Detecting Poisoning Attacks on Federated Learning Using Gradient-Weighted Class Activation Mapping

Jingjing Zheng
zheng@isep.ipp.pt
CISTER Research Centre, ISEP
Porto, Portugal

Kai Li*
kaili@ieee.org
University of Cambridge
Cambridge, United Kingdom

Xin Yuan
xin.yuan@data61.csiro.au
CSIRO
Sydney, Australia

Wei Ni
wei.ni@data61.csiro.au
CSIRO
Sydney, Australia

Eduardo Tovar
emt@isep.ipp.pt
CISTER Research Centre, ISEP
Porto, Portugal

ABSTRACT

This paper proposes a new defense mechanism, namely, GCAMA, against model poisoning attacks on Federated learning (FL), which integrates **G**radient-weighted **C**lass **A**ctivation **M**apping (GradCAM) and **A**utoencoder to offer a scientifically more powerful detection capability compared to existing Euclidean distance-based approaches. Particularly, GCAMA generates a heat map for each uploaded local model update, transforming each local model update into a lower-dimensional, visual representation, thereby accentuating the hidden features of the heat maps and increasing the success rate of identifying anomalous heat maps and malicious local models. We test ResNet-18 and MobileNetV3-Large deep learning models with CIFAR-10 and GTSRB datasets under Non-Independent and Identically Distributed (Non-IID) setting, respectively. The results demonstrate that GCAMA offers superior test accuracy of FL global model compared to the state-of-the-art methods. Our code is available at: https://github.com/jjzgeeks/GradCAM-AE

CCS CONCEPTS

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

KEYWORDS

Federated learning, poisoning attacks, gradient-weighted class activation mapping, autoencoder

ACM Reference Format:
Jingjing Zheng, Kai Li, Xin Yuan, Wei Ni, and Eduardo Tovar. 2024. Detecting Poisoning Attacks on Federated Learning Using Gradient-Weighted Class Activation Mapping. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion), May 13–17, 2024, Singapore, Singapore*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3589335.3651490

*Kai Li is the corresponding author.

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0172-6/24/05
https://doi.org/10.1145/3589335.3651490

1 INTRODUCTION

Federated learning (FL), as a decentralized machine learning approach, enables multiple user devices to cooperatively train a shared model under the orchestration of a server without sharing their local data, 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 upload model updates to a server for aggregation. In response, the server amalgamates local model updates to generate a common global model that is then distributed to the devices for updating their respective local models [7, 14, 15]. Such a communication round repeats until the model achieves a satisfactory accuracy level.

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 [8, 9]. Existing countermeasures, leveraging Euclidean distance-based metrics to discern deviations between malicious and benign models, have demonstrated efficacy against such attacks [1, 13]. 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 GCAMA, against model poisoning attacks on FL.GCAMA leverages a Gradient-weighted Class Activation Mapping (GradCAM [10])-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 heatmap 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 utilized to reproject 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.

We conducted a comprehensive assessment of the proposed GCAMA framework using two public datasets, CIFAR-10 and GTSRB, under Non-Independent and Identically Distributed (Non-IID) settings. Our assessment encompasses two prominent deep learning models, i.e., ResNet-18 [2] and MobileNetV3-Large [3]. Our approach offers test accuracy of FL global model compared to the state-of-the-art methods.

2 PROPOSED GCAMA AGAINST MODEL POISONING ATTACKS

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

2.1 GCAMA Architecture

On the device side, each of the K benign devices utilizes the Deep Neural Network (DNN) tailored specifically for image classification tasks, as shown in Fig.1. 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 [5], ResNet [2], DenseNet [4], and MobileNet [3] 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 [10] visualization is distinguished by its high resolution and high-class discriminative ability compared to other methods, e.g., Class Activation Mapping [16]. 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:

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

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

$$L^{(c)}_{l,GradCAM} = ReLU\Big(\sum_{m=1}^{n} \alpha^{(c)}_{l,m} A^m_l\Big). \tag{2}$$

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.

2.2 Autoencoder for Malicious GradCAM heat map 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 [17]. We use an autoencoder to pinpoint abnormal GradCAM heat maps corresponding to malicious local model updates uploaded by the attackers.

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 $\mathbf{L_{GradCAM}}$ as the input to the encoder. During the autoencoder training, the encoder $e_\theta(\mathbf{L_{GradCAM}})$ with parameter θ compresses the GradCAM heat maps from a high-dimensional space to a low-dimensional space $z = e_\theta(\mathbf{L_{GradCAM}})$, 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., $d_\phi(z) = \mathbf{L}'_{\mathbf{GradCAM}} = d_\phi\big(e_\theta(\mathbf{L_{GradCAM}})\big)$. After the training of (θ, ϕ), the reconstructed GradCAM heat maps are reshaped into the same size as the original GradCAM heat maps.

[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.

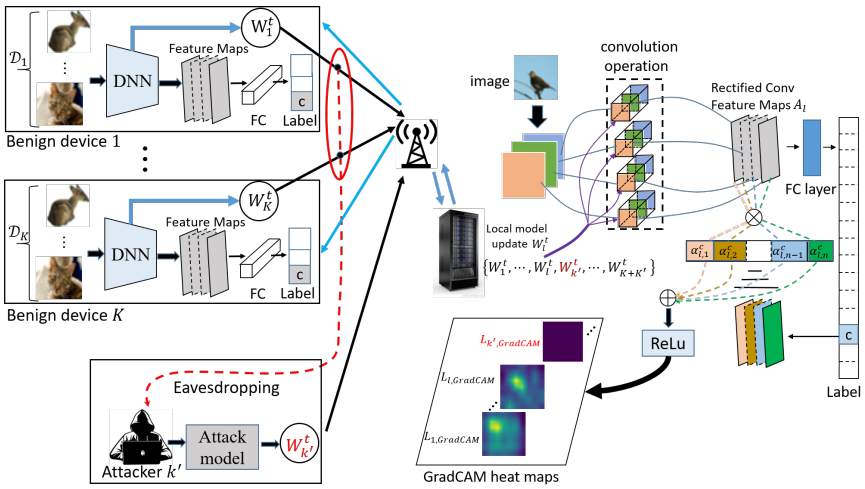

Figure 1: GradCAM-assisted defense against poisoning attacks on FL. The server arbitrarily selects an image (e.g., an image with the label "bird") from the global model testing dataset to create GradCAM heat maps for every uploaded local model update. These GradCAM heat maps flow into an autoencoder for malicious model detection.

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 $\mathbf{L_{GradCAM}}$, and the decoder reconstructed GradCAM heat maps $\mathbf{L^{'}_{GradCAM}}$, i.e.,

$$\begin{aligned} L(\theta,\phi) &= \min_{\theta,\phi} \frac{1}{|K|+|K'|} \|\mathbf{L_{GradCAM}} - \mathbf{L^{'}_{GradCAM}}\|_2^2 \\ &= \min_{\theta,\phi} \frac{1}{K+K'} \sum_{l=1}^{K+K'} \|L_{l,GradCAM} - d_\phi(e_\theta(L_{l,GradCAM}))\|_2^2. \end{aligned} \tag{3}$$

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., $\forall l \in \mathcal{K} \cup \mathcal{M}$,

$$R_l = \frac{\sum_{i=1}^{B}\sum_{j=1}^{H} \left|L_{l,GradCAM}(i,j) - L^{'}_{l,GradCAM}(i,j)\right|}{H \times B}. \tag{4}$$

The average reconstruction error of all GradCAM heat maps is

$$\overline{R} = \frac{1}{R_l} \sum_{l=1}^{K+K'} R_l. \tag{5}$$

A threshold δ is defined as

$$\delta = \overline{R} + \alpha \times \sqrt{\frac{\sum_{l=1}^{K+K'} (R_l - \overline{R})^2}{K+K'}}, \tag{6}$$

where α is an empirically configured coefficient. Here, δ 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. The AE can encounter difficulties in handling anomalies that do not conform to the learned patterns.

Note that the AE is optimized to minimize the reconstruction errors between the input GradCAM heat maps and the reconstructed GradCAM heat maps during training. In other words, the AE learns to reconstruct normal GradCAM heat maps. 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.

3 PERFORMANCE EVALUATION

We use two datasets, i.e., CIFAR-10 [6] and GTSRB [12], to evaluate the performance of our proposed GCAMA. We consider the following two state-of-the-art defense schemes, i.e., **Multi-Krum** [1]: computes a score for each local model update, the score is the sum of its Euclidean distance from its neighbors, those with high scores are regarded as malicious model updates, which are excluded. ; and **FAA-DL** [11]: a lightweight, unsupervised anomaly detection method based on a one-class SVM-based method, support vector machine (SVM), which utilizes an appropriate kernel function and soft margins to estimate a nonlinear decision boundary and separate the benign and malicious local model updates.

Fig. 2 illustrates the test accuracy of ResNet-18 in Non-IID CIFAR-10 and Non-IID GTSRB. Under the Non-IID CIFR10 setting, GCAMA achieves the highest test accuracy (0.8) of FL global model and converges quickly (around the 30th communication round) as it involves more benign devices in FL training. This indicates that GCAMA can accurately filter malicious model updates. **Multi-Krum** 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 reveals 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. **FAA-DL** 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 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.

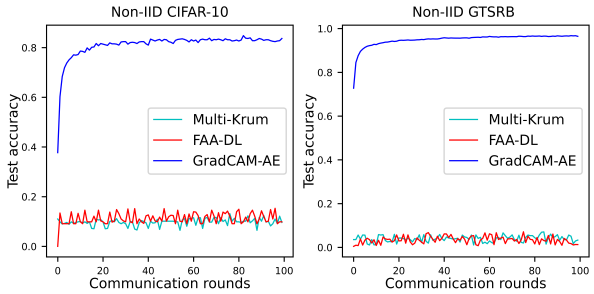

Figure 2: Test accuracy vs. communication rounds for ResNet-18 on Non-IID CIFAR-10 and GTSRB.

We replace the ResNet-18 model with MobileNetV3-Large, and the changing trend of the FL global model test accuracy with the communication round is consistent with Fig. 2, as shown in Fig. 3. 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.

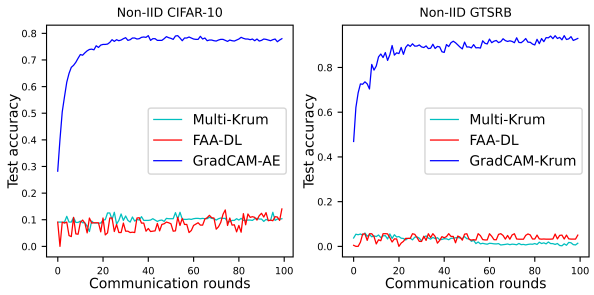

Figure 3: Test accuracy vs. communication rounds for MobileNetV3-Large on Non-IID CIFAR-10 and GTSRB.

4 CONCLUSION

In this paper, we proposed GCAMA, 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 GCAMA significantly outperforms the cutting-edge defense schemes under the same setting tested.

ACKNOWLEDGMENTS

This work was supported by the CISTER Research Unit (UIDP/UIDB/04234/2020) and project ADANET (PTDC/EEICOM/3362/2021), financed by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology).

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] 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)*.

[3] 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)*.

[4] 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.

[5] 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

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

[7] Kai Li, Billy Pik Lik Lau, Xin Yuan, Wei Ni, Mohsen Guizani, and Chau Yuen. 2023. Towards Ubiquitous Semantic Metaverse: Challenges, Approaches, and Opportunities. *IEEE Internet of Things Journal* (2023).

[8] Kai Li, Xin Yuan, Jingjing Zheng, Wei Ni, and Mohsen Guizani. 2023. Exploring adversarial graph autoencoders to manipulate federated learning in the internet of things. In *2023 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 898–903.

[9] 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).

[10] 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.

[11] 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. https://doi.org/10.1109/JSAC.2021.3118347

[12] 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. https://doi.org/10.1016/j.neunet.2012.02.016 Selected Papers from IJCNN 2011.

[13] 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.

[14] Jingjing Zheng, Kai Li, Naram Mhaisen, Wei Ni, Eduardo Tovar, and Mohsen Guizani. 2022. Exploring Deep-Reinforcement-Learning-Assisted federated learning for Online Resource Allocation in Privacy-Preserving EdgeIoT. *IEEE Internet of Things Journal* 9, 21 (2022), 21099–21110.

[15] Jingjing Zheng, Kai Li, Naram Mhaisen, Wei Ni, Eduardo Tovar, and Mohsen Guizani. 2023. Federated Learning for Online Resource Allocation in Mobile Edge Computing: A Deep Reinforcement Learning Approach. In *2023 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

[16] 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.

[17] 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.