No Trust, No Learning? Improving Federated Learning Security and Robustness with Reputation and Trust

Sergei Chuprov[1*], Dmitrii Korobeinikov[2], Raman Zatsarenko[2], Leon Reznik[2]

[1*]Department of Computer Science, The University of Texas Rio Grande Valley, 1201 W University Dr, Edinburg, 78539, TX, USA.
[2]Department of Computer Science, Rochester Institute of Technology, 1 Lomb Memorial Dr, Rochester, 14623, NY, USA.

*Corresponding author(s). E-mail(s): sergei.chuprov@utrgv.edu;
Contributing authors: dk9148@rit.edu; rz4983@rit.edu;
leon.reznik@rit.edu;

Abstract

In this paper, we address the vulnerability of Federated Learning (FL) to data poisoning attacks, where malicious clients can degrade global model performance through manipulation of local data. Existing defenses often face limitations in computational complexity and unrealistic assumptions about attacker's knowledge. To overcome these challenges, we develop a novel FL defense mechanism based on Reputation and Trust metrics. This approach dynamically identifies and excludes malicious clients by detecting statistical anomalies in their model updates and calculating historical metrics. Evaluated on the BloodMNIST dataset under data poisoning attacks, our method demonstrates superior performance compared to Multi-Krum, effectively detecting and removing malicious clients with lower errors. This results in improved model accuracy and robustness with no prior knowledge of the number of attackers. Our key contribution is a practical and effective defense strategy that enhances the security and robustness of FL systems operating in adversarial environments.

Keywords: federated learning security, data poisoning, malicious attacks detection

1 Introduction

Federated Learning (FL) has emerged as a transformative approach to Machine Learning (ML), enabling collaborative model training across multiple decentralized clients without direct data sharing [1]. This paradigm is particularly crucial in privacy-sensitive domains like healthcare and finance, where legal and ethical considerations prohibit centralized data aggregation. While FL offers significant advantages in terms of privacy and data access, a critical challenge is its vulnerability to malicious actors. The inherent distributed nature of FL, while beneficial for privacy, exposes it to two major types of attacks: data poisoning and model poisoning. Data poisoning involves adversaries manipulating the training data on their local devices, while model poisoning involves directly manipulating the model updates sent to the central server. Of these, data poisoning is generally considered more relevant in practical scenarios because it requires less technical expertise and access to the system's inner workings [2].

Existing defense mechanisms in FL often rely on analyzing client updates (gradients or model parameters) using distance-based metrics to identify outliers [3–5]. While offering some protection, these methods can be computationally demanding and risk misclassifying legitimate clients as malicious, potentially reducing the diversity of the training data and hindering the model's overall performance. Furthermore, many of these approaches, especially those like Krum [3], depend on a priori knowledge or estimation of the number of malicious clients, a condition rarely met in realistic scenarios.

This paper introduces a novel defense mechanism based on *Reputation* and *Trust*, designed to address these limitations and assure model effectiveness in the conditions of adversarial attacks. Reputation and Trust are social concepts that were adapted in computing to enhance security in decentralized systems [6–8]. Our approach leverages a dynamic scoring system to identify and exclude malicious clients [9–12]. Specifically, it analyzes the statistical properties of the model updates contributed by each client. Legitimate clients, training on genuine data, are expected to produce updates that, while diverse, exhibit certain statistical consistencies. Malicious clients, injecting poisoned data, will generate updates that are dissimilar from those of honest clients, creating outliers in the distribution of update characteristics. Our method uses this principle, calculating a Reputation score for each client based on the distance of their updates from a central tendency (approximated by the centroid of all updates). Trust, derived from Reputation, reflects the server's confidence in a client's ongoing contributions. Clients consistently exhibiting anomalous behavior, indicated by low Trust scores, are flagged and excluded from model aggregation. This step directly enhances the effectiveness of the resulting model by preventing the integration of malicious, performance-degrading updates, and the security of FL, as malicious clients are detected and excluded.

The key contributions of this work include: **(1)** we introduce a novel FL defense mechanism based on Reputation and Trust, designed to detect and mitigate the influence of malicious clients without requiring prior knowledge of their number; **(2)** we empirically demonstrate that incorporating our defense mechanism into the FL process significantly improves the effectiveness of the trained model in the presence of

data poisoning attacks. The core of our contribution is demonstrating this improvement in model performance resulting from malicious client removal; **(3)** we evaluate our approach on a state-of-the-art medical imaging dataset (BloodMNIST from the Medical MNIST collection) and compare its effectiveness with Multi-Krum [3] defense.

2 Background

FL has rapidly gained traction as a powerful paradigm for collaborative ML, particularly in domains where data privacy is paramount, such as healthcare. The core principle of FL is to train a global model across a network of decentralized clients (e.g., hospitals, mobile devices) without requiring those clients to share their raw data. Instead, each client trains a local model on its own data, and only the model updates (e.g., gradients, weights) are transmitted to a central server for aggregation. This approach significantly reduces the risk of data breaches and facilitates compliance with privacy regulations like HIPAA and GDPR.

Conventional FL, while beneficial for privacy, introduces security vulnerabilities due to its decentralized and asynchronous nature, making it susceptible to poisoning attacks, broadly categorized as data, model, and backdoor attacks. Data poisoning, the most common and practically relevant attack, involves manipulating local training data and is easily executed by malicious clients with minimal expertise [2], aiming to degrade the global model's performance. Model poisoning attacks, conversely, require deeper system understanding to directly manipulate model updates [13, 14], while backdoor attacks, the most sophisticated, embed hidden triggers for specific misbehavior [15, 16], both being more challenging to implement than data poisoning. Given the practical relevance and high probability of untargeted data poisoning attacks, this research focuses on their detection and mitigation.

Malicious client detection is crucial for defending against poisoning attacks in FL, with methods like Krum and Multi-Krum [3], Bulyan [4], and Robust Federated Averaging (RFA) [5] aiming to identify and mitigate malicious updates during server-side aggregation. Krum and its extensions select updates "closest" to the majority, while Bulyan refines this with outlier removal and trimmed means, and RFA uses the robust geometric median for aggregation. However, these defenses are limited by their requirement for prior knowledge of the number of malicious clients and their quadratic computational complexity, ($O(n^2d)$), where n is the number of clients and d is the gradient space's dimension, which becomes a bottleneck in large-scale FL, and fundamentally, they focus on mitigating malicious data's impact without addressing the root cause of client untrustworthiness.

In FL, the concept of *model anomaly* is central to security as even with non-IID data, client model updates are expected to exhibit underlying similarity due to the shared goal of training a global model; therefore, a *model anomaly* is an update that significantly deviates from the typical distribution of updates, indicating potential malicious activity like data poisoning. Aggregating these anomalous updates introduces bias, disrupts learning convergence, and degrades model performance, accuracy,

and reliability, making their detection and exclusion crucial for FL security, unbiased models, and effective learning, necessitating robust and assumption-free defenses adaptable to unpredictable adversarial behavior.

3 Our Defense Methodology

In FL, clients collaboratively train an ML model without direct data sharing, coordinated by a central server, often using the FedAvg aggregation [1], where a global model with weights w_0 is initialized and, in each round t, participating clients receive global weights w_t, train locally, and send updated weights w_t^i back for server aggregation via averaging: $w_{t+1} = \frac{1}{|N|}\sum_{i\in N} w_t^i$. Our defense aims to enhance FL effectiveness by proactively identifying and excluding "malicious" clients in a cross-silo setting with full client participation, focusing on performance improvement through the detection and removal of detrimental contributions, particularly from data poisoning attacks, and introducing *Reputation* and *Trust* metrics to quantify client reliability for informed inclusion decisions.

1. Centroid Calculation. A crucial step in our defense is the calculation of the centroid of the client models. In each round t, after receiving the updated model weights w_t^i from all participating clients $i \in N$, the server computes the centroid μ_t as the average of these weights: $\mu_t = \frac{1}{|N|}\sum_{i\in N} w_t^i$. This centroid, μ_t, represents the "center of mass" of the model updates in the weight space. It serves as a reference point for evaluating the deviation of individual client models.

2. Reputation. Reputation is a distance-based metric that reflects the consistency of a client's model updates with the overall direction of the federated learning process, represented by the centroid. We define a distance metric $d(w_a, w_b)$ that measures dissimilarity between models w_a and w_b.

Initially, each client's Reputation is set based on its distance from the centroid: $R_{t_0}^i = 1 - d(w_{t_0}^i, \mu_{t_0}), R_{t_0}^i \in [0,1]$, where $d(w_{t_0}^i, \mu_{t_0})$ is the distance between client i's initial model $w_{t_0}^i$ and the initial centroid μ_{t_0}. We normalize this distance so that the initial reputation $R_{t_0}^i$ is between 0 and 1.

In each subsequent round t, we calculate $d_t^i = d(w_t^i, \mu_t)$, which is the distance between client i's model w_t^i and the centroid μ_t. Reputation is then updated according to (1) and (2).

$$X_t^i = \begin{cases} R_{t-1}^i + d_t^i - \frac{R_{t-1}^i}{t}, & \text{if } d_t^i \le \alpha, \\ R_{t-1}^i + d_t^i - e^{1-d_t^i\left(\frac{R_{t-1}^i}{t}\right)}, & \text{if } d_t^i > \alpha. \end{cases} \tag{1}$$

$$R_t^i = \begin{cases} \beta + (1-\beta)\cdot R_{t-1}^i, & \text{if } X_t^i \ge 1 \\ (1-\beta)\cdot R_{t-1}^i, & \text{if } X_t^i \le 0 \\ \beta \cdot X_t^i + (1-\beta)\cdot R_{t-1}^i & \text{otherwise} \end{cases}. \tag{2}$$

Here, α is a predefined distance threshold. Clients whose updates are close to the centroid (i.e., $d_t^i \le \alpha$) are rewarded. The term $\frac{R_{t-1}^i}{t}$ diminishes over time. Clients with updates far from the centroid ($d_t^i > \alpha$) are penalized.

3. Trust. Trust builds upon Reputation and serves as the decisive factor for client exclusion. The Trust metric, T_t^i, is calculated for each client i at round t:

$$Y_t^i = \sqrt{(R_t^i)^2 + (d_t^i)^2} - \sqrt{(1 - R_t^i)^2 + (1 - d_t^i)^2} \tag{3}$$

$$T_t^i = \begin{cases} \beta + (1-\beta) \cdot T_{t-1}^i, & \text{if } Y_t^i \geq 1, \\ (1-\beta) \cdot T_{t-1}^i, & \text{if } Y_t^i \leq 0, \\ \beta \cdot Y_t^i + (1-\beta) \cdot T_{t-1}^i & \text{otherwise} \end{cases} \tag{4}$$

Initially, T_0^i is set to 0. Equation (3) calculates Y_t^i based on the current Reputation (R_t^i) and the distance (d_t^i). Equation (4) then updates the Trust value. If a client's Trust (T_t^i) falls below a predefined threshold (we use 0.15 in our experiments), that model provided by the client is excluded from the aggregation in the current and all the consequent rounds.

4 Attack Model

In our study, we investigate untargeted data poisoning attacks within a realistic FL setting, assuming an adversary controls a minority of clients (up to 25%) without knowledge of the global model, communication protocol, or non-compromised client data, and is limited to manipulating training data labels, prioritizing data poisoning due to the challenges of model-level attacks in FL [2]. The attacker's objective is to degrade the global model's performance, formalized as maximizing the average loss difference over rounds T_0 to $T_0 + T$ between the attacked model $w_t^{\mathcal{A}}$ and the non-attacked model w_t, evaluated on a clean validation dataset $\mathcal{D}$: $\max_{\mathcal{A}} \frac{1}{T} \sum_{t=T_0}^{T_0+T} \left[L(w_t^{\mathcal{A}}, \mathcal{D}) - L(w_t, \mathcal{D}) \right]$, where $L(w, \mathcal{D})$ is the loss function evaluated on a clean, held-out validation dataset $\mathcal{D}$.

5 Empirical Study

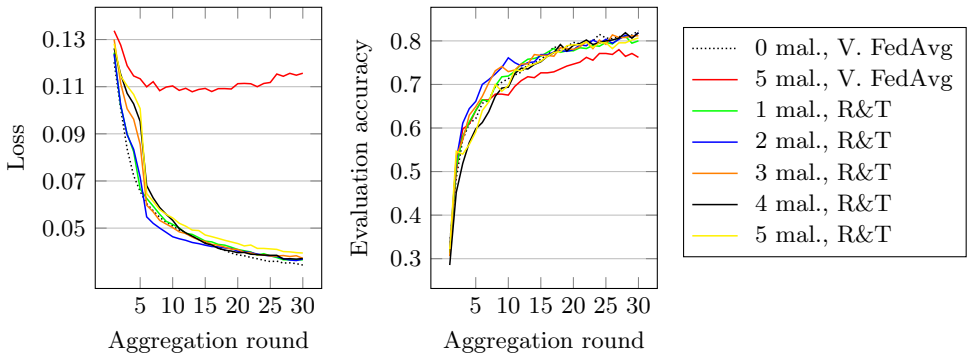
For experimental evaluation, we used a testbed based on the Flower Framework[1] to execute FL and assess our defense mechanism's effectiveness under varying attack intensities and settings, including different aggregation strategies, number of simulation rounds, and amount of participating clients. This testbed collects ML model performance metrics like cross-entropy loss and evaluation accuracy (using a 9:1 train/evaluation dataset split), alongside defense mechanism metrics such as False Positives (FP), False Negatives (FN), and client removal accuracy, as detailed in Table 1, enabling comprehensive evaluation of both model training and defense efficiency in preventing model degradation under attacks.

We executed our experiments with the total of 20 clients participating in the FL simulation. Initially, we randomly partitioned the entire BloodMNIST dataset into 40 subsets, which were then assigned to clients, resulting in the average number of 210 samples per client. Subsequently, we determined that the setup with only 20 participating clients is sufficient and representative enough for the purpose of evaluation of our defense mechanism.

[1] https://flower.ai

Metric	Description
Aggregated loss	Weighted loss of all aggregated client models. Weight of each client's model is defined by the number of samples in the client's subset. In our case, all clients have the same weight
Evaluation accuracy	Average evaluation accuracy of all benign client models. We purposefully calculate the accuracy this way in order to showcase the indirect effects of participating malicious clients on the accuracy of benign clients
Total count of FP and FN	At each aggregation round, FP is if a benign client was excluded from the aggregation. FN is when a malicious client was not excluded from the aggregation
Removal accuracy	The accuracy of the client exclusion process: $$\frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Table 1 Description of the metrics collected at each aggregation round during the experiment execution.



(a) (b)

Fig. 1 Robustness of Reputation and Trust (R&T) defense under varied attack intensities and comparison of the attack impact on the resulting model accuracy and convergence with conventional FL setting with no defense – Vanilla (V.) FedAvg: (a) – loss; (b) - evaluation accuracy of benign clients.

5.1 Robustness of Reputation and Trust Defense Under Various Attack Intensities

We altered the number of malicious clients that aim to participate in aggregation between the experiments. We conducted these experiments with the engaged Reputation and Trust (R&T) defense. Additionally, we executed one experiment with the presence of malicious clients, but without the defense mechanism, using the Vanilla

FedAvg aggregation. This allowed for more clear demonstration of the benefits of timely exclusion of malicious parties.

In Figs. 1(a) and 1(b) we demonstrate the loss and accuracy metrics collected during executions with a varied number of malicious clients participating in the training process, respectively. As one can see, on each figure, the dotted line represents the loss and accuracy in the no-attack scenario: none of the clients were malicious. In this case, the loss converges to the optimal value, indicating that the training process is performing as intended. Subsequently, the evaluation accuracy increases with each round, reaching values of around 0.8 by round 30, demonstrating more than adequate ML model performance for 8-class classification. The red line represents the case where 5 malicious clients were present, but the model training was performed in a conventional FL setting without defense techniques. In Fig. 1(a), the red line represents the loss behavior for this case. As one can see, it does not converge to optimal values, meaning that the model training is not performing well. Although the evaluation accuracy of benign clients, found in Fig. 1(b), does not demonstrate a similarly drastic difference, it still reflects the negative impact of participating malicious clients that is acquired during the local model updates after centralized aggregation.

R&T defense was applied for the rest of the experimental simulations with the number of clients varied between 1 and 5. As clearly demonstrated in Figs. 1(a) and 1(b), this defense mechanism renders more robust FL setup. Given the various number of malicious parties participating in the FL model training, the algorithm was able to detect and remove them in all cases. After the exclusion of these clients, both loss and evaluation accuracy of benign clients shifted towards the values typical for the setup without any active attacks.

5.2 Robustness of Reputation and Trust Compared to Multi-Krum

We conducted a series of experiments that compare the algorithm performance with Multi-Krum (MK) [3]. MK requires prior knowledge of the number of malicious clients participating in the training process. While this approach was shown to be theoretically robust, in our experiments we did not observe the reliable exclusion of malicious clients when their amount is not precisely estimated prior FL simulation execution.

The results of these experiments are presented in Fig. 2. We conducted three experimental scenarios. First, we executed FL training with five malicious clients participating in aggregation while applying the R&T defense mechanism. Since the MK approach necessitates an estimate of the number of malicious clients involved in the model training, we designed two experimental setups to evaluate its performance. In the first setup, we precisely estimated (p.e.) the number of malicious clients. However, in real-world scenarios, such precise knowledge may not be feasible. Instead, it is often only possible to approximate the fraction of malicious clients that may be present in the system. Therefore, in the third experiment, we assumed that no more than 10% of the clients were malicious. This assumption led to an underestimation (u.e.) of the actual proportion of malicious clients, which in our setup was 25%.

As demonstrated in Fig. 2(a), when the MK defense is applied with the underestimated number of malicious clients, the loss does not converge. The accuracy in Fig.

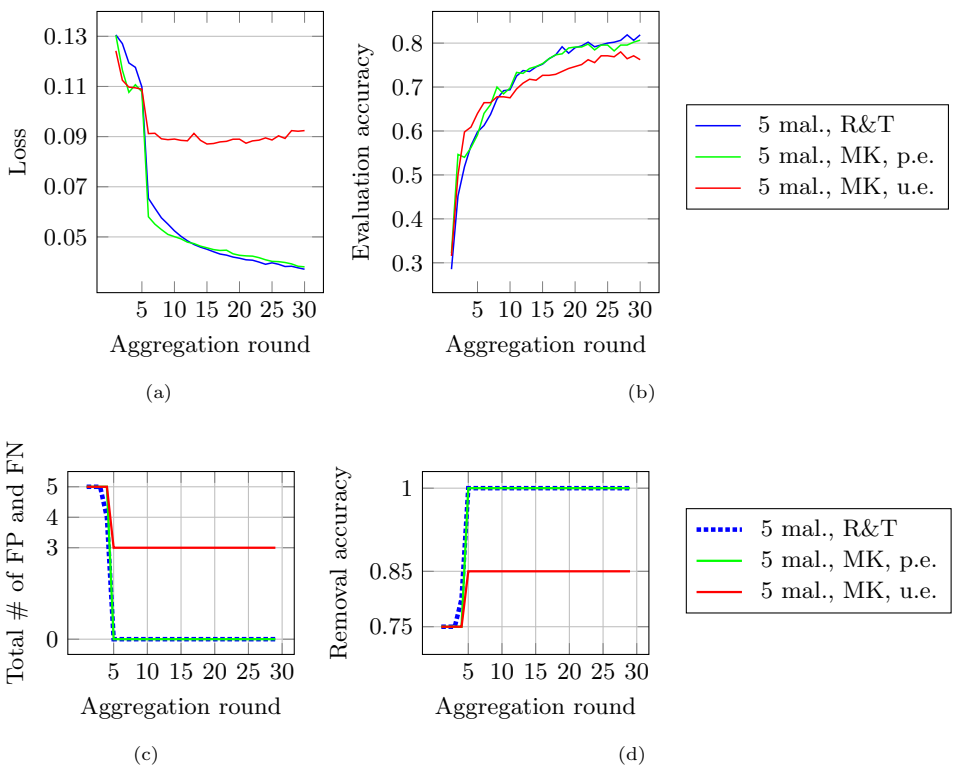


Fig. 2 Comparison of metrics collected during executions with Trust and Reputation (R&T) and Multi-Krum (MK) with the precise estimation of number of malicious clients and with underestimation (10% estimated amount of malicious clients, 25% real amount). (a) – loss; (b) - evaluation accuracy of benign clients; (c) - total count of FP and FP; (d) - removal accuracy.

2(b) also fluctuates and does not provide a stable result. This happens because, in the case of underestimation, MK only excludes the number of clients that is within the provided constraint, in our case, 10%, or two clients. Three malicious clients continue to participate in the aggregation, negatively affecting the training process.

Fig. 2(c) depicts the total count of FP and FN in each aggregation round for all three experiment executions. In the first five aggregation rounds, R&T defense does not exclude any clients, since we consider this to be a reasonable number of rounds for the initial model warm-up. Then, immediately after round five, R&T excludes malicious clients, resulting in the total of 0 FP and FN for the rest of the simulation. However, that is not the case for MK with the underestimation. For the entire training process, it produces a consistent count of 3 FN. This happens because the algorithm does not exclude more than 10% of the clients, an estimate provided prior to the training process. This behavior also reflects on the removal accuracy, demonstrated in Fig. 2(d).

6 Conclusion

In this paper, we developed and implemented Reputation and Trust-based mechanisms to enhance the security of FL against data poisoning attacks, achieving better results compared to the existing Multi-Krum defense strategy. Our defense allowed to effectively identify and exclude malicious clients, leading to fewer detection errors and improved client removal accuracy. This enhanced FL security and translated to a more robust and effective global model produced, evidenced by improved performance metrics under adversarial conditions. The empirical evaluation using the BloodMNIST dataset provides encouraging evidence for the practical applicability of our Reputation and Trust-based defense, suggesting it as a promising approach for enhancing the security of FL systems, particularly due to its ability to operate effectively without prior knowledge of the number of attackers.

Acknowledgments. This research was supported in part by the National Science Foundation (awards #2321652 and #2415299).

References

[1] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282 (2017). PMLR

[2] Shejwalkar, V., Houmansadr, A., Kairouz, P., Ramage, D.: Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In: 2022 IEEE Symposium on Security and Privacy (SP), pp. 1354–1371 (2022). IEEE

[3] Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. Advances in neural information processing systems **30** (2017)

[4] Mhamdi, E.M.E., Guerraoui, R., Rouault, S.: The Hidden Vulnerability of Distributed Learning in Byzantium. arXiv. Issue: arXiv:1802.07927 arXiv: 1802.07927 [cs, stat] (2018). http://arxiv.org/abs/1802.07927

[5] Pillutla, K., Kakade, S.M., Harchaoui, Z.: Robust Aggregation for Federated Learning. IEEE Transactions on Signal Processing **70**, 1142–1154 (2022) https://doi.org/10.1109/TSP.2022.3153135 . arXiv: 1912.13445 [cs, stat]

[6] Chuprov, S., Viksnin, I., Kim, I., Marinenkov, E., Usova, M., Lazarev, E., Melnikov, T., Zakoldaev, D.: Reputation and trust approach for security and safety assurance in intersection management system. Energies **12**(23), 4527 (2019)

[7] Chuprov, S., Viksnin, I., Kim, I., Melnikov, T., Reznik, L., Khokhlov, I.: Improving knowledge based detection of soft attacks against autonomous vehicles with

reputation, trust and data quality service models. In: 2021 IEEE International Conference on Smart Data Services (SMDS), pp. 115–120 (2021). IEEE

[8] Chuprov, S., Viksnin, I., Kim, I., Reznikand, L., Khokhlov, I.: Reputation and trust models with data quality metrics for improving autonomous vehicles traffic security and safety. In: 2020 IEEE Systems Security Symposium (SSS), pp. 1–8 (2020). IEEE

[9] Korobeinikov, D., Chuprov, S., Reznik, L.: Towards more robust federated learning with medical imaging model anomaly detection. In: 2024 IEEE Western New York Image and Signal Processing Workshop (WNYISPW), pp. 1–4 (2024). IEEE

[10] Chuprov, S., Bhatt, K.M., Reznik, L.: Federated learning for robust computer vision in intelligent transportation systems. In: 2023 IEEE Conference on Artificial Intelligence (CAI), pp. 26–27 (2023). IEEE

[11] Chuprov, S., Zatsarenko, R., Korobeinikov, D., Reznik, L.: Robust training on the edge: Federated vs. transfer learning for computer vision in intelligent transportation systems. In: 2024 IEEE World AI IoT Congress (AIIoT), pp. 172–178 (2024). IEEE

[12] Zatsarenko, R., Chuprov, S., Korobeinikov, D., Reznik, L.: Trust-based anomaly detection in federated edge learning. In: 2024 IEEE World AI IoT Congress (AIIoT), pp. 273–279 (2024). IEEE

[13] Fang, M., Cao, X., Jia, J., Gong, N.Z.: Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. arXiv. Issue: arXiv:1911.11815 arXiv: 1911.11815 [cs] (2021). http://arxiv.org/abs/1911.11815

[14] Shejwalkar, V., Houmansadr, A.: Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In: Proceedings 2021 Network and Distributed System Security Symposium. Internet Society, Virtual (2021). https://doi.org/10.14722/ndss.2021.24498 . https://www.ndss-symposium.org/wp-content/uploads/ndss2021_6C-3_24498_paper.pdf

[15] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How To Backdoor Federated Learning. arXiv. Issue: arXiv:1807.00459 arXiv: 1807.00459 (2019). http://arxiv.org/abs/1807.00459

[16] Boenisch, F., Dziedzic, A., Schuster, R., Shamsabadi, A.S., Shumailov, I., Papernot, N.: When the Curious Abandon Honesty: Federated Learning Is Not Private. arXiv (2023). https://doi.org/10.48550/arXiv.2112.02918