Gradient Guard: Robust Federated Learning using Saliency Maps

Ujjwal Gupta
Walmart Global Tech
Bangalore, Karnataka, India

Yeshwanth Nagaraj
Indian Institute of Technology Madras
Chennai, India

ABSTRACT

We presents a groundbreaking approach to counter Directed Deviation Attacks (DDA)[5] in the domain of Federated Learning (FL)[12]. DDAs exploit gradient manipulation, disrupting model learning and increasing test error rates. This study introduces a novel defense mechanism employing Saliency Maps[7]—a tool highlighting influential input regions—to detect gradient anomalies caused by malicious clients.

Existing defenses struggle against DDAs, prompting exploration of an alternative solution. By quantitatively measuring Structural Similarity Index (SSI)[14] between Saliency Maps[7] of benign and potentially malicious client updates, abnormal gradient patterns can be swiftly identified. This method safeguards FL models by isolating contributions of suspicious clients. Its efficacy across diverse algorithms, architectures, and datasets is demonstrated.

Empirical evaluations reveal superiority, with the proposed approach ensuring Byzantine-robustness[9] against up to 50% malicious clients, compared to traditional defenses[3, 15]' 20-30% limit and recent work,FLAIR [13] which offers byzantine-robustness upto a malicious client percentage of 45% . This paper introduces a pioneering technique combining Saliency Maps and SSI[14] to protect FL models and underscores the need for proactive measures in the evolving landscape of decentralized learning.

The rise of Federated Learning (FL) brings decentralized learning to the forefront, enabling efficient and private model training across devices. However, FL's decentralized nature exposes it to adversarial challenges, including Directed Deviation Attacks (DDA). DDAs manipulate gradients to divert models from optimal paths, increasing test errors. Traditional defenses fall short against DDAs, necessitating innovative solutions.

This research harnesses Saliency Maps, traditionally used to understand model behavior, for defense against DDAs. Coupled with the Structural Similarity Index (SSI)[14], it provides a dual visual and quantitative strategy. The paper thoroughly explores this approach's foundations, mechanics, and effectiveness through empirical analysis, highlighting the need for proactive defenses in the decentralized learning era.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**.

ASIA CCS '24, July 01–05, 2024, Singapore, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

KEYWORDS

Federated Learning, Adversarial Attacks, Directed Deviation Attacks (DDA), Gradient Manipulation, Saliency Maps, Structural Similarity Index (SSI), Byzantine Robustness, Decentralized Learning, Model Integrity, Gradient Anomalies,Gradient Patterns, Malicious Client Detection, Machine Learning Security, Distributed Training, Byzantine Attack

ACM Reference Format:
Ujjwal Gupta and Yeshwanth Nagaraj. 2023. Gradient Guard: Robust Federated Learning using Saliency Maps. In *Proceedings of ACM ASIA Conference on Computer and Communications Security (ASIA CCS '24).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In the rapidly evolving domain of Federated Learning (FL), defending against adversarial attacks like Directed Deviation Attacks (DDA) remains paramount. DDA's, which exploit gradient manipulation, pose significant challenges, diverting models from their optimal learning path and increasing test error rates. This research proposes a novel defense mechanism, leveraging the potency of Saliency Maps—a tool that highlights crucial input regions that influence the output—to detect gradient variations and anomalies introduced by malicious clients.

Existing strategies have yet to fully counter the nuances of Directed Deviation Attacks (DDA). While several countermeasures have been introduced, their effectiveness against such attacks remains a topic of discussion. With this in mind, our study explores a different approach to address DDAs in FL. By examining the Saliency Maps of benign and potentially malicious client updates, we adopt the Structural Similarity Index (SSI)[14] to quantitatively measure the similarity between them. This approach allows for a fine-grained understanding of the model's response to different inputs, enabling a swift detection of atypical gradient patterns. By effectively isolating the contributions of potentially malicious clients, the FL models are guarded against DDAs. Our method, emphasizing both visual and quantitative analysis, displays consistent robustness across a plethora of learning algorithms, architectures, and datasets. Empirical evaluations underline the effectiveness of our approach. Comparative tests reveal that, in scenarios where traditional defenses[15] falter with a 20-30% malicious client infiltration, our Saliency Maps comparison using SSI[14] ensures Byzantine-robustness[9] for up to 50%. This paper not only presents a pioneering approach to defend FL models using Saliency Maps[7] and SSI but also underscores the necessity for proactive measures in the ever-challenging landscape of decentralized learning.

Federated Learning (FL) stands as a beacon in the ever-evolving landscape of machine learning, exemplifying the shift towards decentralized learning paradigms. By enabling model training across a myriad of devices while ensuring data remains localized, FL champions both efficiency and privacy. However, this very decentralization

that defines FL also exposes it to a gamut of adversarial challenges, fundamentally altering the security dynamics in play. Among these challenges, the shadow of Directed Deviation Attacks (DDA) looms large. DDAs represent a class of attacks unique in their method and malice. Rather than directly compromising data or the model's architecture, they subtly manipulate the gradients. This covert interference diverts learning models from their intended trajectory, causing them to stray from the optimal path. The repercussions are profound: not only is there an elevation in test error rates, but the integrity and reliability of the model are thrown into disarray. Traditional defenses, while commendable in their design and intent, often find themselves outpaced by the cunning sophistication of DDAs. It's akin to a relentless game of cat and mouse, where every defensive stratagem is met with an equally ingenious offensive countermeasure by adversarial agents. This tug of war underscores the urgency for innovative solutions that are both robust and anticipatory.

In light of this, our research turns to an unexpected ally in this battle: Saliency Maps. Conventionally utilized to decipher model behaviors by emphasizing crucial input regions, we harness their potential in a novel defensive context. Merging this with the precision of the Structural Similarity Index (SSI)[14] offers an unparalleled dual approach: a visual perspective paired with quantitative rigor. This paper embarks on elucidating this integrated strategy, exploring its foundations, mechanics, and efficacy in thwarting DDAs in the realm of Federated Learning. Through comprehensive empirical analysis and rigorous testing, we endeavor to not just present a solution but also emphasize the pressing need for proactive, innovative defenses in this decentralized age of learning.

2 BACKGROUND AND RELATED WORK

In the increasingly interconnected world of machine learning, Federated Learning (FL) has emerged as a transformative paradigm. Contrasting traditional centralized training models, FL offers decentralized training that primarily functions at the data source, addressing data privacy concerns and reducing data centralization-associated bottlenecks. However, the shift to decentralized processing hasn't been without challenges. Among the myriad of obstacles posed by FL, the threat of adversarial attacks, particularly the Directed Deviation Attacks (DDA), has caused significant consternation. These attacks, by manipulating gradients, have the potential to derail models from their optimal learning trajectories, leading to decreased performance and increased test error rates.

Historically, the machine learning community has responded to DDAs with an array of countermeasures, aiming to provide robustness against these attacks. Yet, a thorough review of literature indicates that many of these measures, while innovative, offer only partial protection, often being susceptible to advanced or modified attacks. Their inherent weaknesses stem from a variety of factors: some focus too narrowly on specific attack vectors, leaving them exposed to newer threats, while others might introduce undue computational overheads, making them impractical for real-world deployment.

Some of the recent works include:

FedSGD[12]: FedSGD uses a straightforward approach of aggregating the gradients based on a weighted mean, with the weight determined by the volume of data each client possesses. However, it's vulnerable to attacks from even a single malicious client that sends amplified harmful gradients.

Trimmed Mean and Median[18]: These two methods handle parameter aggregation individually. The Trimmed Mean removes a set number (notated as c_{max}) of extreme values from both ends for every parameter, whereas the Median method simply uses the median value for each parameter from all received gradients. These methods, however, can be undermined by the Full-Trim attack.

Krum[1]: Krum's approach involves choosing a local model to represent the next global model. This decision is based on selecting the client whose model has the smallest Euclidean distance from its closest (m - c_{max} - 2) other clients. However, this method is susceptible to the full-Krum attack .

Bulyan[4]: Bulyan merges methodologies by first applying Krum multiple times to pick a subset of models and then employing Trimmed Mean on this subset. Unfortunately, the Full-Trim attack can be adapted to compromise Bulyan as well.

FABA[16]: This method sequentially eliminates models that deviate the most from the mean of the yet-to-be-filtered models. This filtering happens c_{max} times, after which the mean of the remaining gradients is chosen.

FoolsGold[6]: FoolsGold aims to safeguard against Sybil clone poisoning attacks. It detects and labels clients with high cosine similarity as potentially malicious, demoting their reputation. The gradients are then aggregated using a weighted mean, where the weights are determined by each client's reputation.

FLTrust[2]: FLTrust establishes client trust by presuming the server possesses a limited but clean validation dataset. It aggregates gradients using a weighted mean based on this trust. However, in many scenarios, acquiring such a clean dataset might be impractical, particularly given the non-iid nature of datasets on individual clients.

FLAIR[13]: FLAIR also employs a stateful suspicion model, which maintains a history of each client's activity. This historical data is then used as a weighting factor during gradient aggregation. The core intuition is that in a benign setting, as the model approaches an optimum, the number of gradients that drastically change direction is minimal. This consistent pattern of gradient changes is disrupted during an attack, allowing FLAIR to detect malicious entities. However, it has a heavy reliance on flip-score and its stateful suspicion model for each client might introduce scalability concerns.

Parallel to these developments in FL, another significant advancement in deep learning has been the advent and popularization of Saliency Maps. Originally developed to interpret and understand neural networks, Saliency Maps have the unique ability to highlight the critical input regions that have the most significant influence on model outputs. While their primary application has been in model interpretation, there's growing recognition of their potential in the realm of adversarial defense. The logic is intuitive: if a map can highlight influential regions for benign inputs, it might also spotlight anomalies introduced by malicious inputs, providing a visual cue for adversarial interventions.

Simultaneously, the Structural Similarity Index (SSI)[14] has been making waves in the domain of image processing. Initially conceptualized to measure the similarity between two images (typically an original and a compressed version), the SSI has been recognized for

its adaptability. Beyond mere image quality assessments, its ability to quantify similarities has found applications in diverse domains, from video processing to even some niche areas in machine learning. Given its prowess in measuring similarities, it stands to reason that SSI might offer a quantifiable metric to compare Saliency Maps, determining whether a given input (or set of inputs) deviates from an established norm.

While both Saliency Maps and SSI have individually contributed to their respective fields, their intersection, especially in the context of FL security, remains a largely unexplored frontier. Some preliminary studies have hinted at the synergy between these tools, suggesting that a union of visual interpretation (through Saliency Maps) and quantitative analysis (via SSI) might pave the way for a more comprehensive adversarial defense mechanism. However, such endeavors have been sparse, and a consolidated effort to harness their combined potential in FL remains conspicuously absent.

In essence, while FL presents a promising trajectory for machine learning, its susceptibility to DDAs poses a pressing challenge. Previous defenses, despite their merits, have demonstrated limitations, underscoring the need for fresh perspectives and tools. In this milieu, the promise offered by tools like Saliency Maps and SSI, and more importantly, their combined potential, positions them as likely candidates to forge a new path in securing FL against adversarial threats. This paper ventures into this intersection, aiming to bridge the existing gaps and chart a new course in FL defence.

2.1 Saliency Maps in Federated Learning

Saliency Maps[7] have garnered significant attention in the world of deep learning, offering a window into understanding how neural network models perceive and process data. At their core, these maps serve as visual tools that offer insights into a model's decision-making process. In the vast realm of machine learning, where model interpretability often remains elusive, Saliency Maps have emerged as beacons of clarity, allowing researchers and practitioners to "see" and understand the inner workings of these models. To understand the essence of a Saliency Map, consider an image I passed through a convolutional neural network (CNN)[10] for classification. The model provides a probability distribution over the classes, and let's say it classifies I as a "dog" with a high probability p. The associated Saliency Map for this classification would highlight regions in I that most influenced this decision. Simply put, it answers the question: "Which pixels in I were most crucial for the model to label it as 'dog'?"

$$S(I) = \partial p / \partial I \tag{1}$$

Here, S(I) represents the Saliency Map of image I. Bright regions in S(I) indicate areas where small changes can cause significant changes in the output probability p, suggesting that those regions were pivotal in the model's decision. In the context of Federated Learning (FL), where data is decentralized and model updates are sent from clients to a central server, understanding these influential regions becomes even more critical. Why? Because it can provide insights into whether a client's model update is genuine or potentially adversarial. By comparing the Saliency Maps of benign updates with those of new client updates, deviations can be spotted, offering a visual cue of possible malicious intent.

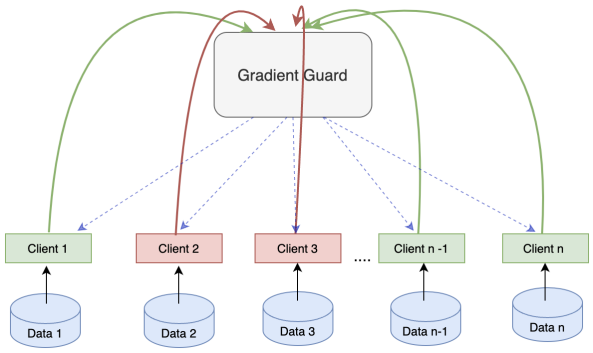

Figure 1: Gradient Guard Architecture

2.2 Leveraging Structural Similarity Index(SSI)

SSI[14], commonly referred to as SSIM, is a metric that quantifies the similarity between two images. Unlike MSE, which measures the absolute difference, SSI compares changes in structural information, luminance, and contrast. The index ranges from -1 to 1, with 1 indicating that the two images being compared are identical. Mathematically, the SSI between two images x and y is defined as:

$$SSI(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2}$$

where μ_x and μ_y are the average of x and y respectively; σ_x^2 and σ_y^2 are the variances ; σ_{xy} is the covariance; and c_1,c_2 are constants to avoid instability.

When leveraging SSI in the context of Saliency Maps, the goal is to measure how structurally similar a client's model update Saliency Map is to a baseline or known benign Saliency Map. If a client's update is genuine, its associated Saliency Map should have a high structural similarity to those of benign updates. Conversely, a low SSI score might indicate potential deviations, signaling malicious or atypical client behavior.

3 PROPOSED METHOLOGY: GRADIENT GUARD

3.1 Overview

Gradient Guard stands at the intersection of the inherent qualities of Saliency Maps and the analytical power of the Structural Similarity Index (SSI). The methodology seeks to fortify Federated Learning against Directed Deviation Attacks (DDA)[5] by scrutinizing updates from various clients through a two-pronged analytical approach.

3.2 Saliency Maps Extraction

Objective: To visualize the most important features or regions of an input image that contribute maximally to the model's output. Step 1: Input Initialization: Begin with client updates and the associated data they trained on. For each training instance or image, initialize the process of extracting its saliency map. Step 2: Forward Propagation: Pass the input data through the client's updated model. Compute the output scores, but focus primarily on the score of the

true class (or target class in case of an adversarial setting). Step 3: Gradient Computation: Calculate the gradient of the true class score with respect to the input image. This gradient signifies how much each pixel of the input image influences the true class score. Step 4: Saliency Determination: For each input image, derive the saliency map by taking the absolute value of the gradient. Highlight regions with larger gradient values, indicating more influence on the output decision.

3.3 SSI Analysis

Objective: To determine the similarity between Saliency Maps of potentially malicious and benign client updates. Step 1: Reference Selection: Choose a reference saliency map, ideally from a trusted or known benign client update. This serves as a benchmark for comparisons. Step 2: Pairwise Comparison: For each client's saliency map, compute the Structural Similarity Index with the reference. The SSI value lies between -1 and 1, with 1 indicating that the two images being compared are identical. Step 3: Threshold Setting: Establish a similarity threshold. If a client's saliency map has an SSI value below this threshold when compared to the reference, it's flagged as potentially malicious.

3.4 Malicious Client Filtering

Step 1: Compilation of SSI Scores: Aggregate all SSI scores from the comparisons. Step 2: Flagging Suspects: Clients with scores below the predefined threshold are earmarked. This doesn't directly classify them as malicious but rather as "under scrutiny". Step 3: Model Update Decision: Updates from clients under scrutiny aren't immediately integrated into the global model. Further tests or fallbacks to previous states of their model can be used to ensure that the collective learning process remains untainted.

3.5 Integration and Continuous Monitoring

Final Model Update: After careful screening, integrate the benign client updates into the global federated model. The process ensures that the model is guarded against any malicious drift caused by DDAs. Continuous Monitoring: With each federated learning round, reapply Gradient Guard to ensure the model's integrity. Over time, with more data and updates, refine the SSI threshold to improve accuracy and reduce false positives.

(1) **Global model** $\mathbf{G}(t+1,\cdot)$:
- Represents the updated global model at a given time $t+1$.

(2) **Local model updates** $\mathbf{w} = \Delta L_i(t+1,\cdot)$:
- Refers to the updates sent by the local model i for a given time $t+1$.

(3) **Parameters:** $m, c_{\mathbf{max}}, \mu_d$:
- m: The total number of clients participating in the federated learning.
- $c_{\max}$: The maximum number of clients that can be malicious.
- μ_d: Decay parameter indicating the degree to which past reputation scores influence the current score.

(4) **Initialize reputation** $RS_i(0) = 0$:
- Initially set the reputation score for every client i to zero.

(5) **Initialize global direction** $s_g(0)$:
- Set the initial gradient direction of the global model to a zero vector.

(6) **Saliency Maps of** $\Delta LM_i(t+1,\cdot)$ **and** $s_g(t,\cdot)$: Saliency maps highlight the regions in input data (like images) that are most influential for a neural network's decision. Here, it's applied to local updates and global direction to derive some form of similarity or distinction.

(7) **SSI** $SSI_i(t+1)$: Represents the Saliency Similarity Index (or a similarly named measure), indicating the similarity between the saliency maps of the local update and the global direction.

(8) **Penalizing and Rewarding Clients**: Clients with particularly high or low SSI scores are penalized or rewarded based on their contributions being in line or deviating from the expected.

(9) **Normalize reputation weights:** WR:
- This converts the reputation scores to a normalized weight, ensuring that they are all between 0 and 1 and their sum is 1. This is important for creating an aggregate model update.

(10) **Aggregate gradients:** $\Delta G(t+1,\cdot)$:
- This process collects and combines all the local updates based on their normalized reputation weights to form a global model update.

(11) **Update global direction:** $s_g(t+1,\cdot)$:
- After the aggregation, the global gradient direction is recalculated to be used in the next round.

(12) **Update global model and broadcast**:
- The global model is updated with the aggregated gradient and then shared with all participating clients.

ALGORITHM 1: Federated Learning With Gradient Guard

Output : Global model $\mathbf{GM}(t+1,\cdot)$
Input : Local model updates $\mathbf{w} = \Delta\mathbf{LM}_i(t+1,\cdot)$
Parameters : m, c_{max}, μ_d

Initialize global direction $s_g(0)$ to a zero vector
for every client i compute SSI score over Saliency Maps:
Compute Saliency Maps of $\Delta\mathbf{LM}_i(t+1,\cdot)$ and $s_g(t,\cdot)$ to obtain SM_i and SM_g respectively.
$SSI_i(t+1) = \text{SSI}(SM_i, SM_g)$
Penalize c_{max} clients with either the highest or lowest SSI scores as: $RS(i,t+1) = \mu_d RS(i,t)$ - $\left(1 - \frac{2c_{max}}{m}\right)$
Reward the rest of the clients as: $RS(i,t+1) = \mu_d RS(i,t) + \frac{2c_{max}}{m}$
Normalize reputation weights: $WR = \frac{e^{RS}}{\sum e^{RS}}$
Aggregate gradients: $\Delta\mathbf{GM}(t+1,\cdot) = \mathbf{w}^T WR$
Update global direction: $s_g(t+1,\cdot) = \text{sign}(\Delta\mathbf{GM}(t+1,\cdot))$
Update global model and broadcast: $\mathbf{GM}(t+1,\cdot) = \mathbf{GM}(t,\cdot) + \Delta\mathbf{GM}(t+1,\cdot)$

4 EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed method, we conducted experiments on various widely recognized datasets in federated learning, including CIFAR-10[8], MNIST[11], and Fashion-MNIST[17]. Our experimental architectures varied according to

the dataset: CIFAR-10[8] employed a convolutional neural network (CNN)[10] while MNIST[11] and Fashion-MNIST were tested with feed-forward neural networks.

4.1 Datasets and Architectures

4.2 Discussion of Results

5 CONCLUSION

REFERENCES

[1] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf

[2] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Gong. 2021. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. https://doi.org/10.14722/ndss.2021.24434

[3] Huili Chen and Farinaz Koushanfar. 2023. Tutorial: Toward Robust Deep Learning against Poisoning Attacks. *ACM Trans. Embed. Comput. Syst.* 22, 3, Article 42 (apr 2023), 15 pages. https://doi.org/10.1145/3574159

[4] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The Hidden Vulnerability of Distributed Learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 3521–3530. https://proceedings.mlr.press/v80/mhamdi18a.html

[5] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2021. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. arXiv:1911.11815 [cs.CR]

[6] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2020. The Limitations of Federated Learning in Sybil Settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. USENIX Association, San Sebastian, 301–316. https://www.usenix.org/conference/raid2020/presentation/fung

[7] Xiaodi Hou and Liqing Zhang. 2007. Saliency Detection: A Spectral Residual Approach. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*. 1–8. https://doi.org/10.1109/CVPR.2007.383267

[8] Alex Krizhevsky. 2012. Learning Multiple Layers of Features from Tiny Images. *University of Toronto* (05 2012).

[9] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* 4, 3 (jul 1982), 382–401. https://doi.org/10.1145/357172.357176

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* 1, 4 (1989), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. https://doi.org/10.1109/5.726791

[12] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2023. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629 [cs.LG]

[13] Atul Sharma, Wei Chen, Joshua Zhao, Qiang Qiu, Saurabh Bagchi, and Somali Chaterji. 2023. FLAIR: Defense against Model Poisoning Attack in Federated Learning. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security* (Melbourne, VIC, Australia) *(ASIA CCS '23)*. Association for Computing Machinery, New York, NY, USA, 553–566. https://doi.org/10.1145/3579856.3582836

[14] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

[15] Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. 2019. FABA: An Algorithm for Fast Aggregation against Byzantine Attacks in Distributed Neural Networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4824–4830. https://doi.org/10.24963/ijcai.2019/670

[16] Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. 2019. FABA: An Algorithm for Fast Aggregation against Byzantine Attacks in Distributed Neural Networks.. In *IJCAI*. 4824–4830.

[17] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747 [cs.LG]

[18] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 5650–5659. https://proceedings.mlr.press/v80/yin18a.html

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009