Mitigating adversarial attacks against IoT profiling

Neto, Euclides Carlos Pinto; Dadkhah, Sajjad; Sadeghi, Somayeh; Molyneaux, Heather

Publisher's version / Version de l'éditeur:
https://doi.org/10.3390/electronics13132646
Electronics, 13, 13, pp. 1-20, 2024-07-05

electronics

MDPI

Article

Mitigating Adversarial Attacks against IoT Profiling

Euclides Carlos Pinto Neto [1,*], Sajjad Dadkhah [1], Somayeh Sadeghi [1] and Heather Molyneaux [2]

[1] Canadian Institute for Cybersecurity (CIC), University of New Brunswick (UNB), Fredericton, NB E3B 9W4, Canada; sdadkhah@unb.ca (S.D.); s.sadeghi@unb.ca (S.S.)
[2] National Research Council (NRC)-Canada, Fredericton, NB E3B 9W4, Canada; heather.molyneaux@nrc-cnrc.gc.ca
* Correspondence: e.neto@unb.ca

Abstract: Internet of Things (IoT) applications have been helping society in several ways. However, challenges still must be faced to enable efficient and secure IoT operations. In this context, IoT profiling refers to the service of identifying and classifying IoT devices' behavior based on different features using different approaches (e.g., Deep Learning). Data poisoning and adversarial attacks are challenging to detect and mitigate and can degrade the performance of a trained model. Thereupon, the main goal of this research is to propose the Overlapping Label Recovery (OLR) framework to mitigate the effects of label-flipping attacks in Deep-Learning-based IoT profiling. OLR uses Random Forests (RF) as underlying cleaners to recover labels. After that, the dataset is re-evaluated and new labels are produced to minimize the impact of label flipping. OLR can be configured using different hyperparameters and we investigate how different values can improve the recovery procedure. The results obtained by evaluating Deep Learning (DL) models using a poisoned version of the CIC IoT Dataset 2022 demonstrate that training overlap needs to be controlled to maintain good performance and that the proposed strategy improves the overall profiling performance in all cases investigated.

Keywords: internet of things (IoT); security; IoT profiling; deep learning (DL); adversarial attacks; data poisoning; label flipping

Citation: Neto, E.C.P.; Dadkhah, S.; Sadeghi, S.; Molyneaux, H. Mitigating Adversarial Attacks against IoT Profiling. *Electronics* **2024**, *13*, 2646. https://doi.org/10.3390/electronics13132646

Academic Editors: Irfan Awan, Andreas Mauthe, Amna Qureshi and Muhammad Shahwaiz Afaqui

Received: 6 May 2024
Revised: 20 June 2024
Accepted: 2 July 2024
Published: 5 July 2024

1. Introduction

Internet of Things (IoT) applications have been helping society in several ways [1–3]. IoT's popularity in areas such as transportation and healthcare has widely fostered its adoption in other industries and IoT devices and new applications are under development [4]. This new concept establishes a globally connected sensor network with multiple devices producing large amounts of traffic [5,6]. As a result of the efforts made by the research community and industrial groups, these devices are becoming more present in our daily lives [7,8].

Various areas have been positively impacted by IoT applications. In healthcare, patients can be continuously monitored [9–11]. In transportation, accident detection and prevention have been supported by IoT [12–14]. In Industrial IoT (IIoT), reliability, low latency monitoring, and collaborative control are enhanced [15]. Finally, there are also applications in areas such as education [16], aviation [17], and forestry [18].

In the last few years, society has experienced a drastic increase in IoT connections [19], which is expected to be even more present in the next few years across different areas [20]. This fosters the creation and development of business models and new paradigms that rely on a highly distributed infrastructure. Various approaches have been proposed to solve potential IoT problems, i.e., the scientific contributions developed in the past few years support the deployment of new services. The benefits include interoperability, security, standards, and server technologies [21–23].

However, several obstacles still need to be overcome to enable efficient and secure IoT operations. New applications may also entail new requirements to the systems, e.g.,

Internet of Vehicles (IoV) applications might require more restrictive response times than usual IoT applications [24]. Besides, detecting and mitigating attacks in IoT environments is challenging due to several factors (e.g., distributed connections and light devices without security mechanisms) [25].

In this context, monitoring the IoT network traffic to classify devices, identify abnormal patterns, and detect malicious activities becomes a pillar for secure operations [26–28]. IoT profiling refers to the service of identifying and classifying IoT devices' behavior based on different features [29]. One possible approach for IoT profiling and device identification relies on using Deep Learning (DL) trained on datasets collected from IoT network traffic.

Moreover, Deep Learning (DL) performance can be affected by different attacks. In this sense, data poisoning attacks represent one of the most challenging attacks to detect and mitigate [30]. Dealing with such a threat is difficult since the learning process assumes the training data can be used to approximate the hidden function, i.e., the training data would prepare the model to perform well in a realistic environment [31]. Thereupon, data poisoning makes it complex for a model to infer if the data used in the learning process is legitimate [32].

As a result of these attacks, the performance of a trained model can be degraded. For example, suppose noisy instances are included in the training dataset of different clients. In that case, the training process can lead the model weights to erroneous configurations and, consequently, to lower classification and regression performance scores. Furthermore, attackers could worsen the performance of such models for particular classes of interest, which could lead to hazardous scenarios (e.g., misclassification of speed limit readings for self-driving cars [33]). In fact, label flipping is one of the most challenging data poisoning attacks [34,35].

Although the adoption of a profiling mechanism is an important aspect of IoT security, this is not intended to be the only countermeasure to deploy. Identifying abnormal behaviors of impersonated devices enhances the defensive posture against advanced attacks (e.g., Advanced Persistent Threats-APT and zero-day attacks). Thereupon, attacks against profiling data can compromise security in different environments. For example, in smart cities, threat actors can take advantage of delayed and less accurate profiling identification to launch attacks against a dynamic topology [36,37]. In the Internet of Vehicles (IoV), impersonated devices can interact with intra-vehicle and inter-vehicle [38,39] networks and target specific protocol characteristics (e.g., CAN bus [40]) while undetected. In the case of the Internet of Medical Things (IoMT), the increased number of false positives can degrade the trust of profiling mechanisms, preventing the use of advanced analytical solutions in healthcare security [41,42]. Finally, the Internet of Industrial Things (IIoT) can be affected by delayed intrusion detection as impersonated devices remain undetected, leading to safety hazards and efficiency shortcomings [43,44].

Thereupon, the main goal of this research is to propose the Overlapping Label Recovery (OLR) framework to mitigate the effects of label-flipping attacks in Deep-Learning-based IoT profiling. OLR uses random forests as internal cleaners to recover labels based on their voting mechanism to accomplish this. After that, the dataset is re-evaluated and new labels are produced to minimize the impact of label flipping. In fact, OLR can be configured using different hyperparameters and we investigate how different values can improve the recovery procedure. Finally, the results are obtained by evaluating Deep Learning (DL) models using a poisoned version of the CIC IoT Dataset 2022 [29]. The main contributions of this research are as follows:

- A framework to mitigate the effects of label flipping attacks in Deep-Learning-based IoT profiling called Overlapping Label Recovery (OLR);
- A novel label recovery mechanism based on overlapping training and data sampling;
- An evaluation model based on the average area between performance curves considering label flipping and recovery procedure.

This paper is organized as follows: Section 2 presents the related works. Secondly, Section 3 presents important topics necessary for the understanding of the proposed

strategy. Then, Section 4 depicts the Overlapping Label Recovery (OLR) framework for IoT profiling. Sections 5 and 6 show the methods adopted in this research and the experiments performed. Finally, Section 7 presents the conclusion of this research.

2. Related Works

This Section reviews works that are related to this research's proposal. First, we review efforts in IoT profiling, highlighting their main aspects and goals. Then, we focus on label-flipping mitigation.

2.1. IoT Profiling

The authors in [45] propose a strategy to generate fingerprints of IoT devices based on the different device manufacturers' network system implementations. The authors emphasize that there is a lack of IoT device discovery applications on a large scale due to the massive number of device models (i.e., types, vendors, and products). This motivates the exploration of IoT feature spaces in three network layers: network, transport, and application. The prototype implemented demonstrated effectiveness in the experiments performed generating device class labels with a 94% precision and 95% recall.

In [46], the authors focus on IoT device behavioral fingerprinting to undertake strong device identification. The security challenges brought by IoT applications include the identification and authentication of several devices. The problem lies in the lack of robust approaches to identify and classify behaviors in IoT operations. Hence, the authors approximate the device's behavior using features extracted from the network traffic and train a machine-learning model that can be used to detect similar device types. The results showed that the proposed approach can achieve an identification rate of 93–100% and a mean accuracy of 99%. Similarly, Thangavelu et al. [47] propose a distributed DEvice Fingerprinting Technique (DEFT) that develops and maintains fingerprint classifiers. This approach is designed to be scalable and dynamic, with intersections with Software-Defined Networking (SDN) [48,49] and network function virtualization [50].

In [51], the authors produce device fingerprints for 20 IoT devices based on 30 features using Wireshark and four Four supervised machine learning algorithms (i.e., Support Vector Machine-SVM-, Decision Tree-DT-, Ensemble Random Forest-RF-, and Gradient Boosting Classifier-GBC). The results showed that the proposed approach is effective and promising for more extensive topologies. In addition to that, Rose et al. [52] explore the potential of using dynamic and active network profiling and machine learning to secure IoT operations against tampering attempts and suspicious network transactions.

In [53], the authors build the Abnormal Behavior Profiling (ABP) of IoT devices to support ML-based abnormal behavior detection in IoT. The authors used the k-Means and SVM algorithms to detect data modification out of four possible data points from one sensor, and the results showed that the k-Means (92%) outperform the SVM (69.5%) in terms of detection accuracy. Furthermore, the first fingerprinting framework used to identify ZigBee and Z-Wave IoT device classes is introduced in [54]. This approach monitors idle network traffic to implement signature-based device-class fingerprinting mechanisms. The experiments conducted showed that the proposed strategy achieves excellent performance in identifying different classes of IoT devices without yielding overhead to IoT devices or network traffic.

The efforts presented in [55–57] focus on IoT Device fingerprinting using Machine Learning (ML) and Deep Learning (DL). The authors in [55] propose a novel idea of device fingerprinting based on graphs of Inter Arrival Time (IAT) for packets, while [56] focuses on training ML algorithms on selected features extracted from the encrypted IoT traffic. Finally, the authors in [57] introduce an IoT device identification platform to improve Internet of Things (IoT) security using different techniques: Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). In all cases, the authors achieved high accuracy and promising results for future applications (e.g., more extensive network topologies).

2.2. Label Flipping Mitigation

The authors in [58,59] evaluate the effects of label flipping in different scenarios. Xiao et al. [58] evaluate the performance of Support Vector Machines (SVMs) to adversarial label noise attacks considering attacks that maximize the SVM's classification error by flipping a number of labels in the training data. The highlight of the proposed strategy can be used to support the development of more secure SVM learning algorithms. Furthermore, Zhang et al. [59] propose two novel label-flipping attacks to evaluate the robustness of the Naive Bayes (NB) algorithm under label noise. The authors' primary goal is to increase the false negative rate without affecting normal mail classification in the spam classification domain. The evaluation conducted showed that the label-flipping attacks reduce the overall classification performance. Similarly, Lukasik et al. [60] investigate whether label smoothing effectively mitigates the effects of label noise. The findings presented in this research indicate that label smoothing can be beneficial and that other aspects of DL training can be considered in future works (e.g., gradient clipping) [61].

In [62], the authors propose (i) an approach to perform optimal label-flipping poisoning attacks and (ii) a mechanism to detect and recover suspicious data points. The paper discusses details on the problem faced and the proposed solution. The experiments demonstrated a significant degradation of the models' performance produced by the proposed attack and the effectiveness of the proposed method in mitigating the effect of label-flipping attacks. In addition to this effort, Ortego et al. [63] mitigate the effects of label noise by proposing a Multi-Objective Interpolation Training (MOIT) approach that exploits contrastive learning and classification. The experiments conducted showed that state-of-the-art results are achieved when training DNNs with different noise distributions and levels.

Authors in [64–66] focus on the effect of data poisoning in Federated Learning (FL) applications [67]. The general idea is to develop mechanisms capable of defending, preventing, detecting, and mitigating such attacks in a federated environment. All efforts presented promising results and pointed out the extension of their approach (e.g., in terms of techniques used) as their future direction.

The authors in [68] solve the optimization problem of minimizing the number of labels flipped using an approximate linear programming algorithm and provide theoretical guarantees on how close its result is to the optimal solution in terms of the number of label flips. Multi-class classification and regression tasks are highlighted as future directions of this research. In addition to this effort, Sharma et al. [69] propose a method to identify the poisoned training samples, a label-flipping attack based on a stochastic hill-climbing search, and a CatBoost-based defense mechanism to detect and recover malicious training samples.

Some recent cybersecurity efforts have also focused on mitigating data poisoning attacks. Yang et al. [70] address the issue of training federated models for IoT security considering label flipping. In addition, Jiang et al. [71] introduce a method to detect malicious clients in Federated Learning (FL) and mitigate label flipping through the recovery of feature characteristics. Finally, Taheri et al. [72] proposes a defensive mechanism to mitigate label flipping in the context of Android malware classification relying on Convolutional Neural Networks (CNNs).

3. Background

This section presents relevant topics in support of this research's proposal. We discuss the main aspects of IoT profiling and how it can be affected by label flipping. After that, the main aspects of Deep Learning (DL) and Random Forest (RL) are presented with a focus on the context of this research.

3.1. IoT Profiling & Label Flipping

IoT profiling is the process of gathering and analyzing information about individual IoT devices and traffic that may include packet characteristics, behaviors, connections and interactions [23,53]. IoT devices are often targeted by malicious procedures due to the limited processing power and security mechanisms [73]. Thus, network-level security

is pivotal to identifying and mitigating attacks in heterogeneous IoT networks. In fact, IoT profiling is an essential approach to identifying and monitoring connected devices, specific behavior, and abnormal traffic within the network. Adopting efficient IoT profiling solutions can prevent malicious network activity by promptly detecting and isolating compromised devices for further investigation.

Moreover, the success of Deep Learning (DL) is possible due to the massive amount of data available. The same applies to IoT profiling, where data can be used to identify devices and abnormal behaviors. One of the main goals during the training of a DL model is to establish a context as close as possible to the real-world application. This entails that the dataset used to train the model is expected to represent the real world and be trustworthy. However, attackers can poison the dataset in a way to transform it into a misleading representation of reality.

One of the most challenging attacks to detect and mitigate is label flipping. This malicious procedure controls labels assigned to training data to significantly diminish the performance of the DL classifiers [72]. Once this attack is executed, it is difficult to recover original labels since it usually requires domain knowledge and massive data.

3.2. Deep Learning (DL) & Random Forest (RF)

In the past few years, Deep Learning (DL) has been applied to several problems across many areas. This subarea of Artificial Intelligence (AI) has attracted the research community's attention due to its extraordinary success in performing tasks considered complex to computational systems due to their reasoning requirements. For example, there are solutions in computer vision [74], Natural Language Processing (NLP) [75], and generative solutions [76].

Thereupon, Deep Neural Networks (DNNs) consist of processing units named neurons and are divided into multiple layers [77]. These processing units are connected to synaptic weights and output the result of an activation function [78]. The goal of DNN is to approximate a hidden function that maps the inputs (features) to the outputs to reduce the estimator error [79].

Although DNNs can be trained using different methods, the combination of the back-propagation and the Stochastic Gradient Descent (SGD) algorithms are used to optimize the weights. Figure 1 illustrates a standard DNN. Indeed, a different number of nodes and layers can be adopted and the flexibility of this model has been vital in its success in solving several problems.

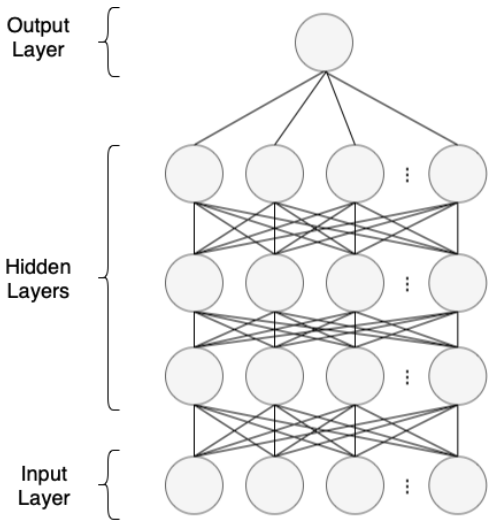

Figure 1. Deep Neural Network (DNN) [80].

As a very successful Machine Learning (ML) method, Random Forest (RF) is a tree-based ensemble that depends on a collection of random variables [81,82]. This method

generates random subsets of training configurations and makes decisions based on the evaluation from several standpoints [83]. RF has been successfully used in data cleaning efforts [84,85]. Finally, its flexibility enables the OLR framework to adopt RF in the internal label recovery process. Figure 2 illustrates a simple RF model.

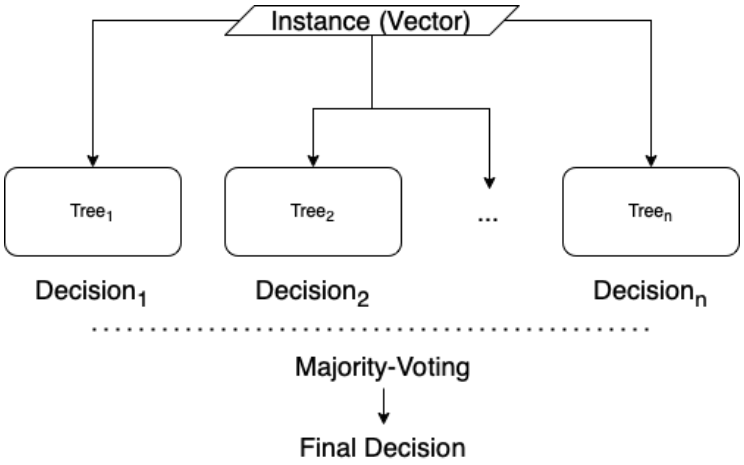

Figure 2. Random Forest (RF) [86].

4. Overlapping Label Recovery for IoT Profiling

This Section introduces the main contribution of this research, i.e., the Overlapping Label Recovery (OLR) framework. The primary objective of this strategy is to recover original labels perturbed by label-flipping attacks based on the use of multiple Random Forest (RF) models, referred to as cleaners, training on overlapping samples.

Assume an IoT profiling dataset is attacked in a way that multiple labels are flipped. In this scenario, the dataset used to train models becomes deceiving since misleading data points are introduced. For example, data related to an IoT speaker can be wrongfully associated with a smart camera. This attack happens due to several reasons and can have a profound impact on classification performance. A framework that can mitigate this problem is vital for IoT profiling applications.

Furthermore, Figure 3 illustrates how the OLR framework supports the training of accurate Deep Learning (DL) methods for IoT profiling. Since the IoT profiling dataset has been compromised, OLR acts as a label recovery strategy to improve the overall data quality. The output dataset, namely the recovered IoT profiling dataset, is then used to train the Deep Learning (DL) model to improve its performance.

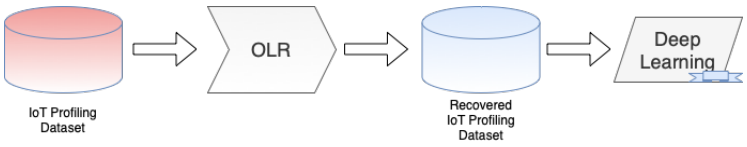

Figure 3. Process of training a Deep Learning (DL) model for IoT profiling classification using the OLR framework to recover flipped labels.

Moreover, the internal OLR process is divided into two phases as illustrated in Figure 4. This framework adopts multiple cleaners that enable a probabilistic profiling label recovery capability with the ultimate goal of mitigating attacks against data integrity. First, mechanisms to recover the flipped labels need to be initialized. In this sense, Figure 5 illustrates the process of training multiple cleaners to recover flipped labels. Firstly, The dataset is used as a baseline to train all cleaners. However, each cleaner is trained in a random sample of the original dataset, with a size equal to a factor ϕ of the original dataset's size (ϕ%, $0 < \phi < 1$). In fact, these are overlapping samples, as they are randomly collected from the same dataset.

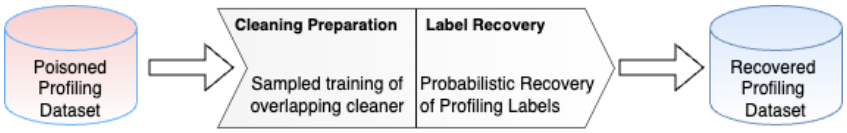

Figure 4. Overlapping Label Recovery (OLR) for IoT Profiling.

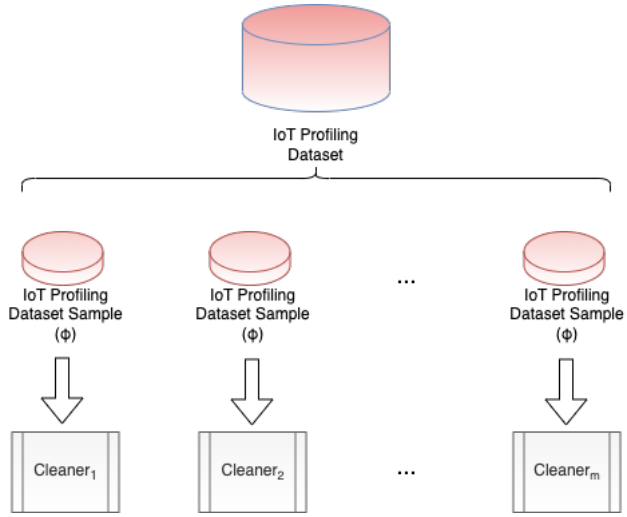

Figure 5. OLR Phase 1: cleaning preparation.

The second OLR phase refers to the actual label recovery, as illustrated in Figure 6. For each data point present in the dataset, all cleaners are used to calculate the probability of this data point belonging to each class defined in the dataset. All cleaners, previously trained with a random sample, produce a vector Vp to describe all probabilities. Cleaners are instances of Random Forest (RF), i.e., the probability of a data point belonging to a class refers to the internal voting mechanism used in this method. Once all outputs are generated, the average probabilities are calculated, and the data point is assigned to the class with the greatest average probability.

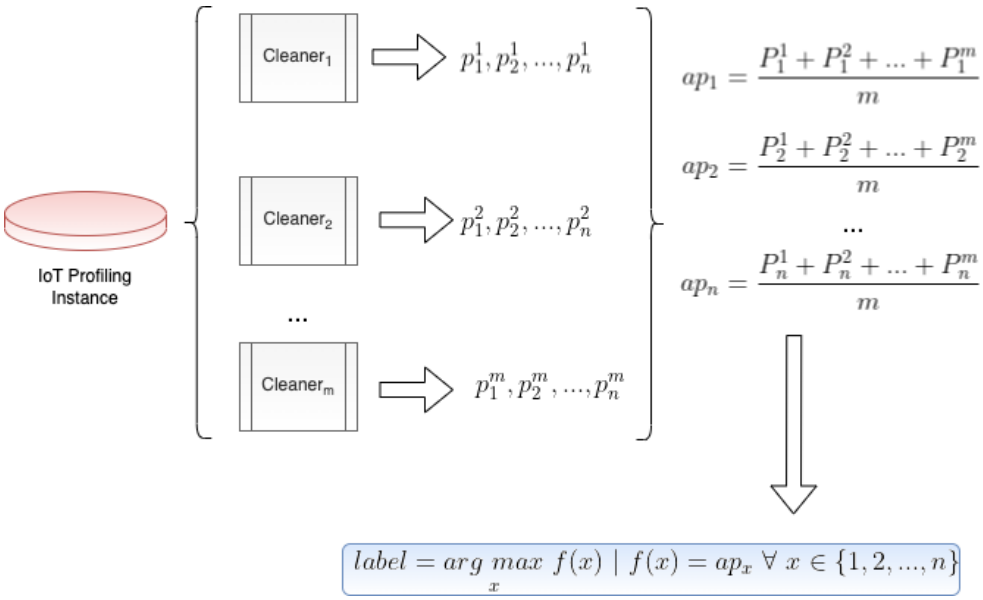

Figure 6. OLR Phase 2: label recovery.

This process is formally described in Algorithm 1. This approach requires the vector of cleaners and its length (Ω and Ω_s), the IoT profiling dataset (D), the sampling factor

(ϕ), and the number of classes (i.e., unique labels) present in the dataset (n) as inputs. Line 1 initializes an empty probabilities vector Vgp to be populated in a for loop (Lines 2 to 10). Then, for each cleaner $\omega \in \Omega$, a randomly generated sample d_ω is used in the training process. After that, ω is used to calculate the probabilities of all data points present in D belonging to each class. Since these probabilities are iteratively stored in the local empty vector of probabilities Vlp, Line 10 adds such vector into the global vector P.

Algorithm 1 Overlapping Label Recovery (OLR) for IoT Profiling

Require: Ω: vector containing cleaners;
Require: Ω_s: length of Ω;
Require: D: IoT Profiling dataset;
Require: ϕ: Sampling factor;
Require: n: Number of classes present in the dataset;
1: Initialize P a global empty vector of probabilities;
2: **for** ω in Ω **do**
3: Randomly sample as a factor ϕ of D, producing d_ω;
4: Train cleaner ω using d_ω;
5: Initialize Vlp a local empty vector of probabilities;
6: **for** i in D **do**
7: Calculate probabilities $p^i \leftarrow [p_0^\omega, p_1^\omega, ..., p_n^\omega]$ of instance i belonging to each class;
8: Add p^i to Vlp;
9: **end for**
10: Add Vlp to P;
11: **end for**
12: Initialize AP_D as an empty vector of average probabilities;
13: **for** i in D **do**
14: $ap \leftarrow \begin{bmatrix} \frac{P_{i,1}^1 + P_{i,1}^2 + ... + P_{i,1}^{\Omega_s}}{\Omega_s} \\ \frac{P_{i,2}^1 + P_{i,2}^2 + ... + P_{i,2}^{\Omega_s}}{\Omega_s} \\ ... \\ \frac{P_{i,n}^1 + P_{i,n}^2 + ... + P_{i,n}^{\Omega_s}}{\Omega_s} \end{bmatrix}$
15: Add ap to AP_D
16: **end for**
17: Initialize empty vector R_l of recovered labels;
18: **for** ap in AP_D **do**
19: $label_{ap} \leftarrow \arg\max_x f(x) \mid f(x) = ap_x \; \forall \; x \in \{1, 2, ..., n\}$;
20: Add $label_{ap}$ to R_l;
21: **end for**
22: Return R_l;

Once P is populated with the probabilities of all instances belonging to all classes calculated by all cleaners, these results are combined from Lines 13 to 16. This process comprises the calculation of average probabilities of a given instance i belonging to a specific class $(1, 2, \ldots, n)$ according to all cleaners $(1, 2, \ldots, \Omega_s)$ and, after that, defining labels based on the classes with the highest probabilities (Lines 17 to 21). Finally, the procedure returns the set of recovered labels Rl.

To compare the performance of the different training sample sizes (ϕ), we adopt the evaluation model shown in Equation (1). The main idea is to produce a DL configuration that maximizes the area between the curves generated by the results with and without OLR support. In this model, θ represents the metrics used to measure the DL performance, i.e., in this research, accuracy, precision, recall, and f1-score. Also, γ_{min} and γ_{max} represent the minimum and maximum label flipping factors adopted in the experiments. Given that D is the IoT profiling dataset used, $\mu_{\gamma,\phi,\theta(D)}$ represents the DL performance for metric θ using the OLR framework to recover labels with training sample size ϕ and label flipping factor

γ. Similarly, $\sigma_{\gamma,\theta(D)}$ represents the DL performance for metric θ without label recovering mechanisms. Finally, the outcome of this summation is multiplied by θ_t^{-1} as an approach to average the area between these two curves in all metrics.

$$f(\phi) = \theta_t^{-1} \sum_{i=1}^{\theta_t} \int_{\gamma_{min}}^{\gamma_{max}} (\mu_{\gamma,\phi,\theta(D)} - \sigma_{\gamma,\theta(D)})\, d_\gamma \quad (1)$$

5. Evaluation Method

To demonstrate the applicability of the proposed approach and show the improvements achieved, we adopt the evaluation methods depicted in this section. Figure 7 shows how the evaluation is conducted. Three processes are initiated to (i) generate baseline results using the original data; (ii) generate results using the poisoned version of the profiling dataset and (iii) generate results using the same poisoned version of the profiling dataset, but considering the recovery capabilities of OLR. After that, a comparison is conducted to analyze the improvements of the proposed strategy regarding accuracy, recall, precision, and F1-score.

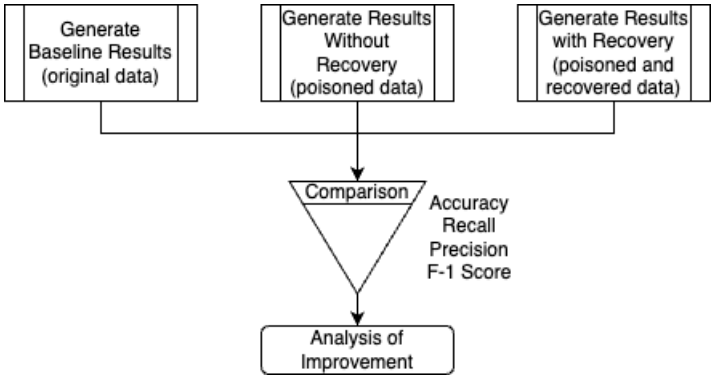

Figure 7. Evaluation and comparison of different methods.

Figure 8 illustrates the method adopted to produce baseline results. Firstly, the IoT profiling dataset is loaded (train split, i.e., 80% of the dataset). Then, it goes through the preprocessing phase, which comprises normalization using the Min-Max approach [87]. After that, a Deep Learning (DL) model is trained and evaluated using the test split (i.e., 20% of the original dataset). Finally, the results are reported.

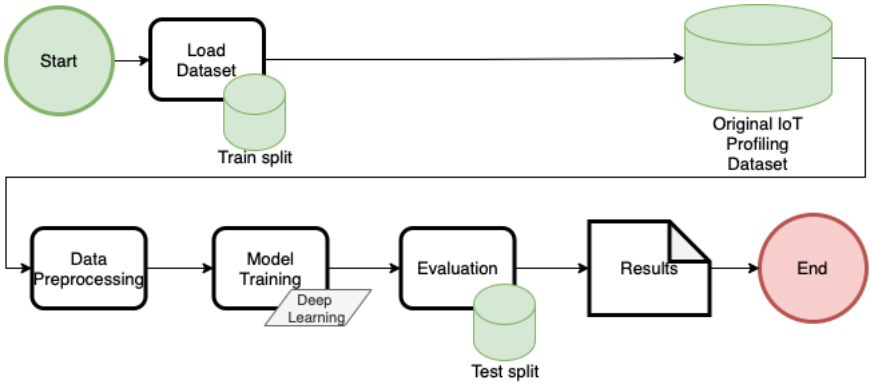

Figure 8. Method adopted to produce baseline results.

Figure 9 depicts the method adopted to produce results when the train set suffers a label-flipping attack. Firstly, the IoT profiling dataset is loaded (train split, i.e., 80% of the dataset) and an attack is performed in a factor of γ of the data ($0 < \gamma < 1$). Then, the data are normalized and a Deep Learning (DL) model is trained and evaluated. Finally, the results are summarized.

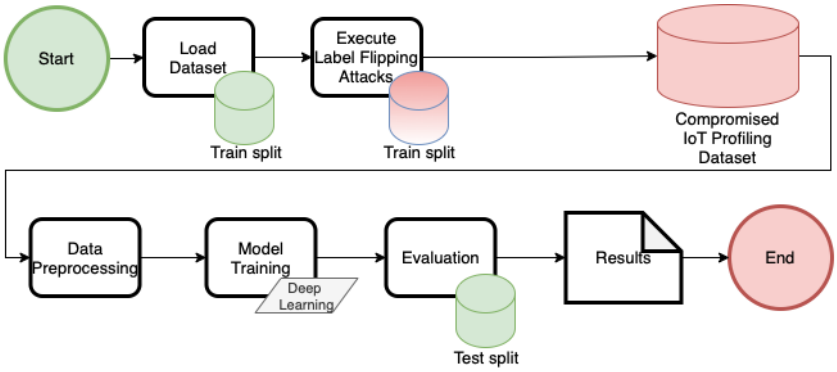

Figure 9. Method adopted to produce results without recovery mechanism.

Finally, Figure 10 describes the method adopted to produce results when the train set suffers a label-flipping attack and the OLR approach is used to recover original labels. Firstly, the IoT profiling dataset is loaded (train split, i.e., 80% of the dataset) and an attack is performed. After that, the data go through a recovery process conducted by OLR and is normalized using the MinMax strategy. Then, a DL model is trained and evaluated, and the results are reported.

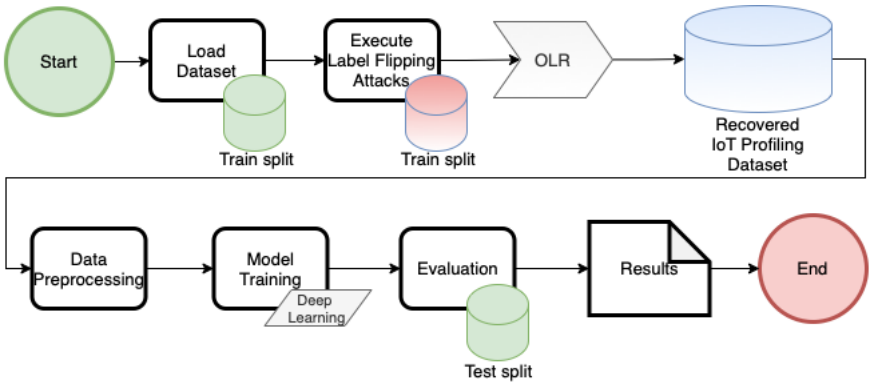

Figure 10. Method adopted to produce results use OLR as the label recovery mechanism.

6. Experiments

This Section describes the details and results of the experiments performed. The primary goal is to show how the proposed approach can mitigate label-flipping attacks in different scenarios. Three experiments are conducted varying the internal sampling factor ϕ adopted by OLR, where in Experiment I, $\phi = 0.15$, in Experiment II, $\phi = 0.25$, and Experiment III adopts $\phi = 0.35$. In terms of models used, a Deep Learning (DL) composed of three hidden layers of 64 nodes is used in each experiment. Besides, Random Forests (RFs) are used as cleaners based on 100 internal classifiers. The results of using 5, 10, and 15 cleaners are discussed in each experiment. Finally, to show the applicability of the proposed approach, we used the CIC IoT Dataset 2022 dataset [29], which comprises network traffic produced by IoT devices to enable the analysis of IoT devices' behavior exhibited operating under different constraints. Additionally, we consider the IoT profiling problem from the perspective of classifying groups of devices, namely audio (18774 traffic readings of Echo Spot, Nest mini, Echo Dot, Sonos, Echo Studio), home automation (19808 traffic readings of Amazon Plug, Roomba, Heim vision lamp, globe lamp, Eufy home base, Atomi cofee maker, Smartboard, Yutron, Phillips Hue, and Teckin), and cameras (77361 traffic readings of Amcrast, Arlo base cam, Arlo qcam, Borun cam, and DLink cam). In all cases, we consider label flipping factors of 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6, i.e., $\gamma_{min} = 0.1$ and $\gamma_{max} = 0.6$. Finally, several features were used in the process, e.g., "L4_tcp", "L4_udp", "L7_http", "L7_https", "ethernet_frame_size", "ttl", and "protocol". The graphs present multiple lines of different colors rather than symbols to simplify the visualization in overlapping areas.

6.1. Experiment I

This experiment considers $\phi = 0.15$ as the sampling factor used in the internal cleaning procedure of the OLR framework. Adopting the baseline, compromised, and recovered methods illustrated in Figure 8, Figure 9, and Figure 10, respectively, the results obtained are depicted in Figure 11. In fact, OLR is capable of improving the IoT profiling performance in all cases. For all metrics (i.e., accuracy, recall, precision, and F1-score) and all OLR configurations, the model performance using the recovered labels lies between the baseline and the flipped label performance. Furthermore, as the label flipping factor increases, it becomes more difficult for labels to be recovered and the performance decreases.

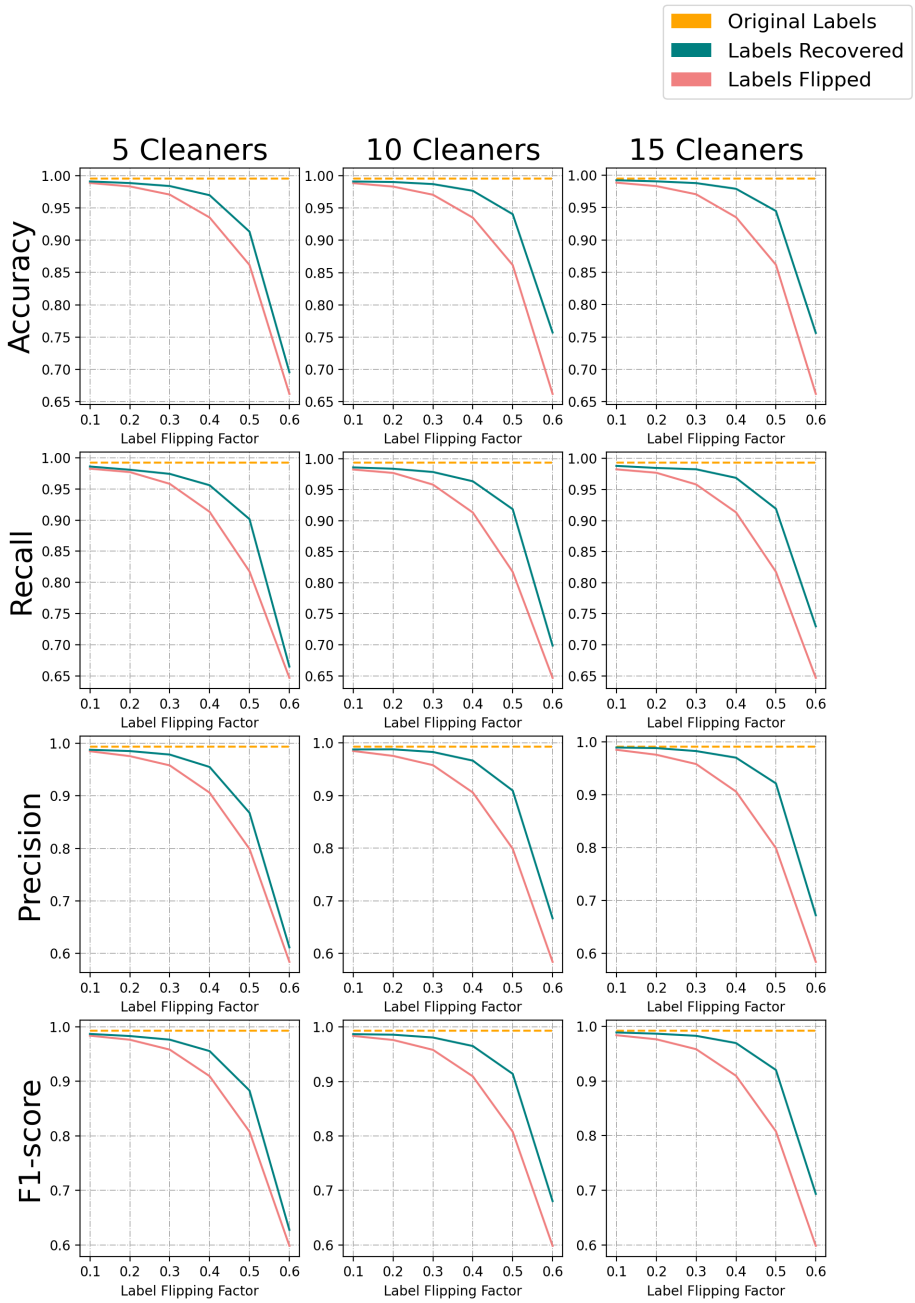

Figure 11. Experiment I: IoT Profiling ($\phi = 0.15$).

Figure 12 compares the performance of IoT profiling adopting different OLR configurations. All configurations present a very similar outcome in cases where the label-flipping

factor is less than 0.3. However, increasing the number of cleaners improves overall performance in more challenging cases.

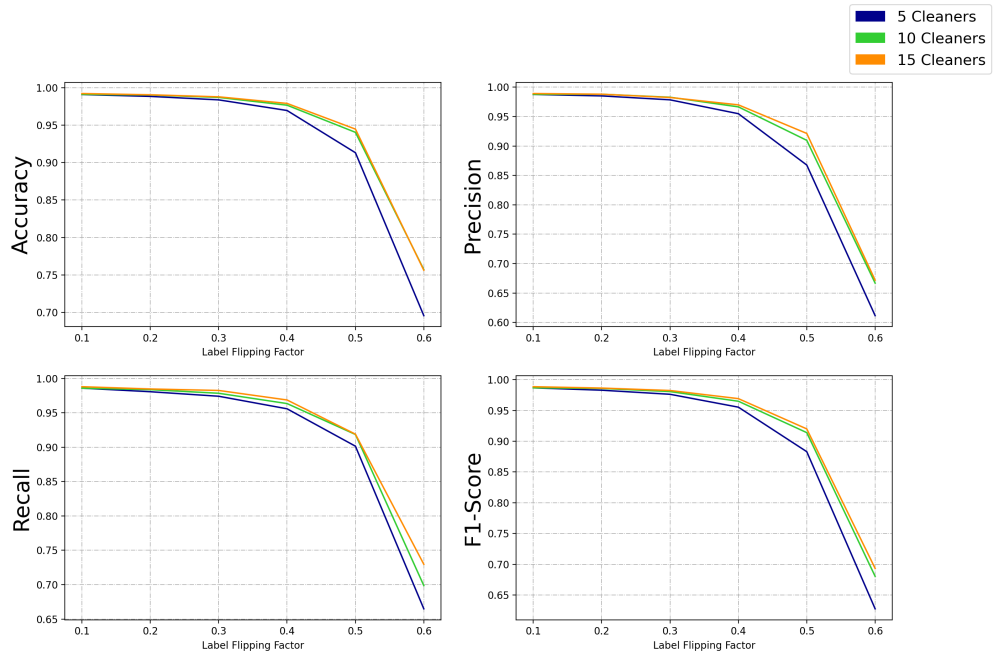

Figure 12. Experiment I: Comparison of OLR configurations ($\phi = 0.15$).

Finally, Figure 13 illustrates the capability of different OLR configurations in terms of label recovery. In all cases, several labels were recovered and used as stated in the original dataset. Furthermore, increasing the number of cleaners improves overall performance in most cases and the adoption of OLR mitigated the impact of label flipping, illustrated by the red line.

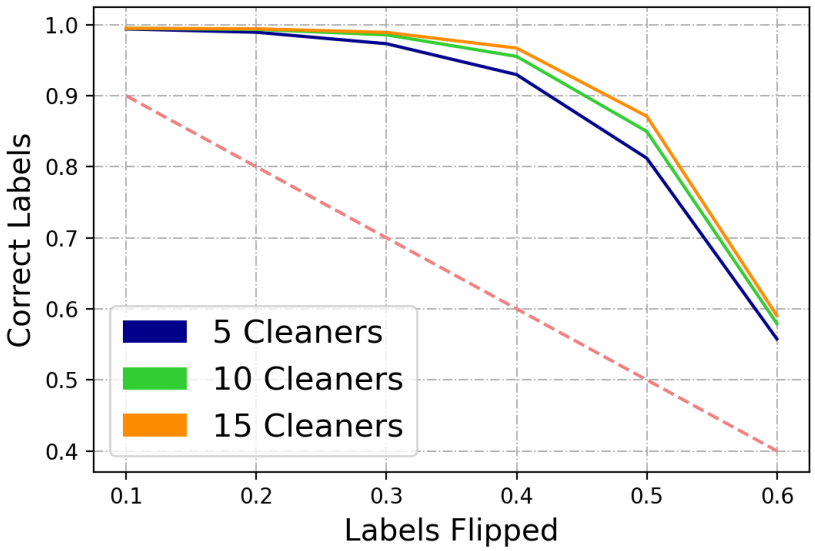

Figure 13. Experiment I: OLR label recovery ($\phi = 0.15$).

6.2. Experiment II

Adopting $\phi = 0.25$ as the sampling factor used in the internal cleaning procedure of the OLR framework, this experiment follows the methods illustrated in Figures 8–10 to produce the results. Figure 14 shows that OLR can improve the IoT profiling performance in

all cases. The performance using recovered labels lies between the baseline and the flipped label performance, although the performance decreases as the label flipping factor increases.

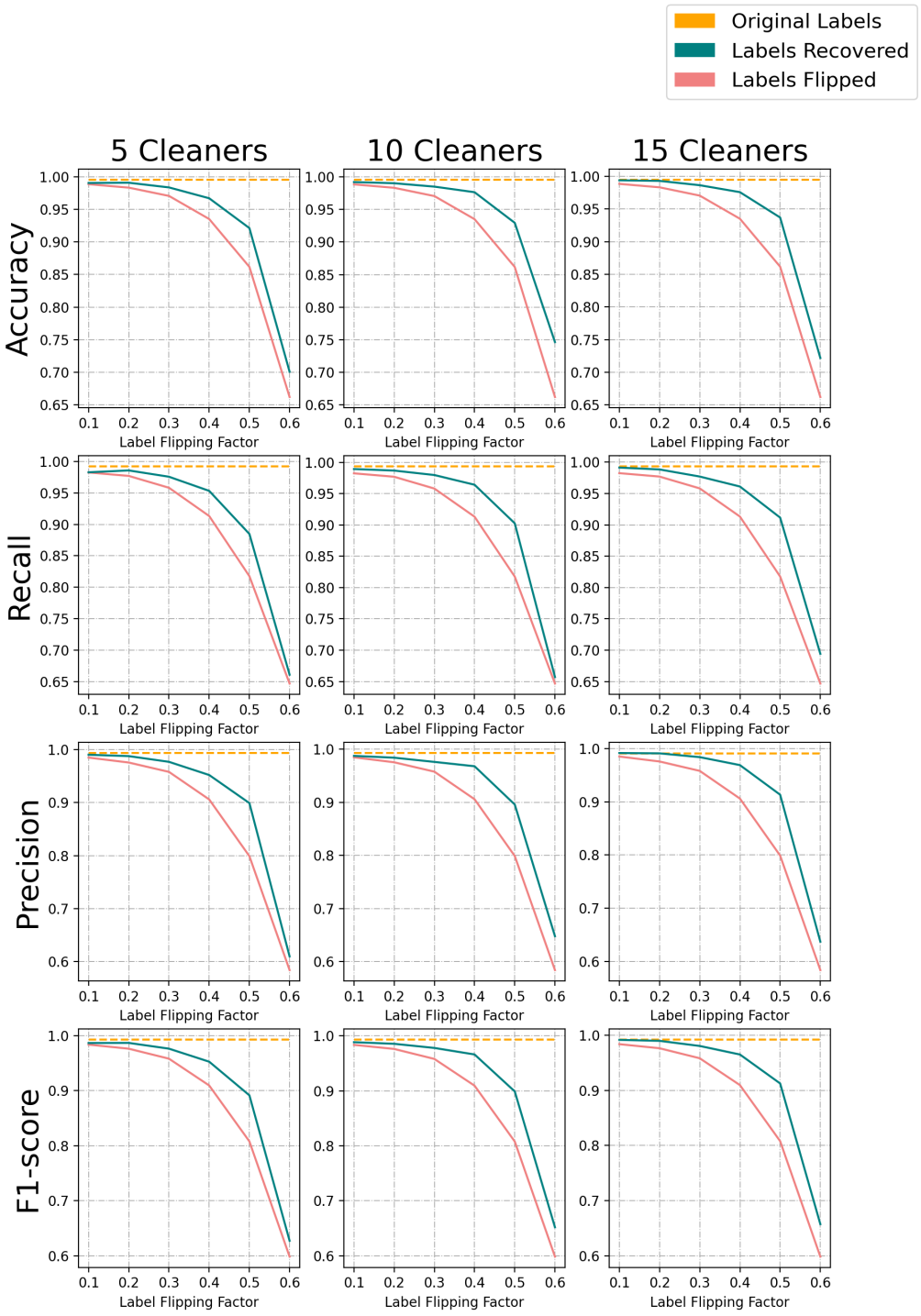

Figure 14. Experiment II: IoT Profiling ($\phi = 0.25$).

A comparison of the IoT profiling performance using different OLR configurations is illustrated in Figure 15. Once again, increasing the number of cleaners improves overall performance in more challenging cases, although all configurations present a very similar outcome when the label flipping factor is less than 0.3.

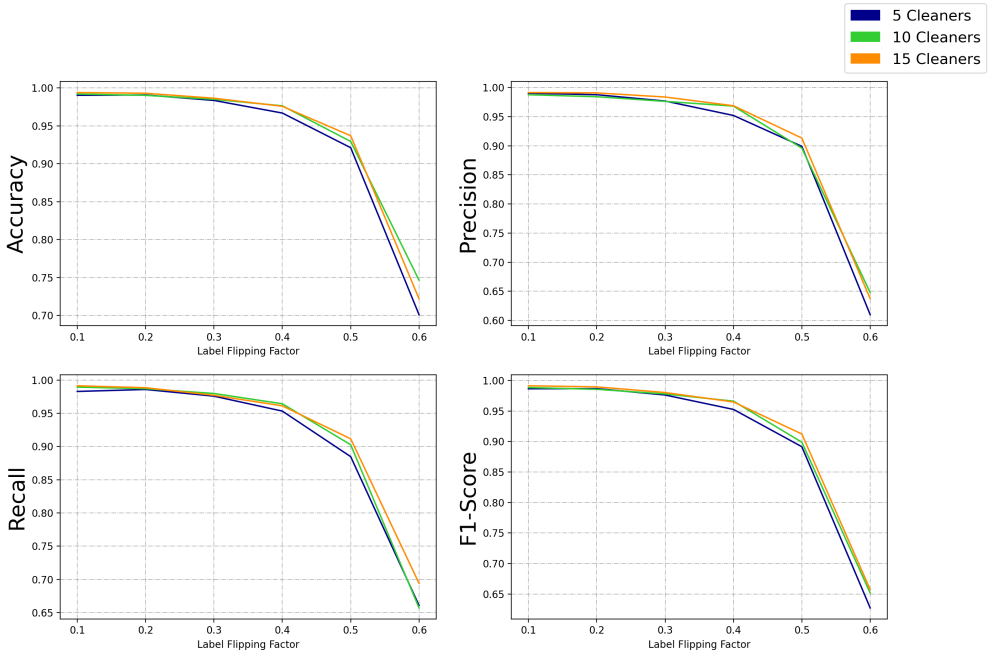

Figure 15. Experiment II: Comparison of OLR configurations ($\phi = 0.25$).

The label recovery capability of different OLR configurations is depicted in Figure 16. Several labels were successfully recovered since increasing the number of cleaners improves overall performance in most cases, mitigating the effects of label flipping illustrated by the red line.

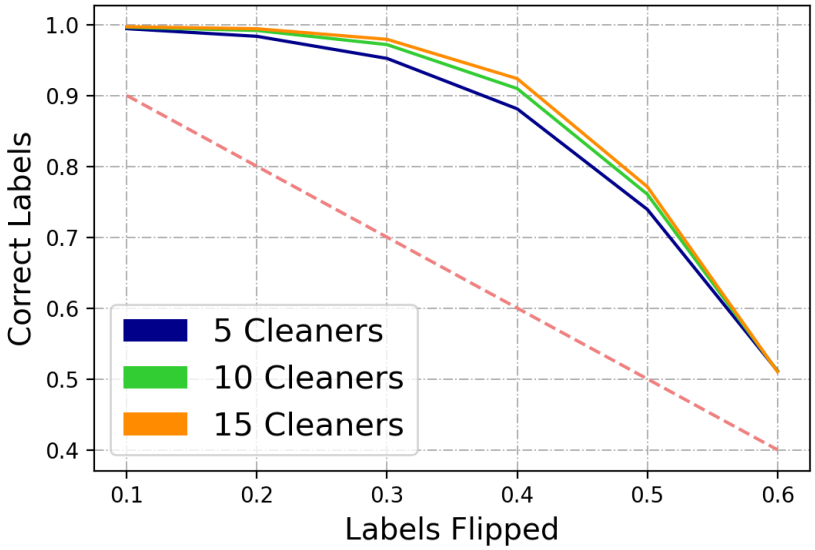

Figure 16. Experiment II: OLR label recovery ($\phi = 0.25$).

6.3. Experiment III

This experiment considers $\phi = 0.35$ as the OLR sampling factor. The results illustrated in Figure 17 are obtained based on the methods illustrated in Figures 8–10. OLR increases accuracy, recall, precision, and F1-score of IoT profiling with labels flipped. Furthermore, it becomes more difficult for labels to be recovered as the label-flipping factor increases.

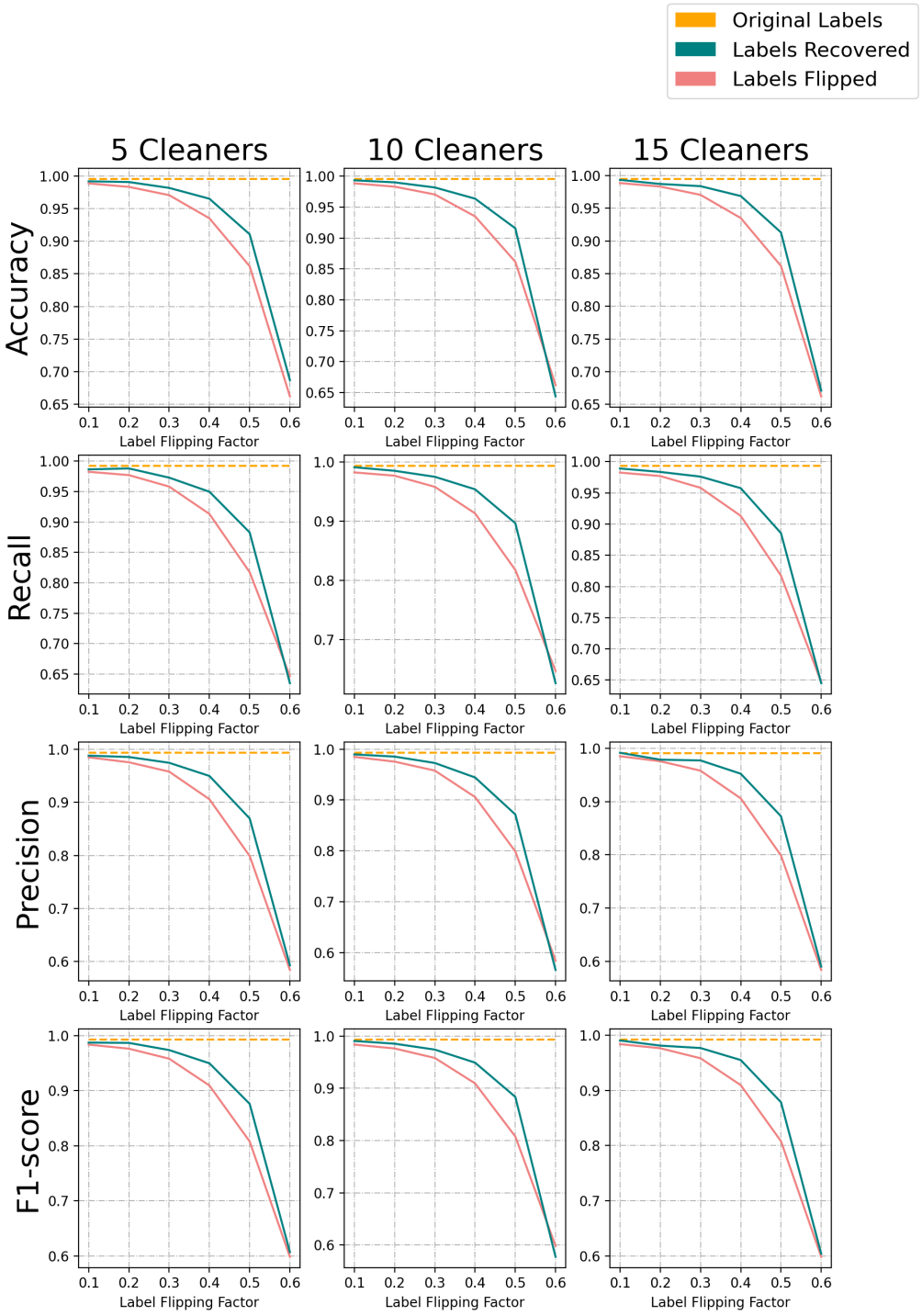

Figure 17. Experiment III: IoT Profiling ($\phi = 0.35$).

Figure 18 compares the performance of IoT profiling using different OLR configurations and shows that having 10 cleaners is slightly better in cases where the label flipping factor is less than 0.3. However, although all configurations can yield high performance, presenting similar outcomes, increasing the number of cleaners improves overall performance in more challenging cases.

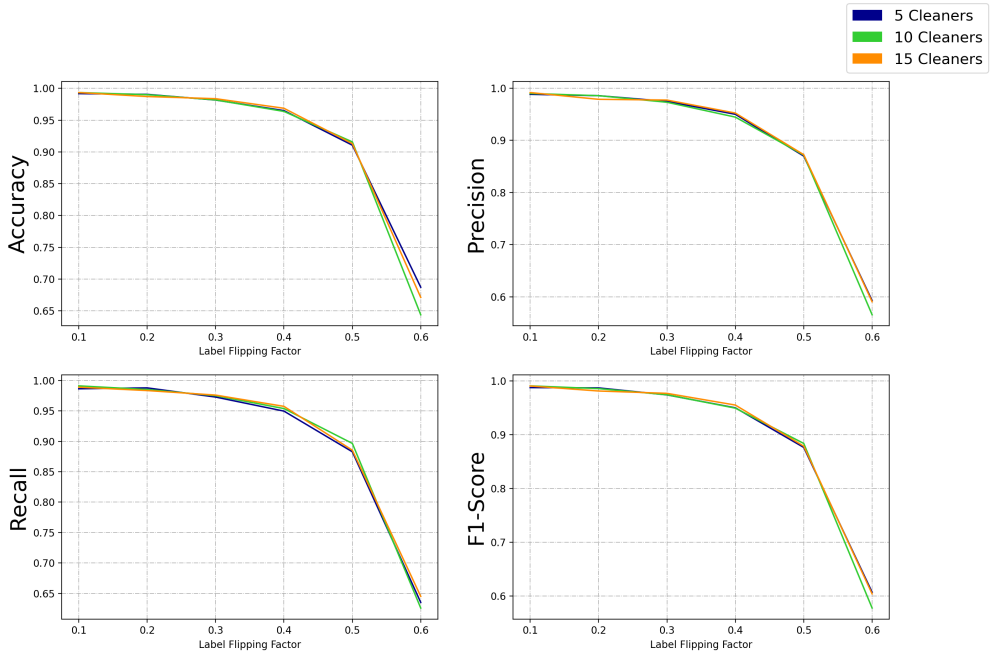

Figure 18. Experiment III: Comparison of OLR configurations ($\phi = 0.35$).

Finally, Figure 19 illustrates the capability of different OLR configurations in terms of label recovery, presenting several labels recovered in all cases. Furthermore, increasing the number of cleaners improves overall performance in most cases, especially when the label-flipping factors vary from 0.2 to 0.4.

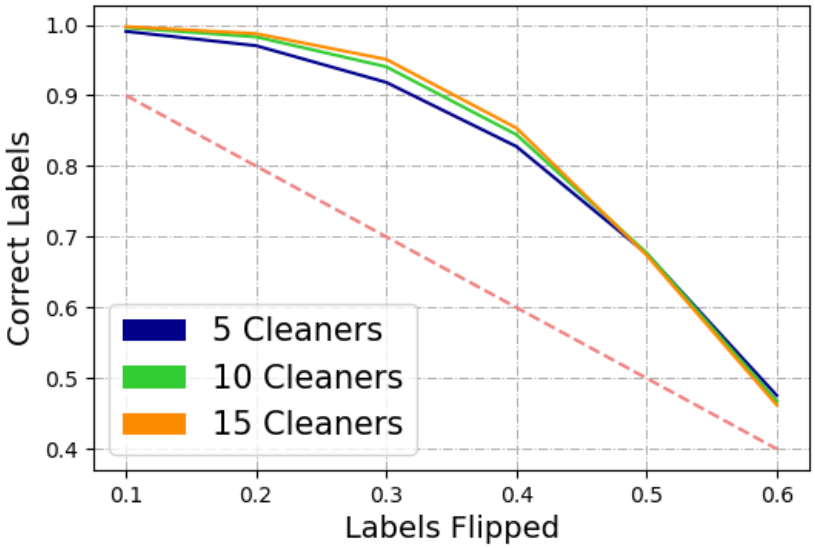

Figure 19. ExperimentIII: OLR label recovery ($\phi = 0.35$).

6.4. Evaluation

Figure 20 shows the performance of the different OLR configurations across all experiments. These graphs show that having more cleaners can be beneficial in many cases. However, as we increase ϕ, reducing the number of cleaners can improve the score and the overall performance. This is an insightful finding since it shows that training overlap needs to be controlled to maintain good performance. On the other hand, considering a few cleaners with small fractions of the training data can be problematic as the underlying learning process will not consider several aspects of the dataset.

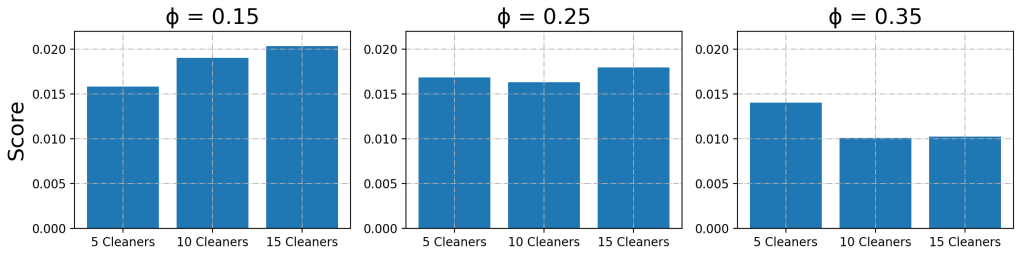

Figure 20. Performance of the different OLR configurations across all experiments (Equation (1)).

7. Conclusions

In this paper, we introduced the Overlapping Label Recovery (OLR) framework to mitigate the effects of label-flipping attacks in Deep-Learning-based IoT profiling. The current security challenges faced by IoT operators can be mitigated by novel methods, such as Deep-learning-based IoT profiling. However, label flipping can compromise the performance of such methods by manipulating data. OLR uses Random Forests (RF) as underlying cleaners to recover labels and re-evaluates the training dataset to recover its labels. The results are obtained by evaluating Deep Learning (DL) models using the CIC IoT Dataset 2022 demonstrating that training overlap needs to be controlled to maintain good performance and that the proposed strategy improves the overall profiling performance in all the cases investigated. In the future directions of this research, the authors intend to investigate the efficiency of other recovery methods for IoT profiling and present a comprehensive comparison with the results presented in this paper. Also, we will investigate how OLR can improve IoT profiling in federated environments.

Author Contributions: Methodology, E.C.P.N.; Validation, H.M.; Formal analysis, S.D.; Investigation, S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Da Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243.
2. Nauman, A.; Qadri, Y.A.; Amjad, M.; Zikria, Y.B.; Afzal, M.K.; Kim, S.W. Multimedia Internet of Things: A comprehensive survey. *IEEE Access* **2020**, *8*, 8202–8250. [CrossRef]
3. Habibzadeh, H.; Dinesh, K.; Shishvan, O.R.; Boggio-Dandry, A.; Sharma, G.; Soyata, T. A survey of healthcare Internet of Things (HIoT): A clinical perspective. *IEEE Internet Things J.* **2019**, *7*, 53–71. [CrossRef] [PubMed]
4. Lee, S.K.; Bae, M.; Kim, H. Future of IoT networks: A survey. *Appl. Sci.* **2017**, *7*, 1072. [CrossRef]
5. Marjani, M.; Nasaruddin, F.; Gani, A.; Karim, A.; Hashem, I.A.T.; Siddiqa, A.; Yaqoob, I. Big IoT data analytics: architecture, opportunities, and open research challenges. *IEEE Access* **2017**, *5*, 5247–5261.
6. Hajjaji, Y.; Boulila, W.; Farah, I.R.; Romdhani, I.; Hussain, A. Big data and IoT-based applications in smart environments: A systematic review. *Comput. Sci. Rev.* **2021**, *39*, 100318. [CrossRef]
7. Madakam, S.; Ramaswamy, R.; Tripathi, S. Internet of Things (IoT): A literature review. *J. Comput. Commun.* **2015**, *3*, 164–173. [CrossRef]
8. Čolaković, A.; Hadžialić, M. Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. *Comput. Netw.* **2018**, *144*, 17–39. [CrossRef]
9. Selvaraj, S.; Sundaravaradhan, S. Challenges and opportunities in IoT healthcare systems: A systematic review. *SN Appl. Sci.* **2020**, *2*, 1–8. [CrossRef]
10. Akkaş, M.A.; Sokullu, R.; Cetin, H.E. Healthcare and patient monitoring using IoT. *Internet Things* **2020**, *11*, 100173. [CrossRef]
11. Mohammed, J.; Lung, C.H.; Ocneanu, A.; Thakral, A.; Jones, C.; Adler, A. Internet of Things: Remote patient monitoring using web services and cloud computing. In Proceedings of the 2014 IEEE International Conference on Internet of Things (IThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom), Taipei, Taiwan, 1–3 September 2014; pp. 256–263.
12. Zantalis, F.; Koulouras, G.; Karabetsos, S.; Kandris, D. A review of machine learning and IoT in smart transportation. *Future Internet* **2019**, *11*, 94. [CrossRef]
13. Uma, S.; Eswari, R. Accident prevention and safety assistance using IOT and machine learning. *J. Reliab. Intell. Environ.* **2022**, *8*, 79–103. [CrossRef]

14. Celesti, A.; Galletta, A.; Carnevale, L.; Fazio, M.; Ĺay-Ekuakille, A.; Villari, M. An IoT cloud system for traffic monitoring and vehicular accidents prevention based on mobile sensor data processing. *IEEE Sens. J.* **2017**, *18*, 4795–4802. [CrossRef]
15. Cheng, J.; Chen, W.; Tao, F.; Lin, C.L. Industrial IoT in 5G environment towards smart manufacturing. *J. Ind. Inf. Integr.* **2018**, *10*, 10–19. [CrossRef]
16. Al-Emran, M.; Malik, S.I.; Al-Kabi, M.N. A survey of Internet of Things (IoT) in education: Opportunities and challenges. In *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 197–209.
17. Pate, J.; Adegbija, T. AMELIA: An application of the Internet of Things for aviation safety. In Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–6.
18. Salam, A. Internet of things for sustainable forestry. In *Internet of Things for Sustainable Community Development*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 147–181.
19. Cisco, U. *Cisco Annual Internet Report (2018–2023) White Paper*; Cisco: San Jose, CA, USA, 2020.
20. Vermesan, O.; Friess, P.; Guillemin, P.; Giaffreda, R.; Grindvoll, H.; Eisenhauer, M.; Serrano, M.; Moessner, K.; Spirito, M.; Blystad, L.C.; et al. Internet of things beyond the hype: Research, innovation and deployment. In *Building the Hyperconnected Society-Internet of Things Research and Innovation Value Chains, Ecosystems and Markets*; River Publishers: Roma, Italy, 2022; pp. 15–118.
21. Shafique, K.; Khawaja, B.A.; Sabir, F.; Qazi, S.; Mustaqim, M. Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios. *IEEE Access* **2020**, *8*, 23022–23040. [CrossRef]
22. Lee, S.H.; Shiue, Y.L.; Cheng, C.H.; Li, Y.H.; Huang, Y.F. Detection and Prevention of DDoS Attacks on the IoT. *Appl. Sci.* **2022**, *12*, 12407. [CrossRef]
23. Safi, M.; Dadkhah, S.; Shoeleh, F.; Mahdikhani, H.; Molyneaux, H.; Ghorbani, A.A. A Survey on IoT Profiling, Fingerprinting, and Identification. *ACM Trans. Internet Things* **2022**, *3*, 1–39. [CrossRef]
24. Sharma, S.; Kaushik, B. A survey on internet of vehicles: Applications, security issues & solutions. *Veh. Commun.* **2019**, *20*, 100182.
25. Abrishami, M.; Dadkhah, S.; Neto, E.C.P.; Xiong, P.; Iqbal, S.; Ray, S.; Ghorbani, A.A. Classification and Analysis of Adversarial Machine Learning Attacks in IoT: A Label Flipping Attack Case Study. In Proceedings of the 2022 32nd Conference of Open Innovations Association (FRUCT), Tampere, Finland, 9–11 November 2022; pp. 3–14.
26. Krishnan, P.; Jain, K.; Buyya, R.; Vijayakumar, P.; Nayyar, A.; Bilal, M.; Song, H. MUD-based behavioral profiling security framework for software-defined IoT networks. *IEEE Internet Things J.* **2021**, *9*, 6611–6622. [CrossRef]
27. Hamza, A.; Ranathunga, D.; Gharakheili, H.H.; Benson, T.A.; Roughan, M.; Sivaraman, V. Verifying and monitoring iots network behavior using mud profiles. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1–18. [CrossRef]
28. Safi, M.; Kaur, B.; Dadkhah, S.; Shoeleh, F.; Lashkari, A.H.; Molyneaux, H.; Ghorbani, A.A. Behavioural Monitoring and Security Profiling in the Internet of Things (IoT). In Proceedings of the 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Haikou, China, 20–22 December 2021; pp. 1203–1210.
29. Dadkhah, S.; Mahdikhani, H.; Danso, P.K.; Zohourian, A.; Truong, K.A.; Ghorbani, A.A. Towards the development of a realistic multidimensional IoT profiling dataset. In Proceedings of the 2022 19th Annual International Conference on Privacy, Security & Trust (PST), Fredericton, NB, Canada, 22–24 August 2022; pp. 1–11.
30. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 480–501.
31. Lyu, L.; Yu, H.; Yang, Q. Threats to federated learning: A survey. *arXiv* **2020**, arXiv:2003.02133.
32. Nuding, F.; Mayer, R. 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, 24–27 April 2022; pp. 24–34.
33. Sitawarin, C.; Bhagoji, A.N.; Mosenia, A.; Chiang, M.; Mittal, P. Darts: Deceiving autonomous cars with toxic signs. *arXiv* **2018**, arXiv:1802.06430.
34. Rosenfeld, E.; Winston, E.; Ravikumar, P.; Kolter, Z. Certified robustness to label-flipping attacks via randomized smoothing. In Proceedings of the International Conference on Machine Learning, Virtual, 12–18 July 2020; pp. 8230–8241.
35. Chan, P.P.; Luo, F.; Chen, Z.; Shu, Y.; Yeung, D.S. Transfer learning based countermeasure against label flipping poisoning attack. *Inf. Sci.* **2021**, *548*, 450–460. [CrossRef]
36. Demertzi, V.; Demertzis, S.; Demertzis, K. An overview of cyber threats, attacks and countermeasures on the primary domains of smart cities. *Appl. Sci.* **2023**, *13*, 790. [CrossRef]
37. Mulero-Palencia, S.; Monzon Baeza, V. Detection of Vulnerabilities in Smart Buildings Using the Shodan Tool. *Electronics* **2023**, *12*, 4815. [CrossRef]
38. Korium, M.S.; Saber, M.; Beattie, A.; Narayanan, A.; Sahoo, S.; Nardelli, P.H. Intrusion detection system for cyberattacks in the Internet of Vehicles environment. *Ad Hoc Netw.* **2024**, *153*, 103330. [CrossRef]
39. Chen, H.; Liu, J.; Wang, J.; Xun, Y. Towards secure intra-vehicle communications in 5G advanced and beyond: Vulnerabilities, attacks and countermeasures. *Veh. Commun.* **2023**, *39*, 100548. [CrossRef]
40. Taslimasa, H.; Dadkhah, S.; Neto, E.C.P.; Xiong, P.; Ray, S.; Ghorbani, A.A. Security issues in Internet of Vehicles (IoV): A comprehensive survey. *Internet Things* **2023**, *22*, 100809. [CrossRef]

41. Hernandez-Jaimes, M.L.; Martinez-Cruz, A.; Ramírez-Gutiérrez, K.A.; Feregrino-Uribe, C. Artificial intelligence for IoMT security: A review of intrusion detection systems, attacks, datasets and Cloud-Fog-Edge architectures. *Internet Things* **2023**, *23*, 100887. [CrossRef]
42. Ahmed, S.F.; Alam, M.S.B.; Afrin, S.; Rafa, S.J.; Rafa, N.; Gandomi, A.H. Insights into Internet of Medical Things (IoMT): Data fusion, security issues and potential solutions. *Inf. Fusion* **2024**, *102*, 102060. [CrossRef]
43. Al-Hawawreh, M.; Alazab, M.; Ferrag, M.A.; Hossain, M.S. Securing the Industrial Internet of Things against ransomware attacks: A comprehensive analysis of the emerging threat landscape and detection mechanisms. *J. Netw. Comput. Appl.* **2023**, *223*, 103809. [CrossRef]
44. Chaudhary, S.; Mishra, P.K. DDoS attacks in Industrial IoT: A survey. *Comput. Netw.* **2023**, *236*, 110015. [CrossRef]
45. Yang, K.; Li, Q.; Sun, L. Towards automatic fingerprinting of IoT devices in the cyberspace. *Comput. Netw.* **2019**, *148*, 318–327. [CrossRef]
46. Bezawada, B.; Bachani, M.; Peterson, J.; Shirazi, H.; Ray, I.; Ray, I. Behavioral fingerprinting of iot devices. In Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security, Toronto, ON, Canada, 19 October 2018; pp. 41–50.
47. Thangavelu, V.; Divakaran, D.M.; Sairam, R.; Bhunia, S.S.; Gurusamy, M. DEFT: A distributed IoT fingerprinting technique. *IEEE Internet Things J.* **2018**, *6*, 940–952. [CrossRef]
48. Xia, W.; Wen, Y.; Foh, C.H.; Niyato, D.; Xie, H. A survey on software-defined networking. *IEEE Commun. Surv. Tutor.* **2014**, *17*, 27–51. [CrossRef]
49. Ali, M.N.; Imran, M.; din, M.S.u.; Kim, B.S. Low Rate DDoS Detection Using Weighted Federated Learning in SDN Control Plane in IoT Network. *Appl. Sci.* **2023**, *13*, 1431. [CrossRef]
50. Yi, B.; Wang, X.; Li, K.; Huang, M. A comprehensive survey of network function virtualization. *Comput. Netw.* **2018**, *133*, 212–262. [CrossRef]
51. Ferman, V.A.; Tawfeeq, M.A. Machine learning challenges for IoT device fingerprints identification. *J. Phys. Conf. Ser.* **2021**, *1963*, 012046. [CrossRef]
52. Rose, J.R.; Swann, M.; Bendiab, G.; Shiaeles, S.; Kolokotronis, N. Intrusion detection using network traffic profiling and machine learning for IoT. In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; pp. 409–415.
53. Lee, S.Y.; Wi, S.r.; Seo, E.; Jung, J.K.; Chung, T.M. ProFiOt: Abnormal Behavior Profiling (ABP) of IoT devices based on a machine learning approach. In Proceedings of the 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, Australia, 22–24 November 2017; pp. 1–6.
54. Babun, L.; Aksu, H.; Ryan, L.; Akkaya, K.; Bentley, E.S.; Uluagac, A.S. Z-iot: Passive device-class fingerprinting of zigbee and z-wave iot devices. In Proceedings of the ICC 2020-2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–7.
55. Aneja, S.; Aneja, N.; Islam, M.S. IoT device fingerprint using deep learning. In Proceedings of the 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS), Bali, Indonesia, 1–3 November 2018; pp. 174–179.
56. Msadek, N.; Soua, R.; Engel, T. Iot device fingerprinting: Machine learning based encrypted traffic analysis. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019; pp. 1–8.
57. Jafari, H.; Omotere, O.; Adesina, D.; Wu, H.H.; Qian, L. IoT devices fingerprinting using deep learning. In Proceedings of the MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018; pp. 1–9.
58. Xiao, H.; Biggio, B.; Nelson, B.; Xiao, H.; Eckert, C.; Roli, F. Support vector machines under adversarial label contamination. *Neurocomputing* **2015**, *160*, 53–62. [CrossRef]
59. Zhang, H.; Cheng, N.; Zhang, Y.; Li, Z. Label flipping attacks against Naive Bayes on spam filtering systems. *Appl. Intell.* **2021**, *51*, 4503–4514. [CrossRef]
60. Lukasik, M.; Bhojanapalli, S.; Menon, A.; Kumar, S. Does label smoothing mitigate label noise? In Proceedings of the International Conference on Machine Learning, Virtual, 12–18 July 2020; pp. 6448–6458.
61. Menon, A.K.; Rawat, A.S.; Reddi, S.J.; Kumar, S. Can gradient clipping mitigate label noise? In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
62. Paudice, A.; Muñoz-González, L.; Lupu, E.C. Label sanitization against label flipping poisoning attacks. In Proceedings of the ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, 10–14 September 2018; Proceedings 18; Springer: Berlin/Heidelberg, Germany, 2019; pp. 5–15.
63. Ortego, D.; Arazo, E.; Albert, P.; O'Connor, N.E.; McGuinness, K. Multi-objective interpolation training for robustness to label noise. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 6606–6615.
64. Zhang, K.; Tao, G.; Xu, Q.; Cheng, S.; An, S.; Liu, Y.; Feng, S.; Shen, G.; Chen, P.Y.; Ma, S.; et al. FLIP: A Provable Defense Framework for Backdoor Mitigation in Federated Learning. *arXiv* **2022**, arXiv:2210.12873.
65. Lv, Z.; Cao, H.; Zhang, F.; Ren, Y.; Wang, B.; Chen, C.; Li, N.; Chang, H.; Wang, W. AWFC: Preventing Label Flipping Attacks Towards Federated Learning for Intelligent IoT. *Comput. J.* **2022**, *65*, 2849–2859. [CrossRef]
66. Li, D.; Wong, W.E.; Wang, W.; Yao, Y.; Chau, M. Detection and mitigation of label-flipping attacks in federated learning systems with KPCA and K-means. In Proceedings of the 2021 8th International Conference on Dependable Systems and Their Applications (DSA), Yinchuan, China, 11–12 September 2021; pp. 551–559.

67. Li, L.; Fan, Y.; Tse, M.; Lin, K.Y. A review of applications in federated learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [CrossRef]
68. Zhang, H.; Tae, K.H.; Park, J.; Chu, X.; Whang, S.E. iFlipper: Label Flipping for Individual Fairness. *arXiv* **2022**, arXiv:2209.07047.
69. Sharma, R.; Sharma, G.; Pattanaik, M. A CatBoost Based Approach to Detect Label Flipping Poisoning Attack in Hardware Trojan Detection Systems. *J. Electron. Test.* **2022**, *38*, 667–682. [CrossRef]
70. Yang, R.; He, H.; Wang, Y.; Qu, Y.; Zhang, W. Dependable federated learning for IoT intrusion detection against poisoning attacks. *Comput. Secur.* **2023**, *132*, 103381. [CrossRef]
71. Jiang, Y.; Zhang, W.; Chen, Y. Data quality detection mechanism against label flipping attacks in federated learning. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1625–1637. [CrossRef]
72. Taheri, R.; Javidan, R.; Shojafar, M.; Pooranian, Z.; Miri, A.; Conti, M. On defending against label flipping attacks on malware detection systems. *Neural Comput. Appl.* **2020**, *32*, 14781–14800. [CrossRef]
73. Mahmoud, R.; Yousuf, T.; Aloul, F.; Zualkernan, I. Internet of things (IoT) security: Current status, challenges and prospective measures. In Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 14–16 December 2015; pp. 336–341.
74. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, *2018*, 7068349. [CrossRef]
75. Otter, D.W.; Medina, J.R.; Kalita, J.K. A survey of the usages of deep learning for natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 604–624. [CrossRef]
76. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
77. Svozil, D.; Kvasnicka, V.; Pospichal, J. Introduction to multi-layer feed-forward neural networks. *Chemom. Intell. Lab. Syst.* **1997**, *39*, 43–62. [CrossRef]
78. Sazli, M.H. A brief review of feed-forward neural networks. *Commun. Fac. Sci. Univ. Ank. Ser.-Phys. Sci. Eng.* **2006**, *50*. [CrossRef]
79. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
80. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT press: Cambridge, MA, USA, 2016.
81. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
82. Cutler, A.; Cutler, D.R.; Stevens, J.R. Random forests. In *Ensemble Machine Learning: Methods and Applications*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 157–175.
83. Robnik-Šikonja, M. Improving random forests. In Proceedings of the Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, 20–24 September 2004; Proceedings 15; Springer: Berlin/Heidelberg, Germany, 2004; pp. 359–370.
84. Liu, J.; Cao, Y.; Li, Y.; Guo, Y.; Deng, W. A big data cleaning method based on improved CLOF and Random Forest for distribution network. *CSEE J. Power Energy Syst.* **2020**, *early access*.
85. Gu, J. *Random Forest Based Imbalanced Data Cleaning and Classification*; Citeseer: Gothenburg, Sweden, 2007.
86. Sapountzoglou, N.; Lago, J.; Raison, B. Fault diagnosis in low voltage smart distribution grids using gradient boosting trees. *Electr. Power Syst. Res.* **2020**, *182*, 106254. [CrossRef]
87. Gajera, V.; Gupta, R.; Jana, P.K. An effective multi-objective task scheduling algorithm using min-max normalization in cloud computing. In Proceedings of the 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Bangalore, India, 21–23 July 2016; pp. 812–816.