, Xuebin Shao, Geguang Pu, and Yan Zhang. [n. d.]. Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing. 70, 6 ([n. d.]), 6073–6084. https://doi.org/10.1109/TVT.2021.3076780

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. [n. d.]. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (2017-04-20/2017-04-22) *(Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, 1273–1282. https://proceedings.mlr.press/v54/mcmahan17a.html

[21] Mohamed Amine Merzouk, Frédéric Cuppens, Nora Boulahia-Cuppens, and Reda Yaich. [n. d.]. Parameterizing Poisoning Attacks in Federated Learning-Based Intrusion Detection. In *Proceedings of the 18th International Conference on Availability, Reliability and Security* (New York, NY, USA, 2023-08-29) *(ARES '23)*. Association for Computing Machinery, 1–8. https://doi.org/10.1145/3600160.3605090

[22] Nour Moustafa and Jill Slay. [n. d.]. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). In *2015 Military Communications and Information Systems Conference (MilCIS)* (2015-11). 1–6. https://doi.org/10.1109/MilCIS.2015.7348942

[23] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N. Asokan, and Ahmad-Reza Sadeghi. 2019. DÏoT: A Federated Self-learning Anomaly Detection System for IoT. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)* (2019-07), Vol. 2019-July. IEEE, 756–767. https://doi.org/10.1109/ICDCS.2019.00080

[24] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza

Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. 2022. FLAME: Taming Backdoors in Federated Learning. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 1415–1432. https://www.usenix.org/conference/usenixsecurity22/presentation/nguyen

[25] Thien Duc Nguyen, Phillip Rieger, Markus Miettinen, and Ahmad-Reza Sadeghi. [n. d.]. Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System. In *Proceedings 2020 Workshop on Decentralized IoT Systems and Security* (San Diego, CA, 2020). Internet Society. https://doi.org/10.14722/diss.2020.23003

[26] ntop. [n. d.]. nProbe documentation. https://www.ntop.org/guides/nprobe/index.html

[27] Florian Nuding and Rudolf Mayer. [n. d.]. Data Poisoning in Sequential and Parallel Federated Learning. In *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics* (Baltimore MD USA, 2022-04-18). ACM, 24–34. https://doi.org/10.1145/3510548.3519372

[28] Segun I. Popoola, Guan Gui, Bamidele Adebisi, Mohammad Hammoudeh, and Haris Gacanin. [n. d.]. Federated Deep Learning for Collaborative Intrusion Detection in Heterogeneous Networks. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)* (2021-09). 1–6. https://doi.org/10.1109/VTC2021-Fall52928.2021.9625505

[29] Nuria Rodríguez-Barroso, Daniel Jiménez-López, M. Victoria Luzón, Francisco Herrera, and Eugenio Martínez-Cámara. [n. d.]. Survey on Federated Learning Threats: Concepts, Taxonomy on Attacks and Defences, Experimental Study and Challenges. 90 ([n. d.]), 148–173. https://doi.org/10.1016/j.inffus.2022.09.011

[30] Mohanad Sarhan, Siamak Layeghy, and Marius Portmann. [n. d.]. *Towards a Standard Feature Set for Network Intrusion Detection System Datasets*. arXiv:2101.11315 [cs] http://arxiv.org/abs/2101.11315

[31] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. [n. d.]. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy* (Funchal, Madeira, Portugal, 2018). SCITEPRESS - Science and Technology Publications, 108–116. https://doi.org/10.5220/0006639801080116

[32] Shiqi Shen, Shruti Tople, and Prateek Saxena. [n. d.]. Auror: Defending against Poisoning Attacks in Collaborative Deep Learning Systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications* (New York, NY, USA, 2016-12-05). ACM, 508–519. https://doi.org/10.1145/2991079.2991125

[33] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, Lingjuan Lyu, and Ji Liu. [n. d.]. Data Poisoning Attacks on Federated Machine Learning. 9, 13 ([n. d.]), 11365–11375. https://doi.org/10.1109/JIOT.2021.3128646

[34] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. [n. d.]. A Detailed Analysis of the KDD CUP 99 Data Set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (2009-07). IEEE, 1–6. Issue Cisda. https://doi.org/10.1109/CISDA.2009.5356528

[35] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. [n. d.]. Data Poisoning Attacks Against Federated Learning Systems. In *Computer Security – ESORICS 2020* (Cham, 2020) *(Lecture Notes in Computer Science)*, Liqun Chen, Ninghui Li, Kaitai Liang, and Steve Schneider (Eds.). Springer International Publishing, 480–501. https://doi.org/10.1007/978-3-030-58951-6_24

[36] Emmanouil Vasilomanolakis, Shankar Karuppayah, and Mathias Fischer. [n. d.]. Taxonomy and Survey of Collaborative Intrusion Detection. 47, 4 ([n. d.]), 33. https://doi.org/10.1145/2716260

[37] Nguyen Chi Vy, Nguyen Huu Quyen, Phan The Duy, and Van-Hau Pham. [n. d.]. Federated Learning-Based Intrusion Detection in the Context of IIoT Networks: Poisoning Attack and Defense. In *Network and System Security*, Min Yang, Chao Chen, and Yang Liu (Eds.). Vol. 13041. Springer International Publishing, 131–147. https://doi.org/10.1007/978-3-030-92708-0_8

[38] J. Beau W. Webber. [n. d.]. A Bi-Symmetric Log Transformation for Wide-Range Data. 24, 2 ([n. d.]), 027001. https://doi.org/10.1088/0957-0233/24/2/027001

[39] Omry Yadan. [n. d.]. Hydra - A Framework for Elegantly Configuring Complex Applications. Github. https://github.com/facebookresearch/hydra

[40] Run Yang, Hui He, Yulong Wang, Yue Qu, and Weizhe Zhang. [n. d.]. Dependable Federated Learning for IoT Intrusion Detection against Poisoning Attacks. 132 ([n. d.]), 103381. https://doi.org/10.1016/j.cose.2023.103381

[41] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. [n. d.]. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *Proceedings of the 35th International Conference on Machine Learning* (2018-07-03). PMLR, 5650–5659. https://proceedings.mlr.press/v80/yin18a.html

[42] Yuemeng Zhang, Yong Zhang, Zhao Zhang, Haonan Bai, Tianyi Zhong, and Mei Song. [n. d.]. 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)* (2022-12). 2235–2242. https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00330

[43] Zhao Zhang, Yong Zhang, Da Guo, Lei Yao, and Zhao Li. 2022. SecFedNIDS: Robust Defense for Poisoning Attack against Federated Learning-Based Network Intrusion Detection System. 134 (2022), 154–169. https://doi.org/10.1016/j.future.2022.04.010

[44] Lingchen Zhao, Shengshan Hu, Qian Wang, Jianlin Jiang, Chao Shen, Xiangyang Luo, and Pengfei Hu. [n. d.]. *Shielding Collaborative Learning: Mitigating Poisoning Attacks through Client-Side Detection*. arXiv:1910.13111 [cs] http://arxiv.org/abs/1910.13111

Received March, 13 2024; revised –; accepted –