This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

FedDMC: Efficient and Robust Federated Learning via Detecting Malicious Clients

Xutong Mu, Ke Cheng, Yulong Shen, *Member, IEEE,* Xiaoxiao Li, *Member, IEEE,* Zhao Chang, Tao Zhang, Xindi Ma, *Member, IEEE,*

***Abstract*—Federated learning (FL) has gained popularity in the field of machine learning, which allows multiple participants to collaboratively learn a highly-accurate global model without exposing their sensitive data. However, FL is susceptible to poisoning attacks, in which malicious clients manipulate local model parameters to corrupt the global model. Existing FL frameworks based on detecting malicious clients suffer from unreasonable assumptions (e.g., clean validation datasets) or fail to balance robustness and efficiency. To address these deficiencies, we propose FedDMC, which implements robust federated learning by efficiently and precisely detecting malicious clients. Specifically, FedDMC first applies principal component analysis to reduce the dimensionality of the model parameters, which retains the primary parameter feature and reduces the computational overhead for subsequent clustering. Then, a binary tree-based clustering method with noise is designed to eliminate the effect of noisy points in the clustering process, facilitating accurate and efficient malicious client detection. Finally, we design a self-ensemble detection correction module that utilizes historical results via exponential moving averages to improve the robustness of malicious client detection. Extensive experiments conducted on three benchmark datasets demonstrate that FedDMC outperforms state-of-the-art methods in terms of detection precision, global model accuracy, and computational complexity.**

***Index Terms*—Federated learning, Poisoning attack, Malicious clients, Clustering**

I. Introduction

Federated learning (FL) [1] is an emerging collaborative machine learning paradigm that is widely used in areas such as medical imaging diagnosis [2], [3], risk identification of banking transactions [4], and autopilot training [5]. FL's key advantage is enabling the central server to build highly-accurate global models while keeping the client's private data confidential. Specifically, the FL process involves three iterative steps: 1) The clients train their local data to generate local models individually and transmit them to the central server. 2) The central server aggregates the local models based on predefined aggregation rules to update the global model. 3) The updated global model is transmitted back to the clients for the next learning round.

However, FL is vulnerable to model poisoning attacks due to its distributed nature [6]–[13], in which malicious clients send tamper model updates to the server resulting in global model corruption. Adversaries can tamper with the local training phase of malicious clients in two ways: 1) by modifying local data, known as *data poisoning attacks*, and 2) by manipulating local model updates, referred to as *local model poisoning attacks*. In data poisoning attacks, some attackers randomly flip normal labels to reduce global accuracy or purposefully flip to target labels to achieve targeted attacks [6]. Other attackers add a triggered backdoor to the image and specifically tag the data [7]–[9], known as the backdoor attack. In local model poisoning attacks [10]–[13], attackers typically manipulate local model parameters by exploiting information such as aggregation rules and the distribution of local models. This manipulation aims to evade detection by defensive mechanisms while maximizing the detrimental impact on the global model.

Various efforts have been made to defend against malicious client attacks, primarily from three perspectives. The first category [7], [14] is based on differentiated privacy techniques [15] that dilute the impact of poisoning attacks by adding noise. The limitation of such approaches is that they would reduce the accuracy of the primary task. The second category is based on aggregation rule approaches, which achieve robust federated learning by changing the aggregation rules [16], [17]. However, malicious clients can circumvent these rules by designing covert models in which the poisoned parameters are selectively incorporated into the global model. The third type is to distinguish between benign and malicious clients by using clean validation datasets [18], [19] or clustering model parameters [13], [20]–[24]. Compared to the first two approaches, the third category offers the benefit of more accurately detecting malicious behavior, thereby minimizing the impact of malicious models on the global model. However, all of approaches require strong prior knowledge (e.g., clean validation dataset, preset number of malicious clients) or fail to balance robustness and efficiency. We list the detailed drawbacks of the third category of approaches in TABLE I.

In this paper, we present FedDMC, an FL framework that aims to defend against poisoning attacks by efficiently and precisely detecting malicious clients. FedDMC addresses the limitations of existing detection methods, such as the requirement of clean validation datasets or presets for the number of malicious clients. Moreover, FedDMC takes care of both

Xutong Mu, Ke Cheng, Yulong Shen, Zhao Chang, Tao Zhang, and Xindi Ma are with the School of Computer Science and Technology, Xidian University, Xi'an, 710071, China. (e-mail: xtmu@stu.xidian.edu.cn, chengke@xidian.edu.cn, ylshen@mail.xidian.edu.cn, changzhao@xidian.edu.cn, taozhang@xidian.edu.cn, xdma1989@gmail.com)

Xiaoxiao Li is with Electrical and Computer Engineering, the University of British Columbia, Vancouver, BC V6T 1Z4 Canada. (e-mail: xiaoxiao.li@ece.ubc.ca)

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

TABLE I
EXISTING APPROACHES BASED ON MALICIOUS CLIENT DETECTION IN FEDERATED LEARNING

Solution	Detection approach	No clean validation dataset	No preset the number of malicious clients	Byzantine Robustness	Efficiency
FLTrust [18]	Cosine	○	●	●	◐
VAE [19]	Autoencoder	○	●	◐	◐
Multi-Krum [21]	Krum	●	○	◐	●
DnC [13]	SVD	●	○	◐	◐
Auror [20]	K-means	●	●	◐	◐
FoolsGold [22]	Cosine	●	●	◐	●
SecFedNIDS [23]	SOS	●	●	◐	◐
FedPRC [26]	K-means++	●	●	◐	◐
FLdetector [24]	Model consistency	●	●	●	○
FedDMC (our work)	PCA+Clustering	●	●	●	●

* The symbol ● indicates YES, ○ indicates NO, ◐ indicates Partially YES. In the column of Byzantine Robustness, the half circle ◐ indicates that these methods still exhibit limitations when facing certain known attacks. A full circle ● indicates that the method is better defended against a wide range of known attacks. In the Efficiency column, symbols are divided into three levels: the empty circle ○ represents the slowest efficiency, the half circle ◐ represents relatively slow efficiency, and the full circle ● represents high efficiency.

the accuracy and efficiency of malicious client detection. Our fundamental intuition is that, due to the high dimensionality of the model parameters, a malicious client can introduce a significant change to a single parameter without significantly affecting the ℓ_p norm (Euclidean distance), rendering the model ineffective. Based on this intuition, FedDMC detects malicious clients by performing the dimension reduction of the model parameters before applying clustering methods, thereby mitigating the influence of irrelevant dimensions and highlighting the differentiation of malicious model parameters.

Specifically, FedDMC is achieved by three modules: 1) *dimensionality reduction* (DR). Classical Principal Component Analysis (PCA) [25] is utilized to reduce the dimensionality of model parameters. This method preserves the main features of the parameters while lowering the computational overhead for subsequent clustering. 2) *binary tree-based clustering with noise* (BTBCN). A simple and effective noise-removable clustering method is proposed that can eliminate the effect of noisy points to facilitate accurate and efficient malicious client detection. 3) *self-ensemble detection correction* (SEDC). A trust score is formulated, integrating the detection outcomes from historical rounds. This strategy aims to address the unreliability of single-round detection, thereby enhancing the overall robustness of the detection process. We evaluate FedDMC on three benchmark datasets, demonstrating its superiority over existing state-of-the-art methods in terms of detection accuracy, global model accuracy, and computational complexity. Moreover, we conduct separate analyzes for each of the three components, providing insights into their contributions to the overall performance of FedDMC.

In summary, we make the following contributions.

- We propose FedDMC, an FL framework to defend against poisoning attacks by efficiently and precisely detecting malicious clients. FedDMC achieves detection with high accuracy and low computational overhead. After removing a majority of the malicious clients, FedDMC is able to learn an accurate global model.
- To construct FedDMC, we design three simple and effective modules: the DR module to remove irrelevant parameters, the BTBCN module to detect malicious clients, and the SEDC module to enhance the robustness of the detection results.
- In the experimental evaluation, we comprehensively assess the proposed framework on three benchmark datasets under four widely-adopted attack scenarios. Additionally, we validate the effectiveness of each of the three components, providing insights into their impact on the overall performance.

Organization. The rest of this paper is organized as follows. In Section II, we summarise the existing FL defense methods. In Section III, we describe the federated learning system model and the defense objectives. Section IV provides the motivation and details of the proposed framework. Section V presents the experimental setup, and Section VI provides a detailed analysis of the experimental results. Finally, we conclude the paper and discuss future work in Section VII.

II. RELATED WORKS

A. Robust Federated Learning

Existing works defend against poisoning attacks primarily from three perspectives, respectively, differential privacy-based approaches [7], [14], [27], aggregation rule-based approaches [16], [17], and malicious client detection-based approaches [13], [18]–[24]. Our research belongs to the third category, which is orthogonal to the first two and can be employed synergistically with them in the aggregation phase.

1) Differential Privacy-based Approaches: Differential Privacy (DP) [15] is a technique for protecting data privacy by adding noise while keeping the data as effectively as possible. Inspired by differential privacy (DP) techniques, [7] and [27] dilute the effect of the poisoning model on the global model by clipping the individual model parameters to a maximum threshold and adding random noise to the model parameters. FLAME [14] proposes filtering out poisoned models with high attack impact before using differential privacy. The limitation of these methods is that they reduce the accuracy of the global model.

2) Aggregation Rule-based Approaches: Aggregation rule-based approaches guarantee robust aggregation by changing aggregation rules. Krum [21] selects the local model most similar to other models from a set of local models to serve

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

as the global model. Trimmed-Mean [16] aggregates each parameter of the model separately by removing the largest and smallest several values to avoid the effect of outliers. Median [16] selects the median value of all dimensions as its aggregate global model. Bulyan [17] is essentially a combination of Krum [21] and a variant of Trimmed-Mean [16]. Bulyan first iteratively applies Krum to select some of the locals model and then uses a variant of the Trimmed-Mean to aggregate these local models. RFA [28] propose a geometric median-based robust aggregation method, aimed at enhancing resistance to corrupted updates. FedRoC [29] employs bootstrap median-of-means for resilient client clustering against non-IID data and model poisoning, utilizing a dual-layer optimization and stochastic expectation maximization for enhanced effectiveness. Ditto [30] employs a personalized federated multi-task learning framework that balances global and local models through regularization, allowing individualized model learning that mitigates the impact of malicious devices. Roughly speaking, these methods apply statistical methods to estimate the aggregated model updates. Nevertheless, adversaries may bypass such defenses by introducing stealth models, selectively incorporating poisoned parameters into the global model.

3) Malicious Clients Detection-based Approaches: Malicious clients detection-based approaches aim to distinguish between benign and malicious clients by using clean validation datasets or clustering model parameters.

Using clean validation datasets: FLTrust [18] maintains a clean validation dataset on the server to generate an update direction. Specifically, higher trust scores are given when local model updates closely match server model updates. The server then discriminates between malicious and benign clients using these scores. Li et al. [19] utilize a variational autoencoder (VAE) to capture model update statistics, assuming the server has access to a clean validation dataset from the overall training data distribution for model training. In brief, these approaches require a clean validation dataset with a distribution closely matching the local clients' training data. However, this requirement is difficult to meet in real-world applications, limiting the practical usability of these approaches.

Clustering model parameters: Multi-Krum [21], an extension of Krum, selects multiple local models that exhibit the highest similarity to others. It then computes their average to form the global model. DnC [13] employs random sampling for dimensionality reduction. Subsequently, it uses a spectral approach, specifically Singular Value Decomposition (SVD), to identify and remove malicious clients. However, both Multi-Krum and DnC require a preset number of malicious clients, leading to limitations in practical applications. Auror [20] utilizes K-means [31] clustering to detect malicious clients. FoolsGold [22] uses client-contributed similarity to detect malicious clients based on the diversity of client updates. SecFedNIDS [23] selects important model parameters based on gradient-based changes. It distinguishes malicious clients using the Stochastic Outlier Selection (SOS) algorithm. SOS is a method that evaluates the probability of outliers based on similarity scores between data points, especially useful for quickly identifying outliers in large datasets. FedPRC [26] employs K-means++ [32] for initial clustering and density-based anomaly detection to exclude malicious clients, enhancing robustness against model poisoning in federated learning. These methods, while innovative, still exhibit limitations when tackling specific attacks and do not fully guarantee the reliability of the defense. FLDetector [24] identifies malicious clients by leveraging model consistency. However, it has a drawback of high computational overhead.

TABLE I summarises the strengths and weaknesses of approaches based on malicious client detection in federated learning. As can be seen from the table, our proposed method overcomes the limitations of existing methods and achieves robust and effective federated learning without requiring prior knowledge.

B. Clustered Federated Learning

Existing some Clustered Federated Learning (CFL) methods utilize clustering techniques to address the challenge of non-independent and identically distributed (non-IID) data. Sattler et al. [33] designed a novel federated multi-task learning framework, which leverages geometric properties to cluster client populations based on their data distributions, addressing the issues brought by non-IID data. Ghosh et al. [34] proposed the Iterative Federated Clustering Algorithm (IFCA), which alternates between estimating user clustering identities and optimizing model parameters via gradient descent, catering to user distribution heterogeneity in federated learning. Long et al. [35] developed a multi-center aggregation mechanism that aggregates clients through model parameters and learns multiple global models as cluster centers, simultaneously solving the optimal user-center matching problem. Briggs et al. [36] introduced a hierarchical clustering into federated learning, separating clients into different groups based on the similarity of local updates, and conducting independent training on dedicated models. Lastly, Ma et al. [37] revisited CFL, formalizing it as a bi-level optimization framework and proposed a new algorithm called Weighted Clustered Federated Learning (WeCFL), proving its convergence in non-IID settings. Although both the aforementioned methods and FedDMC utilize clustering algorithms, there are significant differences in the challenges addressed and the underlying methodological framework.

III. System Model

A. Federated Learning

A typical FL system consists of n clients and a cloud server. Each client holds a local training dataset $\mathcal{D}_i, i = 1, \cdots, n$. The server is not allowed to access the private training data held by the clients, and the clients collaborate to learn a global model maintained on the server. Specifically, in the t-th iteration of FL, the server broadcasts the current aggregation model w^t to the clients. To minimize the local empirical loss $\mathcal{L}(\mathcal{D}_i, w^t)$, clients fine-tune individual local model parameters w_i^t based on the received global model w^t over their local training dataset $\mathcal{D}_i$ using stochastic gradient descent as

$$w_i^{t+1} = w^t - \eta \Delta \mathcal{L}(w^t, \mathcal{D}_i) \tag{1}$$

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

where η is the learning rate. Subsequently, the trained local model parameters w_i^{t+1} are returned to the server. The server aggregates the local model parameters received from the clients according to a proper aggregation rule $\mathcal{A}$, i.e.,

$$w^{t+1} = \mathcal{A}(w_1^{t+1}, w_2^{t+1}, \cdots, w_n^{t+1}) \tag{2}$$

The client-server interaction is repeated until the global model converges or reaches a predetermined number of communication rounds.

In this study, the FL system involves one honest server that is not compromised and two types of clients, benign and malicious. The benign clients honestly train and report the local model parameters w_i^{t+1}. In contrast, malicious clients tamper with the local training phase by modifying local data or manipulating local model parameters to corrupt the global model. Assume that the number of malicious clients is M, not more than half of the total number of clients.The attacker is agnostic about the server's aggregation rule $\mathcal{A}$, thus, cannot perform specific attacks based on the aggregation rule.

B. Defense Objectives

A generic FL framework that is effective against poisoning attacks needs to achieve the following three objectives:

1) **High detection accuracy.** A superior defense method should detect all malicious clients ($M < \lfloor \frac{n}{2} \rfloor$) as far as possible to ensure that the federated learning system is not compromised.
2) **Low computational overhead.** An effective defense method should be easy to deploy and fast to compute. It must avoid imposing an excessive computational load on the server while ensuring minimal disruption to the overall operational timeline.
3) **Broad applicability.** An effective defense mechanism should prevent attackers within a broad threat model. Essentially, it should not require prior knowledge of the method of attack, the proportion of clients being attacked, and various data settings, such as whether the data distribution follows an IID or non-IID distribution.

IV. Method

In this section, we first discuss the motivation for designing the FedDMC method, followed by a comprehensive overview and design description of the proposed framework. Finally, we analyze the computational complexity of the FedDMC approach.

A. Motivation

In the FL framework based on malicious client detection, clustering methods are widely regarded as potent detection mechanisms due to their capacity to automatically classify clients with similar characteristics. Nevertheless, traditional clustering methods encounter challenges when dealing with high-dimensional data, such as high computational complexity and susceptibility to noise contamination. Due to the inherently high-dimensional parameter space of deep learning models, malicious clients are able to significantly alter individual parameters but with a negligible impact on the Euclidean distance between that model and other models. This situation creates favorable conditions for adversarial attacks [10], [17]. Consequently, it is desirable to reduce the dimensionality of the data before adopting clustering methods.

Algorithm 1: The FedDMC framework

Input: Local datasets $\mathcal{D}_i$, number of communication rounds T, number of clients n, number of local epochs E, learning rate η, Ensemble parameter α

Output: The final model w^T

Server executes:
initialize w^0
for $t = 0, 1, \cdots, T-1$ **do**
  for $i = 1, 2, \cdots, n$ ***in parallel*** **do**
    send the global model w^t to C_i
    $w_i^{t+1} \leftarrow$ **ClientLocalTraining**$(i,\ w^t)$
  $\boldsymbol{W} \leftarrow (w_0^{t+1}, \cdots, w_n^{t+1})$
  $\widetilde{\boldsymbol{W}} \leftarrow \boldsymbol{P}_{\boldsymbol{W}}^k(\boldsymbol{W})$ `// PCA`
  $\boldsymbol{S}_{[t]} \leftarrow$ **BTBCN**$(\widetilde{\boldsymbol{W}})$ `// Clustering`
  $\mathbf{C}_B, \mathbf{C}_M \leftarrow$ **SEDC** $(\boldsymbol{S}_{[t]})$ `// Ensemble`
  $w^{t+1} \leftarrow \frac{1}{|\mathbf{C}_B|} \sum_j w_j^{t+1},\ j \in \mathbf{C}_B$
return w^T

ClientLocalTraining$(i,\ w^t)$:
$w_i^t \leftarrow w^t$
if C_i *is malicious* **then**
  Performing poisoning attacks. `// e.g., LF-attack, GS-attack, Scaling-attack.`
else
  for *epoch* $= 1, 2, \cdots, E$ **do**
    $w_i^{t+1} = w^t - \eta \Delta \mathcal{L}(w_i^t, \mathcal{D}_i)$
Attackers perform attacks by combining malicious clients. `// LIT-attack.`
return w_i^{t+1} to server

B. The Design of FedDMC

Based on the above motivations, we design a novel approach to detect malicious clients in federated learning systems, dubbed FedDMC. FedDMC consists of three modules: i) dimensionality reduction (DR), ii) binary tree-based clustering with noise (BTBCN), and iii) self-ensemble detection correction (SEDC). First, we use PCA to project model parameters into a lower-dimensional space, preserving the main features while reducing computational complexity for subsequent clustering. In the dimensionality reduction process, PCA helps to highlight differences between clients' models, allowing the clustering algorithm to more easily distinguish between benign and malicious clients. Subsequently, we design the BTBCN module to effectively detect malicious clients. Finally, we introduce the SEDC module that utilizes historical detection

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

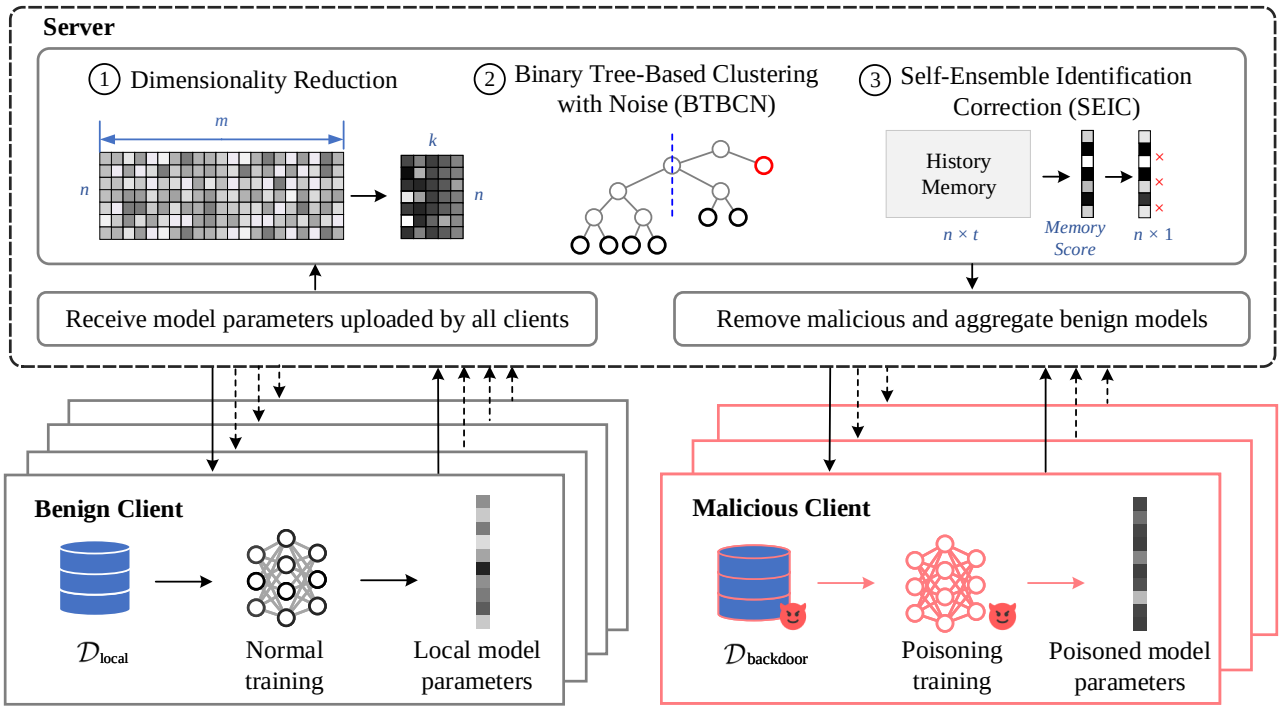
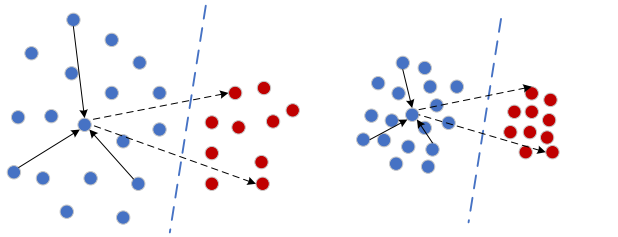


Fig. 1. Overview of FedDMC. The server receives the model parameters from the clients and performs malicious client detection with three modules, i) dimensionality reduction (DR), ii) binary tree-based clustering with noise (BTBCN), and iii) self-ensemble detection correction (SEDC). Afterward, the parameters of the malicious clients are removed, and the benign models are aggregated.



(a) Cluster distribution before PCA (b) Cluster distribution after PCA

Fig. 2. Schematic diagram of cluster distribution.

results through exponential moving averages to gradually correct current detection results. These modules and the workflow of FedDMC are shown in Fig. 1. The process of FedDMC is summarised in Algorithm 1. Details of these modules are described subsequently.

*1) **Dimensionality Reduction**:* Principal Component Analysis (PCA) was selected as the dimensionality reduction technique in this framework due to its effectiveness in facilitating cluster analysis. PCA is computationally efficient compared to methods, such as t-SNE [38], isomap [39], and UMAP [40], making it particularly appropriate for large-scale parameters in federated learning. As a linear technique, PCA adeptly handles data with linear relationships and offers robust interpretability compared to random projection [41] and Singular Value Decomposition (SVD) [42].

Crucially, PCA highlights the differences between benign and malicious clients. Considering the model parameters matrix $\boldsymbol{W} = \{w_1, w_2, \cdots, w_n\}$ for n clients in a specific round, where each w_i is a d-dimensional vector representing client i's model parameters. PCA is applied to this parameter matrix $\boldsymbol{W} \in \mathbb{R}^{d\times n}$, and the process can be expressed as $\widetilde{\boldsymbol{W}} = \boldsymbol{P}^k_{\boldsymbol{W}}(\boldsymbol{W})$. Here, $\boldsymbol{P}^k_{\boldsymbol{W}}$ represents the PCA projection operator, k is the reduced dimension, significantly smaller than d, and its value can be adjusted based on the specific client data distribution and computational overhead. $\widetilde{\boldsymbol{W}} \in \mathbb{R}^{k\times n}$ is the reduced-dimensional model parameters matrix obtained after PCA projection. The detailed procedure for applying PCA is provided in Algorithm 3 of Appendix A.

*2) **Binary Tree-Based Clustering with Noise (BTBCN)**:* In FL scenarios, the heterogeneity of local data distributions and the uncertainty of malicious client behavior can result in noisy points besides benign and malicious clusters, as illustrated in Fig. 4. The accuracy of clustering methods such as K-means [43] and hierarchical clustering [44] can be easily affected by these noisy points. Existing clustering approaches, such as DBSCAN [45], OPTICS [46], and HDBSCAN [47], can effectively exclude noisy points. However, distinguishing between malicious and benign clients is a binary classification task. These methods have limited performance when dealing with binary classification tasks. Robust K-means++ [32] can be applicable for binary classification that can eliminate noise, but it incurs a huge computational overhead. To address these challenges, we propose a binary tree-based clustering method with noise (BTBCN) suitable for classifying malicious and benign clients.

Specifically, we construct a binary tree based on the Euclidean distance of the model parameters after dimension reduction. Each leaf node represents a client, non-leaf nodes

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

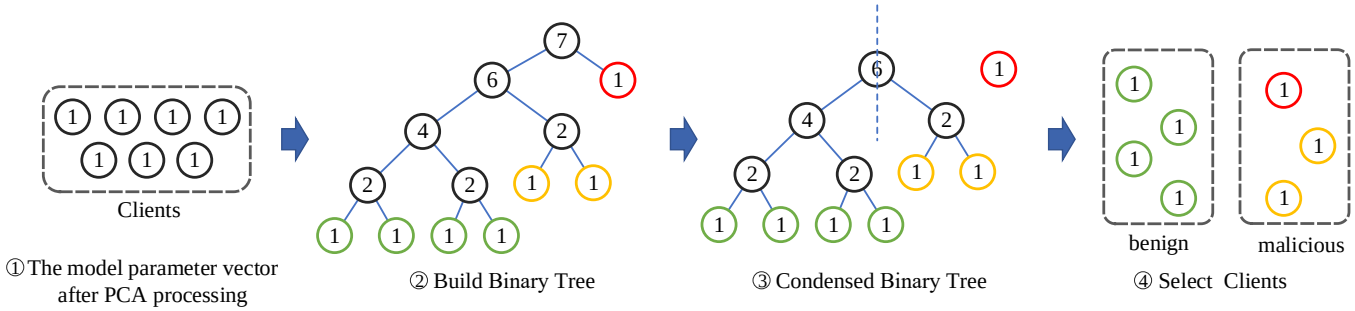


Fig. 3. Illustration of the workflow for **BTBCN**. The workflow consists of 4 steps: ① Obtain the model parameter vectors after PCA processing. ② Construct a hierarchical clustering tree based on the Euclidean distance between these model parameter vectors. ③ Set the value of $min\_cluster\_size$ and execute the Condensed Clustering Tree algorithm to obtain the clustering tree. Assume $min\_cluster\_size$ is 2. Splitting from top to bottom, the first split gives two clusters of 6 and 1. 1 is smaller than $min\_cluster\_size$, so it is considered as an outlier. ④ Finally, the detection results are obtained based on the clustering results.

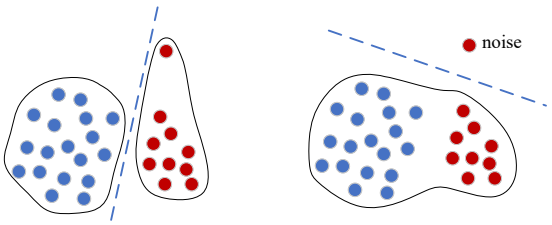


Fig. 4. Schematic diagram of cluster distribution.

represent subsets of clients, and the root node represents the set of all clients. The number in each node represents the number of leaf nodes in the subset. From top to bottom, each node splits into two clusters with the maximum distance between them, which are viewed as ideal benign and malicious client clusters, as shown in Fig. 4(a). However, noisy points can cause incorrect clustering, as shown in Fig. 4(b). Here, we propose a core step called a condensed the binary tree, which filters out noisy points before obtaining the clusters. We need to determine the minimum cluster size, denoted as $min\_cluster\_size$, which serves as a parameter for this module. If the number of leaf nodes in a split cluster is less than $min\_cluster\_size$, these leaf nodes are considered to be noisy points. These noisy points are pruned from this binary tree. Pruning stops until the number of nodes in both clusters of the binary tree split is more than $min\_cluster\_size$. After the clustering tree splits into two clusters, we regard the larger cluster as the benign client. The workflow of the clustering algorithm is shown in Fig. 3, and Algorithm 2 details the BTBCN.

*3) **Self-Ensemble Detection Correction (SEDC)**:* Although the BTBCN approach effectively distinguishes between malicious and benign clients, it remains vulnerable to the challenges posed by the uncertain data distribution of clients and the agnostic attacks of malicious clients. Detection of malicious clients may be influenced by random factors in the current round, leading to inaccurate detection results. As a result, relying on a single round of detection may be unreliable. To address these challenges, we propose a simple and effective method called Self-Ensemble Detection Correction (SEDC).

Algorithm 2: Binary Tree-Based Clustering with Noise (BTBCN)

Input: The model parameters after dimensionality reduction $\widetilde{\boldsymbol{W}}$
Output: Detection results, $\boldsymbol{S} = \{s_0, s_1, \cdots, s_n\}$, $s_i \in \{0, 1\}$.

```
// Build Binary Tree
Initialize tree 𝒯
Initialize each vector w̃_i as a cluster.
B ← EUCLIDEANDISTANCE(w̃_1, ··· , w̃_n)
while ∃ w̃_i ∉ 𝒯 do
    Merge the two closest clusters.
    Update the distance matrix B.
// Condensed Binary Tree
r ← RootNode(𝒯)
Lchild, Rchild ← NodeSplitting(r)
while Lchild or Rchild is outliers do
    if Lchild is outliers then
        Mark the Lchild as outlier
        r ← Rchild
    if Rchild is outliers then
        Mark the Rchild as outlier
        r ← Lchild
    if outliers ≥ ⌊N/2⌋ then
        break
    Lchild, Rchild ← NodeSplitting(r)
S ← ClientsEvaluation(Lchild, Rchild, outlier)
return S // detection results
```

This method integrates historical detection outcomes from multiple rounds to reduce the inaccuracy of single detection instances, leading to more reliable and robust detection results.

Specifically, we represent the detection results of round t with $\boldsymbol{S}_{[t]} = \{s_0^t, s_1^t, \cdots, s_n^t\}$, where $s_i \in 0, 1$ indicates the detection outcome for the i^{th} client, with 1 representing a benign client and 0 denoting a malicious one. Considering the varying credibility of detection results across rounds, SEDC summarizes historical detection outcomes and generates a trust

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

score $\hat{\boldsymbol{S}}_{[t]}$ for each client, initialized at 0.5. As the FL process iterates, we consider the most recent round's results to be relatively reliable and thus adjust the ratio of the current round's score to the accumulated historical scores using the hyperparameter $\alpha, 0 \leq \alpha < 1$. At round t, we have ensemble detection results

$$\hat{\boldsymbol{S}}_{[t]} = \begin{cases} 0.5 & \text{if} \quad t = 0 \\ \alpha \hat{\boldsymbol{S}}_{[t-1]} + (1-\alpha)\boldsymbol{S}_{[t]} & \text{if} \quad t > 0 \end{cases} \tag{3}$$

Based on Equation (3), we can derive the ensemble detection for round t as $\hat{\boldsymbol{S}}_{[t]} = 0.5 \cdot \alpha^t + \sum_{i=1}^{t}(1-\alpha)\alpha^{t-i}\boldsymbol{S}_{[i]}$. To classify clients as benign or malicious, we use the trust scores and establish a threshold. Specifically, clients with trust scores **above** this threshold are classified as **benign**, while those with trust scores **below** this threshold are classified as **malicious**. Based on the previous research [48], the threshold value is set to 0.5 to produce the most efficient classification results.

C. Theoretical Analysis

*1) **Effectiveness Analysis of FedDMC**:* FedDMC adopts the well-established FedAvg [1] mechanism for model aggregation, inheriting its convergence properties. Therefore, this paper focuses on demonstrating the effectiveness of FedDMC in detecting malicious clients, rather than on the convergence aspects of the model aggregation. To substantiate the effectiveness of FedDMC in the detection of malicious clients, we delve into its core concept: the enhancement of clustering algorithms through the application of PCA, thereby elevating the precision of detection. Theorem 1 provides a rigorous theoretical foundation, elucidating how PCA affects the distances between data points in clustering models.

Definition 1 (k-PC Relative Compression)**.** *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be two vectors in the dataset A. The k-PC relative compression of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined by the ratio:*

$$\Delta_{A,k}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\|\boldsymbol{u} - \boldsymbol{v}\|}{\|P_A^k(\boldsymbol{u}) - P_A^k(\boldsymbol{v})\|}$$

Theorem 1. *Let $\hat{A}$ be a $d \times n$ dataset in the random vector clustering model, where each column vector $\boldsymbol{u}, \boldsymbol{v} \in \hat{A}$ resides in $[0,1]^d$. The variance of all coordinate wise random variables in all cluster distributions is upper bounded by σ^2. Let s_k denote the k-th singular value of $\hat{A}$, and assume there exists a constant C_0 such that $\sigma \geq C_0 \frac{\log n}{n}$. Then, with probability at least $1 - \mathcal{O}(n^{-3})$, the following compression ratio bounds hold for PCA:*

1) *For all intra-cluster pairs of points $\boldsymbol{u}, \boldsymbol{v} \in V_j$, the k-PC compression ratio $\Delta_{A,k}(\boldsymbol{u}, \boldsymbol{v})$ is lower bounded by*

$$\frac{\sqrt{2d\sigma_j^2 - \sqrt{\frac{3d\log n}{2}}}}{\sqrt{8k}\left(\sigma + 4\sqrt{\log(nk)}\right) + C_0 \frac{\sigma^2\sqrt{8d(d+n)}}{s_k}}. \tag{4}$$

2) *For all inter-cluster pairs of points $\boldsymbol{u} \in V_j, \boldsymbol{v} \in V_{j'}, j \neq j'$, the k-PC compression ratio $\Delta_{A,k}(\boldsymbol{u}, \boldsymbol{v})$ is upper bounded by*

$$\frac{\sqrt{d(\sigma_j^2 + \sigma_{j'}^2) + \|\boldsymbol{c}_j - \boldsymbol{c}_{j'}\|^2 + \sqrt{\frac{3d\log n}{2}}}}{\sqrt{2}\|c_j - c_{j'}\| - 2\sqrt{2k}\left(\sigma + 4\sqrt{\log(nk)}\right) - R} \tag{5}$$

where R is $\frac{8C_0\sigma^2\sqrt{(d+n)(\|c_j - c_{j'}\|^2 + d(\sigma_j^2 + \sigma_{j'}^2))}}{s_k}$.

For a detailed proof of Theorem 1, please refer to Appendix B. Theorem 1 suggests that the compression ratio within clusters is significantly greater than the ratio between clusters. Before PCA preprocessing, intra-cluster and inter-cluster distances are highly similar. This similarity indicates that the distribution $D^{(j)}$ significantly influences the pre-PCA distances, leading to a high resemblance between the numerators in Eqs. (4) and (5). Regarding the denominators, a noticeable difference between intra-cluster and inter-cluster distances emerges when the size of $\|c_j - c_j'\|$ substantially exceeds $\sqrt{k}$, and when the k^{th} singular value of $\hat{A}$ significantly surpasses $\sqrt{d+n}$. Many natural clustering models meet these criteria [49]. Hence, this demonstrates that PCA markedly reduces the distances between data points within clusters while only marginally reducing the distances between clusters.

*2) **Complexity Analysis**:* FedDMC requires the server to store n client model parameters, incurring a storage overhead of $\mathcal{O}(nm)$. The space complexity of PCA is typically $\mathcal{O}(nm)$ for an $n \times m$ matrix. For a binary tree consisting of n k-dimensional data, the space complexity is $\mathcal{O}(kn\log_2 n)$. The ensemble detection results require an overhead of $\mathcal{O}(n)$. Thus the total storage overhead of FedDMC is $\mathcal{O}(2nm + kn\log_2 n + n)$. We analyze the time complexity of FedDMC in three aspects. 1) When SVD is used for PCA, the time complexity mainly depends on the computational complexity of SVD. For an $n \times m$ matrix $\boldsymbol{W}$, the time complexity of its singular value decomposition is typically $\mathcal{O}(nm^2)$. During the calculation process, SVD is performed on $\boldsymbol{W}$, and the SVD to the top k singular values is selected to project the original data points onto these vectors. The time complexity of this process is $\mathcal{O}(nmk)$. Therefore, the total time complexity is $\mathcal{O}(nm^2 + nmk)$. 2) The time complexity of performing the BTBCN module on a matrix $\widetilde{\boldsymbol{W}}$ of $n \times k$ depends primarily on the number of distance calculations and cluster merges. The time complexity of distance computation is usually $\mathcal{O}(nk^2)$, where n is the number of vectors, and k is the dimensionality of the vectors. As BTBCN performs $\mathcal{O}(n^2)$ merging operations, the total time complexity is $\mathcal{O}(n^3k^2)$. 3) The time complexity of performing the SEDC module is $\mathcal{O}(n)$. Overall, the total time complexity of FedDMC in each round of detection is $\mathcal{O}(nm^2 + nmk + n^3k^2 + n)$.

V. Experimental Setup

In this section, we present the experimental setup, which includes details of the dataset, model structure, attack methods, detection methods used for comparison, and the metrics to be evaluated.

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

A. Datasets and Model Architectures

As with previous work [24], we evaluated the performance of FedDMC on three standard benchmark datasets. To test the robustness of the method, we also investigated three distinct neural network architectures.

- **MNIST** [50] is a 10-class digital image classification dataset consisting of 60,000 training examples and 10,000 test examples. For MNIST, we use a fully connected network (FC) with layer sizes {784, 100, 10} as the global model architecture.
- **EMNIST** [51] poses a more challenging task for classification involving letters and numbers. The dataset includes 62 classes and 697,932 grayscale images, sharing the same image structure and parameters as the original MNIST task. For EMNIST, we employed a four-layer convolutional neural network (CNN) with two 3×3 convolutional layers with rectified linear unit (ReLU) activation, followed by 2×2 max-pooling (the first with 30 channels and the second with five channels) and one fully-connected layer with ReLU activation (with 100 units).
- **CIFAR10** [52] is a 10-class color image classification dataset consisting of a predefined set of 50,000 training and 10,000 test examples. Each class has an equal number of samples, i.e., 6,000 images per class. For CIFAR10, we utilized the widely used ResNet18 architecture [53] as the global model.
- **COVIDx** [54] is a large-scale, multinational chest radiograph dataset, containing over 30,000 images from 16,400 patients. For COVIDx, we utilized the widely used ResNet50 architecture [53] as the global model.

We set up the FL system with 100 clients and employed the Dirichlet distribution to create a non-IID data distribution among local clients, following prior research [55], [56]. Specifically, we extract $q^j \sim DirN(\beta)$ from a Dirichlet distribution and assign a q_i^j percentage of the examples of class j to client i, where β is the concentration parameter that determines client sameness in the range of 1 to 10. Unless otherwise noted, we set the non-IID degree β to 5.

B. Attack Settings

As in the previous work [8], [24], we randomly selected 28% of the clients as malicious clients by default. We consider both data poisoning attacks and local model poisoning attacks. For data poisoning attacks, we consider the popular Label flipping (LF)-attack and Scaling-attack [7]. For local model poisoning attacks, we consider GS-attack and LIT-attack [10].

- **Label flipping (LF)-attack. [18]** The adversary can be ignorant of the distribution of the training data. This attack flips the labels of each training sample on each manipulated malicious client. Specifically, we randomly flip the label l to l', where $l' \in L,\ l \neq l'$, and L is the set of all labels.
- **LIT-attack [10].** This is a targeted attack. The attacker manipulates all malicious clients to provide a range of perturbations. The attacker uses the maximum value $z^{max} \leftarrow max_z \left(\Phi(z) < \frac{N-M-s}{N-M}\right)$ to circumvent the defense.
- **Scaling-attack [7].** This method uses constraint and scale techniques to incorporate evasive anomaly detection into the attacker's loss function. The malicious clients scale the weights of the model to what the detector allows in order to keep the backdoor from being faded by the server average parameters.
- **Gaussian (GS)-attack [57]:** The existence of Gaussian attacks that send gradients following a certain distribution, which is easily identified by defense methods. We enhance the invisibility of the Gaussian attack by replacing the unpoisoned data with Gaussian noise generated with the mean and variance of the model parameters at each layer.
- **Adaptive-attack [13]:** We assume that attackers know that FedDMC is used to detect malicious clients, and thus they adapt their strategies to avoid detection. Specifically, the adaptive attack strategy considers Euclidean distance constraints and dimension-specific differences. The model parameters provided by the malicious client i in the t-th iteration are denoted as w_i^t and the global model parameters are w_g^t. The malicious client i attempts to solve the following optimization problem:

$$w_i^t = \arg\min_{w_i^t} \quad \lambda||w_i^t - w_g^t||^2 - (1-\lambda)\sum_{j\in d}|w_{i,j}^t - w_{g,j}^t|$$

The first term, $\lambda||w_i^t - w_g^t||^2$, ensures that the model parameters provided by the malicious client closely align with the global model in terms of overall Euclidean distance, thereby enhancing the stealthiness of the attack. The second term, $(1-\lambda)\sum_{j\in d}|w_{i,j}^t - w_{g,j}^t|$, is designed to emphasize significant differences between the malicious model parameters and the global model to ensure the attack's effectiveness. The hyperparameter λ plays a pivotal role in balancing these two aspects; a value of λ close to 1 increases the attack's stealthiness by maintaining overall similarity to the global model, while a lower λ value places greater emphasis on discrepancies in specific dimensions, thereby enhancing the attack's effectiveness.

TABLE II
THE DEFAULT CONFIGURATION OF OUR WORK

Parameter	Default value
Learning rate (η)	0.01
Batch size (B)	128
Number of clients (N)	100
Number of malicious clients (M)	28
Communication rounds (T)	200
Number of local epochs (E)	1
Concentration parameter (β)	5
Ensemble parameter (α)	0.8

C. Compared Detection Methods

The defense methods we consider include Muti-Krum [21], Auror [20], and FLDetector [24]. These methods do not require the server to have a trusted dataset and can output a

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

TABLE III
THE RESULTS (DAR, DPR AND RR) OF MALICIOUS CLIENT DETECTION FOR DIFFERENT ATTACKS, DETECTION METHODS UNDER THREE BENCHMARK DATASETS.

Dataset	Detector	LF-attack			GS-attack			LIT-attack			Scaling-attack			Adaptive-attack		
		DAR	DPR	RR	DAR	DPR	RR	DAR	DPR	RR	DAR	DPR	RR	DAR	DPR	RR
MNIST	Multi_krum	0.96	0.93	0.93	0.88	0.79	0.79	0.44	0.06	0.07	0.82	0.65	0.79	0.44	0.03	0.04
	Auror	0.93	0.89	0.86	0.72	0.50	0.29	0.70	0.46	0.39	0.72	0.50	0.64	0.62	0.22	0.14
	FoolsGold	0.37	0.07	0.11	0.13	0.15	0.46	0.79	**1.00**	0.25	0.22	0.11	0.25	0.40	0.14	0.21
	FLDetector	0.52	0.33	0.71	0.91	0.83	0.86	0.32	0.08	0.14	0.73	0.52	0.54	0.88	0.90	0.64
	FedDMC	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.93**	**0.89**	**0.86**	**0.93**	**0.82**	**0.96**
EMNIST	Multi_krum	0.89	0.81	0.79	0.92	0.86	0.86	0.44	0.26	0.54	0.95	0.93	0.89	0.46	0.20	0.32
	Auror	0.91	0.85	0.82	0.72	0.50	0.75	0.71	0.48	0.39	0.56	0.33	0.57	0.56	0.19	0.18
	FoolsGold	0.25	0.00	0.00	0.14	0.09	0.21	0.96	**1.00**	0.86	0.18	0.05	0.11	0.52	0.29	0.50
	FLDetector	0.95	0.93	0.9	0.89	0.84	0.75	0.53	0.12	0.11	0.89	0.81	0.79	0.76	0.70	0.25
	FedDMC	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.98**	**0.93**	**1.00**
CIFAR10	Multi_krum	0.64	0.35	0.32	0.60	0.29	0.29	0.45	0.25	0.46	0.92	0.86	0.86	0.62	0.31	0.29
	Auror	0.82	0.75	0.54	0.73	0.56	0.18	0.63	0.36	0.43	0.97	0.90	1.00	0.70	0.43	0.21
	FoolsGold	0.38	0.16	0.29	0.71	0.49	0.71	0.29	0.14	0.29	0.26	0.13	0.29	0.46	0.09	0.11
	FLDetector	0.45	0.30	0.71	0.92	0.92	0.79	0.49	0.34	0.89	0.43	0.09	0.11	0.84	0.65	0.93
	FedDMC	**0.99**	**1.00**	**0.96**	**0.97**	**0.93**	**0.96**	**0.92**	**0.81**	**0.93**	**0.99**	**0.97**	**1.00**	**0.88**	**0.71**	**0.96**
COVIDx	Multi_krum	0.69	0.47	0.75	0.76	0.61	0.39	0.81	0.85	0.39	0.85	0.74	0.71	0.47	0.25	0.46
	Auror	0.76	0.56	0.68	0.81	0.71	0.54	0.79	0.62	0.64	0.91	0.79	0.93	0.39	0.27	0.68
	FoolsGold	0.38	0.23	0.61	0.62	0.42	0.89	0.47	0.31	0.71	0.46	0.28	0.57	0.58	0.37	0.71
	FLDetector	0.49	0.34	0.89	0.94	0.87	0.93	0.58	0.35	0.57	0.51	0.32	0.64	0.71	0.49	0.86
	FedDMC	**0.94**	**0.96**	**0.82**	**0.99**	**0.97**	**1.00**	**0.96**	**0.81**	**0.93**	**0.92**	**0.79**	**0.96**	**0.87**	**0.71**	**0.89**

* DAR indicates detection accuracy rate, DPR indicates detection precision rate, RR indicates recall rate. 28 malicious clients are used. The best results are bolded.

list of malicious clients to evaluate detection effectiveness. For Multi-Krum, in general, the server is unaware of how many malicious clients there are. Here, we relax the requirement and set the $Multi\_value$ as the number of benign clients given. For Auror, FoolsGold and FLDetector, we keep the same settings as in previous work [20], [21], [24] to ensure the fairness of the experiment. We only changed the detection method for detecting malicious users and using the same local network model structure, learning rate (η), batch size (B), concentration parameter (β), ensemble parameter (α), and local epoch number (E) for all methods, TABLE II summarizes the default configuration of the experiments. We use PyTorch to implement FedDMC and replicate the attack methods and defenses described above. The code is publicly available.

D. Evaluation Metrics

Before introducing the evaluation metrics, we give the following definitions. True Positive (TP): a client that is malicious and is predicted to be malicious; False Positive (FP): a client that is benign but is predicted to be malicious. True Negative (TN): a client that is benign and is predicted to be benign; False Negative (FN): a client that is malicious but predicted to be benign; We consider a set of metrics to evaluate the validity of defense techniques.

Detection Accuracy Rate (DAR), the proportion of all correct predictions to the total number of samples (including malicious and benign clients), as

$$DAR = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

Although the accuracy rate can determine the total correct rate, it is not a suitable metric to measure the results. This is because affect the aggregated results differently. For example, a benign client identified as malicious only slightly affects the global accuracy, while a malicious client identified as benign significantly reduces the global accuracy.

Detection Precision Rate (DPR), the proportion of correctly predicted malicious to all malicious, as

$$DPR = \frac{TP}{TP + FP} \tag{7}$$

Recall Rate (RR), the proportion of correctly predicted malicious to the proportion of predicted malicious, as

$$RR = \frac{TP}{TP + FN} \tag{8}$$

To evaluate the learned global model, we use Test Accuracy ($TACC$), the proportion of test instances the global model correctly classifies. In addition, for target model poisoning attacks, such as LIT-attack and Scaling-attack, we use the attack success rate (ASR) to evaluate the global model. Specifically, we embedded triggers into each test sample and considered the attack successful if the test sample was classified as a target label by the global model. ASR is the proportion of test inputs that are successfully attacked, and a lower ASR means that the target model poisoning attack is less successful. The experimental result reported is an average of over five trials.

VI. EXPERIMENTAL RESULTS

In this section, we first evaluate the FedDMC framework, then analyze the effectiveness of the three modules and finally compare the efficiency of the extant detection methods.

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

TABLE IV
THE RESULTS ($TACC$ AND ASR) OF THE GLOBAL MODEL FOR THREE BENCHMARK DATASETS WITH DIFFERENT ATTACKS AND DIFFERENT DETECTION METHODS.

Dataset	Detector	No-attack	LF-attack	GS-attack	LIT-attack		Scaling-attack		Adaptive-attack
		TACC	TACC	TACC	TACC	ASR	TACC	ASR	TACC
MNIST	Multi_krum	98.05	97.09	97.22	87.16	8.81	97.95	90.18	80.59
	Auror	98.05	97.16	93.26	98.07	5.75	97.90	89.09	93.41
	FoolsGold	98.05	28.94	9.49	98.00	41.59	9.72	89.94	83.64
	FLDetector	98.05	87.29	97.06	97.02	60.30	97.94	90.19	94.77
	FedDMC	98.05	**97.57**	**98.12**	**98.10**	**0.34**	**98.06**	**54.51**	**98.15**
EMNIST	Multi_krum	85.63	84.25	85.17	85.22	13.24	84.49	69.95	68.62
	Auror	85.63	85.19	75.47	85.32	6.49	83.59	95.01	79.47
	FoolsGold	85.63	4.74	0.67	85.56	7.19	4.96	95.03	80.29
	FLDetector	85.63	84.40	85.24	83.28	32.75	84.08	88.96	83.49
	FedDMC	85.63	**85.47**	**85.81**	**85.77**	**1.98**	**85.89**	**2.19**	**85.43**
CIFAR10	Multi_krum	79.76	76.93	35.98	70.67	5.72	74.48	84.59	68.51
	Auror	79.76	77.18	36.34	75.74	4.50	74.91	89.51	74.69
	FoolsGold	79.76	64.12	45.13	45.33	56.24	22.57	86.61	47.36
	FLDetector	79.76	79.70	74.68	73.26	5.37	51.78	87.56	80.17
	FedDMC	79.76	**85.85**	**75.94**	**77.88**	**1.32**	**80.56**	**44.41**	**82.38**
COVIDx	Multi_krum	84.02	77.16	37.96	71.80	6.82	74.91	57.55	70.81
	Auror	84.02	78.82	58.96	66.58	8.77	66.43	91.90	66.05
	FoolsGold	84.02	76.53	45.83	48.31	9.41	22.85	56.61	50.01
	FLDetector	84.02	80.70	76.04	76.24	4.45	52.88	9.63	81.28
	FedDMC	84.02	**86.91**	**75.95**	**78.51**	**2.64**	**82.09**	**6.43**	**84.01**

* $TACC$ indicates the testing accuracy and ASR indicates the attack success rate. 28 malicious clients are used. The best results are bolded.

TABLE V
COMPUTATIONAL OVERHEAD OF DIFFERENT DEFENSE METHODS

Detector	MNIST	EMNIST	CIFAR10
Multi_krum	**0.73s**	**0.19s**	75.31s
Auror	1.45s	0.56s	128.28s
FoolsGold	0.87s	0.21s	**59.91s**
FLDetector	6.55s	6.31s	160.51s
FedDMC	0.98s	0.34s	64.72s

A. FedDMC Evaluation

*1) **Detection Results**:* TABLE III shows the detection results of three data sets under different attack and defense methods. FedDMC achieves optimal results under these attack settings. In particular, FedDMC achieves 100% detection accuracy for LF-attack, GS-attack, and LIT-attack on MNIST and FMNIST datasets. We analyze these defense methods. i) Multi_krum is unable to defend against LIT-attack. For LIT-attack, the adversary can manipulate all malicious clients and constrain the uploaded model parameters. The adversary sets all malicious model parameters the same to amplify the effect of the attack. Multi_krum selects M model parameters that are most similar to other clients so that the LIT-attack can escape detection by Multi_krum. ii) Auror is a defense method based on K-means clustering to detect malicious clients. K-means has apparent defects, such as the poor clustering effect of K-means when the data is in a certain distribution. This is why Auror is only effective against LF-attack and is ineffective against other attack methods. iii) FoolsGold is only effective with LIT-attack and has little effect on other attack methods. This is because FoolsGold assumes that the angle between malicious clients is smaller than the angle between benign and malicious clients. FoolsGold reflects this similarity in terms of cosine similarity. However, the parameters of malicious clients are unknown and impossible to estimate in most cases, so the angle between malicious users is not necessarily slight. For example, in GS attacks, the parameters of benign clients are more similar, while the parameters of malicious clients are more scattered in distribution. Therefore, the detection accuracy of FoolsGold on MNIST and EMNIST datasets is 0. iv) FLDetector is the state-of-the-art among the existing defense methods. FLDetector is defensive against these attacks and can detect most malicious clients. However, FLDetector does not work well in specific experimental settings, such as LIT-attack. We reimplemented FedDMC following the experimental setup in FLDetector, and the experimental results are shown in Appendix C. Experimental results show that FedDMC excels on the complex COVIDx dataset, outperforming existing methods. This underscores our method 's effectiveness and practicality for large, complex datasets. In general, FedDMC outperforms existing defense methods in varying degrees of accuracy and can defend against a wider range of attacks.

*2) **Performance of the Global Models**:* TABLE IV shows the TACC and ASR of the global model for the three benchmark datasets under different attacks and detection methods. "No-attack" implies learning the global model using the remaining 72% of benign clients. The experimental setup for the other four attacks remains 28% malicious clients and 72% benign clients. The LIT-attack and the Scaling-attack are targeted backdoor attacks. Therefore, the ASR of each detection method is tested against these two attacks.

We observe that the global model is even more accurate under some attacks than under no attacks. This is because there is still residual value in the malicious client. The models learned by FedDMC under several other attacks reach the same

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

TABLE VI
THE IMPACT OF THE NUMBER OF MALICIOUS CLIENTS ON FEDDMC UNDER DIFFERENT ATTACKS.

Number of malicious clients	LF-attack		GS-attack		LIT-attack			Scaling-attack		
	DAR	TACC	DAR	TACC	DAR	TACC	ASR	DAR	TACC	ASR
10	1.00	86.13	1.00	86.17	1.00	85.96	0.89	1.00	85.68	1.15
15	1.00	85.94	1.00	85.91	1.00	85.83	0.97	1.00	85.34	1.56
20	1.00	85.68	1.00	85.73	1.00	85.17	1.41	1.00	85.16	2.07
25	1.00	85.56	1.00	85.69	1.00	84.87	1.62	1.00	84.92	2.12
30	1.00	85.44	1.00	85.61	1.00	84.49	1.93	1.00	84.73	2.19
35	1.00	85.11	1.00	85.56	1.00	84.38	2.69	1.00	84.67	1.96
40	1.00	85.31	1.00	85.53	1.00	84.34	2.16	1.00	84.49	2.44
45	1.00	84.99	1.00	85.50	1.00	84.04	2.12	1.00	84.39	2.13

* DAR indicates detection accuracy rate, $TACC$ indicates the testing accuracy, and ASR indicates the attack success rate. The dataset for the experiments is MNIST.

level as the models learned under No-attack. The accuracy of the global model learned by FedDMC is the highest compared to other defense methods. In the LIT-attack and Scaling-attack, FedDMC achieves the lowest ASR, which means that FedDMC effectively suppresses backdoor infiltration by removing malicious clients. Scaling-attack amplifies the parameters in order to make the backdoor not be faded through the global average. If a small number of malicious clients are undetected, the backdoor can be easily implanted in the global model. FedDMC benefits from excellent detection, thus weakening the backdoor implantation of the Scaling-attack.

*3) **Computational Overhead**:* We conducted experiments on the defense methods on three datasets to show the superiority of the computational overhead of FedDMC. Even though servers possess greater computational resources, the efficiency of processing and aggregating high-dimensional parameter models from hundreds or thousands of clients simultaneously is critical, especially in frequent aggregation and updating scenarios. FedDMC focuses primarily on server-side optimization in federated learning, specifically on efficiently processing data from clients to detect and remove malicious clients, thereby ensuring the security of the global model. Here we only consider the time consumed to execute the detection algorithm on the server side, excluding the local training time and the parameter transfer time. TABLE V shows the experimental results of these defense methods, and the computational overhead of FedDMC is within an acceptable range. Compared to the state-of-the-art method FLDetector, FedDMC shows advantages not only in terms of detection accuracy but also achieves significant improvements in computational efficiency.

Moreover, we measured the computation cost and overhead of the FedDMC components under the COVIDx dataset, which corresponds to a local client with a network model of ResNet50. Specifically, the computation time of the PCA is 26.8 seconds, the time of the BTBCN module is 0.99×10^{-3} seconds, and the processing time of the SEDC module is is less than 10^{-6} seconds. The experimental results demonstrate that FedDMC is highly feasible and practical in resource-constrained environments.

B. Impact of Key Parameters

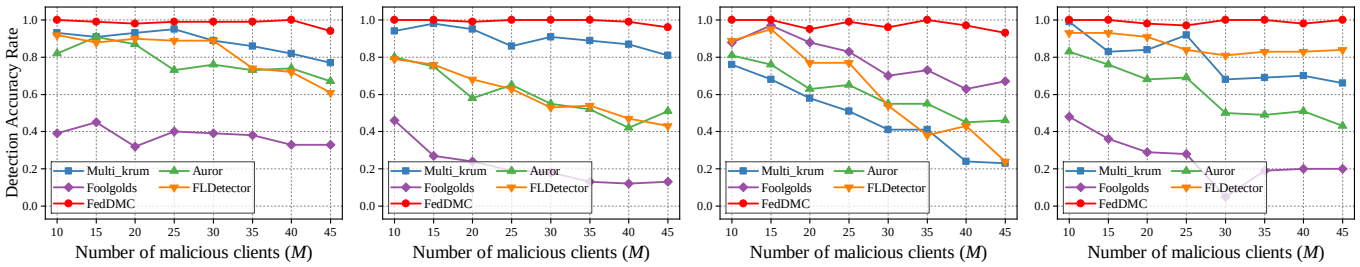
*1) **Impact of the Number of Malicious Clients**:* We explore the impact of different numbers of clients on FedDMC. We compare different defense methods under different attacks affected by the number of malicious clients, as shown in Fig 5. We observe that, except for FedDMC, other defense methods decrease in detection accuracy as the number of malicious clients increases. Some defense methods work only under certain attacks, e.g., Foolsgolds is effective only for LIT-attack, and Auror is effective only for LF-attack. Because of space limitation, we only show the experimental results for the dataset EMNIST in TABLE VI. We can get some observations. i) FedDMC shows strong stability under different attacks. ii) FedDMC detection can always reach 100% accuracy under the condition that the number of malicious clients $M < \lfloor \frac{N}{2} \rfloor$ is satisfied. iii) The accuracy of the global model decreases as the number of malicious clients increases. iv) In targeted attacks, the ASR increases a little with the increase of malicious clients, but it is still guaranteed to be no more than 3%.

*2) **Impact of the Different Degrees of Non-IID**:* We explore the impact of different degrees of non-IID on FedDMC and conduct experiments on the dataset EMNIST. From Fig. 6, we observe that the DAR of all defense methods increases as the Concentration parameter (β) increases (The smaller the β, the more unbalanced the distribution). This indicates that the degree of non-IID affects the detection accuracy of defense methods. FedDMC maintains a high DAR for different attacks, and FedDMC achieves 100% detection accuracy after the Concentration parameter (β) reaches a certain threshold. For example, the detection accuracy of FedDMC starts to decrease for LF-attack with $\beta < 3$ and for GS-attack with $\beta < 5$. Specifically, FedDMC exhibits better detection than other defense methods at different levels of non-IID.

*3) **Impact of Different Degrees of Dimensionality Reduction**:* We explored the impact of the degree of dimensionality reduction (k) on FedDMC. We conduct experiments on three datasets and set k as $\{100, 20, 10, 5, 2\}$, the attack method as LF-attack, and the number of malicious clients as 28, respectively. For each set of experiments, we compared the experimental results for different concentration parameters (β). As shown in Fig. 7, the overall trend is that as the degree of dimensionality reduction (k) decreases, the detection accuracy (DAR) increases. However, it is clear that the detection accuracy decreases as the skew of the data distribution increases. In the case of skewed data distribution, it is not that the lower the dimension, the higher the detection accuracy. For example, in Fig. 7(c), when $\beta = 1$, this set of experiments achieves the

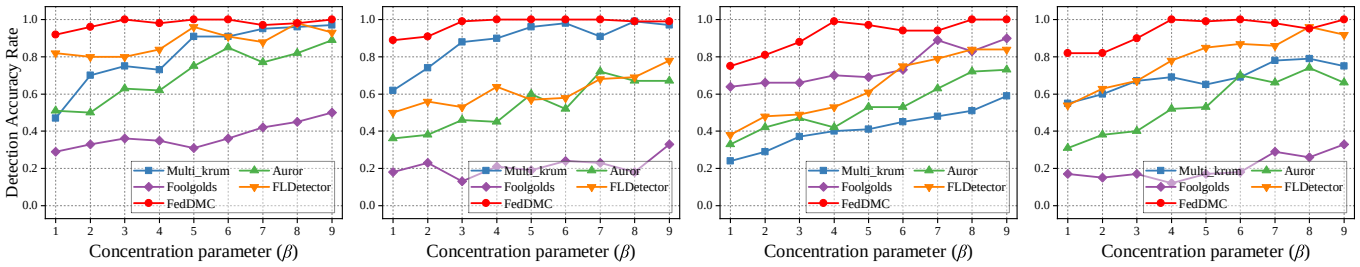
Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634



(a) LF-attack (b) GS-attack (c) LIT-attack (d) Scaling-attack

Fig. 5. The impact of the number of malicious clients on the defense methods under different attacks.



(a) LF-attack (b) GS-attack (c) LIT-attack (d) Scaling-attack

Fig. 6. The impact of the degree of non-IID on the EMNIST dataset under different attacks.

TABLE VII
THE DISTANCES OF LOCAL MODEL PARAMETERS IN DIFFERENT ROUNDS AFTER DIMENSIONALITY REDUCTION TO DIFFERENT DIMENSIONS.

Dimension	$t=15$			$t=25$			$t=35$			$t=45$			$t=55$			$t=65$		
	D_{bb}	D_{mb}	RE	D_{bb}	D_{mb}	RE	D_{bb}	D_{mb}	RE	D_{bb}	D_{mb}	RE	D_{bb}	D_{mb}	RE	D_{bb}	D_{mb}	RE
k=78500	20.23	28.18	**0.28**	18.89	28.59	**0.34**	18.04	28.02	**0.36**	17.86	27.82	**0.36**	17.39	28.02	**0.38**	17.21	27.83	**0.38**
k=100	20.23	28.18	**0.28**	18.89	28.6	**0.34**	18.04	28.01	**0.36**	17.86	27.83	**0.36**	17.38	28.03	**0.38**	17.23	27.84	**0.38**
k=20	10.03	22.61	**0.56**	9.03	23.49	**0.62**	8.35	23.19	**0.64**	8.06	23.14	**0.65**	7.67	23.53	**0.67**	7.24	23.54	**0.69**
k=10	6.44	21.82	**0.71**	5.98	22.52	**0.74**	5.53	22.15	**0.75**	5.28	22.03	**0.76**	5.27	22.24	**0.76**	4.89	22.25	**0.78**
k=5	3.6	21.4	**0.83**	3.37	22.07	**0.85**	3.32	21.6	**0.85**	3.13	21.4	**0.85**	3.17	21.71	**0.85**	3.03	21.48	**0.86**
k=2	1.29	21.12	**0.94**	1.26	21.8	**0.94**	1.11	21.31	**0.95**	1.03	21.1	**0.95**	1.31	21.43	**0.94**	1.06	21.21	**0.95**

* We set up the FL system with 100 clients, among which 20 are malicious, and these malicious clients are manipulated by the attacker to execute the label flipping attack. The dataset for the experiments is MNIST and the network structure is a fully connected network (FC) with layers of size {784, 100, 10}. D_{bb} represents the distance between benign clients and benign clients, D_{mb} represents the distance between benign clients and malicious clients, and RE represents the relative error between the two distances.

highest detection accuracy when $k = 10$. The reason for the misclassification is that when the data is unbalanced, the model parameters of the clients also drift, and the dimensionality reduction can also widen the distance between these clients and other clients. The goal of PCA is to transform the original data into a set of linearly uncorrelated representations, which are sorted according to variance. However, calculating distance based solely on the first component isn't necessarily the best approach. The Euclidean distance takes into account the differences across all components, offering a comprehensive measure of similarity. Given the heterogeneity of client data, there might be biases in the model parameters. This suggests that relying solely on the first component might not ensure robust clustering.

*4) **Impact of Different** $min\_cluster\_size$:* We conducted experiments to determine the most appropriate minimum clustering size threshold. As shown in Fig. 9(a), for $min\_cluster\_size = 3$, the detection accuracy is optimal on multiple datasets. For future work, we will utilize machine learning techniques to automatically adjust the thresholds based on the distribution of data points. This strategy is expected to enhance the adaptability of FedDMC.

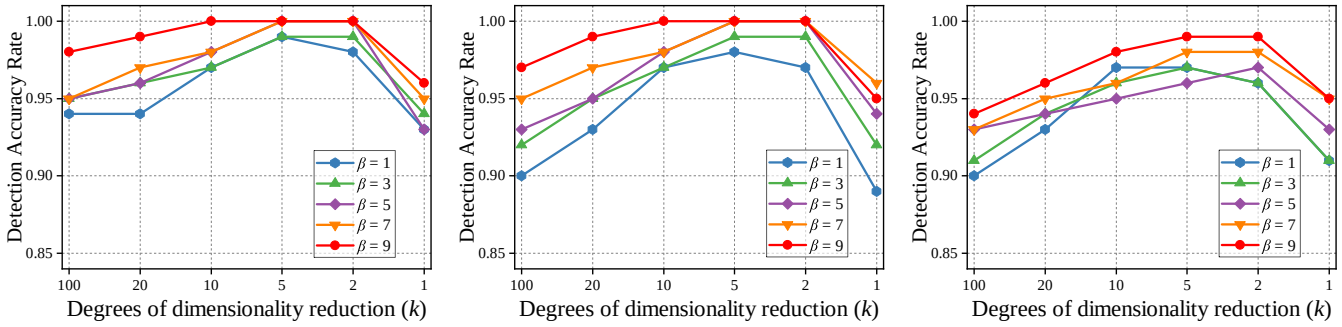
*5) **Impact of the Adaptive Attack**:* Fig. 9(b) shows the performance when applying the adaptive attack to FedDMC. As the hyperparameter λ increases, the DAR exhibits a decreasing trend. This indicates that larger values of λ can effectively improve the stealthiness of the attack, making it more difficult to detect malicious clients. However, the TACC does not significantly decrease, which is due to the fact that the pursuit of attack covertness correspondingly diminishes the destructive potency of the attack. In other words, although malicious clients are not recognized by the defense mechanisms, the malicious parameters they provide do not significantly affect the accuracy of the global model.

C. Module Evaluation

*1) **Evaluation of PCA Dimensionality Reduction Performance**:* PCA significantly decreases the distance of data points belonging to the same cluster while relatively mildly

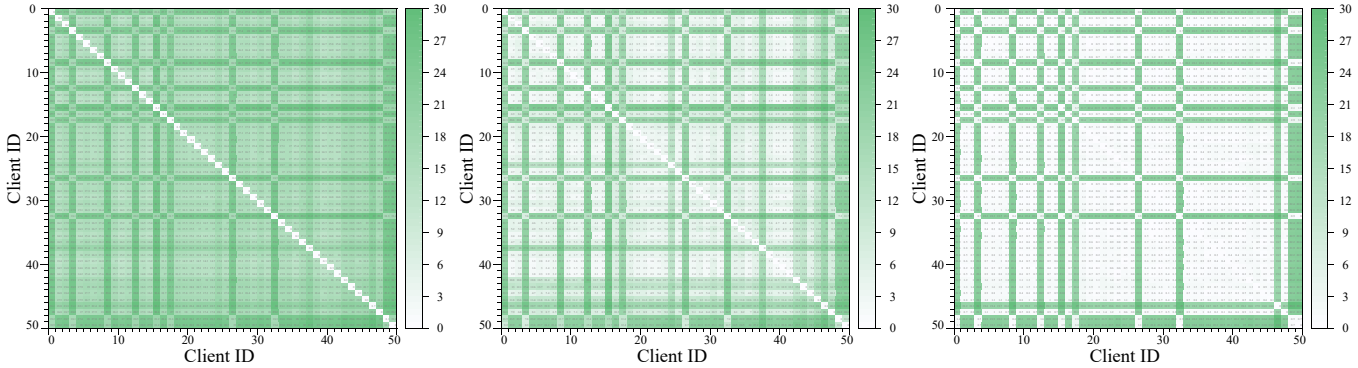
Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634



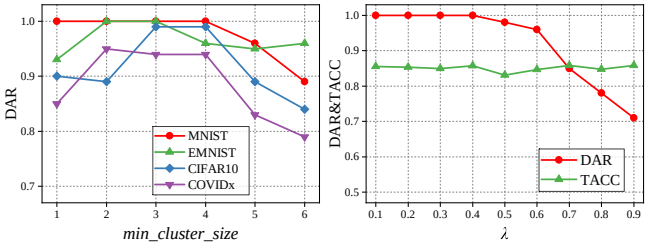
(a) MNIST (b) EMNIST (c) CIFAR10

Fig. 7. The impact of different degradation levels on the effectiveness of FedDMC defense.



(a) $k = 100$ (b) $k = 20$ (c) $k = 2$

Fig. 8. The distance relationship between the 50 clients at different degrees of dimensionality reduction.



(a) (b)

Fig. 9. (a) The impact of different $min\_cluster\_size$; (b) The impact of different λ on the effectiveness of FedDMC defense in adaptive attack.

decreasing the inter-cluster distance. This means that the processing with PCA relatively increases the distance between clusters. TABLE VII describes this observation visually. We set up the FL system with 100 clients, among which 20 are malicious, and these malicious clients are manipulated by the attacker to execute the LF-attack. For the sake of description, the average distance between benign clients is defined as D_{bb} and the average distance between benign and malicious clients as D_{mb}, and the relative error between D_{bb} and D_{mb} is denoted by RE, which is calculated as $RE = (D_{mb} - D_{bb})/D_{mb}$. What is clear from this table is that before the dimensionality reduction, D_{bb} and D_{mb} are similar. What stands out is that as the dimensionality decreases (k), the relative error rises rapidly, as shown in each column of TABLE VII. The distance between benign clients decreases significantly, while the distance between benign and malicious clients decreases slightly. Apart from that, it can be noticed from TABLE VII that D_{bb} tends to decrease with the training iterations (t), which is due to the gradient between benign clients getting closer as the model converges. However, it is clear from the relative error RE that this does not contribute much to distinguishing between benign and malicious clients. In other words, as the dimensionality decreases, it becomes more obvious to distinguish between benign and malicious clients.

We visualize the distance relationship among 50 clients in Fig. 8. It can be clearly seen that as the dimension decreases ($k : 100 \rightarrow 20 \rightarrow 5$), the direct distance distinction between malicious clients and benign clients becomes more obvious. This also shows that PCA significantly reduces the distance between data points belonging to the same cluster while relatively mildly reducing the distance between clusters.

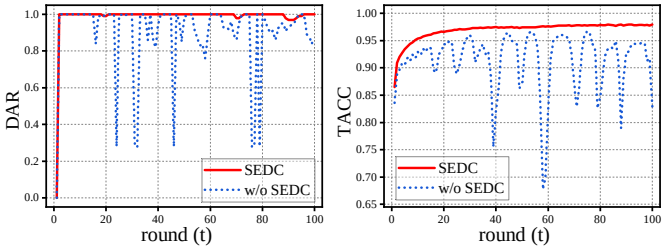
We conducted tests on the BTBCN module for different values of k, specifically $k = 10^5, 10^4, 10^3, 10^2$, and 10. The time overheads for these tests were 2.53 s, 0.217 s, 0.0229 s, 3.98×10^{-3} s, and 1.06×10^{-3} s, respectively. A significant increase in computation time for distance calculations was observed when the dimensionality of points increased from 10 to 10^5. This increase in computation time significantly impacted the overall runtime of the algorithm. The time overhead for the SEDC module remained below 10^{-6} seconds for different values of k, thus it was considered negligible.

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

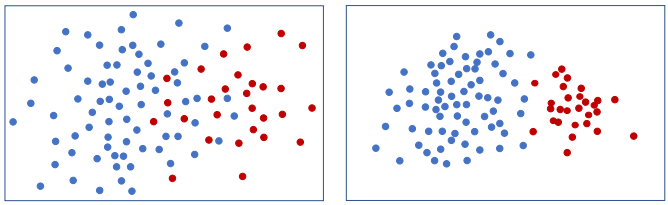
TABLE VIII
COMPARISON OF THE RESULTS OF THE EXISTING CLUSTERING METHODS REPLACING BTBCN IN THE FEDDMC FRAMEWORK, AND THE COMPUTATIONAL OVERHEAD OF EACH METHOD.

Clustering methods	LF-attack		GS-attack		LIT-attack		Scaling-attack		Computational overhead
	RR	TACC	RR	TACC	RR	TACC	RR	TACC	
K-means [43]	0.73	65.21	0.90	75.13	0.56	72.81	0.91	77.17	77.63S
Hierarchical clustering [44]	0.85	73.68	0.91	75.42	0.42	64.32	0.84	76.85	**70.54s**
DBSCAN [45]	0.86	84.16	0.95	75.85	0.74	75.84	0.97	79.46	106.41s
HDBSCAN [47]	0.95	85.77	0.96	75.78	0.79	76.45	1.00	80.42	132.93 s
Robust K-means++ [32]	0.95	85.51	**0.97**	75.92	0.80	77.59	1.00	**80.63**	112.49 s
BTBCN (our work)	**0.96**	**85.85**	0.96	**75.94**	**0.82**	**77.88**	**1.00**	80.56	**71.82s**



(a) Detection Precision Rate (b) Global Model Test Accuracy

Fig. 10. The detection precision rate and global model test accuracy for all rounds. The dataset is MNIST and the attack setting is LF-attack.



(a) Client distribution before PCA (b) Client distribution after PCA

Fig. 11. Client distribution using t-SNE visualization.

*2) **Evaluation of BTBCN Effectiveness**:* To evaluate the efficiency of the proposed BTBCN module in terms of accuracy and computational cost, we replaced the clustering module in FedDMC with various clustering methods, such as K-means [43], Hierarchical clustering [44], DBSCAN [45], and HDBSCAN [47]. We conducted experiments on the CIFAR-10 dataset, and the results presented in TABLE VIII demonstrate that the BTBCN module outperforms other clustering methods. Traditional clustering methods, such as K-means and Hierarchical clustering, are susceptible to extremely malicious values, resulting in suboptimal performance. DBSCAN and HDBSCAN outperform traditional methods but at the cost of extensive computational resources. The Robust K-means++ algorithm is comparable to BTBCN in terms of accuracy in detecting malicious clients. However, BTBCN has a significant advantage in terms of Computational overhead. This advantage is derived from BTBCN's implementation of a binary tree structure for clustering, which offers faster performance. In contrast, the Robust K-means++ algorithm necessitates a significant amount of computational resources during the initialization of cluster centers. In summary, the BTBCN demonstrates significant advantages in malicious client detection due to its robustness to noise and computational efficiency. We utilize BTBCN to address the limitations of existing clustering-based malicious client detection implementations, minimizing false positives in poisoned model detection. We visualize the model parameters for the 100 local clients using the t-SNE method. Fig. 11(a) shows the result of direct visualization using t-SNE, while Fig. 11(b) shows the result of first reducing the dimensionality to 5 using PCA and then visualizing it using t-SNE.

*3) **Evaluation of SEDC Effectiveness**:* We evaluate the effectiveness of the Self-Ensemble Detection Correction (SEDC) module in enhancing the overall stability and accuracy of malicious client detection. To assess the impact of the SEDC module, we compare the performance of our proposed method, FedDMC, with and without the SEDC module. We conducted experiments using the MNIST dataset and under LF-attack as the adversarial setting. Considering the stability of the learning process, the evaluation metrics include detection accuracy and global model testing accuracy for all rounds. As depicted in Fig. 10, the experimental results demonstrate that incorporating the SEDC module into FedDMC significantly improves in detection accuracy, along with an increased and stabilized global model accuracy. By leveraging historical detection results, the SEDC module effectively mitigates the effects of single-round detection inaccuracies, resulting in a more reliable and robust malicious client detection process. This highlights the effectiveness of the SEDC module in enhancing the performance of our proposed method FedDMC under challenging conditions such as different data distributions and uncertain attacks from malicious clients.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented FedDMC, a novel approach for detecting malicious clients in federated learning settings, which combines PCA dimensionality reduction, the *binary tree-based clustering with noise* (BTBCN), and the *self-ensemble detection correction* (SEDC). Our experimental results demonstrate the effectiveness of our proposed method in achieving high detection accuracy while maintaining reasonable time and space complexity. Although our proposed approach shows promising results, it is essential to consider the privacy protection aspect when detecting malicious clients. Detecting malicious clients becomes more challenging under the premise of privacy protection. For future work, we plan to explore the integration of privacy-preserving techniques such as secure shuffling [58] and secure multi-party computation

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

[59] into our framework. This would allow us to maintain a high level of privacy protection while effectively detecting and mitigating the impact of malicious clients in federated learning scenarios.

REFERENCES

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

[2] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen *et al.*, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[3] X. Mu, Y. Shen, K. Cheng, X. Geng, J. Fu, T. Zhang, and Z. Zhang, "Fedproc: Prototypical contrastive federated learning on non-iid data," *Future Generation Computer Systems*, vol. 143, pp. 93–104, 2023.

[4] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated learning*. Springer, 2020, pp. 240–254.

[5] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

[6] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 070–16 084, 2020.

[7] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.

[8] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International Conference on Learning Representations*, 2019.

[9] X. Cao and N. Z. Gong, "Mpaf: Model poisoning attacks to federated learning based on fake clients," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3396–3404.

[10] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[11] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[12] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*. PMLR, 2019, pp. 634–643.

[13] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021.

[14] T. D. Nguyen, P. Rieger, R. De Viti, H. Chen, B. B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen *et al.*, "{FLAME}: Taming backdoors in federated learning," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1415–1432.

[15] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[16] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[17] R. Guerraoui, S. Rouault *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3521–3530.

[18] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," in *NDSS*, 2021.

[19] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," *arXiv preprint arXiv:2002.00211*, 2020.

[20] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, 2016, pp. 508–519.

[21] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[22] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings." in *RAID*, 2020, pp. 301–316.

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

[24] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 2545–2555.

[25] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[26] J. Ma, M. Xie, and G. Long, "Personalized federated learning with robust clustering against model poisoning," in *International Conference on Advanced Data Mining and Applications*. Springer, 2022, pp. 238–252.

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

[28] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[29] M. Xie, J. MA, G. Long, and C. Zhang, "Robust clustered federated learning with bootstrap median-of-means," in *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. Springer, 2022, pp. 237–250.

[30] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 6357–6368.

[31] G. Hamerly and C. Elkan, "Learning the k in k-means," *Advances in neural information processing systems*, vol. 16, 2003.

[32] A. Deshpande, P. Kacham, and R. Pratap, "Robust k-means++," in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 799–808.

[33] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[34] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 586–19 597, 2020.

[35] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: clients clustering for better personalization," *World Wide Web*, vol. 26, no. 1, pp. 481–500, 2023.

[36] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–9.

[37] J. Ma, G. Long, T. Zhou, J. Jiang, and C. Zhang, "On the convergence of clustered federated learning," *arXiv preprint arXiv:2202.06187*, 2022.

[38] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[39] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[40] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[41] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 245–250.

[42] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[43] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*. University of California Los Angeles LA USA, 1967, pp. 281–297.

[44] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[45] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[46] M. Ankerst, M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, "Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD*, vol. 99, 2008.

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2024.3372634

[47] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17*. Springer, 2013, pp. 160–172.

[48] Y. Lu and W. He, "Selc: self-ensemble label correction improves learning with noisy labels," in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org, 2022, pp. 3278–3284. [Online]. Available: https://doi.org/10.24963/ijcai.2022/455

[49] C. S. Mukherjee and J. Zhang, "Compressibility: Power of pca in clustering problems beyond dimensionality reduction," *arXiv preprint arXiv:2204.10888*, 2022.

[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[51] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2921–2926.

[52] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[54] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific reports*, vol. 10, no. 1, p. 19549, 2020.

[55] X. Mu, Y. Shen, K. Cheng, X. Geng, J. Fu, T. Zhang, and Z. Zhang, "Fedproc: Prototypical contrastive federated learning on non-iid data," *arXiv preprint arXiv:2109.12273*, 2021.

[56] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7252–7261.

[57] X. Ma, Q. Jiang, M. Shojafar, M. Alazab, S. Kumar, and S. Kumari, "Disbezant: secure and robust federated learning against byzantine attack in iot-enabled mts," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[58] K. Cheng, Y. Hou, and L. Wang, "Secure similar sequence query over multi-source genomic data on cloud," *IEEE Transactions on Cloud Computing*, 2022.

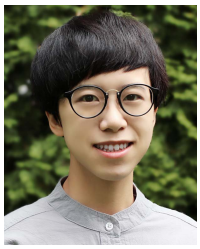
[59] K. Cheng, J. Fu, Y. Shen, H. Gao, N. Xi, Z. Zhang, and X. Zhu, "Manto: A practical and secure inference service of convolutional neural networks for iot," *IEEE Internet of Things Journal*, 2023.

[60] V. VU, "A simple svd algorithm for finding hidden partitions," *Combinatorics, Probability and Computing*, vol. 27, no. 1, p. 124–140, 2018.

Xutong Mu received his B.S. degree from North University of China, China, in 2019. He is currently pursuing the Ph.D degree with the School of Computer Science and Technology, Xidian University, China. His research interests include machine learning security and federated learning.

Ke Cheng is a lecturer in the School of Computer Science and Technology at Xidian University. He received his B.S. and M.S. degree from Anhui University, Hefei, China, in 2015 and 2018, respectively. He received Ph.D. degree in computer science and technology from Xidian University in 2022. His research interests include cloud computing security, data security, and privacy protection.

Yulong Shen (Member, IEEE) received the B.S. and M.S. degrees in computer science and the Ph.D. degree in cryptography from Xidian University, Xi'an, China, in 2002, 2005, and 2008, respectively. He is currently a Professor with the School of Computer Science and Technology, Xidian University, where he is also an Associate Director of the Shaanxi Key Laboratory of Network and System Security and a member of the State Key Laboratory of Integrated Services Networks. His research interests include wireless network security and cloud computing security. He has also served on the technical program committees of several international conferences, including ICEBE, INCoS, CIS, and SOWN.



Xiaoxiao Li (Member, IEEE) received the Ph.D. degree from Yale University, New Haven, CT, USA, in 2020. She was a Post-Doctoral Research Fellow with the Computer Science Department, Princeton University, Princeton, NJ, USA. Since August 2021, she has been an Assistant Professor at the Department of Electrical and Computer Engineering (ECE), University of British Columbia (UBC), Vancouver, BC, Canada. In the last few years, she has over 30 papers published in leading machine learning conferences and journals, including NeurIPS, ICML, ICLR, MICCAL, IPML, BMVC, IEEE TRANSACTIONS ON MEDICAL LMAGING, and Medical lmage Analysis. Her research interests include range across the interdisciplinary fields of deep learning and biomedical data analysis, aiming to improve the trustworthiness of artificial intelligence (AI) systems for health care. Dr. Li's work has been recognized with several best paper awards at international conferences.

Zhao Chang received a BS degree in computer science and technology from Peking University in 2013, and a PhD degree in computing from University of Utah in 2021. He currently works as an associate professor in the School of Computer Science and Technology at Xidian University. His research interests focus on security and privacy issues in large-scale data management.

Tao Zhang received the M.S. and Ph.D. degrees in computer science from Xidian University, Xi'an, China, in 2011 and 2015, respectively. He is currently an Assistant Professor with the School of Computer Science and Technology, Xidian University. His research interests include network security and privacy protection.

Xindi Ma (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from Xidian University in 2013 and 2018, respectively. He is currently an Associate Professor at the School of Cyber Engineering, Xidian University. His current research interests include machine learning and location-based service and recommender system with focus on the security and privacy issues.

Authorized licensed use limited to: Iowa State University Library. Downloaded on March 06,2024 at 23:25:47 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.