Federated Learning for Robust Malware Detection under Noisy Labels and Malicious Sybils

Andrea Augello[1,*], Alessandra De Paola[1] and Giuseppe Lo Re[1]

[1] *Department of Engineering, University of Palermo, Italy*

Abstract

Malware poses a significant threat to computing systems, and federated learning (FL) has emerged as a promising privacy-preserving approach to collaboratively train detection models through crowdsourcing. However, FL relies on local client labels that can be noisy or corrupted, and some clients may behave maliciously to poison the global model, leading to severely degraded performance. While techniques for either noisy-label robust FL or defenses against malicious clients exist, few approaches jointly address both challenges, which are likely to co-occur in realistic cybersecurity use-cases. This work presents a federated learning system for robust malware detection in this challenging setting. The proposed pipeline leverages properties of the client update gradients to determine a core set of reliable participants. A cluster-based, distance-aware scoring mechanism isolates low-quality and malicious contributors. Subsequently, the identified clients undergo label refinement through a semi-supervised per-client noisy-label detection and correction using the global model as a reference. Experiments on a publicly available Android Malware dataset show the system maintains high detection accuracy even with high fractions of noisy and adversarial clients, with up to 80% of the training data corrupted, outperforming representative baselines.

Keywords

Federated Learning, Malware Detection, Noisy Labels, Sybil Attack, Robust Aggregation

1. Introduction

Connected devices, cloud platforms, and mobile services have become an essential part of everyday life. Their widespread deployment has, however, expanded the available attack surface, enabling a growing range of malware that threatens the confidentiality, integrity, and availability of systems and data [1]. This evolution has driven a persistent arms race where attackers continually refine and diversify their tactics to evade detection and compromise systems in novel ways [2].

Machine learning techniques have become central to malware detection, enabling the discovery of patterns associated with malicious behaviors and representing a major focus in cybersecurity research [3]. Yet, in practical deployments these ML-based detectors often fail to meet expectations. Such gaps stem both from weaknesses in experimental practice [3],[4] and from constraints encountered in real-world operational environments. An enduring difficulty is that trained detectors quickly become outdated [5]. Adversaries continuously evolve their evasion strategies, leading to model aging that gradually undermines the accuracy of static models [6]. Consequently, models need frequent retraining on up-to-date samples to remain effective against the current threat landscape [7, 8].

However, high-quality labeled examples are scarce, limiting the efficacy of retraining. While gathering data is relatively straightforward, for instance by periodically crawling app stores, generating trustworthy labels for the collected samples remains costly and labor-intensive [9]. For this reason, Federated Learning (FL) has emerged as an enabling technology for crowdsourcing data and collaboratively train malware detectors in a privacy-preserving manner [10], [9]. Federated Learning [11] is a distributed learning framework that enables multiple clients to collaboratively train a shared model without exchanging raw data [12]. In FL, multiple clients perform local training of copies of the same model on their private datasets and send model weight updates to a central server, which aggregates

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT

*Corresponding author.

andrea.augello01@unipa.it (A. Augello); alessandra.depaola@unipa (A. De Paola); giuseppe.lore@unipa.it (G. Lo Re)

0000-0001-9085-4218 (A. Augello); 0000-0002-7340-1847 (A. De Paola); 0000-0002-8217-2230 (G. Lo Re)

them to update the global model. Over multiple rounds of communication, the global model converges to a state that reflects the collective knowledge of all clients while never exposing the local data of any participant. The FL approach is particularly appealing for the malware detection domain, where data privacy and regulatory compliance are paramount. Information on the installed applications and their behaviors is sensitive, as it may reveal user habits, preferences, and private information [13], and clients may be reluctant to share raw data with a central authority.

Nevertheless, despite the advantages brought by FL, the practical deployment of FL-based systems for malware detection faces two intertwined challenges:

1. Participating clients are not necessarily expert annotators of malware samples. Their local datasets may contain **noisy labels** due to misclassification of stealthy malware as benign or mislabeling of goodware as malicious software.
2. Some clients may be **malicious** and attempt to poison the global model via label flipping or crafted updates (e.g., in order to evade detection of their own malware).

These two challenges both stem from the fact that the server has no direct access to client data, and cannot reliably assess the quality of local labels or the integrity of client updates. Despite the growing body of work on either noisy-label robust FL or defenses against malicious clients, few approaches address both issues simultaneously, yet in the malware detection domain they are likely to co-occur. Non-expert crowd-sourced annotators lead to accidental noise. Adversaries, meanwhile, have a clear incentive to perform targeted attacks (like label flipping) to evade detection. These two sources of corruption are fundamentally independent and must be addressed concurrently to ensure robust model training.

In order to tackle these limitations, this paper proposes a **Fed**erated learning pipeline for **Ro**bust **Se**lection and **C**orrection (Fed-RoSeC) that identifies and mitigates both noisy labels and malicious clients. Fed-RoSeC leverages update gradients behavior to identify reliable clients, which train a robust global model; suspicious clients are isolated and undergo a semi-supervised label correction process using the global model as a reference. The main contributions of this work are:

- Fed-RoSeC includes a novel distance-based client scoring and clustering procedure that separates honest from suspicious participants based on the statistical properties of their model updates, enabling effective isolation of malicious and low-quality clients;
- By exploiting the multimodality of per-sample loss distributions under a reasonably-trained global model, Fed-RoSeC implements a per-client noisy-label detection and correction mechanism that conservatively relabels likely-mislabeled samples using the global model as an oracle, improving local data quality without compromising privacy;
- An extensive experimental evaluation on a large-scale Android malware dataset demonstrates that Fed-RoSeC can be used to train a malware detection model that maintains high detection accuracy even with high fractions of noisy and adversarial clients, outperforming representative baselines across a wide range of noise and attack intensities.

The rest of the paper is organized as follows. Section 2 reviews related work on robust federated learning and malware detection. Section 3 details the proposed Fed-RoSeC algorithm. Finally, Section 4 presents the experimental evaluation, and Section 5 concludes the paper.

2. Related Works

Early malware detection relied on signature-based scanning; these methods are fragile to obfuscation and morphing techniques. Subsequent generations of detectors introduced heuristics, behavioral detection, sandbox-based dynamic analysis, and combined systems that integrate multiple signals to improve recall on sophisticated threats [14]. Dynamic analysis and sandboxing remain central for uncovering evasive behaviors, although time and resource constraints limit exhaustive execution of long-lived evasive payloads [15, 16].

Many machine learning approaches for malware detection have been designed over the last decade, leveraging static features [10], dynamic traces [16], and hybrid representations [17]. Models range from feature-engineered classical classifiers to deep networks trained on raw traces; however, all such approaches depend on label quality. In many malware corpora labels are obtained via automated antivirus scanners. While an effective heuristic to generate large-scale labels for scientific research, AV-based labeling is noisy and lags behind emerging threats; thus, models trained on such data can be exploited by adversaries that adopt novel malware variants.

To address the scarcity of up-to-date labeled data, both for malware detection and in other cybersecurity scenarios, the scientific community has explored several approaches to mitigate accuracy degradation due to model aging, such as active learning [18, 19]. Nevertheless, these approaches still require periodic access to fresh labeled samples. Crowdsourcing through Federated learning (FL) is thus a promising avenue to enable collaborative model retraining while preserving data privacy [10]. When FL is applied to security domains such as mobile malware detection or threat intelligence, two practical problems surface simultaneously: clients may hold noisy labels (e.g., AV heuristics or weak crowd labels), and some clients may behave adversarially to poison the global model. A large body of recent work addresses these concerns from complementary angles.

One strand of work augments the standard FedAvg pipeline with modules for label calibration and selective relabeling. Representative methods estimate per-client noise and apply conservative relabeling or knowledge distillation to avoid propagating errors; examples include schemes that use local loss statistics to identify likely-noisy samples [20]. These approaches typically use the global model as a reference to relabel only high-confidence samples, and introduce weighted regularization or post-correction selection to limit adverse effects of incorrect corrections. Another class of methods leverages small, trusted benchmark datasets hosted at the server to estimate client credibility and to guide aggregation weights [21, 22]. These techniques trade off the need for a tiny trusted set against stronger robustness guarantees: benchmark-driven credibility scores allow the server to down-weight clients whose local statistics diverge from the trusted distribution. Nevertheless, the requirement for a trusted dataset may not always be feasible in practice and is in contrast with the initial data scarcity motivation for the use of FL in the cybersecurity domain.

Robust aggregation techniques provide an orthogonal defense by reducing the influence of extreme updates. Approaches such as coordinate-wise median [23], trimmed mean, and reputation-based weighting [24] have been studied extensively; more recent proposals adapt aggregation using public data augmentation or reweighting strategies to detect and suppress corrupt contributionsi [25]. These defenses are effective against untargeted (global) poisoning but can be evaded by sophisticated, targeted attacks when adversaries distribute their influence or mimic benign update statistics.

Clustering and representation-based defenses form a further line of work to defend FL against malicious clients. These methods exploit structure in the population of updates: server-side clustering of update-derived statistics [26], centroid alignment, or K-modes style categorization has been used to separate coherent groups of clients and identify outliers whose updates warrant further inspection or exclusion [27, 28]. Likewise, anomaly detectors trained on low-dimensional embeddings of updates have been proposed to flag malicious model submissions [29, 30].

Sampling- and selection-based schemes try to mitigate noisy labels by selecting higher-quality clients or higher-confidence samples during aggregation and local updates. Excluding contributions deemed low-quality can improve robustness to both label noise and adversarial manipulation [31]. Several works focus specifically on detecting malicious clients. Methods based on gradient diversity and history [32] detect groups of correlated updates indicative of Sybil or colluding attackers [24, 33]. Other works deploy dedicated anomaly detectors (including deep detectors and spectral methods) that detect updates with atypical reconstruction errors in a learned latent space [29, 30]. Feature-significance and explainability tools (e.g., SHAP) have also been applied to identify clients with anomalous influential features and to cluster clients based on those features [34].

Generative and meta-learning defenses have also been proposed to harden FL against label-poisoning. Generative adversarial label poisoners can be used to synthesize poisoned examples to make training robust to these samples [35], and meta-learning schemes have been employed to produce model initial-

izations that are less sensitive to noisy labels or adversarial perturbations by introducing synthetically corrupted samples [36]. These techniques aim to produce models and training schemes that generalize despite clients with corrupted data, but still require a majority of honest clients with clean data to be effective.

Noise-robust learning has been advanced via two complementary directions: model-level robust aggregation and data-level correction. Aggregation defenses attempt to limit the influence of extreme updates but can fail when adversaries concentrate their influence across many coordinates. Data-level methods operate by detecting and correcting or down-weighting corrupted labels; methods that use loss-based separation, co-training, or consensus predictions have been effective in centralized settings and have been adapted to FL with varying success. Fed-RoSeC bridges both directions by first isolating suspicious contributors and then applying conservative, model-guided relabeling at the client level.

3. The Fed-RoSeC algorithm

This work considers a FL deployment that aims to leverage crowdsourcing to build a malware detection model. A central server coordinates a population of N clients, each holding a private local dataset $\mathcal{D}_k = \{(x_i, y_i)\}$ of labeled samples, where x_i is a feature vector extracted from an application and $y_i \in \{0, 1\}$ is a binary label indicating whether the sample is benign (0) or malicious (1). Given that the clients are not assumed to be expert annotators of malware samples, their local datasets may contain label noise: they might not realize that some software is stealthily malicious and thus label it as benign, and some goodware might be mislabeled as malicious e.g., due to poor performance, such as crashes caused by software bugs.

The FL system must also contend with bad-faith participants that seek to undermine the learning process. A malicious adversary may control a subset of clients, which can behave arbitrarily to poison the global model. In this work, malicious clients are assumed to be unable to directly manipulate the model parameters sent to the server; instead, they can create arbitrarily many Sybil identities and use them to perform label flipping attacks [24]: all their malware samples are labeled as benign, and all their goodware samples are labeled as malicious, thereby maximizing label noise and attempting to mislead the global model. Three categories of clients are thus considered:

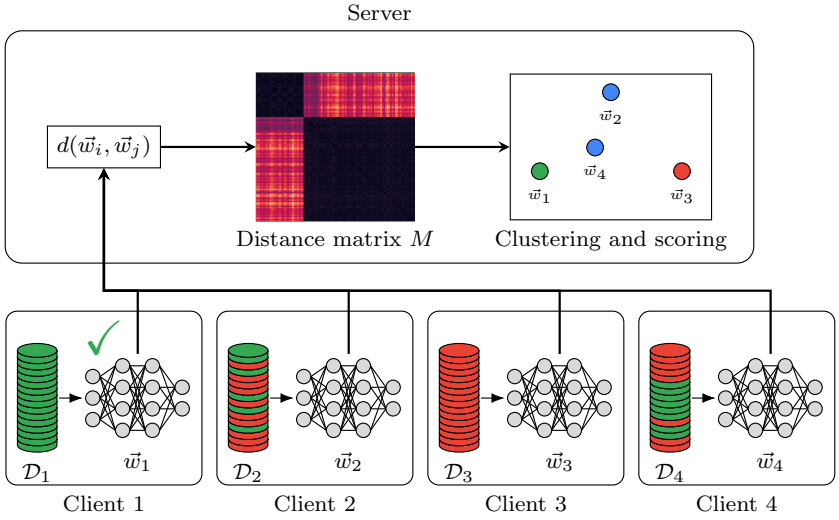
- **Honest clients**: they possess high-quality datasets with no label noise and follow the prescribed FL protocol.
- **Noisy clients**: they are honest participants that hold datasets with systematic label noise (i.e., a fraction of their samples are mislabeled) but follow the FL protocol.
- **Malicious clients**: they are controlled by an adversary and perform label-flipping attacks to poison the global model, potentially using multiple Sybil identities, but cannot directly manipulate model parameters.

The objective of Fed-RoSeC twofold: first, to identify and mitigate the effect of adversarial clients that attempt to poison the global model; second, to detect and correct systematic label noise present in honest clients so that the final global classifier maintains high detection performance. To achieve these objectives, a multi-stage pipeline is employed: identification of suspicious clients, per-client noisy-sample detection, conservative label correction using the global model as an oracle, and final training on filtered or corrected clients.

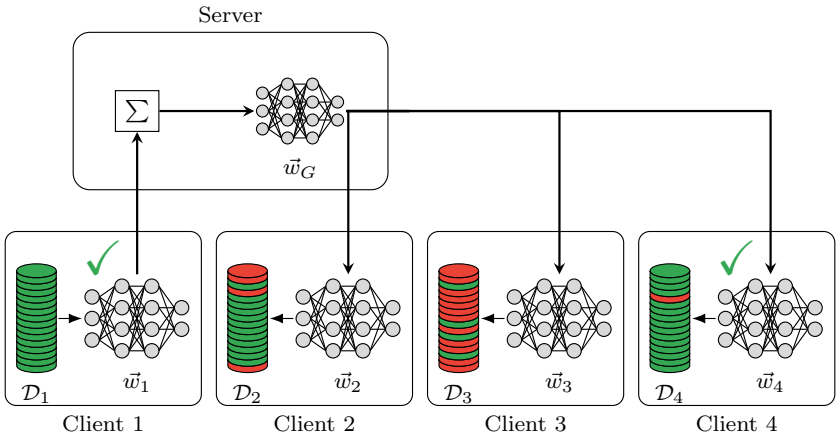
A high-level overview of the Fed-RoSeC pipeline is illustrated in Figure 1.

3.1. Noisy and malicious client identification

The first stage of Fed-RoSeC aims to identify clients that are either malicious or have low-quality datasets due to noisy labels. Initially, all clients participate in a few rounds of federated training to establish a baseline global model. In this phase, the loss function used for local training is the binary cross-entropy loss, with an additional proximal loss term to limit divergence from the global model,



(a) Model initialization and client clustering



(b) Label correction

Figure 1: Overview of Fed-RoSeC. (a) Initially, all clients participate in federated training to establish a baseline global model. Clustering is applied to groups similar clients, and clusters are scored to identify suspicious clients. (b) Client with high-quality data continue participating in training to refine the global model. Suspicious clients leverage the global model to correct noisy labels. The process iterates until convergence, after which a final training phase is executed with all clients using their corrected datasets.

as proposed in FedProx [37] to address data heterogeneity. The proximal term helps stabilize local updates in the presence of heterogeneous data distributions across clients, such as the varying levels of label noise considered in this work. By doing so, the local updates remain more consistent and comparable across clients, which aids in the subsequent identification of suspicious participants. The training objective for client k at round t is thus:

$$\mathcal{L}_k^{(t)} = \frac{1}{|\mathcal{D}_k^{(t)}|} \sum_{(x_i, y_i) \in \mathcal{D}_k^{(t)}} \mathrm{CE}\big(f_{\vec{w}_k^{(t)}}(x_i), y_i\big) + \frac{\mu}{2} \|\vec{w}_k^{(t)} - \vec{w}_G^{(t)}\|^2. \tag{1}$$

Inspired by [38], to ensure comparability of client updates, clients are selected for participation through a selection without replacement strategy. Hence, Fed-RoSeC ensures that, before selecting a client for participation after it has already contributed, all other clients are also selected for participation in an update step; thus, the global model never strays too far from the global average.

After this initial training phase has converged, the latest client updates are collected at the server for analysis. These updates have been computed starting from different global models, hence they are not easily comparable. In order to analyze the relationships among client updates, an appropriate

distance metric is required to capture both magnitude and overall directional differences. Accordingly, Fed-RoSeC employs the DCFL distance metric [39], which accounts for both the Euclidean distance between the local models and the angular alignment between their update directions. Specifically, given two clients i and j, who have last participated in rounds t_i and t_j respectively, their distance is computed as follows:

$$d(\vec{w}_i^{(t_i)}, \vec{w}_j^{(t_j)}) = \|\vec{w}_j^{(t_j)} - \vec{w}_i^{(t_i)}\| \exp\left(2\frac{\vec{w}_i^{(t_i)} - \vec{w}_j^{(t_j)}}{\|\vec{w}_i^{(t_i)} - \vec{w}_j^{(t_j)}\|} \cdot \frac{\vec{w}_i^{(t_i)} - \vec{w}_G^{(t_i)} - \vec{w}_j^{(t_j)} + \vec{w}_G^{(t_j)}}{\|\vec{w}_i^{(t_i)} - \vec{w}_G^{(t_i)}\| + \|\vec{w}_j^{(t_j)} - \vec{w}_G^{(t_j)}\|}\right). \quad (2)$$

The pairwise distances among all client updates are thus computed using Equation (2), which scales the Euclidean distance by an exponential factor that reflects the angular alignment of the updates relative to their respective starting global models. The resulting distance matrix M has entries $m_{i,j} = d(\vec{w}_i^{(t_i)}, \vec{w}_j^{(t_j)})$ for clients i and j. In the resulting distance matrix, each row corresponds to a client and encodes its relationship to all other participants. The underlying hypothesis is that honest clients will exhibit similar update patterns and, likewise, malicious clients will cluster together due to their correlated label-flipping behavior. These two groups are expected to be distinct in the distance space, while clients with noisy labels may exhibit intermediate behavior, resembling honest or malicious clients depending on the noise level. This hypothesis is supported by empirical observation, as shown in Figure 2, which illustrates distance matrices under different noise and attack mixes.

The key insight is that honest and malicious clients are clearly separable in the distance space, while noisy clients tend to cluster between these two groups depending on the noise level. Even at high noise levels, however, noisy clients remain distinguishable from malicious ones even if they do not resemble honest clients.

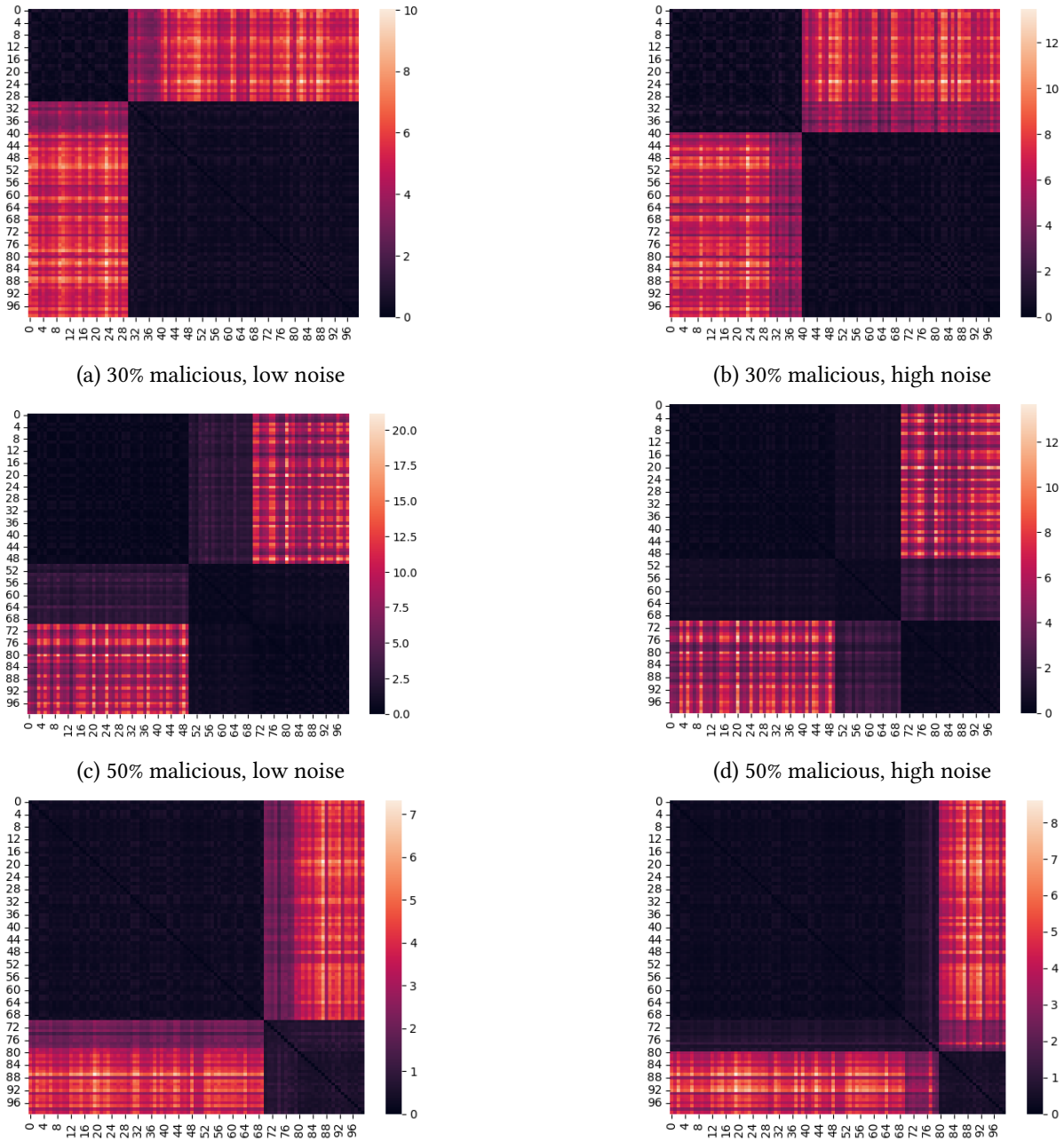
From the distance matrix M, a quantized matrix M_q is derived by discretizing distances into three ordinal levels (low, medium, high) using percentile-based binning (33rd and 66th percentiles as thresholds). This quantization step reduces sensitivity to minor fluctuations and emphasizes categorical relationships among clients: low distances indicate similar update behavior (honest - honest or malicious - malicious), while high distances suggest divergence (honest - malicious). The medium level captures intermediate relationships, often associated with noisy clients.

A K-Modes clustering [40] is then applied to the rows of M_q to group clients into $\sqrt{N}$ clusters $\{C_1, C_2, \ldots, C_{\sqrt{N}}\}$. The choice of $\sqrt{N}$ is a simple heuristic that balances granularity and statistical reliability: with N clients it yields moderately-sized clusters that avoid over-fragmentation (which would produce many noisy, small clusters) while retaining enough resolution to separate coherent groups, and it keeps computational cost manageable. K-Modes is chosen for its suitability to categorical data and its efficiency in handling large datasets. At this stage, there is no a priori labeling of clusters; they are simply groups of clients with similar distance profiles. To identify which clusters correspond to suspicious clients, an additional scoring mechanism is employed. Given a cluster C_i, its score Ω_i is computed through equation 3:

$$\Omega_i = \sum_{k \in C_i} \sum_{j \notin C_i} \frac{M_{k,j}}{|C_i||N - |C_i|}, \quad (3)$$

This score captures the average inter-cluster distance, normalized by the cluster size.

Once cluster-level scores are computed, a Gaussian Mixture Model (GMM) with three components is fit over the set of scores $\{\Omega_1, \Omega_2, \ldots, \Omega_{\sqrt{N}}\}$ to group clusters into homogeneous score categories so that there is a macro-cluster of high-cohesion client groups, a macro-cluster of low-cohesion groups, and an intermediate macro-cluster. The intermediate macro-cluster contains the clients with noisy labels, while which of the high- or low-cohesion macro-clusters corresponds to honest or malicious clients depends on the overall noise/attack mix. To resolve this ambiguity, Fed-RoSeC relies on the empirical observation that honest clients tend to produce more diverse updates [24], resulting in fatter tails in the distance distribution, while malicious clients produce more homogeneous updates, leading to a more peaked distance distribution. To quantify this diversity, Fed-RoSeC computes the Fisher excess



(a) 30% malicious, low noise

(b) 30% malicious, high noise

(c) 50% malicious, low noise

(d) 50% malicious, high noise

(e) 70% malicious, low noise

(f) 70% malicious, high noise

Figure 2: Distance matrices among client updates under different concentrations of malicious (top left in the matrices) and honest (bottom right in the matrices) clients. The noisy clients (between malicious and honest ones) are a fixed 15% of the population with 20% and 80% label noise. At low noise levels (left column), noisy clients are almost indistinguishable from honest ones, but their distance from malicious clients is lower than that of honest clients. At high noise levels (right column), noisy clients have a larger distance from honest clients, but they still have a distinct distance profile from malicious clients.

kurtosis κ of the distance distribution for all clients:

$$\kappa = \frac{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}(m_{i,j}-\bar{m})^4}{\left(\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}(m_{i,j}-\bar{m})^2\right)^2} - 3, \tag{4}$$

A negative excess kurtosis indicates that most clients have moderate distances to others, with few extreme values, suggesting that clusters with higher mean scores correspond to the anomalous group

(malicious clients). On the other hand, a positive excess kurtosis indicates that most clients have either very low or very high distances to others, suggesting that clusters with higher mean scores correspond to honest clients. Based on the sign of κ, Fed-RoSeC selects the appropriate macro-cluster as suspicious and isolates the clients within it for label correction. This dynamic selection mechanism allows Fed-RoSeC to adaptively identify suspicious clients across a range of noise and attack intensities, maintaining its effectiveness even with overwhelming proportions of malicious participants.

3.2. Semi-supervised label correction

Once suspicious clients have been isolated, Fed-RoSeC proceeds with a semi-supervised label correction phase so that even initially contaminated datasets can contribute useful information to the final global model. First, the global model is retrained using only the known honest clients, removing the influence of incorrectly labeled samples and producing a more reliable reference model for label correction. Following this, a per-sample loss analysis is performed locally by the clients with the incorrect labels to identify and correct likely noisy samples.

For each sample in the local dataset, the client computes the training loss under the current global model. High loss samples are likely to be mislabeled, as the global model (trained on mostly honest clients with clean data) should predict them correctly if their labels were accurate.

A GMM with two components is fit to the distribution of per-sample losses. Samples belonging to the high-loss component are flagged as likely noisy. Out of those, a conservative fraction of those with the highest losses are selected as candidates for relabeling. The fraction is chosen as to limit the risk of incorrect relabeling. In this work, the fraction is chosen to achieve a false positive rate (i.e., relabeling an already clean sample) of 5% given the parameters of the fitted GMM. Briefly, if the GMM components have means $\mu_1 < \mu_2$, standard deviations σ_1 and σ_2, and priors π_1 and π_2, the loss threshold τ is chosen such that:

$$\frac{(\tau-\mu_1)^2}{2\sigma_1^2} - \frac{(\tau-\mu_2)^2}{2\sigma_2^2} = \log\left(\frac{1-0.05}{0.05}\cdot\frac{\pi_1}{\pi_2}\right) - \log\frac{\sigma_1}{\sigma_2}. \tag{5}$$

The conservative nature of this selection aims to minimize the propagation of incorrect labels. These candidate samples are then relabeled using the global model's predictions.

After the relabeling step, each client estimates its post-correction noise rate $\hat{\mu}_k$ based on the fraction of samples in the dataset that are assigned to the high-loss component of the GMM after correction. Once the local correction step is complete for all suspicious clients, they evaluate their estimated post-correction noise rates. Clients whose estimated post-correction noise rate $\hat{\mu}_k$ falls closer to the clean side of the noise spectrum than to the noisy side, according to the average loss of the initial honest and suspicious client sets, are deemed to have sufficiently improved their dataset quality and restart participating again in the federated training rounds. This process of identification and correction is iterated until convergence, progressively improving the quality of the global model and the local datasets. In this phase, training is augmented using mixup data augmentation to further enhance robustness against residual label noise [41]. The mixup technique generates synthetic training samples by interpolating between pairs of examples and their labels, which helps the model generalize better in the presence of noisy labels. The loss function used during this phase combines the standard cross-entropy loss with a mixup regularization term, resulting in Eq. (6):

$$\mathcal{L}_k^{(t)} = (1-\lambda)\cdot\frac{1}{|\mathcal{D}_k^{(t)}|}\sum_{(x_i,y_i)\in\mathcal{D}_k^{(t)}}\mathrm{CE}\big(f_{\vec{w}_k^{(t)}}(x_i),y_i\big) + \lambda\cdot\frac{1}{|\tilde{\mathcal{D}}_k^{(t)}|}\sum_{(\tilde{x}_i,\tilde{y}_i)\in\tilde{\mathcal{D}}_k^{(t)}}\mathrm{CE}\big(f_{\vec{w}_k^{(t)}}(\tilde{x}_i),\tilde{y}_i\big), \tag{6}$$

with $\tilde{\mathcal{D}}_k^{(t)}$ being the mixup-augmented dataset and λ a hyperparameter controlling the balance between standard and mixup losses. The mixup augmentation is applied only during this correction phase to enhance robustness without affecting the initial identification of suspicious clients.

Once the correction process converges and no further clients change their labels, the last remaining clients in the high-noise group set all their labels to those predicted by the global model, effectively outsourcing their labeling to the server. Compared to the conservative per-sample relabeling performed

during the iterative correction phase, this final step is more aggressive but is applied only after all possible refinements have been made and the global model is expected to be of high quality and reliable, and these clients have already failed to improve their datasets through partial corrections. This intuition is supported by empirical evidence showing that, after iterative corrections, the global model achieves high accuracy even when a significant fraction of clients are noisy or malicious, as demonstrated in Section 4, Figure 4. Finally, a last training phase is executed where all clients, with the now clean datasets, participate through standard FedAvg aggregation to produce the final global model.

The overall Fed-RoSeC pipeline is summarized in Algorithm 1.

Algorithm 1 Fed-RoSeC algorithm

Require: Client datasets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$, initial global model $\vec{w}_G^{(1)}$
Ensure: Final global model $\vec{w}_G^{(T)}$
for round $t = 1$ to T_{init} **do**
  $\mathcal{C} \leftarrow$ Select clients without replacement
  $\vec{w}_G^{(t+1)} \leftarrow \frac{1}{|\mathcal{C}|} \sum_{k \in \mathcal{C}} \vec{w}_k^{(t)}$ updated using Eq. (1)
$M \leftarrow$ Compute pairwise DCFL distances using Eq. (2)
$M_q \leftarrow$ Quantize M into 3 levels
$\mathcal{C} \leftarrow$ K-Modes(M_q)
for $C_i \in \mathcal{C}$ **do**
  $\Omega_i \leftarrow \sum_{k \in C_i} \sum_{j \notin C_i} \frac{M_{k,j}}{|C_i||N-|C_i|}$ using Eq. (3)
$\kappa \leftarrow$ excess kurtosis using Eq. (4)
Macro-clusters $\leftarrow$ GMM($\{\Omega_i\}$)
if $\kappa < 0$ **then**
  Select low-score macro-cluster as honest
else
  Select high-score macro-cluster as honest
while not converged **do**
  $\vec{w}_G^{(t)} \leftarrow$ Retrain global model using only honest clients with Eq. (6)
  for each non-honest client k **do**
    Compute per-sample losses under current global model
    Fit GMM to losses and identify high-loss samples
    for x_i, y_i in samples flagged as high-loss with high confidence **do**
      $y_i \leftarrow \vec{w}_G^{(t)}(x_i)$ ▷ Relabel using global model
    $\hat{\mu}_k \leftarrow$ Estimate post-correction noise rate
    if $\hat{\mu}_k$ indicates clean data **then**
      Client k rejoins training
for each remaining non-honest client k **do**
  for each sample $(x_i, y_i) \in \mathcal{D}_k$ **do**
    $y_i \leftarrow \vec{w}_G^{(t)}(x_i)$ ▷ Relabel all samples using global model
for round $t = 1$ to T_{final} **do**
  $\vec{w}_G^{(t+1)} \leftarrow$ Standard FedAvg aggregation over all clients

4. Experimental Evaluation

In order to assess the effectiveness of Fed-RoSeC, an extensive experimental evaluation was conducted on the Kronodroid dataset [42]. This dataset contains 78 137 Android applications, of which 41 382 are labeled as malware and 36 755 as goodwares, covering most years of Android malware evolution from

2008 to 2020. The dataset contains both static and dynamic analysis features; in this work, only static features were used to simulate a realistic FL scenario where client devices are average smartphones with insufficient resources to perform dynamic analysis on virtualized sandboxes. The 200 static features include permissions, intent filters, and other manifest attributes.

To ensure a realistic evaluation, avoiding temporal snooping which might inflate performance [4], training and testing splits were created based on the application release date. Specifically, all applications released before 2016 were used for training, while those released from 2016 onward were reserved for testing.

Fed-RoSeC was compared against several representative baselines, also tuned on the same dataset and experimental conditions:

- **FedAvg** [11]: the most commonly adopted federated averaging algorithm without any robustness enhancements. FedAvg was experimentally demonstrated to yield good performance in malware detection scenarios even at moderate noise levels [10].
- **FoolsGold** [24]: a similarity-based aggregation scheme that down-weights clients with similar update patterns, aiming to mitigate Sybil attacks. This method is particularly relevant as it directly targets malicious clients but cannot handle noisy labels.
- **FedCorr** [43]: a federated learning method that tackles heterogeneous label noise through prediction space dimensionality, with noisy label correction. FedCorr has state-of-the-art performance in noisy-label scenarios, but does not explicitly address scenarios with label flipping attacks (i.e., intentional noise).

By combining these baselines, the evaluation covers methods that address either noisy labels or malicious clients, allowing for a comprehensive assessment of Fed-RoSeC's joint handling of both challenges.

Noisy labels were synthetically introduced in honest clients by flipping labels with a certain probability ϵ, which varied across experiments in the range $[0.0, 0.9]$. Different concentrations of malicious and noisy clients were also simulated, with the fraction of honest clients with clean data varying from 100% down to 20%. These parameters were varied systematically through factorial experimental design to assess the robustness of Fed-RoSeC and the baselines across a wide spectrum of challenging scenarios. The overall noise level in the system is thus a combination of the fraction of malicious clients and the label noise rate in noisy clients. This noise, denoted as η, is computed as:

$$\eta = \frac{\sum_{i \in malicious} |\mathcal{D}_i| + \epsilon \sum_{i \in noisy} |\mathcal{D}_i|}{\sum_{i \in clients} |\mathcal{D}_i|} \tag{7}$$

where $|\mathcal{D}_i|$ is the size of the dataset held by client i. For the sake of clarity, results are reported as a function of the overall noise level η with binning to smooth out fluctuations due to different combinations of malicious and noisy clients that yield the same η.

The local model used in experiments was implemented as a 4-layer feed-forward network with batch normalization and dropout to limit overfitting. The model was trained using SGD with a 0.01 learning rate, and batch size of 32. Training hyperparameters were tuned on a held-out validation split. Each federated round involved 5 local training epochs on 20 clients selected without replacement from a total of 100. Inputs were preprocessed to obtain fixed-size feature vectors from static analysis features; details were kept consistent across baselines to ensure a fair comparison.

4.1. Fed-RoSeC performance

Overall performance Figure 3 shows system performance as the overall noise increases, comparing Fed-RoSeC against the selected baselines. Our method degrades gracefully and remains significantly more accurate than baselines at high noise levels. The only comparable performance at $\eta = 0.8$ is achieved by FoolsGold, which, however, cannot handle low to moderate noise levels, where it is consistently the worst performer. FoolsGold's down-weighting mechanism, designed to counter Sybil attacks, is ineffective at low noise levels where noisy clients dominate performance degradation and

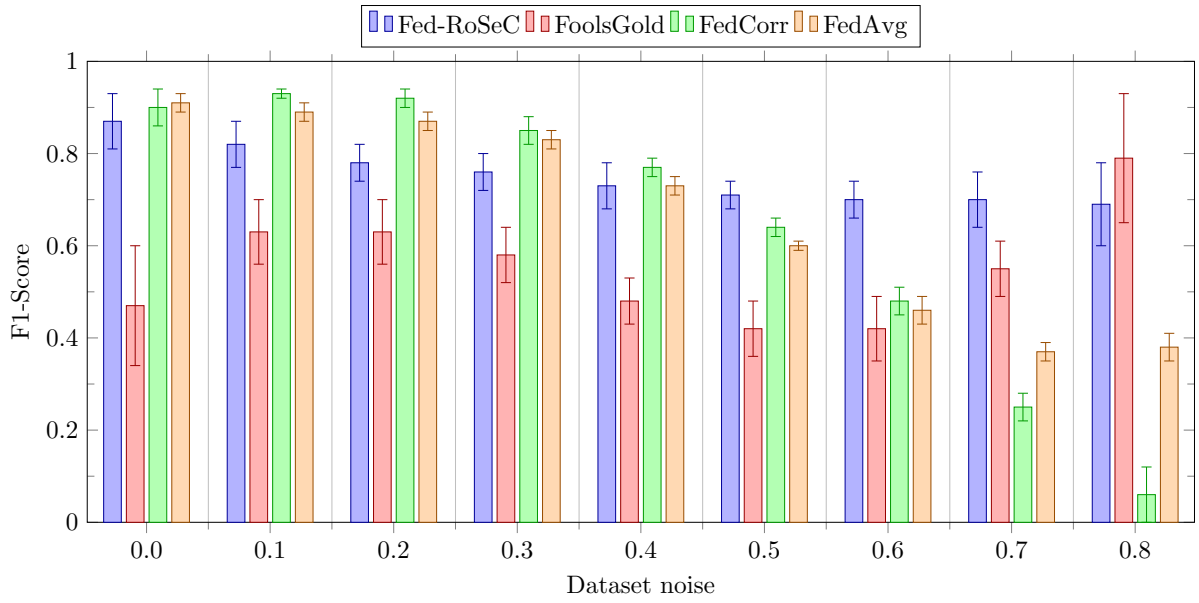


Figure 3: Comparison with representative baselines (FedAvg, FedCorr, FoolsGold).

create a less crisp separation in the update space between normal and malicious clients. FedCorr performs better than for $\eta < 0.3$, showing less degradation than FedAvg, but its performance collapses at higher noise levels, where label correction flips correct labels to incorrect ones due to the prevalence of noisy samples. By contrast, Fed-RoSeC shows performance not significantly different from those of FedCorr at low noise levels ($p = 0.07$ in a non-parametric Wilcoxon paired signed-rank test) and significantly outperforms it at high noise levels ($p < 1.3 \cdot 10^{-10}$), demonstrating the effectiveness of the combined clustering and conservative relabeling strategy.

Label correction efficacy In order to provide a more detailed understanding of Fed-RoSeC's operation, Figure 4 illustrates the evolution of model performance and overall label noise throughout the correction iterations at different representative noise levels. The same client configurations are also tested with FedCorr for comparison of the two label correction strategies. At low noise levels ($\eta = 0.2$), both methods effectively reduce label noise and achieve comparable performance, albeit Fed-RoSeC attains slightly lower noise levels. Instead, at higher noise levels ($\eta = 0.5$), Fed-RoSeC quickly reduces label noise below 10% as it did with lower noise, while FedCorr actually increases the overall noise by following the majority of incorrect labels. This result highlights the advantage of Fed-RoSeC's conservative relabeling strategy and malicious client isolation, which avoids overcorrecting samples when the global model is uncertain. Finally, at very high noise levels ($\eta = 0.8$), FedCorr's performance further degrades. In this scenario, Fed-RoSeC still manages to reduce label noise below 10%, halving the noise at the first step of correction (high-confidence relabeling) and further lowering it after the initial correction process converges, demonstrating its robustness even in extreme conditions where clean labels are a minority.

Clustering effectiveness In order to illustrate the effectiveness of the clustering stage in isolating suspicious clients, Figure 5 presents representative client clusters obtained via K-Modes on quantized distances, sorted by their cluster score Ω. The three chosen examples correspond to different overall noise levels ($\eta = 0.2, 0.55, 0.76$). In all cases, the clusters exhibit clear separation between honest, noisy, and malicious clients, with noisy clients being clustered either as malicious (when noise is high) or in separate intermediate clusters (when noise is lower). The scoring mechanism effectively ranks clusters so that honest clients are grouped together, facilitating their identification and isolation of suspicious participants for subsequent label correction. Specifically, at medium-low values of η (Figure 5

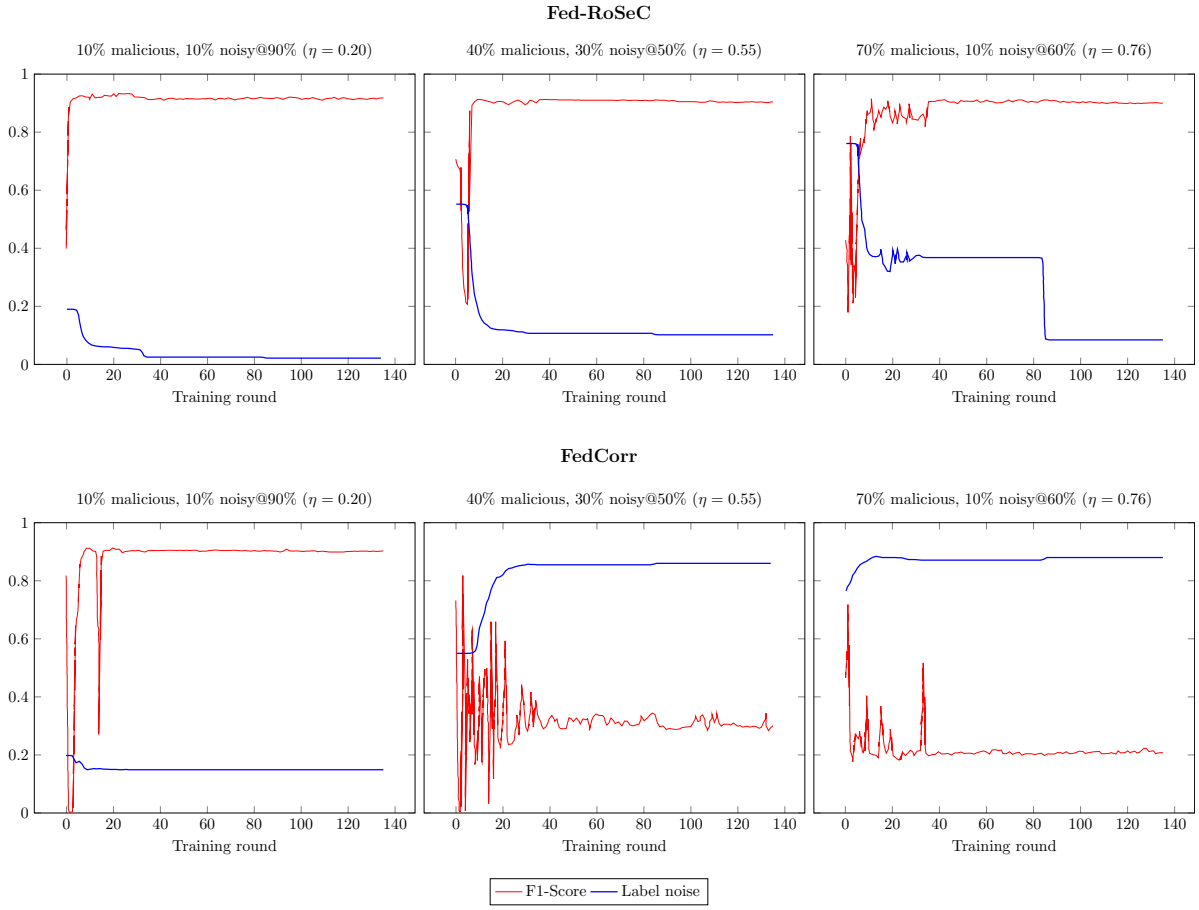


Figure 4: Model performance and overall label noise at different representative noise levels throughout the correction iterations. Fed-RoSeC effectively reduces label noise below 10% in all scenarios. FedCorr's correction, when clean labels are the minority, worsens label quality by miscorrecting honest samples.

left, middle), honest clients have higher scores due to their diverse updates, while at high η (Figure 5 right), honest clients have lower scores as malicious clients dominate the update space.

While not reported for the sake of space, analogous results were observed when noisy clients have low label noise rates (≈10-20%): in those scenarios, noisy clients tend to cluster with honest clients, as their updates mostly contain correct label information.

4.2. Ablation testing

A limitation of Fed-RoSeC is the reliance on heuristics on the client behavior in the distance space to accurately identify suspicious clients. While empirical evidence supports the assumptions made, it is important to assess the sensitivity of the method to these design choices. Specifically, three key design choices were evaluated through ablation testing:

- What is the maximum possible performance of Fed-RoSeC (i.e., an **oracle** version where the server knows exactly which clients are the honest ones with correct labels)? This version skips the clustering in rows 6-14 of Algorithm 1 and directly uses only honest clients for the subsequent correction and training phases.
- What happens if, instead of picking the most confident samples for relabeling, samples are selected through **random** sampling among those flagged as high-loss by the GMM? Given that the GMM already separates high-loss samples, this ablation tests whether the confidence-based selection is necessary or if it introduces unnecessary complexity.
- In $\approx 71\%$ of the experiments, the excess kurtosis κ was such that honest clients had higher

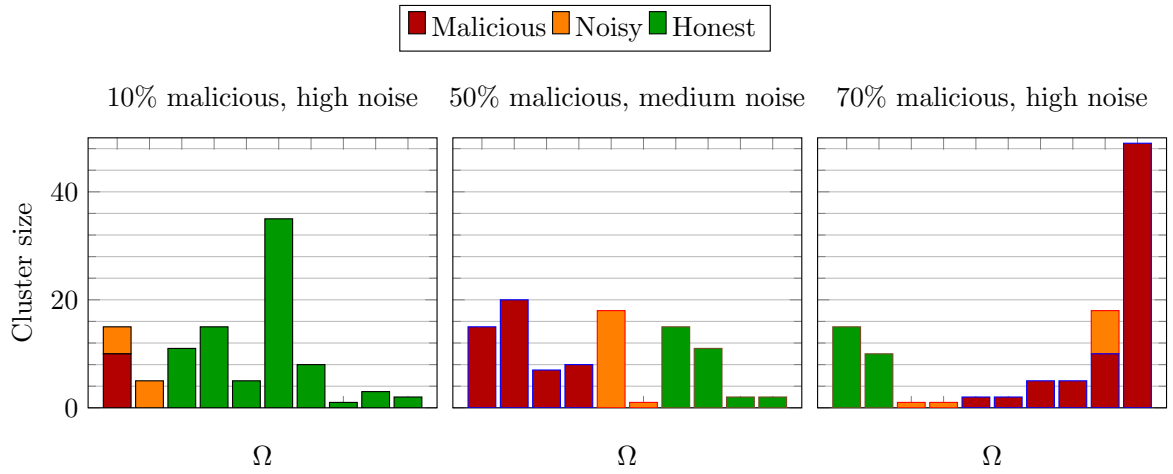


Figure 5: Representative client clusters obtained via K-Modes on quantized distances, sorted by cluster score Ω.

cluster scores than malicious ones. What happens if this selection is **static**, i.e., honest clients are always assumed to have higher scores regardless of κ?

The results of these ablation tests are summarized in Figure 6.

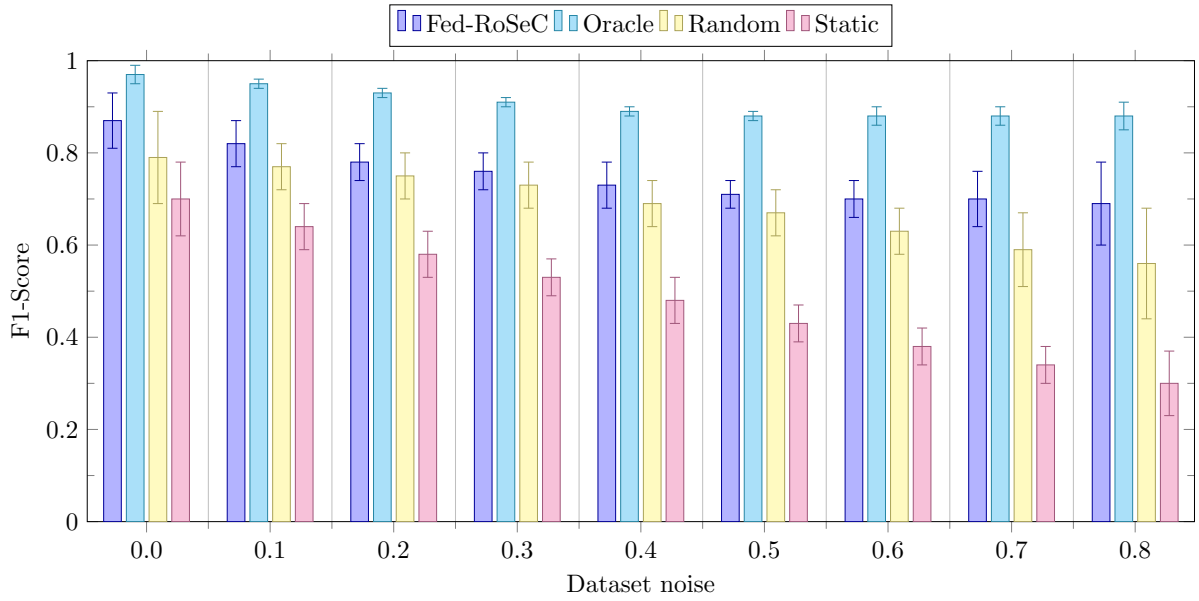


Figure 6: Comparison of Fed-RoSeC against its ablated versions (oracle, random sampling, static selection) at different noise levels. The oracle version sets an upper bound on performance, while the other two ablations show significant performance drops, especially at high noise levels, validating the design choices of Fed-RoSeC.

The oracle version of Fed-RoSeC unsurprisingly achieves the best performance, as it avoids any misclassification of clients and uses only clean data for correction and training. The slight downward trend as noise increases is due to the reduced amount of clean data available for training, which limits the global model's capacity to correct noisy labels. The random sampling ablation shows a noticeable performance drop and variability increase compared to the full Fed-RoSeC, especially at high noise levels (significant with $p = 0.014$). This result highlights the importance of confidence-based selection in ensuring that only the most likely noisy samples are relabeled, thereby minimizing the risk of incorrect corrections. Finally, the static selection ablation performs worse of all three, particularly at high noise levels. This outcome indicates that the dynamic selection based on excess kurtosis is

crucial for accurately identifying honest clients, as static assumptions can lead to misclassification and subsequent degradation in label correction and model performance, especially at high noise levels. Thus, while having some margin for improvement compared to the oracle, Fed-RoSeC's design choices are validated as effective and necessary for robust performance, and the assumptions made are generally sound.

5. Conclusions

This work presented Fed-RoSeC, a novel federated learning algorithm designed to enhance robustness against both malicious clients and noisy labels in the context of collaborative malware detection through crowdsourcing. Fed-RoSeC is designed to overcome the challenges posed by non-expert annotators and adversarial participants by employing a distance-aware clustering mechanism to isolate suspicious clients, followed by a conservative, model-guided label correction strategy. Experimental validation using a large-scale Android malware dataset confirmed its efficacy, demonstrating competitive detection performance while dramatically increasing robustness to incorrectly labeled samples in local datasets without sacrificing privacy. Empirically, Fed-RoSeC has shown versatility and effectiveness across a wide range of settings with varying levels of label noise and adversarial presence, making it a promising solution for real-world federated learning deployments in malware detection. Future research directions include exploring adaptive mechanisms for dynamic noise estimation and extending the Fed-RoSeC framework to settings with more complex adversarial behaviors and partially unlabeled data.

Acknowledgments

Part of the experimental evaluation performed for this work was carried out during the Master's thesis of Mattia Sidoti at University of Palermo, Italy. The authors would like to thank him for his dedication and assistance in collecting experimental data and implementing the initial version of the framework.

This work was partially funded by the European Union Next-Generation EU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3) – Progetto "Future Artificial Intelligence - FAIR" PE00000013, CUP J73C24000060007

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. M. Ansari, M. Nazir, K. Mustafa, Smart homes app vulnerabilities, threats, and solutions: a systematic literature review, Journal of Network and Systems Management 32 (2024) 29.

[2] L. Tang, X. Chen, S. Wen, L. Li, M. Grobler, Y. Xiang, Demystifying the evolution of android malware variants, IEEE Transactions on Dependable and Secure Computing 21 (2023) 3324–3341.

[3] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, K. Rieck, Dos and don'ts of machine learning in computer security, in: 31st USENIX Security Symposium (USENIX Security 22), 2022, pp. 3971–3988.

[4] F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder, L. Cavallaro, TESSERACT: Eliminating experimental bias in malware classification across space and time, in: 28th USENIX security symposium (USENIX Security 19), 2019, pp. 729–746.

[5] V. Agate, A. De Paola, P. Ferraro, G. Lo Re, MIDES: A multi-layer intrusion detection system using ensemble machine learning, International Journal of Intelligent Networks (2025).

[6] A. Augello, A. De Paola, G. Lo Re, Hybrid multilevel detection of mobile devices malware under concept drift, Journal of Network and Systems Management 33 (2025) 36. doi:10.1007/s10922-025-09906-3.

[7] V. Agate, A. De Paola, S. Drago, P. Ferraro, G. Lo Re, Enhancing iot network security with concept drift-aware unsupervised threat detection, in: 2024 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2024.

[8] A. De Paola, S. Drago, P. Ferraro, G. Lo Re, Detecting zero-day attacks under concept drift: An online unsupervised threat detection system, in: CEUR Workshop Proceedings, volume 3731, 2024.

[9] A. Augello, A. De Paola, M. Jestin, G. Lo Re, A stackelberg approach to federated learning for malware detection, in: CEUR Workshop Proceedings, volume 3962, ceur-ws. org/Vol-3962/paper36. pdf, 2025.

[10] A. Augello, A. De Paola, G. Lo Re, M2FD: Mobile malware federated detection under concept drift, Computers & Security (2025) 104361. doi:10.1016/j.cose.2025.104361.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, in: A. Singh, J. Zhu (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, volume 54 of *Proceedings of Machine Learning Research*, 2017, pp. 1273–1282.

[12] F. Concone, C. Ferdico, G. Lo Re, M. Morana, A federated learning approach for distributed human activity recognition, in: Proceedings - 2022 IEEE International Conference on Smart Computing, SMARTCOMP 2022, 2022, p. 269 – 274. doi:10.1109/SMARTCOMP55677.2022.00066.

[13] S. Seneviratne, A. Seneviratne, P. Mohapatra, A. Mahanti, Predicting user traits from a snapshot of apps installed on a smartphone, ACM SIGMOBILE Mobile Computing and Communications Review 18 (2014) 1–8.

[14] S. Jamalpur, Y. S. Navya, P. Raja, G. Tagore, G. R. K. Rao, Dynamic malware analysis using cuckoo sandbox, in: 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1056–1060. doi:10.1109/ICICCT.2018.8473346.

[15] D. Keragala, Detecting malware and sandbox evasion techniques, SANS Institute InfoSec Reading Room 16 (2016).

[16] B. Anderson, D. Quist, J. Neil, C. Storlie, T. Lane, Graph-based malware detection using dynamic analysis, Journal in computer Virology 7 (2011) 247–258.

[17] A. De Paola, S. Gaglio, G. L. Re, M. Morana, A hybrid system for malware detection on big data, in: INFOCOM 2018 - IEEE Conference on Computer Communications Workshops, 2018, p. 45 – 50. doi:10.1109/INFCOMW.2018.8406963.

[18] Y. Chen, Z. Ding, D. Wagner, Continuous learning for android malware detection, in: 32nd USENIX Security Symposium (USENIX Security 23), 2023, pp. 1127–1144.

[19] F. Camarda, A. De Paola, S. Drago, P. Ferraro, G. Lo Re, Managing concept drift in online intrusion detection systems with active learning, in: CEUR Workshop Proceedings, volume 3962, 2025.

[20] N. Wu, L. Yu, X. Jiang, K.-T. Cheng, Z. Yan, Fednoro: Towards noise-robust federated learning by addressing class imbalance and label noise heterogeneity, arXiv preprint arXiv:2305.05230 (2023).

[21] T. Tuor, S. Wang, B. J. Ko, C. Liu, K. K. Leung, Overcoming noisy and irrelevant data in federated learning, in: 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 5020–5027. doi:10.1109/ICPR48806.2021.9412599.

[22] Y. Chen, X. Yang, X. Qin, H. Yu, P. Chan, Z. Shen, Dealing with label quality disparity in federated learning, Federated Learning: Privacy and Incentive (2020) 108–121.

[23] Y. Chen, L. Su, J. Xu, Distributed statistical machine learning in adversarial settings: Byzantine gradient descent, Proceedings of the ACM on Measurement and Analysis of Computing Systems 1 (2017) 1–25.

[24] C. Fung, C. J. Yoon, I. Beschastnikh, Mitigating sybils in federated learning poisoning, arXiv preprint arXiv:1808.04866 (2018).

[25] X. Fang, M. Ye, X. Yang, Robust heterogeneous federated learning under data corruption, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 5020–5030.

[26] P. Ranjan, A. Gupta, S. K. Das, Securing federated learning from distributed backdoor attacks via maximal clique and dynamic reputation system, in: 2025 IEEE International Conference on Smart Computing (SMARTCOMP), IEEE, 2025, pp. 130–137.
[27] S. Yang, H. Park, J. Byun, C. Kim, Robust federated learning with noisy labels, IEEE Intelligent Systems 37 (2022) 35–43. doi:10.1109/MIS.2022.3151466.
[28] Y. Jiang, W. Zhang, Y. Chen, Data quality detection mechanism against label flipping attacks in federated learning, IEEE Transactions on Information Forensics and Security 18 (2023) 1625–1637. doi:10.1109/TIFS.2023.3249568.
[29] S. Li, Y. Cheng, W. Wang, Y. Liu, T. Chen, Learning to detect malicious clients for robust federated learning, arXiv preprint arXiv:2002.00211 (2020).
[30] C. Ma, J. Li, M. Ding, K. Wei, W. Chen, H. V. Poor, Federated learning with unreliable clients: Performance analysis and mechanism design, IEEE Internet of Things Journal 8 (2021) 17308–17319. doi:10.1109/JIOT.2021.3079472.
[31] Z. Wang, T. Zhou, G. Long, B. Han, J. Jiang, Fednoil: A simple two-level sampling method for federated learning with noisy labels, arXiv preprint arXiv:2205.10110 (2022).
[32] A. Gupta, T. Luo, M. V. Ngo, S. K. Das, Long-short history of gradients is all you need: Detecting malicious and unreliable clients in federated learning, in: European Symposium on Research in Computer Security, Springer, 2022, pp. 445–465.
[33] S. Awan, B. Luo, F. Li, Contra: Defending against poisoning attacks in federated learning, in: Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26, Springer, 2021, pp. 455–475.
[34] C. Sandeepa, B. Siniarski, S. Wang, M. Liyanage, Sherpa: Explainable robust algorithms for privacy-preserved federated learning in future networks to defend against data poisoning attacks, in: 2024 IEEE Symposium on Security and Privacy (SP), IEEE Computer Society, 2024, pp. 204–204.
[35] E. Hallaji, R. Razavi-Far, M. Saif, E. Herrera-Viedma, Label noise analysis meets adversarial training: A defense against label poisoning in federated learning, Knowledge-Based Systems 266 (2023) 110384.
[36] S. Park, S. Han, F. Wu, S. Kim, B. Zhu, X. Xie, M. Cha, Feddefender: Client-side attack-tolerant federated learning, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1850–1861. doi:10.1145/3580305.3599346.
[37] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proceedings of Machine learning and systems 2 (2020) 429–450.
[38] J. Xu, Z. Chen, T. Q. Quek, K. F. Ernest Chong, Fedcorr: Multi-stage federated learning for label noise correction, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 10174–10183. doi:10.1109/CVPR52688.2022.00994.
[39] A. Augello, G. Falzone, G. Lo Re, DCFL: Dynamic clustered federated learning under differential privacy settings, in: 2023 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), 2023, pp. 614–619. doi:10.1109/PerComWorkshops56833.2023.10150285.
[40] Z. Huang, Extensions to the k-means algorithm for clustering large data sets with categorical values, Data mining and knowledge discovery 2 (1998) 283–304.
[41] E. Arazo, D. Ortego, P. Albert, N. O'Connor, K. Mcguinness, Unsupervised label noise modeling and loss correction, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 312–321. URL: https://proceedings.mlr.press/v97/arazo19a.html.
[42] A. Guerra-Manzanares, H. Bahsi, S. Nõmm, Kronodroid: Time-based hybrid-featured dataset for effective android malware detection and characterization, Computers & Security 110 (2021) 102399. doi:https://doi.org/10.1016/j.cose.2021.102399.
[43] J. Xu, Z. Chen, T. Q. Quek, K. F. E. Chong, Fedcorr: Multi-stage federated learning for label noise correction, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 10184–10193.