CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Journal Paper

GradCAM-AE: A New Shield Defense against Poisoning Attacks on Federated Learning

Jingjing Zheng*

Kai Li*

Xin Yuan

Wei Ni*

Eduardo Tovar*

Ozgur B. Akan

*CISTER Research Centre

CISTER-TR-250803

2025

GradCAM-AE: A New Shield Defense against Poisoning Attacks on Federated Learning

Jingjing Zheng*, Kai Li*, Xin Yuan, Wei Ni*, Eduardo Tovar*, Ozgur B. Akan

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: zheng@isep.ipp.pt, kai@isep.ipp.pt, xin.yuan@data61.csiro.au, wei.ni@cister-labs.pt, emt@isep.ipp.pt, oba21@cam.ac.uk

https://www.cister-labs.pt

Abstract

Recent poisoning attacks on federated learning (FL) generate malicious model updates that circumvent widely adopted Euclidean distance-based detection methods. This paper proposes a new defense mechanism, namely, GradCAM-AE, against model poisoning attacks on FL, which integrates Gradient-weighted Class Activation Mapping (GradCAM) and autoencoder (AE) to offer a substantially more powerful detection capability compared to existing Euclidean distance-based approaches. Particularly, GradCAM-AE generates a heat map for each uploaded local model update, transforming each local model update into a lower-dimensional, visual representation. An autoencoder further reprojects the GradCAM heat maps of all local module updates with improved distinguishability, thereby accentuating the hidden features of the heat maps and increasing the success rate of identifying anomalous heat maps and malicious local models. A comprehensive evaluation of the proposed GradCAM-AE framework is conducted using the CIFAR-10 and GTSRB datasets under both Independent and Identically Distributed (IID) and Non-IID settings. The ResNet-18 and MobileNetV3-Large models are tested. The results substantiate that GradCAM-AE offers superior detection rates and test accuracy of FL global model, juxtaposed with contemporary state-of-the-art methods. Our code is available at: https://github.com/jjzgeeks/GradCAM-AE.

GradCAM-AE: A New Shield Defense against Poisoning Attacks on Federated Learning

JINGJING ZHENG, Real-Time and Embedded Computing Systems Research Centre (CISTER), Porto, Portugal
KAI LI*, University of Cambridge, Cambridge, U.K., and CISTER Research Centre, Porto, Portugal
XIN YUAN, Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, Australia
WEI NI, Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, Australia
EDUARDO TOVAR, Real-Time and Embedded Computing Systems Research Centre (CISTER), Porto, Portugal
ÖZGÜR B. AKAN, University of Cambridge, Cambridge, U.K., and Center for NeXt-Generation Communications (CXC), Koç University, Istanbul, Turkey

Recent poisoning attacks on federated learning (FL) generate malicious model updates that circumvent widely adopted Euclidean distance-based detection methods. This paper proposes a new defense mechanism, namely, GradCAM-AE, against model poisoning attacks on FL, which integrates Gradient-weighted Class Activation Mapping (GradCAM) and autoencoder (AE) to offer a substantially more powerful detection capability compared to existing Euclidean distance-based approaches. Particularly, GradCAM-AE generates a heat map for each uploaded local model update, transforming each local model update into a lower-dimensional, visual representation. An autoencoder further reprojects the GradCAM heat maps of all local module updates with improved distinguishability, thereby accentuating the hidden features of the heat maps and increasing the success rate of identifying anomalous heat maps and malicious local models. A comprehensive evaluation of the proposed GradCAM-AE framework is conducted using the CIFAR-10 and GTSRB datasets under both Independent and Identically Distributed (IID) and Non-IID settings. The ResNet-18 and MobileNetV3-Large models are tested. The results substantiate that GradCAM-AE offers superior detection rates and test accuracy of FL global model, juxtaposed with contemporary state-of-the-art methods. Our code is available at: https://github.com/jjzgeeks/GradCAM-AE

CCS Concepts: • **Security and privacy Code → Social aspects of security and privacy**; • **Computer systems organization →** *Neural networks.*

Additional Key Words and Phrases: Federated learning, poisoning attacks, gradient-weighted class activation mapping, autoencoder

ACM Reference Format:
Jingjing Zheng, Kai Li, Xin Yuan, Wei Ni, Eduardo Tovar, and Özgür B. Akan. 2025. GradCAM-AE: A New Shield Defense against Poisoning Attacks on Federated Learning. 1, 1 (August 2025), 25 pages. https://doi.org/X

*Corresponding author

Authors' Contact Information: Jingjing Zheng, zheng@isep.ipp.pt, Real-Time and Embedded Computing Systems Research Centre (CISTER), Porto, Porto, Portugal; Kai Li, University of Cambridge, Cambridge, Cambridge, U.K., and CISTER Research Centre, Porto, Portugal, kaili@ieee.org; Xin Yuan, Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, NSW, Sydney, Australia, xin.yuan@data61.csiro.au; Wei Ni, Commonwealth Scientific and Industrial Research Organization (CSIRO), Sydney, NSW, Sydney, Australia, wei.ni@data61.csiro.au; Eduardo Tovar, Real-Time and Embedded Computing Systems Research Centre (CISTER), Porto, Porto, Portugal, emt@isep.ipp.pt; Özgür B. Akan, University of Cambridge, Cambridge, Cambridge, U.K., and Center for NeXt-Generation Communications (CXC), Koç University, Istanbul, Turkey, oba21@cam.ac.uk.

1 Introduction

Federated Learning (FL) has surfaced as a notably distributed machine learning framework, where user devices collaboratively engage in the training of a shared machine learning model, supervised by a server. User devices in FL consecutively train local model updates (e.g., weight parameters or gradients) utilizing their proprietary data. Rather than transmitting raw, private data, user devices forward model updates to a server for amalgamation. In response, the server integrates local model updates to generate a comprehensive global model that is then dispatched to the devices for updating their respective local models [11, 21]. Such a communication cycle repeats until the model achieves a satisfactory accuracy level. FL prevents the potential unauthorized dissemination of private data [42]. For instance, FL allows multiple medical institutions to collaboratively train a unified machine learning model without directly sharing sensitive patient data. By retaining patient data locally, hospitals can contribute to model improvements by processing data on-site and sharing only model updates, thereby enhancing diagnostic tools or treatment algorithms while adhering to data protection regulations [14].

The distributed architecture of FL makes it particularly susceptible to poisoning attacks. User devices compromised by adversarial actors can alter model update parameters, subsequently contaminating the global FL models [3, 7, 24]. Existing countermeasures, leveraging Euclidean distance-based metrics to discern deviations between malicious and benign models, have demonstrated efficacy against such attacks [1, 36]. However, sophisticated adversaries can craft malicious model updates such that the Euclidean distances to benign counterparts remain below a designated threshold, thereby eluding detection by defenses reliant on this metric.

This paper proposes a new defense mechanism, dubbed as GradCAM-AE, against model poisoning attacks on FL, which employs a Gradient-weighted Class Activation Mapping (GradCAM [23])-based approach in coupling with an autoencoder (AE) to offer a substantially more powerful detection capability compared to existing Euclidean distance-based approaches. Specifically, GradCAM is applied at the server to create GradCAM heat maps for every uploaded model update. An autoencoder is applied to reconstruct the heat maps, while magnifying the discernible features of the heat maps. The reconstruction errors of the GradCAM heat maps are measured, and a threshold is created based on the statistics of the reconstruction errors. A reconstructed GradCAM heat map with a reconstruction error surpassing the threshold is categorized as atypical, and the corresponding model update as malicious.

The key contributions are summarized as follows:

- We propose a novel defense method against model poisoning attacks on FL, where GradCAM and autoencoder are orchestrated for the successful detection of subtle attacks.
- GradCAM is adopted to produce heat maps for each uploaded local model, hence transforming each local model into a lower-dimensional, visual representation. This provides a conduit for pinpointing malicious model updates by singling out anomalous GradCAM heat maps.
- An autoencoder is used to remap the GradCAM heat maps to accentuate the hidden features of the heat maps, and improve the distinguishability of the heat maps and the success rate of identifying anomalous heat maps and malicious local models.
- To mitigate potential misclassifications, we also design a voting algorithm, where model updates that consistently manifest as potentially malicious over several communication rounds are conclusively identified as malicious and removed from FL.

We conducted a comprehensive assessment of the proposed GradCAM-AE framework using two public datasets, CIFAR-10 and GTSRB, under both IID and non-IID settings. Our assessment encompasses two prominent deep learning models, i.e., ResNet-18 [9] and MobileNetV3-Large [10]. Our approach offers superior detection rates and test accuracy of FL global model compared to the state-of-the-art methods. The rest of this work is organized as follows. Section 2 discusses the literature on defense models against poisoning attacks on FL. Section 3 presents FL for image classification with deep neural networks. In Section 5, we detail the proposed GradCAM-AE against model poisoning attacks. Performance evaluation is presented in Section 6. This work is concluded in Section 7.

2 Related Work

This section reviews the literature on defense models against poisoning attacks on FL.

Euclidean distance-based defense. Several extant solutions, predicated upon Euclidean distance, have been devised to detect the poisoning attack within FL contexts. Within the framework, Krum [1, 37] or Multi-Krum [1, 37] computes a score for each local model update, the score is the sum of its Euclidean distance from its neighbors. Multi-Krum considers those with high scores as malicious model updates, which are excluded. In stark contrast, Trimmed-mean methodology [33] is a coordinate-wise aggregation rule that aggregates each coordinate of the model update separately. For each coordinate, first sort the values of the corresponding coordinates in the users' model updates. The largest and the smallest k values are removed, then the trimmed mean calculates the average of the remaining values as the corresponding coordinate of the aggregated model update. To defend against poisoning attacks with a large number of malicious users, FLDetector [36] is developed to predict a device's model update in each communication round based on historical model updates. The received model update from the user and the predicted model update are inconsistent in multiple communication rounds are flagged as malicious. A defense strategy is developed against poisoning attacks on FL [4], where users are categorized into distinct groups. A global model is trained for each user group according to an existing FL aggregation rule. Based on the global models of all the groups, a majority vote mechanism is used to identify whether a test input is poisoned by the attacker.

Cosine-similarity based defense. [2] uses cosine similarity to measure the alignment of local model updates with the trusted global model update that is trained by clean dataset. The local model updates with low similarity are scaled down or ignored. [22] utilizes cosine similarity to measures the angular similarity between local model updates and a reference direction (e.g., the aggregated global model or the average of local model updates). The local model updates are clustered based on their angular similarity. The clusters with low cosine similarities to the majority of local model updates are flagged as suspicious and excluded from aggregation. Similarly, [43] keeps the underlying assumption is that the local gradients of benign device align closely with the direction of global gradient, whereas malicious gradients deviate significantly. The authors use cosine similarity to measure the angular alignment between the local gradients of devices and the global gradients maintained by the server. The local gradients with obtuse angles (cosine similarity is less than zero) are flagged as malicious because they indicate opposing or disruptive directions. The server discards these local gradients to prevent them from affecting the global model's update.

Machine learning-based defense. To detect the malicious local devices while generating an accurate model, a statistical mechanism called Auror is introduced [25]. AUROR is based on the observation that the most important model features from the majority of honest devices will exhibit a similar distribution, while those from malicious devices will exhibit an anomalous distribution. AUROR then uses k-means to cluster uploaded local model updates over training rounds and discards the malicious model updates, i.e., contributions from small clusters that exceed a threshold distance are considered malicious. The authors of [26] presented the Federated Anomaly Analytics enhanced Distributed Learning (FAA-DL) framework, which is a lightweight, unsupervised anomaly detection method based on support vector machine (SVM). The main idea of FAA-DL is to utilize an appropriate kernel function and soft margins to estimate a nonlinear decision boundary and separate the benign and malicious local model updates. Detecting network attacks is crucial for preventing and alleviating the consequences of network attacks. To detect and filter malicious local model updates before global model aggregation, the authors of [17] developed the FL framework, which is an integrated isolation forest algorithm. The authors argue that the leaf nodes representing the malicious model are closer to the root. The malicious local model updates can be detected by employing deep reinforcement learning methods to adjust the detection threshold dynamically. To resist generative adversarial network (GAN) attacks, the authors of [5] developed a system that isolates local model update parameters of the network from all devices, derailing any chance that the attacker can set up GAN to carry out attacks.

The existing defense mechanisms for malicious local model updates, particularly those predicated on Euclidean distances, notably encounter challenges pertaining to the "curse of dimensionality" within the context of deep neural networks (DNNs), especially considering that local model update parameters encompass millions or potentially billions of parameters (encompassing both weights and biases). In the milieu of high-dimensional spaces, Euclidean distances may inadvertently overestimate the distances between model updates, thereby diminishing the efficacy of detecting malicious model updates. Predominantly, machine-learning-based detection mechanisms significantly hinge upon the meticulous tuning of hyperparameters and the astute setting of thresholds. Recent experimental findings [8, 39] contend that this approach potentially verges on ineffective anomaly detection.

In contrast, the proposed GradCAM-AE pioneers the exploration of GradCAM-based malicious local model detection in FL, developing a methodology that eschews traditional Euclidean distance measurement. The GradCAM-AE defense strategy against poisoning attacks in FL is capable of converting the local model parameters-ranging in the millions—into a low-dimensional GradCAM heat map. Additionally, an autoencoder is designed with GradCAM to enhance the separability of the heat maps, thereby increasing the success rate in identifying anomalous heat maps generated by malicious model updates.

3 Federated Learning

In this section, we consider image classification as an example to depict the training process of FL. The system consists of a server and K benign users where $\mathcal{K} \triangleq \{1, \cdots, K\}$, as shown in Fig. 1. Each benign user device trains a Deep Neural Network (DNN) that is particularly well-suited for the image classification. The DNN analyzes an input image (e.g., deer), extracts features (e.g., edges, corners, and textures) through several convolutional layers, feeds the feature maps into fully connected (FC) layers, and assigns one of the labels (such as "bird", "dog", "horse" or "deer"). The DNN model is learned based on training data and represented by the parameter vector $\mathbf{W}$ (weights and biases). The user device possesses D_k training data pairs $\{(\mathbf{x}_s, y_s)\}_{s=1}^{D_k}$, consisting of a feature vector $\mathbf{x}_s$ (e.g., the pixels of an image) as the input of the model and a corresponding scalar value y_s (e.g., the real label of the image) that is the desired output of the model [30]. The dataset of the user device is denoted by $\mathcal{D}_k$, and we have

$$\mathcal{D}_k = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_s, y_s), \cdots, (\mathbf{x}_{D_k}, y_{D_k})\}, \tag{1}$$

where D_k denotes the cardinality of $\mathcal{D}_k$, i.e., $D_k \triangleq |\mathcal{D}_k|$.

Let $f_s(\mathbf{W}, \mathbf{x}_s, y_s)$ denote the loss function of each training data sample s, which captures the approximation errors between the input $\mathbf{x}_s$ and desired output y_s. For instance, $f_s(\mathbf{W}, \mathbf{x}_s, y_s) = -y_s \times \log\left(\frac{1}{1+\exp(-\mathbf{W}^{\mathrm{T}}\mathbf{x}_s)}\right) - (1 - y_s)\log\left(1 - \frac{1}{1+\exp(-\mathbf{W}^{\mathrm{T}}\mathbf{x}_s)}\right)$ accounts for cross entropy function in this paper. Here, $\mathbf{W}^{\mathrm{T}}$ is the transpose of $\mathbf{W}$. For

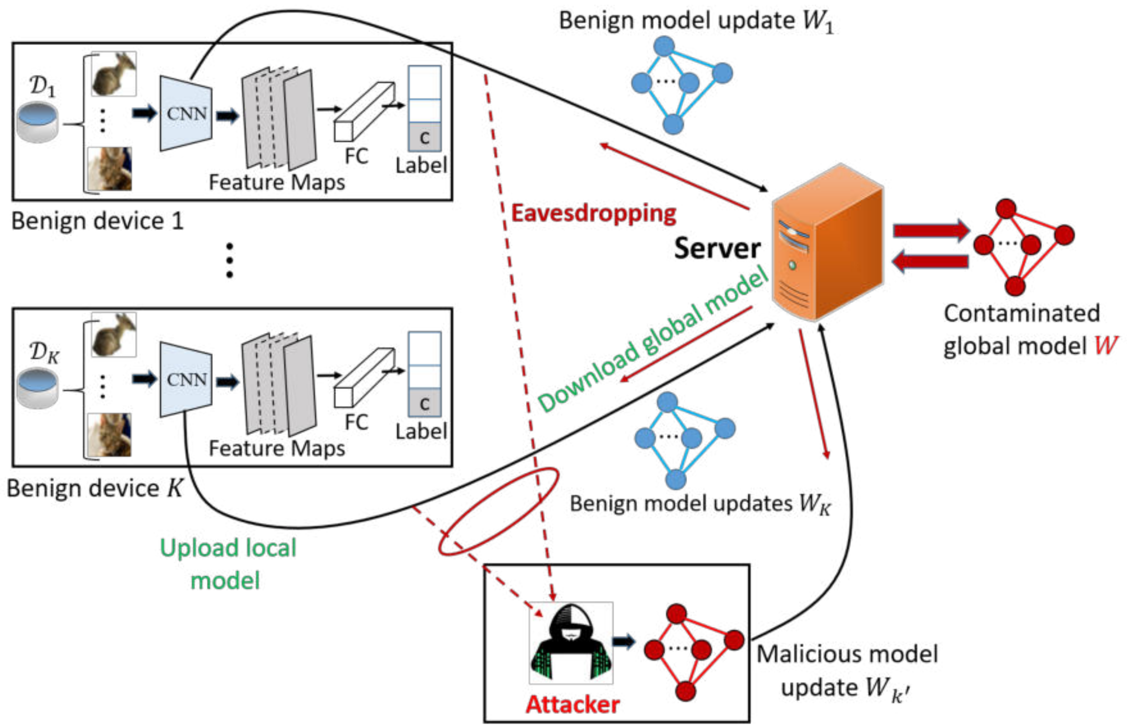

Fig. 1. FL with poisoning attack, where the attacker eavesdrops on the benign local models uploaded by the devices. The attacker fabricates a malicious local model update based on its eavesdropped local model updates. The Euclidean distances between the malicious and benign local models fall within a preset threshold distance.

each device k, given the dataset $\mathcal{D}_k$, the local loss function of $\mathbf{W}$ on the collection of data samples is defined as

$$F_k(\mathbf{W}) \triangleq \frac{1}{D_k} \sum_{s \in \mathcal{D}_k} f_s(\mathbf{W}, \mathbf{x}_s, y_s). \tag{2}$$

Accordingly, the global loss function on all the distributed datasets evaluated the parameter vector $\mathbf{W}$ is given by

$$F(\mathbf{W}) = \sum_{k \in \mathcal{K}} \frac{D_k}{D} F_k(\mathbf{W}), \tag{3}$$

where $D = \sum_{k \in \mathcal{K}} D_k$. The goal of the training process is to find the optimal parameter $\mathbf{W}^*$ such that the global loss function $F(\mathbf{W})$ is minimized, i.e.,

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} F(\mathbf{W}). \tag{4}$$

To solve problem (4) while preserving the data privacy for each device, a canonical gradient-descent technique is widely used in state-of-the-art FL systems [18, 21, 34], as follows.

Let T_{FL} denote the total number of communication rounds. $\mathcal{T} \triangleq \{1, \cdots, T_{FL}\}$. In each round $t \in \mathcal{T}$, there are three steps:

- *Device selection and broadcast:* The server samples K devices satisfying eligibility requirements. The selected devices download the global model $\mathbf{W}^t$ from the server.

- *Local model training and uploading:* Each user device trains its local model based on its local dataset $\mathcal{D}_k$ for L local training epochs, starting with the global model $\mathbf{W}^t$, i.e.,

$$\mathbf{W}_{k,\ell+1}^t = \mathbf{W}_{k,\ell}^t - \eta_t \nabla F_k(\mathbf{W}_{k,\ell}^t, \mathcal{D}_k),\ \ell \in \{1, \cdots, L\}, \tag{5}$$

where η_t denotes the learning rate in the t-th communication round, and $\nabla F_k(\mathbf{W}_{k,\ell}^t, \mathcal{D}_k)$ defines the local gradient estimate over $\mathcal{D}_k$ at local training epoch ℓ. After L local training epochs, each user device k uploads its updated local model $\mathbf{W}_k^t = \mathbf{W}_{k,L}^t$ to the server.

- *Global model aggregation:* The server aggregates the K devices' updated local models $\{\mathbf{W}_k^t\}_{k=1}^K$ to obtain the updated global model, given by

$$\mathbf{W}^{t+1} = \sum_{k \in \mathcal{K}} \frac{D_k}{D} \mathbf{W}_k^t, \tag{6}$$

where $\mathbf{W}^{t+1}$ is sent back to all K devices.

The above process repeats until convergence is achieved or the maximum number of communication rounds T_{FL} is met. Without loss of generality, we assume all K user devices are involved in FL.

4 Threat Model

Malicious devices may attempt to compromise the global and local models by injecting poisoned data into their local models or intentionally tampering with the local model parameters during training. We consider that the model poisoning attack is launched by the attackers whose goal is to impair or degrade the accuracy and convergence of FL. A general model poisoning attack in [7, 24] is considered in our paper as the threat model. Specifically, the considered threat model presents a general model poisoning attack in which the attacker may appear as a legitimate participant that receives the global model at each communication round, the attacker is assumed to have complete knowledge of the global model's architecture, parameters, and the aggregation rules (such as FedAvg), allowing the attacker to craft malicious local model updates to manipulate the benign FL process. By progressively injecting malicious parameters into the aggregation process, the attacker seeks to degrade the global model's accuracy over time.

During the FL process, the attackers collected by the set of $\mathcal{M} \triangleq \{1', \cdots, K'\}$ ($K' < K$) maintain the same DNN structure as benign devices, as shown in Fig. 1. Meanwhile, the attacker eavesdrops on C ($C \leq K$) benign local model updates during the model update uploading to the server. Inspired by projected gradient descent attacks [20, 44], an attacker may find the perturbation on the eavesdropped local model updates that maximize the loss of the global model while keeping the size of the perturbation smaller than a specified threshold.

If the server is unaware of the malicious local models, it unintentionally creates a contaminated global model. The total size of the local training data reported to the server is $S = \sum_{k \in \mathcal{K}} D_k + \sum_{k' \in \mathcal{M}} D_{k'}$, where $D_{k'}$ is the claimed data size by attacker k'. The contaminated global model is given by

$$\mathbf{W}_g^{t+1} = \sum_{k \in \mathcal{K}} \frac{D_k}{S} \mathbf{W}_k^t + \sum_{k' \in \mathcal{M}} \frac{D_k'}{S} \mathbf{W}_{k'}^t. \tag{7}$$

The server broadcasts $\mathbf{W}_g^{t+1}$ to all user devices. The FL training turns to minimize the following global loss function:

$$\min_{\mathbf{W}_g^{t+1}} F(\mathbf{W}_g^{t+1}) = \sum_{k \in \mathcal{K}} \frac{D_k}{S} F_k(\mathbf{W}_g^{t+1}) + \sum_{k' \in \mathcal{M}} \frac{D_{k'}}{S} F_{k'}(\mathbf{W}_g^{t+1}), \tag{8}$$

where $F_{k'}(\cdot)$ is the claimed local loss function of the k'-th attacker.

The attackers aim to maximize $F(\mathbf{W}_g^{t+1})$, i.e., equivalent to minimizing the training accuracy of FL, while keeping $\mathbf{W}_{k'}^t$ imperceptible to the server that may adopt the state-of-the-art Euclidean distance-based [1, 33] or similarity-based detection methods [2] to detect and identify malicious local model updates. Suppose that an attacker can measure the Euclidean distance between $\mathbf{W}_{k'}^t$ and $\mathbf{W}_g^{t+1}$ and make it smaller than a threshold r to evade detection:

$$\max_{\mathbf{W}_{k'}^t} F(\mathbf{W}_g^{t+1}) \tag{9a}$$

$$\text{s.t } \mathfrak{D}\big(\mathbf{W}_{k'}^t, \mathbf{W}_g^{t+1}\big) \leq r, \forall k' \in \mathcal{M}, \tag{9b}$$

where $\mathfrak{D}(\cdot,\cdot)$ is the Euclidean distance function $\mathfrak{D}\big(\mathbf{W}_{k'}^t, \mathbf{W}_g^{t+1}\big) = \|\mathbf{W}_{k'}^t - \mathbf{W}_g^{t+1}\|_2$. The optimal malicious local model updates are

$$\big(\mathbf{W}_{k'}^t\big)^* = \underset{\mathbf{W}_{k'}^t}{\operatorname{argmax}}\ F(\mathbf{W}_g^{t+1}). \tag{10}$$

Note that the solution to optimization (9) can be found in the latest literature [16], and the goal of this paper is not to solve this optimization problem. We aim to defend against such attacks. In particular, the threat model described in [16] represents a specific instance of the general model poisoning attack considered in our paper, where the malicious update is generated to minimize the global model's accuracy. By exploiting the feature correlation between benign local and global models, the attacker in [16] introduces subtle perturbations to the local model updates, while the malicious model remains undetected given existing Euclidean distance-based or similarity-based defense strategies. While the threat model in [16] as a concrete example, our proposed defense model remains effective even against the broader class of general model poisoning attacks in [7, 24].

5 Proposed GradCAM-AE against Model Poisoning Attacks

In this section, we elaborate on the GradCAM-AE, where the GradCAM and Autoencoder are leveraged on the server side for pinpointing the malicious local models.

5.1 GradCAM-AE Architecture

On the device side, each of the K benign devices utilizes the DNN tailored specifically for image classification tasks, as shown in Fig.2. The DNN model extracts relevant features from an input image (e.g., a bird) and subsequently maps them to the corresponding classes. The architecture of a typical DNN comprises multiple layers, each with a specific function. (a) The first layer is a convolutional layer, which applies a set of filters to the input image, thereby extracting intrinsic features, such as edges, corners, and textures. The output of the convolutional layer is a set of feature maps, each representing different aspects of the image. (b) The following layer is a pooling layer, which reduces the dimensionality of the feature maps while retaining essential information, such as the salient features of the bird's beak or feather. Various pooling methods can be employed, including max pooling or average pooling, all with the objective of downsampling the feature maps while retaining their salient features. Some DNN architectures may differ from traditional pooling operations. For example, models like SqueezeNet [13], ResNet [9], DenseNet [12], and MobileNet [10] employ alternative strategies. (c) Subsequent to several convolution and pooling layers, the extracted features are fed into one or multiple fully connected layers, where the final classification is performed.

Upon receiving the local model updates from the devices, the server aggregates the local models, where the benign local models can be mingled with malicious local models. We aim to design a defense countermeasure that can pinpoint and filter malicious devices. GradCAM [23] visualization is distinguished by its high resolution and high-class

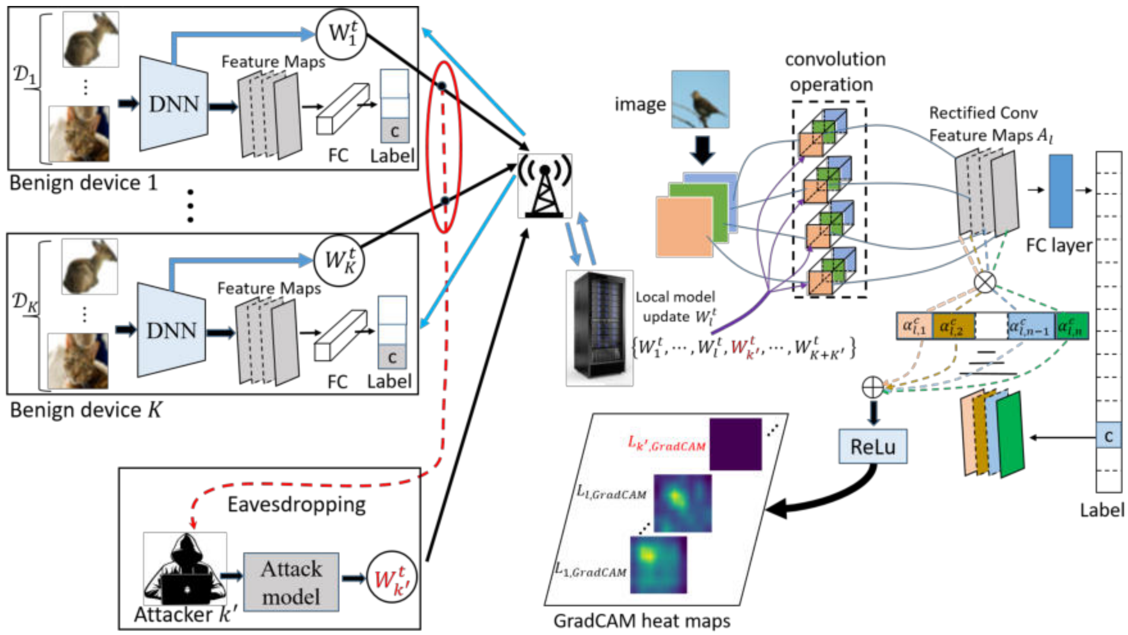

Fig. 2. GradCAM-assisted defense against poisoning attacks on FL. The server arbitrarily selects an image (e.g., an image with the label "bird") from the global model testing dataset to create GradCAM heat maps for every uploaded local model update. These GradCAM heat maps flow into an autoencoder for malicious model detection.

discriminative ability compared to other methods, e.g., Class Activation Mapping [40]. GradCAM is leveraged in our design to detect malicious local model updates. Specifically, the server randomly picks an image from the global model testing dataset that incorporates all categories of the devices dataset as input and passes through the convolutional layers with weight and bias parameters that are replaced by each model update $\mathbf{W}_l, l \in \mathcal{K} \cup \mathcal{M}$ (l is the index of local model updates, including benign and malicious), obtains the feature maps[1] A_l that has n channels. The feature maps A_l are fed into a fully connected layer for final classification.

To obtain the class discriminative localization map of $L^{(c)}_{l,GradCAM} \in R^{B\times H}$ with width B and height H for any class c, GradCAM first computes the gradient of the score for class c, $Y^{(c)}$ (before softmax) with respect to each feature map A^m_l of a convolutional layer, i.e., $\frac{\partial Y^{(c)}}{\partial A^m_l}$. $m \in [1, n]$ is the index of channels. The neuron importance weights $\alpha^{(c)}_{l,m}$ can be obtained through global average pooling:

$$\alpha^{(c)}_{l,m} = \frac{1}{B\times H}\sum_{i=1}^{B}\sum_{j=1}^{H}\frac{\partial Y^{(c)}}{\partial A^m_l(i,j)}, \forall l \in \mathcal{K} \cup \mathcal{M}, \tag{11}$$

where $A^m_l(i,j)$ is the activation at location (i,j) of the feature map A^m_l. We apply a rectified linear unit (ReLU) to the weighted linear combination of the forward activation. $L^{(c)}_{l,GradCAM}$ is given by

$$L^{(c)}_{l,GradCAM} = \mathrm{ReLU}(\sum_{m=1}^{n}\alpha^{(c)}_{l,m}A^m_l). \tag{12}$$

[1]In this paper, we focus on extracting the feature maps after the final convolution layer of deep learning model in that can capture high-level features and hold information regarding the important regions of the input image.

The server uses the image and local model updates to obtain a single-channel GradCAM heat map that is input into the autoencoder for identification. For conciseness, we suppress the indicator of classes c and rewrite $L^{(c)}_{l,GradCAM}$ as $L_{l,GradCAM}$ in what follows.

5.2 Autoencoder for Malicious GradCAM Heat Maps Identification

Considering all GradCAM heat maps are unlabeled, an autoencoder is an efficient tool in unsupervised learning to discover non-linear features across anomaly detection systems [41]. We use an autoencoder to pinpoint abnormal GradCAM heat maps corresponding to malicious local model updates uploaded by the attackers.

As shown in Fig. 3, a canonical autoencoder consists primarily of three components: an encoder, a code (or a latent space) and a decoder. Each GradCAM heat map $L_{l,GradCAM}$ with size of $H \times B$ is flattened into a vector with size of $1 \times H \times B$, which is further concatenated with the vectors of other GradCAM heat maps to form $\mathbf{L}_{\text{GradCAM}}$ as the input to the encoder. During the autoencoder training, the encoder $e_\theta(\mathbf{L}_{\text{GradCAM}})$ with parameter θ compresses the GradCAM heat maps from a high-dimensional space to a low-dimensional space $z = e_\theta(\mathbf{L}_{\text{GradCAM}})$, also called the code or the latent space. The code learns the underlying features or representation of the GradCAM heat maps, which are input to the decoder $d_\phi(z)$ with parameter ϕ. The decoder further reconstructs the input GradCAM heat maps from the code, i.e., $d_\phi(z) = \mathbf{L}'_{\text{GradCAM}} = d_\phi\left(e_\theta(\mathbf{L}_{\text{GradCAM}})\right)$. After the training of (θ, ϕ), the reconstructed GradCAM heat maps are reshaped into the same size as the original GradCAM heat maps.

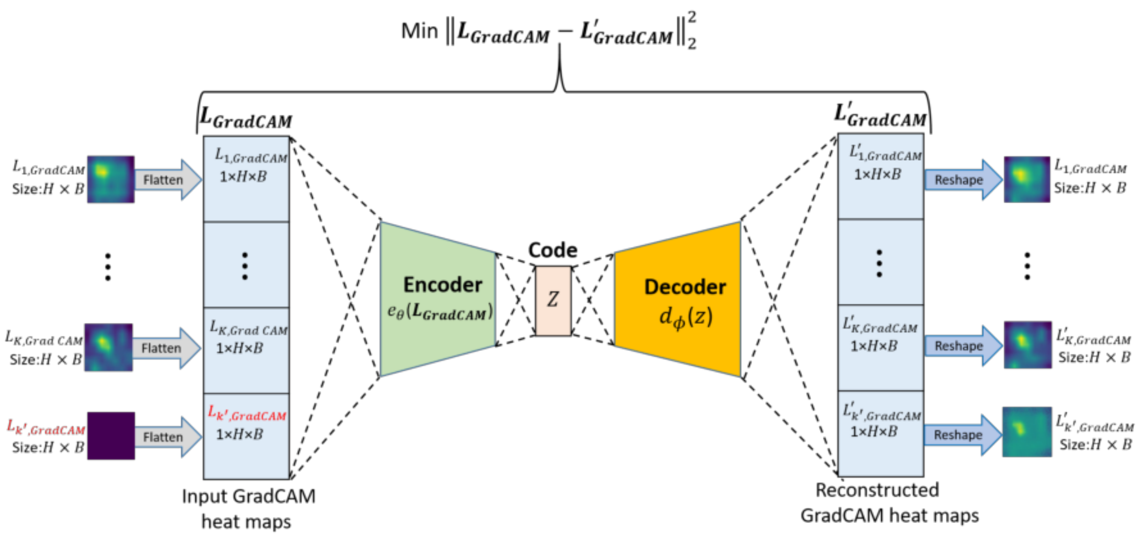

Fig. 3. Autoencoder-based abnormal GradCAM heat map identification. The server flattens and concatenates GradCAM heat maps as input to the encoder neural network, which compresses the GradCAM heat maps from a high dimension to a low dimension Z. The decoder neural network takes Z as its input to reconstruct the original input GradCAM heat maps.

Loss Function. To minimize the difference between the original input GradCAM heat maps and reconstructed GradCAM heat maps, the autoencoder loss function is defined as the mean squared error (MSE) between the encoder input GradCAM heat maps $\mathbf{L}_{\mathbf{GradCAM}}$, and the decoder reconstructed GradCAM heat maps $\mathbf{L}^{'}_{\mathbf{GradCAM}}$, i.e.,

$$\begin{aligned} L(\theta,\phi) &= \min_{\theta,\phi} \frac{1}{|K| + |K'|} \|\mathbf{L}_{\mathbf{GradCAM}} - \mathbf{L}^{'}_{\mathbf{GradCAM}}\|_2^2 \\ &= \min_{\theta,\phi} \frac{1}{K+K'} \sum_{l=1}^{K+K'} \|L_{l,GradCAM} - d_{\phi}(e_{\theta}(L_{l,GradCAM}))\|_2^2. \end{aligned} \tag{13}$$

Once the autoencoder completes training, the server computes the reconstruction errors between each reconstructed GradCAM heat map and its corresponding input GradCAM heat map and obtains the mean reconstruction error, i.e., $\forall l \in \mathcal{K} \cup \mathcal{M}$,

$$R_l = \frac{\sum_{i=1}^{B} \sum_{j=1}^{H} \left|L_{l,GradCAM}(i,j) - L^{'}_{l,GradCAM}(i,j)\right|}{H \times B}. \tag{14}$$

The average reconstruction error of all GradCAM heat maps is

$$\overline{R} = \frac{1}{R_l} \sum_{l=1}^{K+K'} R_l. \tag{15}$$

A threshold δ is defined as

$$\delta = \overline{R} + \alpha \times \sqrt{\frac{\sum_{l=1}^{K+K'} (R_l - \overline{R})^2}{K + K'}}, \tag{16}$$

where α is an empirically configured coefficient. Here, δ is used to distinguish between the benign and malicious GradCAM heat maps. Typically, the reconstruction errors with regard to the GradCAM heat maps follow a normal distribution, the reconstruction errors of GradCAM heat maps lie outside α standard deviation of the mean reconstruction errors are statistically rare and considered potential anomalies [28, 31]. As a result, (16) is used to distinguish between the benign and malicious GradCAM heat maps. If the mean reconstruction error of each GradCAM heat map is greater than the threshold δ, the corresponding input of the GradCAM heat map is considered potentially abnormal. Otherwise, it is a potential normal GradCAM heat map because the AE learns to capture variations in normal GradCAM heat maps during training.

Note that the autoencoder is trained on both normal and abnormal (or malicious) GradCAM heat maps. The autoencoder is designed to minimize the reconstruction error between the input heat maps and their reconstructed counterparts. The reconstruction error at the autoencoder is a widely adopted measure of anomaly or abnormality, given that the autoencoder can learn the distribution of normal data and thus reconstruct such data with minimal error. In contrast, out-of-distribution or abnormal inputs tend to exhibit larger reconstruction errors since the autoencoder has not learned the patterns characteristic of such inputs, which means autoencoder encounter difficulties in handling anomalies that do not conform to the learned patterns. This phenomenon is well-substantiated in various domains, ranging from industrial defect detection to imaging, where high reconstruction errors have repeatedly been shown to strongly correlate with anomalies [6, 38]. As a result, it is reasonable to expect that the GradCAM heat maps that can be reconstructed with low reconstruction errors are considered normal, while those with high reconstruction errors are potentially abnormal.

Also note that δ is time-varying with communication rounds as the GradCAM heat maps change over the communication rounds. The detection results are stored in a buffer:

$$O_l^t = \begin{cases} 1, & \text{if } R_l \leq \delta; \\ 0, & \text{if } R_l > \delta, \end{cases} \tag{17}$$

where “1” and “0” indicate a benign and a malicious local model update, respectively.

A voting mechanism is further designed to further reduce the possibility of misclassifying a benign local model update. The key idea is that the server makes final decisions every fixed ξ communication rounds to determine which devices are malicious. Specifically, the local model updates ϵ out of ξ communication rounds are detected as potentially malicious. They are determined malicious and removed from global aggregations. Finally, (7) is rewritten as

$$\mathbf{W}_g^{t+1} = \begin{cases} \sum_{l \in \mathcal{K} \cup \mathcal{M}} O_l^t \times \frac{D_l}{S} \times \mathbf{W}_l^t, & \text{if } t \text{ modulo } \xi \neq 0; \\ \sum_{l \in \mathcal{K} \cup \mathcal{M}} Y_l^t \times \frac{D_l}{S} \times \mathbf{W}_l^t, & \text{if } t \text{ modulo } \xi = 0, \end{cases} \tag{18}$$

where Y_l^t is given by

$$Y_l^t = \begin{cases} 1, & \text{if } \sum_{t=N\xi-\xi+1}^{N\xi} O_l^t < \epsilon, N = 1, \cdots, \lfloor \frac{T_{FL}}{\xi} \rfloor; \\ 0, & \text{if } \sum_{t=N\xi-\xi+1}^{N\xi} O_l^t \geq \epsilon, N = 1, \cdots, \lfloor \frac{T_{FL}}{\xi} \rfloor. \end{cases} \tag{19}$$

The updated global model $\mathbf{W}_g^{t+1}$ is sent back to all devices. The detailed pseudo-code is given in **Algorithm 1**.

6 PERFORMANCE EVALUATION

6.1 Experiment Setup

Datasets. Two datasets, i.e., CIFAR-10 [15] and GTSRB [27], are used to evaluate the performance of our proposed GradCAM-AE.

- CIFAR-10: It consists of 60000 32×32 low-resolution color images in 10 different classes, with 6000 images per class. There are 50,000 training images allocated to all benign devices for local model training, and 10,000 test images are allocated to the server for the global model testing until the end of each communication round.
- GTSRB: It contains 43 different classes of German traffic signs. The images range in size from 15×15 pixels to 250×250 pixels. There are 39209 training images, and 12630 test images are assigned to all local devices for training and the server for testing. Notice that we resize the images of different sizes to a fixed resolution of 128×128 pixels for training and testing.

Benchmarks. We consider the following state-of-the-art defense schemes, i.e., Multi-Krum [1], AUROR [25], Trimmed-mean [33], FAA-DL [26], FLTrust [2], FLCert [4], and FLDetector [36]. Specifically, FLTrust utilizes cosine similarity to measure the alignment of local model updates with the trusted global model update trained by a clean dataset. The local model updates with low similarity are excluded. FLCert divides devices into multiple groups and trains a global model for each group using a standard FL algorithm. FLDetector uses K-means to detect malicious devices by checking the consistency between the received and predicted model updates of the devices. In addition, we study a GradCAM-Krum anomaly detection scheme in which Krum is applied to identify the malicious GradCAM heat maps that are generated at the server.

Experimental setting. The total communication rounds $T_{FL} = 100$. For each communication round, 18 benign devices train the local models for 20 epochs using the Adam optimizer, with a batch size of 64, a learning rate of 0.0001, and a weight decay of 0. The autoencoder trains the GradCAM heat maps for 200 epochs using the Adam optimizer, with a hidden-layer size of 128, a learning rate of 0.001, and a weight decay of 0.00001. In the voting process, the communication round interval is $\xi = 3$, and the threshold is $\epsilon = 2$.

Algorithm 1: The proposed GradCAM-AE for detecting malicious local model updates on FL

input : $\mathcal{K}$ - the set of benign devices, $\mathcal{M}$ - the set of attackers, T_{FL} the total number of communication round, L - the number of local epochs, $\mathcal{D}_k$ - data size of device, η_t - the learning rate of devices, $\mathbf{W}^0$ - the initialized global model, $\mathbf{M}$ - buffer with size $\xi \times (K + K')$, ξ - the interval of communication rounds, ϵ - voting threshold, I_0 - an image picked form testing dataset.

output: The final global model weight $\mathbf{W}^{T_{FL}}$

for $t = 1, \cdots, T_{FL}$ **do**
    Server broadcasts $\mathbf{W}^t$ to all devices;
    $\mathbf{M} \leftarrow \mathbf{0}, \mathbf{M} \in \mathbf{R}^{\xi \times (|K+K'|)}$;
    for $k \in \mathcal{K}$ **do**
        $\mathbf{W}_k^t \leftarrow \underline{LocalUpdate(k, \mathbf{W}^t)}$;
    Solve (9a) with constraint (9b);
    $\left(\mathbf{W}_{1'}^t\right)^*, \cdots, \left(\mathbf{W}_{K'}^t\right)^* \leftarrow$ (10).
    if t *modulo* $\xi \neq 0$ **then**
        $A_l \leftarrow \mathbf{W}_l^{t+1}$ and I_0 perform convolution operation ;
        $L_{l,GradCAM} \leftarrow$ according to (11) and (12) ;
        $\mathbf{L}_{\mathbf{GradCAM}} \leftarrow$ concatenate $(L_{l,GradCAM})$;
        $\mathbf{L}'_{\mathbf{GradCAM}} \leftarrow$ Autoencoder$(\mathbf{L}_{\mathbf{GradCAM}})$;
        $O_l^t \leftarrow$ (17) ;
        Store O_l^t into $\mathbf{M}$;
    else
        $A_l \leftarrow \mathbf{W}_l^{t+1}$ and I_0 perform convolution operation ;
        $L_{l,GradCAM} \leftarrow$ according to (11) and (12) ;
        $\mathbf{L}_{\mathbf{GradCAM}} \leftarrow$ concatenate $(L_{l,GradCAM})$;
        $\mathbf{L}'_{\mathbf{GradCAM}} \leftarrow$ Autoencoder$(\mathbf{L}_{\mathbf{GradCAM}})$;
        $O_l^t \leftarrow$ (17) ;
        Store O_l^t into $\mathbf{M}$;
        Stat $\leftarrow sum(\mathbf{M}, axis = 0)$;
        $Y_l^t \leftarrow$ (19)
    Update global model according to (18) and (19).

$\underline{LocalUpdate(k, \mathbf{W}^t)}$;
for $\ell = 1, \cdots, L$ **do**
    for $b_k \in \mathcal{D}_k$ **do**
        (5).

6.2 Performance under IID Datasets

Fig. 4 shows the comparison of FL test accuracy under our proposed GradCAM-AE defense framework with existing benchmarks, where the DNN model of ResNet-18 performs image classification on CIFAR-10 and GTSRB with IID settings, respectively. GradCAM-AE can achieve the highest test accuracy of the global model as the autoencoder can accurately exclude the abnormal GradCAM heat maps such that more benign local model updates are involved in the FL training process. While GradCAM-Krum only selects an optimal local model update as the global model to train FL, it also achieves higher test accuracy (0.79) compared with other benchmarks. The reason is that the data is IID, and the difference among the local model updates is not significant. Multi-Krum, AUROR and Trimmed-mean all

Table 1. Detection rates of ResNet-18 on IID and Non-IID CIFAR-10.

ResNet-18	IID CIFAR-10					Non-IID CIFAR-10				
Methods	Recall	Precision	ACC	F1 score	AUC	Recall	Precision	ACC	F1 score	AUC
Multi-Krum	0.045	0.09	0.859	0.06	0.497	0.085	0.085	0.817	0.085	0.492
AUROR	0.015	0.006	0.736	0.009	0.416	0.035	0.017	0.75	0.023	0.432
Trimmed-mean	0.08	0.08	0.816	0.08	0.489	0.09	0.09	0.818	0.09	0.494
FAA-DL	0.61	0.098	0.412	0.168	0.50	0.55	0.109	0.514	0.182	0.53
GradCAM-Krum	1.0	1.0	1.0	1.0	1.0	0.625	0.625	0.925	0.625	0.792
GradCAM-AE	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.943**	**0.986**	**0.961**	**0.992**

directly detect millions, even many more, parameters of local model updates based on Euclidean distance. The Euclidean distance between the crafted malicious local model updates and benign ones is within the threshold designated by the server. This indicates that the malicious local model updates can elude the detection of the server and participate in the FL training process through multiple communication rounds, resulting in the global model being corrupted. Table 1 shows that although the ACCs of Euclidean distance-based defense schemes are relatively high (0.859, 0.736 and 0.816), the Precision is close to zero. In other words, they can correctly classify a few benign devices but fail to identify malicious user devices. Similar performances are observed on GTSRB dataset, the detection rates of GrdaCAM-Krum still outperform the benchmarks, as shown in Table 2. The detailed detection rates of MobileNetV3-Large on CIFAR-10 and GTSRB with IID settings, are given in Table 3 and Table 4, respectively.

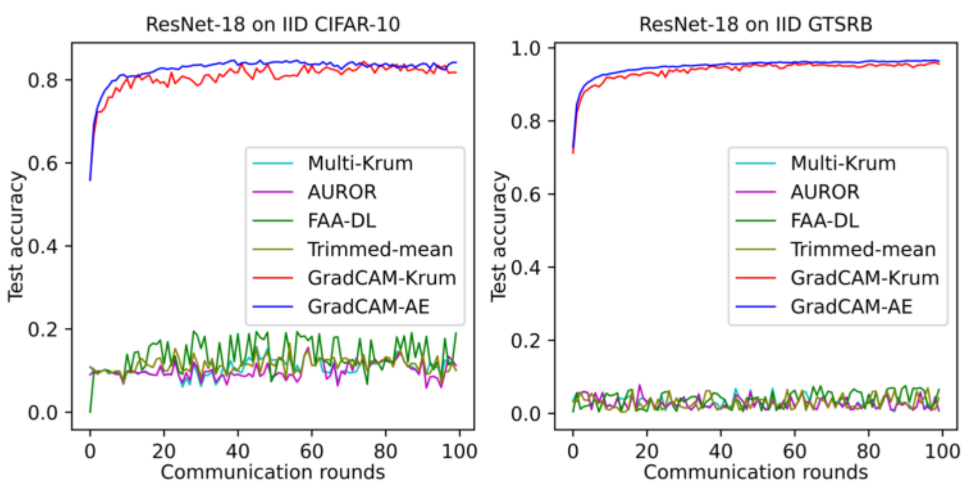

Fig. 4. Test accuracy vs. communication rounds for **ResNet-18** on IID CIFAR-10 and GTSRB.

FAA-DL, a one-class SVM-based method, also directly classifies the local model updates aggregated by the server as benign and malicious. However, there are two key reasons why FAA-DL fails in the experiments. First, the local model updates have the characteristics of high-dimensional feature spaces. The curse of dimensionality can lead to data sparsity, making it challenging for FAA-DL to find a suitable margin that separates benign from malicious local model updates. Second, FAA-DL is sensitive to class imbalance (18 benign local model updates and 2 malicious local

Table 2. Detection rates of ResNet-18 on IID and Non-IID GTSRB.

ResNet-18	IID GTSRB					Non-IID GTSRB				
Methods	Recall	Precision	ACC	F1 score	AUC	Recall	Precision	ACC	F1 score	AUC
Multi-Krum	0.045	0.09	0.859	0.06	0.497	0.045	0.08	0.857	0.06	0.497
AUROR	0.035	0.016	0.739	0.02	0.427	0.035	0.016	0.755	0.0229	0.435
Trimmed-mean	0.08	0.08	0.816	0.08	0.489	0.07	0.07	0.814	0.07	0.483
FAA-DL	0.685	0.092	0.295	0.162	0.468	0.43	0.093	0.517	0.152	0.478
GradCAM-Krum	0.625	0.625	0.925	0.625	0.792	0.55	0.55	0.91	0.55	0.75
GradCAM-AE	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.990**	**0.998**	**0.994**	**0.999**

Table 3. Detection rates of MobileNetV3-Large on IID and Non-IID CIFAR-10.

MobileNetV3-Large	IID CIFAR-10					Non-IID CIFAR-10				
Methods	Recall	Precision	ACC	F1 score	AUC	Recall	Precision	ACC	F1 score	AUC
Multi-Krum	0.05	0.1	0.860	0.06	0.5	0.035	0.07	0.857	0.047	0.492
AUROR	0.02	0.0095	0.74	0.013	0.42	0.015	0.006	0.736	0.009	0.416
Trimmed-mean	0.1	0.1	0.82	0.1	0.50	0.50	0.08	0.816	0.08	0.489
FAA-DL	0.735	0.092	0.257	0.163	0.469	0.7	0.094	0.309	0.166	0.483
GradCAM-Krum	0.86	0.86	0.972	0.86	0.922	0.94	0.94	0.988	0.94	0.967
GradCAM-AE	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 4. Detection rates of MobileNetV3-Large on IID and Non-IID GTSRB.

ResNet-18	IID GTSRB					Non-IID GTSRB				
Methods	Recall	Precision	ACC	F1 score	AUC	Recall	Precision	ACC	F1 score	AUC
Multi-Krum	0.03	0.06	0.856	0.04	0.489	0.135	0.135	0.827	0.135	0.519
AUROR	0.035	0.019	0.745	0.02	0.43	0.015	0.0083	0.7375	0.011	0.416
Trimmed-mean	0.08	0.08	0.816	0.08	0.489	0.065	0.065	0.813	0.065	0.481
FAA-DL	0.79	0.096	0.236	0.17	0.483	0.765	0.098	0.28	0.173	0.496
GradCAM-Krum	0.995	0.995	0.999	0.995	0.997	0.58	0.58	0.916	0.58	0.767
GradCAM-AE	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**0.990**	**0.998**	**0.994**	**0.998**

model updates), which means the FAA-DL biases its decision boundary towards the majority class, making it struggle to detect malicious local model updates effectively. Notwithstanding, we replace the DNN model with MobileNetV3-Large. Similar performances are observed; see Fig. 5. This demonstrates that GradCAM-AE is applicable to different DNN models for detecting malicious local model updates.

Table 5 shows the detection rates of GradCAM-AE, where the poisoning ratio of the local model (the proportion of local model parameters modified over the total number of parameters) increases from 0.2 to 0.5. Recall, Precision, ACC, F1 socre and AUC of GradCAM-AE grows from 0.955 to 1.0, 0.960 to 1.0, 0.992 to 1.0, 0.953 to 1.0, and 0.976 to 1.0, respectively, while the false positive rate (FPR) drops from 0.003 to 0. The decrease in FPR indicates that the number of benign devices mistakenly classified as malicious devices by GradCAM-AE is reduced. This is because the more the malicious local models are modified, the more it deviates from the benign local models, which makes it easier for GradCAM-AE to detect the malicious local model. When the poisoning ratio increases to 0.8, the detection rates of GradCAM-AE maintain to 1.0, as all the malicious local models have been detected by GradCAM-AE.

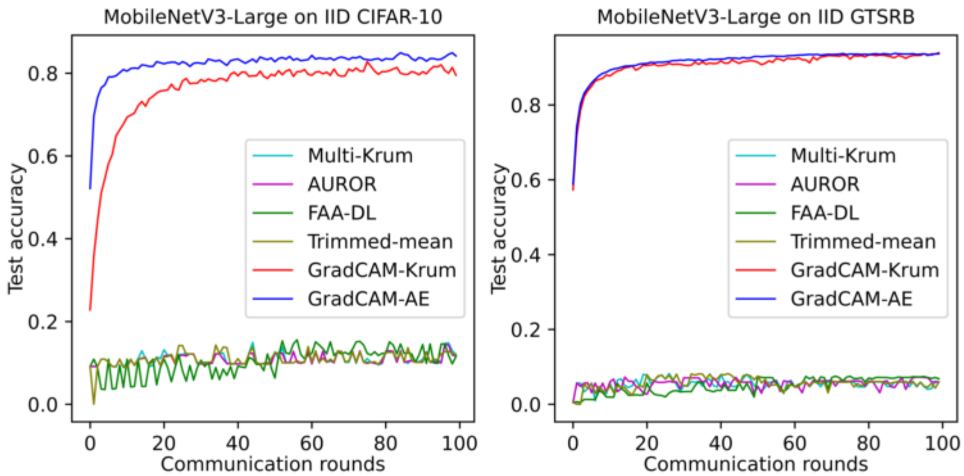

Fig. 5. Test accuracy vs. communication rounds for **MobileNetV3-Large** on IID CIFAR-10 and GTSRB.

Table 5. The detection rate of GradCAM-AE with regards to the poisoning ratios.

Poisoning ratios	Recall	Precision	FPR	ACC	F1 score	AUC
0.2	0.955	0.960	0.003	0.992	0.953	0.976
0.5	1.0	1.0	0.0	1.0	1.0	1.0
0.8	1.0	1.0	0.0	1.0	1.0	1.0

6.3 Performance under non-IID Datasets

Compared with IID, the Non-IID setting is much more challenging for the server to identify the malicious local model updates. This reason is that the local model updates tend to be more divergent under non-IID settings; in turn, the generated heat maps on the server are more diverse. To simulate the statistical data heterogeneity in practical FL scenarios, we adopt one of the Non-IID partition schemes, i.e., latent Dirichlet allocation (LDA), which are widely used in the latest literature [29, 35]. As a biased probability, the β parameter of LDA controls the distribution difference of the local training data. A larger β denotes a higher level of Non-IID among the local training data. Here, we set $\beta = 0.5$.

Fig. 6 illustrates the test accuracy of ResNet-18 in non-IID CIFAR-10 and non-IID GTSRB. Under the Non-IID CIFR10 setting, the test accuracy of GradCAM-AE is much higher than that of GradCAM-Krum. GradCAM-AE can quickly converge (around the 30th communication round) as it involves more benign devices in FL training. This indicates that GradCAM-AE can accurately filter malicious model updates, as can also be confirmed by the detection indicators in Table 1. Although GradCAM-Krum can restrict malicious devices from participating in FL, it sacrifices accuracy and robustness, especially under non-IID CIFAR-10. The test accuracy of GradCAM-Krum fluctuates significantly and likely diverges, as it only selects one local model update as the global model. The more divergent the model updates, the more diverse the GradCAM heat maps. GradCAM-Krum struggles to screen malicious model updates, which coincides with the precision of 0.625 for GradCAM-Krum on Non-IID CIFAR-10 in Table 1. The remaining defense methods still undergo poor anomaly detection as on the IID dataset. The detection rates of MobileNetV3-Large on CIFAR-10 and GTSRB under Non-IID settings are given in Table 3 and Table 4, respectively..

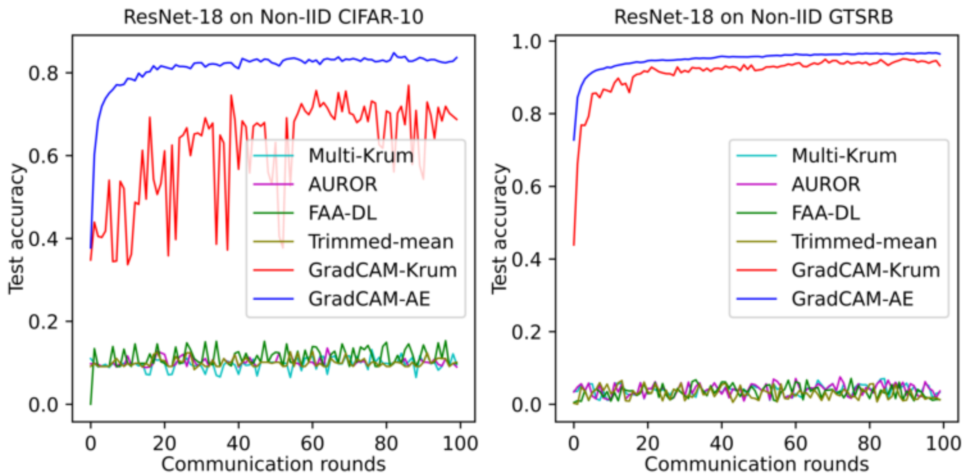

Fig. 6. Test accuracy vs. communication rounds for **ResNet-18** on Non-IID CIFAR-10 and GTSRB.

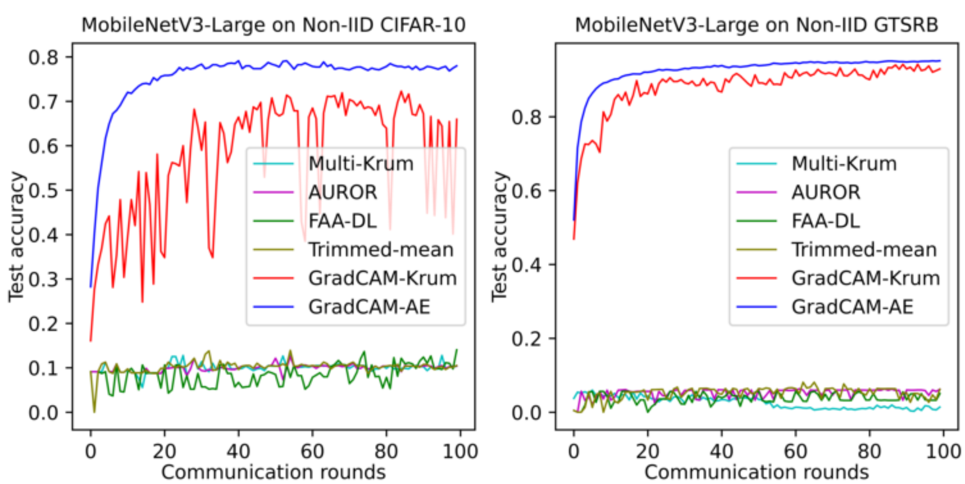

Fig. 7. Test accuracy vs. communication rounds for **MobileNetV3-Large** on Non-IID CIFAR-10 and GTSRB.

We replace the ResNet-18 model with MobileNetV3-Large, and the changing trend of the test accuracy with the communication round is consistent with Fig. 6, as shown in Fig. 7. This means that the complexity of the DNN model itself has a limited impact on the malicious detection of GradCAM-Krum, while the complexity of local data plays a crucial role. GTSRB has fewer image features than CIFAR-10, and the local model updates learned by FL have less diversity, making GradCAM-Krum converge.

This significant difference is quantified in Fig. 8, where the server averages the reconstruction errors of 20 user devices in 100 communication rounds one by one. For the purpose of visualization, Fig. 9 only shows the original GradCAM heat maps and the reconstructed heat maps by the autoencoder under ResNet-18 on Non-IID CIFAR-10

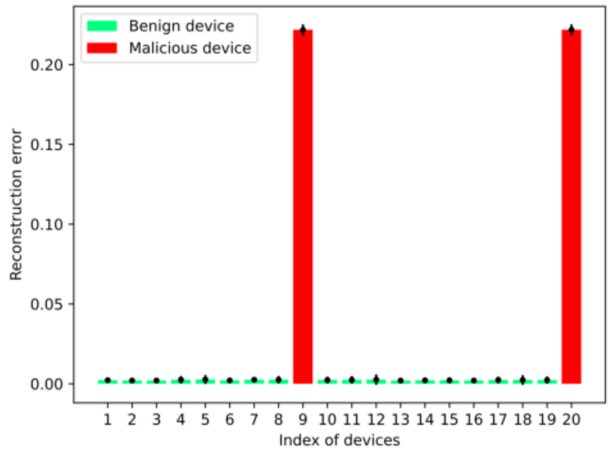

Fig. 8. Reconstruction errors for ResNet18 on Non-IID CIFAR-10.

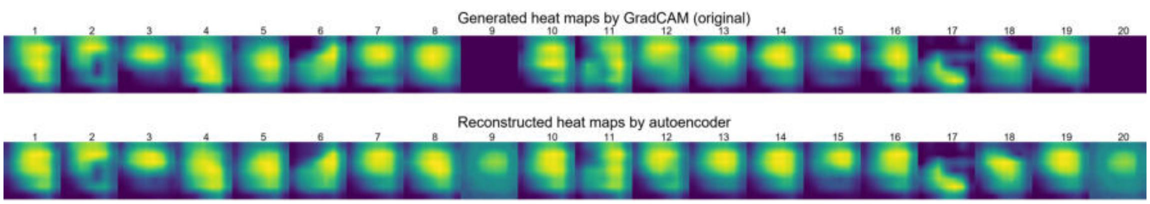

Fig. 9. Comparison between generated heat maps by GradCAM and reconstructed heat maps by the autoencoder.

setting in one communication round. Strikingly, it can be noticed that the reconstruction errors of device 9 and device 20 are obviously larger than those of other user devices. In particular, we use the generated heat maps to train the autoencoder, and after the training, the final reconstructed heat maps are used to compare with the originally generated heat maps, the heat maps with high reconstruction errors are considered abnormal. We conduct 100 communication rounds for FL training with 20 devices, and in each communication round, autoencoder performs 200 epochs for training. Thus, we use $200 \times 20 = 4000$ heat maps to train the autoencoder in each communication round, which does not result in overfitting.

Moreover, Fig. 9 shows that the reconstructed heat maps of malicious devices are significantly different from the heat maps reconstructed by benign devices. This is because the autoencoder captures the key features of normal heat maps during the encoding process. Anomalous heat maps with different patterns, shapes, or structures result in latent representations that cannot be decoded accurately. The abnormal heat maps often fall outside the distribution of normal heat maps in the latent space, making reconstruction difficult. In particular, training an effective autoencoder involves meeting specific criteria that ensure the model effectively learns a meaningful latent representation and reconstructs the input heat maps accurately. The primary criterion for training a high-quality autoencoder is achieving a reconstruction error on the training heat maps that is less than the mean reconstruction error plus α times the standard deviation of all reconstruction errors.

Fig. 10 compares GradCAM-AE, FLTrust [2], FLCert [4], and FLDetector [36]. In terms of the test accuracy of the global model, GradCAM-AE is 0.7 better than FLTrust defense. The reason is that the malicious local models generated by attackers are highly correlated with benign local models that can evade FLTrust but can be detected by GradCAM-AE. GradCAM-AE can map local models into visible heat maps, which are remapped by the autoencoder to highlight the hidden features. The utilization of autoencoder improves the distinguishability of the heat maps and the success rate of identifying anomalous heat maps and malicious local models. Likewise, FLCert inevitably involves malicious devices in FL training, leading to the corruption of the global model. FLDetector can prevent attackers from participating in FL training, there are still benign devices who are misclassified as malicious. FLDetector uses K-means for defense, nevertheless, it still inevitably involves malicious devices in FL training. The test accuracy of the global model under FLDetector defense is 0.68 lower than the that under our proposed GradCAM-AE defense.

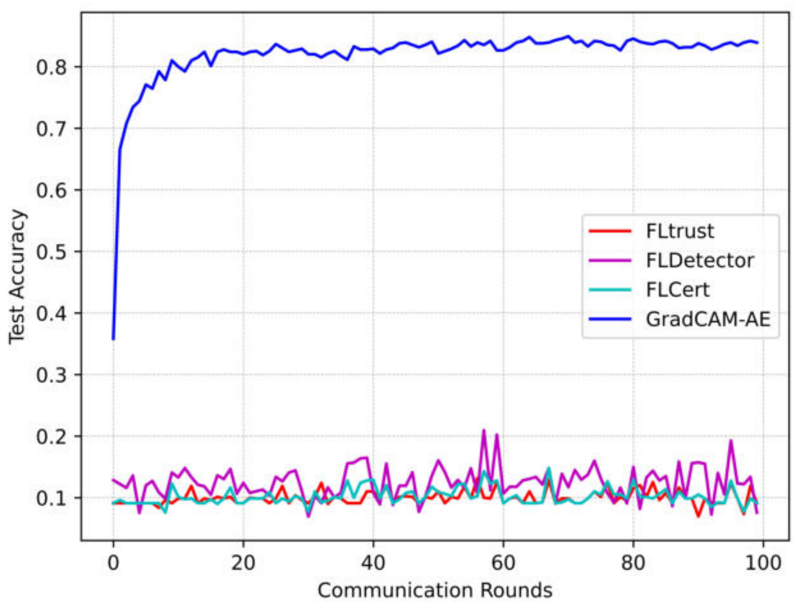

Fig. 10. Comparison of global model test accuracy between GradCAM-AE and benchmarks.

Fig. 11 shows the test accuracy of the global model, where β is set to 0.1, 0.5, or 0.9. Generally, the test accuracy of the global model increases with the growth of β. Given $\beta = 0.1$, the low detection rates of GradCAM-AE for malicious devices result in the low test accuracy of the global model, which is 0.6. When β increases from 0.5 to 0.9, the divergence degree of the local model updates is reduced, and the detection rates of GradCAM-AE for malicious devices increase, leading to the global model's test accuracy increasing from 0.83 to 0.86.

To evaluate the scalability of GradCAM-AE, we define the attack rate as the ratio of the number of attackers to the total number of local devices, i.e., $\mathrm{Att} = \frac{\#\mathrm{attackers}}{\#\mathrm{all\ devices}}$. GradCAM-AE and GradCAM-Krum are compared under Non-IID CIFAR-10 with 2 and 8 attackers. The total number of local devices is 20, i.e., Att = 0.1 and Att = 0.4. Fig. 12 shows that as the attack rate increases, GradCAM-Krum fails to identify attackers. The maximum test accuracy gap of 0.652 appears

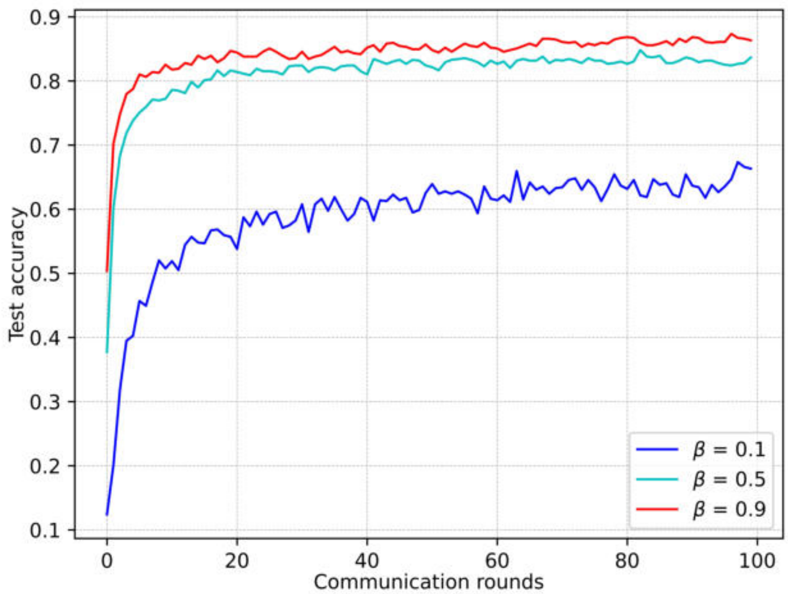

Fig. 11. The test accuracy of FL with GradCAM-AE, where β is set to 0.1, 0.5 or 0.9.

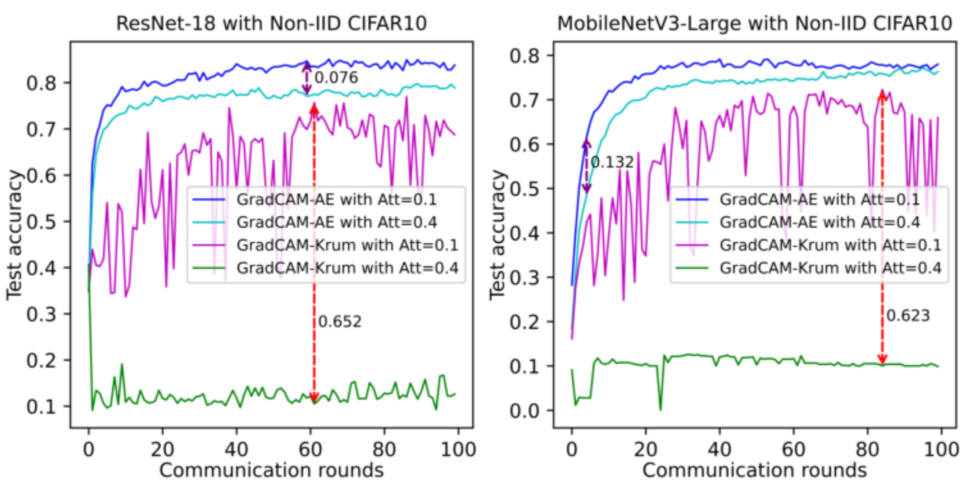

Fig. 12. Scalability of GradCAM-AE vs. GradCAM-Krum

in the 61st communication round. In contrast, the test accuracy of GradCAM-AE changes, and the maximum gap of 0.076 appears in the 61st communication round. Similarly, in the case of MobileNetV3-Large, the tendency of test accuracy is consistent, offering scalability and robustness. Fig. 13 shows detection rates of ResNet-18 and MobileNetV3-Large on

Non-IID CIFAR-10 with Att = 0.4, respectively. The results indicate that the detection rates of GradCAM-AE outperform GradCAM-Krum no matter the number of attackers, offering scalability and robustness.

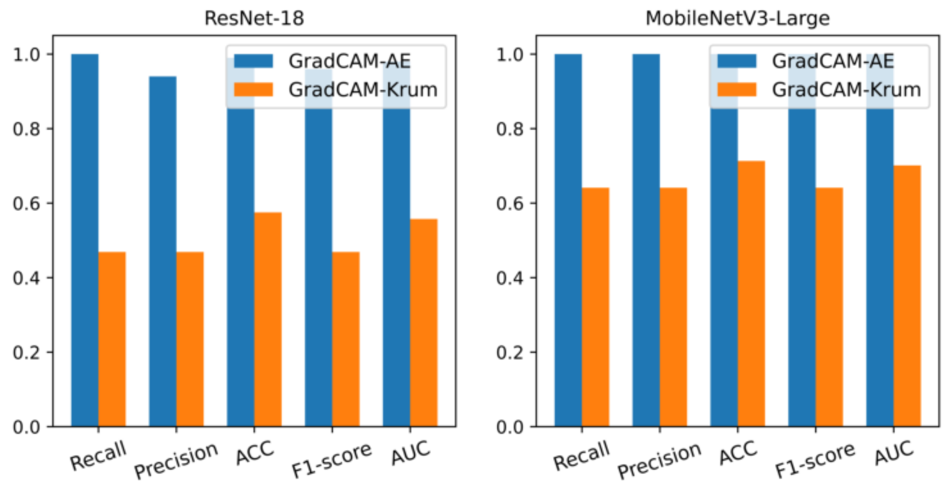

Fig. 13. Detection rates of ResNet-18 and MobileNetV3-Large on Non-IID CIFAR-10 with Att = 0.4

Fig. 14 demonstrates that FL maintains a high test accuracy of 0.83 and 0.8 under the proposed GradCAM-AE, when the number of devices is set to 20 (18 benign devices and 2 malicious devices) and 100, respectively. The decrease in test accuracy of global model is because given the total dataset size remains same, assigning the dataset to more devices reduces the amount of data per device under non-IID (i.e., different devices have data from different distributions) scenario, smaller local datasets exacerbate this heterogeneity, which causes the detection rates of GradCAM-AE on 100 devices in detecting malicious local models to decrease.

Table 6 presents the detection rates of GradCAM-AE under MPAF [3] and Fang attack [7]. In particular, the MPAF [3] generates malicious local models that are aligned with an attacker-specified base model. These malicious local models are scaled up before being transmitted to the server, thereby amplifying their adversarial impact and reducing the test accuracy of the global model. Given MPFA, the Recall, Precision, ACC, F1 score and AUC of GradCAM-AE are respectively. Moreover, our proposed GradCAM-AE is applicable to Fang attack in [7]. As shown in Table 6, all the malicious local models of Fang attack are detected by GradCAM-AE. This is because Fang attack alters the direction of the local model updates. As a result, generated heat maps based on the malicious local models are different from the benign ones, which are identified as malicious one.

Table 6. Detection rates of GradCAM-AE under MPFA and Fang attack.

References	Recall	Precision	FPR	ACC	F1 score	AUC
MPAF [3]	0.985	0.990	0.002	0.997	0.984	0.992
Fang attack [7]	1.0	1.0	0.0	1.0	1.0	1.0

Runtime. As shown in Fig. 15, the average runtime of our proposed GradCAM-AE (35.71 s) is higher than that of Euclidean distance-based defense methods, i.e., Multi-Krum (29.68 s) and Trimmed-mean (32.89 s). This is because the

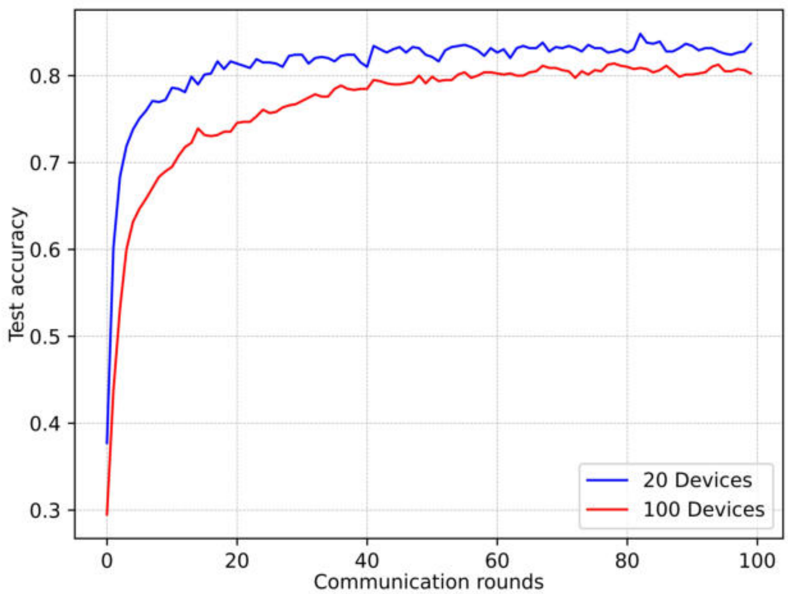

Fig. 14. The test accuracy of FL global model under the defense of GradCAM-AE with 20devices vs 100 devices train ResNet-18 on Non-IID CIFAR-10 dataset.

Euclidean distance-based defense does not require training, while GradCAM-AE takes a little longer time to train the autoencoder. Nevertheless GradCAM-AE is still faster than that of the machine learning-based AUROR (35.97 s) and FAA-DL (40.53 s). It is noting that although runtime of GradCAM-AE is not the fastest, detection rates of GradCAM-AE are the highest.

6.4 Discussion

Untargeted model poisoning attacks are generally easier to detect compared to semantic targeted model poisoning or backdoor attacks, as the former often leads to significant decreases in global model accuracy and can result in a denial-of-service situation. In the paper, our primary focus is on defending against untargeted model poisoning attacks due to their disruptive nature and potential for large-scale impact. However, our proposed GradCAM-AE has the potential to be extended to defend against more advanced attacks, such as semantic targeted model poisoning [19], backdoor attacks, and even byzantine-robust [7] or distributed attacks [32]. At a high level, both untargeted and targeted poisoning attacks share the goal of modifying the behavior of the global model, either overtly or covertly. While untargeted model poisoning attacks aim to disrupt model performance broadly, targeted and backdoor attacks focus on embedding specific malicious objectives into the model without compromising overall performance. Specifically, our method can adapt to the detection of these advanced attacks as follows:

Semantic Targeted and Backdoor Attacks: These attacks often introduce subtle, targeted changes in the local model updates that are harder to detect. GradCAM-AE, which leverages heat map generation via GradCAM and anomaly

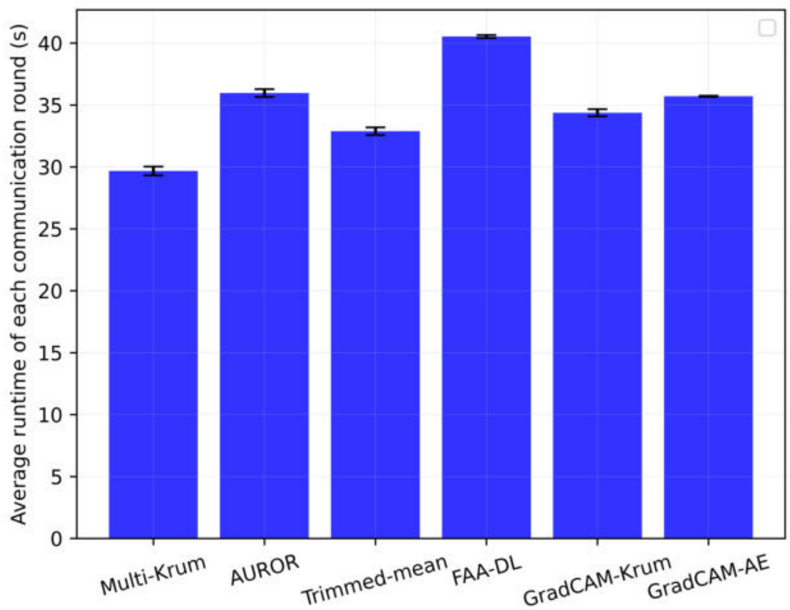

Fig. 15. Comparison of GradCAM-AE's average runtime with other benchmarks.

detection using an autoencoder, is capable of identifying discrepancies in the activation patterns of malicious local updates. Even in scenarios where backdoor triggers are embedded, the heat maps produced by malicious devices will deviate significantly from those of benign devices due to the targeted intent, allowing our autoencoder to flag such updates as anomalous. For instance, the ability to capture nuanced, target-specific activations makes GradCAM-AE a promising approach for these attack types.

Byzantine-Robust Attacks: In these attacks, adversaries aim to manipulate the direction of local model updates to mislead the aggregation process without completely disrupting the global model. GradCAM-AE can still identify such manipulations because malicious updates often result in distinct heat maps that differ from those generated by benign devices. These discrepancies can be detected by the autoencoder as anomalies, highlighting the adaptability of our method to various threat models.

Distributed Attacks: In distributed poisoning scenarios, where multiple adversarial devices coordinate to evade detection, our approach remains effective due to its reliance on per-device heat map analysis. The autoencoder is designed to detect patterns in these heat maps, which allows it to isolate anomalous contributions even in coordinated attack settings.

In future work, we will extend our focus to thoroughly evaluate GradCAM-AE against semantic targeted attacks, backdoor attacks, and other advanced poisoning techniques, ensuring robustness against a wide range of adversarial behaviors.

7 Conclusion

In this paper, we proposed GradCAM-AE, a novel and robust shield defense against poisoning attacks on FL. GradCAM was leveraged to process the received local model updates with a selected image from the test dataset, generating the corresponding GradCAM heat maps. An autoencoder was incorporated to accentuate the hidden features of the GradCAM heat maps. It was demonstrated experimentally that GradCAM-AE significantly outperforms the cutting-edge defense schemes under all settings tested. In the future, we will investigate defense against multimodal FL attacks.

Acknowledgments

This Paper was supported by the CISTER Research Unit (UIDP/UIDB/04234/2020), financed by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), by project ADANET (PTDC/EEICOM/3362/2021), and by project Aero.Next Portugal (ref. C645727867-00000066), funded by the EU/Next Generation, within call n.o 02/C05-i01/2022 of the Recovery and Resilience Plan (RRP); and also supported in part by the AXA Research Fund (AXA Chair for Internet of Everything at Koç University).

References

[1] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 118–128.

[2] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2021. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society.

[3] Xiaoyu Cao and Neil Zhenqiang Gong. 2022. MPAF: Model Poisoning Attacks to Federated Learning Based on Fake Clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 3396–3404.

[4] Xiaoyu Cao, Zaixi Zhang, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. Flcert: Provably secure federated learning against poisoning attacks. *IEEE Transactions on Information Forensics and Security* 17 (2022), 3691–3705.

[5] Zhenzhu Chen, Anmin Fu, Yinghui Zhang, Zhe Liu, Fanjian Zeng, and Robert H. Deng. 2021. Secure Collaborative Deep Learning Against GAN Attacks in the Internet of Things. *IEEE Internet of Things Journal* 8, 7 (2021), 5839–5849. doi:10.1109/JIOT.2020.3033171

[6] Jun Kang Chow, Zhaoyu Su, Jimmy Wu, Pin Siang Tan, Xin Mao, and Yu-Hsing Wang. 2020. Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Advanced Engineering Informatics* 45 (2020), 101105.

[7] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*. 1623–1640.

[8] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. 2022. ADBench: Anomaly Detection Benchmark. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 32142–32159. https://proceedings.neurips.cc/paper_files/paper/2022/file/cf93972b116ca5268827d575f2cc226b-Paper-Datasets_and_Benchmarks.pdf

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. 2019. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[11] Yuke Hu, Wei Liang, Ruofan Wu, Kai Xiao, Weiqiang Wang, Xiaochen Li, Jinfei Liu, and Zhan Qin. 2023. Quantifying and Defending against Privacy Threats on Federated Knowledge Graph Embedding. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 2306–2317. doi:10.1145/3543507.3583450

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[13] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *CoRR* abs/1602.07360 (2016). arXiv:1602.07360 http://arxiv.org/abs/1602.07360

[14] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima Jr, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, et al. 2021. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence* 3, 6 (2021), 473–484.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[16] Kai Li, Jingjing Zheng, Xin Yuan, Wei Ni, Ozgur B Akan, and H Vincent Poor. 2024. Data-Agnostic Model Poisoning against Federated Learning: A Graph Autoencoder Approach. *IEEE Transactions on Information Forensics and Security* (2024).

[17] Wenxin Liu, Hui Lin, Xiaoding Wang, Jia Hu, Georges Kaddoum, Md. Jalil Piran, and Atif Alamri. 2023. D2MIF: A Malicious Model Detection Mechanism for Federated-Learning-Empowered Artificial Intelligence of Things. *IEEE Internet of Things Journal* 10, 3 (2023), 2141–2151. doi:10.1109/JIOT.2021.3081606

[18] Hidde Lycklama, Lukas Burkhalter, Alexander Viand, Nicolas Küchler, and Anwar Hithnawi. 2023. Rofl: Robustness of secure federated learning. In *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 453–476.

[19] Xiaoting Lyu, Yufei Han, Wei Wang, Jingkai Liu, Bin Wang, Jiqiang Liu, and Xiangliang Zhang. 2023. Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 9020–9028.

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJzIBfZAb

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

[22] Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al. 2022. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*. 1415–1432.

[23] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[24] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.

[25] Shiqi Shen, Shruti Tople, and Prateek Saxena. 2016. Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*. 508–519.

[26] Siping Shi, Chuang Hu, Dan Wang, Yifei Zhu, and Zhu Han. 2022. Federated Anomaly Analytics for Local Model Poisoning Attack. *IEEE Journal on Selected Areas in Communications* 40, 2 (2022), 596–610. doi:10.1109/JSAC.2021.3118347

[27] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks* 32 (2012), 323–332. doi:10.1016/j.neunet.2012.02.016 Selected Papers from IJCNN 2011.

[28] Marek Wadinger and Michal Kvasnica. 2024. Adaptable and Interpretable Framework for Anomaly Detection in SCADA-based industrial systems. *Expert Systems with Applications* 246 (2024), 123200.

[29] Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. 2022. Federated Unlearning via Class-Discriminative Pruning. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 622–632. doi:10.1145/3485447.3512222

[30] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K. Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221. doi:10.1109/JSAC.2019.2904348

[31] Peter H Westfall and Kevin SS Henning. 2013. *Understanding advanced statistical methods*. Vol. 543. CRC Press Boca Raton, FL.

[32] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2020. DBA: Distributed backdoor attacks against federated learning. In *International conference on learning representations*.

[33] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*. PMLR, 5650–5659.

[34] Kai Yue, Richeng Jin, Chau-Wai Wong, Dror Baron, and Huaiyu Dai. 2023. Gradient obfuscation gives a false sense of security in federated learning. In *32nd USENIX Security Symposium (USENIX Security 23)*. 6381–6398.

[35] Yanjun Zhang, Guangdong Bai, Mahawaga Arachchige Pathum Chamikara, Mengyao Ma, Liyue Shen, Jingwei Wang, Surya Nepal, Minhui Xue, Long Wang, and Joseph Liu. 2023. AgrEvader: Poisoning Membership Inference against Byzantine-Robust Federated Learning. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 2371–2382. doi:10.1145/3543507.3583542

[36] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. 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*. 2545–2555.

[37] Zhiyuan Zhang, Qi Su, and Xu Sun. 2022. Dim-Krum: Backdoor-Resistant Federated Learning for NLP with Dimension-wise Krum-Based Aggregation. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, 339–354. doi:10.18653/v1/2022.findings-emnlp.25

[38] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. 2017. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*. 1933–1941.

[39] Yue Zhao, Ryan Rossi, and Leman Akoglu. 2021. Automatic Unsupervised Outlier Model Selection. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 4489–4502. https://proceedings.neurips.cc/paper_files/paper/2021/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf

[40] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.

[41] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 665–674.

[42] Hao Zhou, Geng Yang, Hua Dai, and Guoxiu Liu. 2022. PFLF: Privacy-Preserving Federated Learning Framework for Edge Computing. *IEEE Transactions on Information Forensics and Security* 17 (2022), 1905–1918. doi:10.1109/TIFS.2022.3174394

[43] Tengteng Zhu, Zehua Guo, Chao Yao, Jiaxin Tan, Songshi Dou, Wenrun Wang, and Zhenzhen Han. 2024. Byzantine-robust Federated Learning via Cosine Similarity Aggregation. *Computer Networks* 254 (2024), 110730.

[44] Haomin Zhuang, Yihua Zhang, and Sijia Liu. 2023. A Pilot Study of Query-Free Adversarial Attack Against Stable Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2384–2391.