Improving Federated Learning Security with Trust Evaluation to Detect Adversarial Attacks

Harshil Patel, Sergei Chuprov, Dmitrii Korobeinikov, Raman Zatsarenko, and Leon Reznik
Rochester Institute of Technology, Rochester, NY, USA,
Email: hp8231@rit.edu, sc1723@rit.edu, dk9148@rit.edu, rz4983@rit.edu, leon.reznik@rit.edu

ABSTRACT

Federated Learning (FL), an emerging decentralized Machine Learning (ML) approach, offers a promising avenue for training models on distributed data while safeguarding individual privacy. Nevertheless, when imple- mented in real ML applications, adversarial attacks that aim to deteriorate the quality of the local training data and to compromise the performance of the resulting model still remaining a challenge. In this paper, we propose and develop an approach that integrates Reputation and Trust techniques into the conventional FL. These techniques incur a novel local models' pre-processing step performed before the aggregation procedure, in which we cluster the local model updates in their parameter space and employ clustering results to evaluate trust towards each of the local clients. The trust value is updated in each aggregation round, and takes into account retrospective evaluations performed in the previous rounds that allow considering the history of updates to make the assessment more informative and reliable. Through our empirical study on a traffic signs classification computer vision application, we verify our novel approach that allow to identify local clients compromised by adversarial attacks and submitting updates detrimental to the FL performance. The local updates provided by non-trusted clients are excluded from aggregation, which allows to enhance FL security and robustness to the models that might be trained on corrupted data.

Keywords: Federated learning, reputation, trust, adversarial attacks, intelligent transportation system.

I. INTRODUCTION

In today's Machine Learning (ML) research, Federated Learning (FL) has become increasingly popular because it offers a way to balance the need for valuable insights from data while also protecting privacy and security [1]. FL works by training models on multiple devices without sharing the actual data, keeping it private. This decentralized approach means that local model training is performed on the individual devices, and only the models are communicated to get combined centrally. This allows for additional valuable insights to be gained from different models while keeping individual data confidential.

However, the decentralized nature of FL brings complex issues to the security and communication. Exchange between edge devices and central aggregator may face challenges such as limited bandwidth, fluctuating network latency, and variations in device capabilities [2]. Moreover, maintaining the integrity and confidentiality of the FL process is crucial, as it is vulnerable to malicious activities such as ML model poisoning and data breaches [3]. Addressing these challenges is essential to ensure the effectiveness and security of FL systems.

In this paper, we delve into the nuanced intersection of the security and communication within FL frameworks, advocating for the incorporation of Reputation and Trust techniques as a means to mitigate malicious clients challenge [4], [5]. Reputation and Trust techniques, widely studied in the context of decentralized systems [6], hold promise in fostering collaborative environments while mitigating risks inherent in such distributed settings. By assigning Reputation and Trust values based on historical behavior and the quality of contributions, we aim to identify and exclude malicious clients from the aggregation process.

We verify our novel approach on Intelligent Transportation System (ITS) computer vision-based traffic signs classification use case. In our empirical study, we employ real-world traffic sign dataset within the FL framework and purposefully poisoning a specified number of clients by adversarial attacks such as label flipping and introducing noise into training data. Through our experiments, we demonstrate the capacity of Reputation and Trust techniques to effectively detect compromised local clients, which contributes to improving FL security and robustness to adversarial attacks against the local data, as well as to reducing the drop in performance of the resulting global model.

II. RELATED FEDERATED LEARNING SECURITY AND ROBUSTNESS IMPROVEMENT TECHNIQUES

FL is a promising approach for training ML models. Its primary advantage is the preservation of client data privacy. In FL only the model updates instead of the data itself get transmitted to an aggregator agent over a network [7].

Despite such feature as the ability to preserve the local data privacy, FL still faces various security threats relevant to ML systems in general. Particularly, concerns arise regarding data integrity and the validity of the transferred model. On the one hand, data poisoning may occur if the data utilized in an ML system becomes corrupted, either due to adversarial attacks against it, or because of the unintentional harmful conditions, such as a poor network connection or a storage damage. On the other hand, the risk of reverse engineering attacks against the ML model itself poses a threat to FL

. Given that the model is transmitted over a network, questions arise about the validity of the acquired model on the aggregation agent.

Various attempts have been made to mitigate these issues. One area of research is focused on investigating the impact of modifying server aggregation strategies on ML model robustness. An adjustable aspect of FL is the model aggregation algorithm employed on the server. The selection of the aggregation strategy significantly influences the robustness of the whole system [8], impacting the model's capabilities of withstanding possible outliers in the training data. Consequently, utilizing a specific aggregation strategy may serve as a potential way to enhance the overall robustness of the ML model.

Federated Average (FedAvg) stands as a widely employed conventional aggregation strategy. McMahan *et al.* [9] examined the FL setup with FedAvg on the CIFAR dataset, employing various neural network architectures, including two distinct convolutional neural networks (CNN), a two-layer character long short-term memory (LSTM), and a large-scale word-level LSTM. The experimental results demonstrated that the FedAvg enables the convergence of client models in fewer rounds compared to federated stochastic gradient descent (SGD) strategy.

Multi-Krum [10] represents an aggregation strategy that is specifically designed to mitigate the malicious parties in FL. The essence of the approach is in excluding a parameterized number of ML model contributors whose models deviate the farthest from the mean value during centralized model aggregation. The Euclidean distance serves as the metric for calculating the distance between the aggregated models, allowing the identification of the potentially compromised contributors.

A novel approach to the FL aggregation process – robust federated aggregation – was introduced by Pillutla *et al.* [8]. The core idea behind it is to enhance FL resilience to corrupted updates, thereby mitigating the impact of poisoned data in ML systems, as the sensitivity to the corrupted model updates poses a vulnerability. Robust federated aggregation is based on the Geometric Median (GM) method and is proposed by the authors as a solution which offers greater robustness to ML model outliers compared to the FedAvg and the pure GM. Additionally, two other approaches to FL aggregation – trimmed mean and FedMGDA – were proposed in [11] and [12], respectively. These aggregation algorithms also aim to address limitations and bottlenecks of the conventional FL aggregation.

In addition to examining the effects of applying various aggregation strategies, other studies concentrate on the network exchange process in ML systems utilizing FL. In [13] and [14], the authors aim to reduce communication burdens.

Lu *et al.* [15] in their study aimed at enhancing the overall robustness of ML in Industrial Internet of Things (IoT) applications from an architectural standpoint. Authors explore FL as a component of the data sharing mechanism along with the blockchain technology. A significant difference from the traditional data sharing in this context lies in the role of the blockchain module in establishing secure connections among IoT devices. However, the incorporation of blockchain into the data transmission process introduces additional threats typical of blockchain systems. Among these concerns is the increased complexity involved in setting up and deploying such architectures. Additionally, as noted by the authors in their study, there is a potential challenge regarding data privacy within the blockchain itself.

Li *et al.* [16] in their work focused on addressing the challenges stemming from systems and statistical heterogeneity within FL networks. Authors propose the FedProx framework, seeking to mitigate the impact of such heterogeneity on federated optimization. FedProx allows the variable amounts of computations among participating clients and utilizes a proximal term to stabilize the optimization process. Empirical evaluations that were conducted across diverse FL datasets validated the theoretical analysis, demonstrating that the FedProx framework significantly enhances convergence behavior in heterogeneous networks that are close to real-world conditions. Kang *et al.* [17] focused on addressing privacy concerns, particularly regarding the differential privacy (DP) requirements in SGD-based FL frameworks. Authors introduced NbAFL – a novel framework that ensures DP under distinct protection levels by adapting various amounts of artificial noises, thereby offering a tradeoff between convergence performance and privacy protection. Researchers provided theoretical convergence bounds for the loss function of the trained FL model in NbAFL, which helped to identify the optimal aggregation times for a given protection level. Proposed K-random scheduling strategy also demonstrated effectiveness in retaining convergence performance while preserving privacy.

Felix *et al.* [18] investigated the resilience of clustered FL systems against Byzantine attacks, where some participants behave maliciously. Through analysis of different clustering strategies and their impact on system robustness, the authors provided techniques for enhancing the security and scalability of FL frameworks. The study provides practical guidance for designing robust and efficient FL systems capable of withstanding adversarial behavior in real-world scenarios.

In [19], authors conducted the overview the past five years of FL research, investigating attacks in FL systems ranging from privacy to data poisoning attacks. Different attack vectors and their implications were identified, which would help to develop robust defense mechanisms.

Our novel approach offers Reputation and Trust techniques for the client ML models. This design aims at mitigating reverse engineering attacks on the FL by identifying compromised ML models before the centralized aggregation process. A key advantage is that, on the one hand, ML models of malicious participants can be completely excluded from the aggregation, mitigating the harmful effect on the resulting centralized model. On the other hand, no additional network communication overhead is introduced, which is crucial for the ML-based applications that are often dealing with the poor network conditions. The approach can also be combined

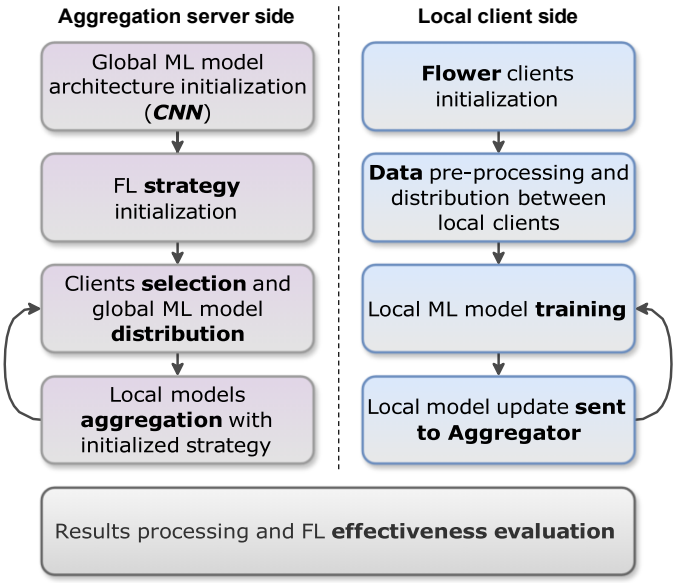

Fig. 1. Flower framework workflow employed in our empirical study. The diagram illustrates the collaborative training cycle between local clients and the central aggregation server, showcasing the FL process major training stages

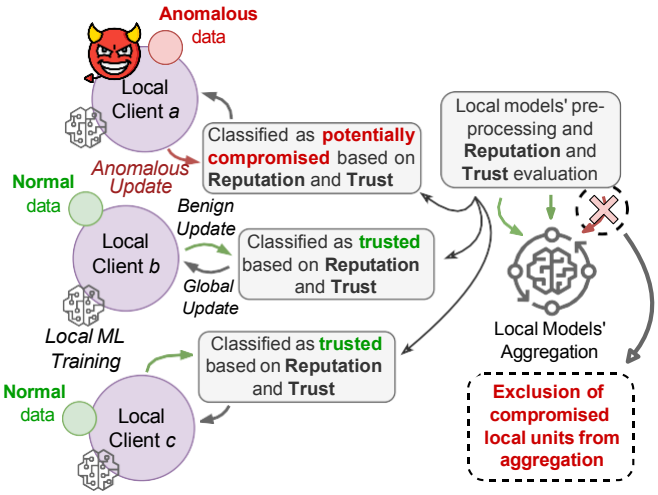

Fig. 2. Reputation and Trust-based techniques for detecting and excluding anomalous local models updates from FL aggregation procedure. The Reputation and Trust indicators calculated based on local models' clustering. If Trust towards local client drops below an established threshold, their updates are excluded from FL aggregation

with the encryption-based strategies for FL. Additionally, this design allows for more in-depth behavioral analysis of potentially malicious participants. Such parties can not only simply be excluded from the aggregation process but also receive the model that would help to adjust their Reputation and Trust score based on their further response in the next aggregation round.

III. System Architecture and Workflow

The system architecture described herein represents a comprehensive simulation framework for FL, implemented on a single local machine utilizing the Flower framework [20]. Our architecture encompasses six Python files, each serving distinct functionalities crucial for orchestrating FL system locally. Leveraging Flower, our system facilitates the emulation of FL scenarios within a controlled environment. This framework enables the evaluation of FL algorithms, the study of Reputation and Trust techniques, and the visualization of experimental outcomes, all within the confines of a single local machine environment. Below, we provide detailed explanations of each module within the system, elucidating their roles and functionalities in orchestrating FL system.

Load Dataset is tasked with the initial loading and preprocessing of datasets from separate folders corresponding to individual clients. Its functionality extends to pre-processing tasks, such as injecting noise or introducing poisoned data into specific clients' training datasets.

Flower Client embodies the Flower client class, equipped with methods to manage local model parameters and communicate with the aggregation server. These methods include setting and retrieving parameters, training and testing local models, and transmitting updates to the aggregation server.

Network defines the structure of the local neural network utilized by Flower clients for training models on their private datasets. This component plays a crucial role in enabling clients to independently train models on their local data.

Reputation And Trust Strategy introduces FL strategy that extends the FedAvg strategy within the Flower framework. It incorporates techniques to calculate the Reputation and Trust of each client, aggregate model parameters, determine client participation in aggregation based on Reputation and Trust metrics, and evaluate aggregate loss and accuracy.

Plots serves as the visualization hub, offering functions to represent output data graphically. This includes visualizing client accuracies across rounds and tracking the Reputation and Trust of clients throughout the FL process.

Controller acts as the central orchestrator, coordinating interactions between the various components of the system. It initiates the loading of datasets, creation of Flower clients, setup of the Reputation and Trust strategy, execution of FL system using the Flower framework, and presentation of experimental outcomes through visualizations generated by various functions in Plots.

Collectively, these Python files form a comprehensive and modular system architecture, enabling the implementation and execution of FL system using Flower framework. This architecture facilitates the evaluation of different FL strategies and provides a platform for analyzing and visualizing results of the empirical study.

The workflow diagram illustrates the training process collaboration between local clients and the central aggregation server within the Flower framework, as applied in our study (Figure 1). It consists of two critical verticals: the aggregation server side and the local client side, each representing key stages of the FL process.

At the aggregation server side, the global model architecture, CNN in our case, is set up at the aggregation server. Then, the FL strategy is initialized to establish the rules for the collaborative training which is Reputation and Trust strategy in our case (Figure 2). Next, the aggregation server selects clients to join and sends out the initial global ML model for local training. Generally all clients are selected in the first round unless specified. After local training, clients send their ML model updates back to the aggregation server for combining and configuring clients for the next round according to the FL strategy. The clients with anomalous updates are removed from the aggregation during the configuration process. The process of receiving clients' model parameters and selecting clients for next round keeps on repeating for each round.

On the local client side, all the flower clients are initialized. The data is pre-processed and segregated among the clients for training and validation. After assigning data, clients receive global model parameters from the aggregation server, and train their ML models using the global model architecture. Clients then send their updates back to the aggregation server for integration. Even at the client side receiving global model parameters and sending local model parameters from and to the aggregation server happens in each round. Finally, the results processing and FL effectiveness evaluation stage examines how well the FL process achieves its goals by visualizing various evaluation matrices.

IV. Reputation and Trust-based Aggregation in Federated Learning

Our proposed solution to enhance security within FL systems involves leveraging Reputation and Trust techniques [6], [4], [5]. At the aggregation server, before computing the final model, we employ K-means clustering on the clients' model parameters to group them based on similarity. Subsequently, we calculate the reputation R for each client based on the normalized Euclidean distance d from the major cluster center. Initially, R is determined in (1), where R_i^{t0} represents the reputation value for the i-th unit after the first local training round, and d_i denotes the truth value of the client which is one minus normalized Euclidean distance. R is updated in each aggregation round; if $d \geq \alpha$ where α is a specified threshold ($\alpha = 0.5$ in our case), R increases linearly, otherwise if $d < \alpha$, it decreases exponentially. Lastly, we utilize exponential smoothing to update parameters, blending the current round's reputation with the previous one based on a smoothing factorβ(0.75 in our case). The reputation for the current time moment t is calculated as defined in (2), (3) and (4), adjusting R based on the previous reputation value R_i^{t-1}. This approach penalizes units providing low-quality models and requires consistent positive contributions over time to build reputation.

$$R_i^{t0} = d_i \quad R, d \in [0,1] \subset \mathbb{R} \tag{1}$$

$$R_i^t = \begin{cases} (R_i^{t-1} + d_i) + (R_i^{t-1}/t), & \text{if } d \geq \alpha, \\ (R_i^{t-1} + d_i) - e^{-(1-d)(R_i^{t-1}/t)} & \text{if } d < \alpha \end{cases} \tag{2}$$

$$R_i^t = \begin{cases} 1, & \text{if } R \geq 1 \\ 0, & \text{if } R \leq 0 \end{cases} R_i^t = \beta \cdot R_i^t + (1-\beta) \cdot R_i^{t-1} \tag{3}$$

The trust indicator is derived from R and regulates how changes in R influence trust towards the local unit. If the trust falls below a predefined threshold β, the client's model is excluded from current and subsequent aggregation rounds. In (4), (5) and (6), we depict the calculation of the trust indicator $Trust^t_i$ for the i-th local unit at time t, considering R, the trust value d_i and previous rounds trust value. For the first round, the previous round's trust value is assigned to 0 for all clients. We use the same exponential smoothing formula for the trust calculation as well but with β value as 0.85. This mechanism ensures that trust is maintained towards units consistently providing high-quality contributions, while excluding those with lower reputation or questionable models from participating further in the FL process.

$$Trust_i^t = \sqrt{(R_i^t)^2 + d_i^2} - \sqrt{(1-R_i^t)^2 + (1-d_i)^2} \tag{4}$$

$$Trust_i^t = \beta \cdot Trust_i^t + (1-\beta) \cdot Trust_i^{t-1} \tag{5}$$

$$Trust_i^t = \begin{cases} 1, & \text{if } Trust \geq 1, \\ 0, & \text{if } Trust \leq 0 \end{cases} \tag{6}$$

V. Dataset and Experimental Setup

For the experimental studies, we investigate the ITS computer vision-based traffic signs classification use case [21]. We employ a traffic sign dataset containing images with two distinct labels: 0 for "stop sign" and 1 for "other traffic sign", each with dimensions of 224x224 pixels. The dataset is distributed across 12 different clients, with each client allocated approximately 120 images, ensuring a relatively equal distribution of images for both labels. To perform empirical study of adversarial scenarios, we intentionally poison the training data for clients with IDs 11 and 12 by flipping their labels. For training and validation purposes, each client utilize 90% of its allocated data for training and the remaining 10% for validation. This dataset composition and partitioning scheme facilitates the evaluation of our FL framework's performance under realistic conditions, including the presence of malicious clients.

We utilize a CNN architecture for the local training process on each client. The model architecture consists of two convolutional layers, each followed by max-pooling layers, and three fully connected layers. Dropout layers are incorporated to prevent overfitting, and ReLU activation functions are applied to introduce non-linearity.

Our FL process is performed over 10 rounds, mimicking the iterative nature of FL in real-world applications. During each round, clients perform local training on their respective datasets and communicate their model parameters and gradient updates to the aggregation server.

To enhance the security and reliability of FL, we integrate Reputation and Trust techniques into the FL framework. These

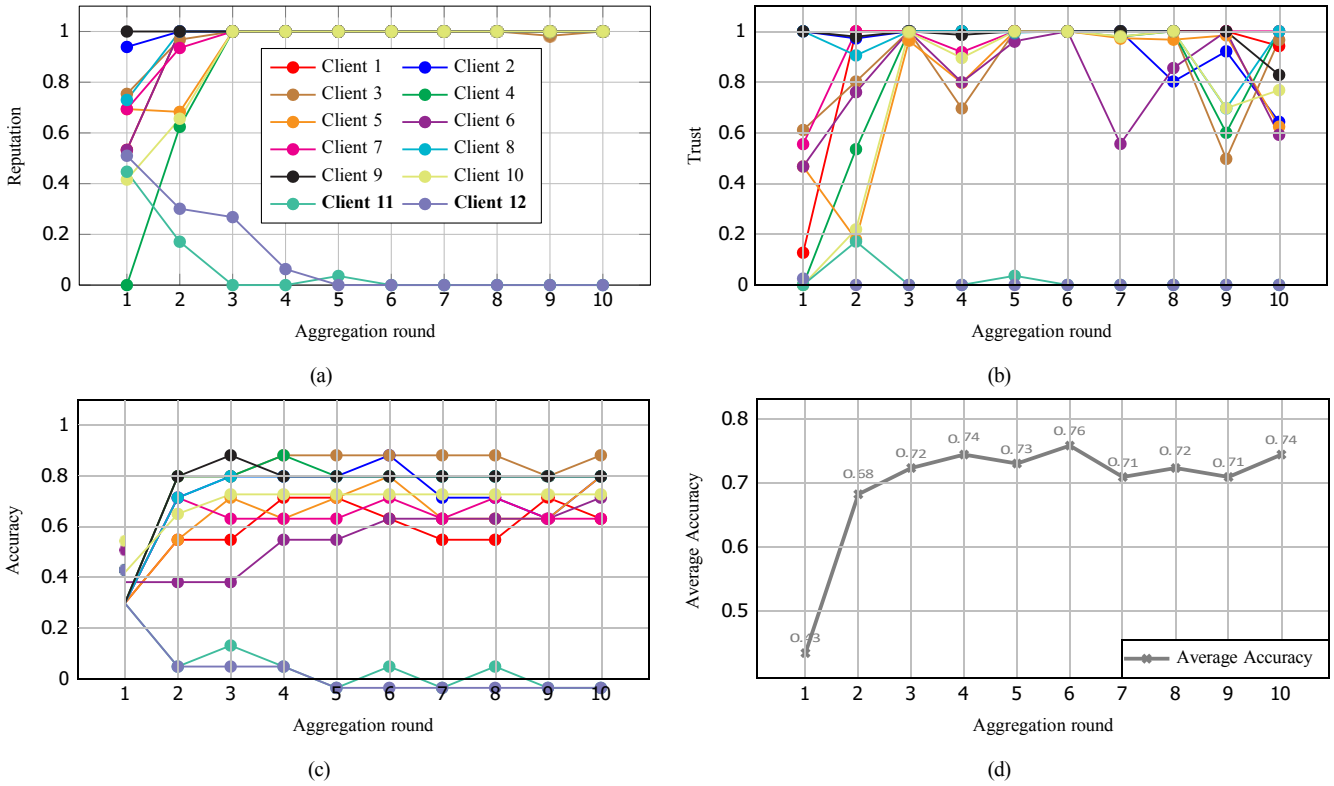

Fig. 3. Results demonstrated by the FL integrated with our Reputation and Trust indicators to manage the aggregation procedure: (a) – reputation values for each of the clients, calculated in each subsequent aggregation round; (b) – trust values for each of the clients, calculated in each subsequent aggregation round; (c) – accuracy over the local validation data for each of the clients, demonstrated after each subsequent aggregation round; (d) – average accuracy accross all the local clients, demonstrated after each subsequent aggregation round. Local data possessed by clients 11 and 12 is poisoned by the adversarial label flipping attack

techniques aim to evaluate the performance and behavior of individual clients based on their contributions to the global model and adherence to expected norms. Clients with higher Reputation and Trust scores are prioritized during model aggregation, while those exhibiting suspicious behavior are subject to penalties or exclusion from aggregation process.

VI. Reputation and Trust Strategy Verification and Results

To integrate Reputation and Trust techniques into our FL environment, we implemented a systematic approach that leveraged client behavior analysis and dynamic adaptation. Below is the description of our implementation.

A. Warm-up Rounds for Reputation and Trust Calculation

We initiated the FL process with three warm-up rounds, allowing the system to initialize and calculate Reputation and Trust scores for each client. During these rounds, clients continued to participate in the aggregation process, but no actions were taken based on their trust values. This phase served as a preparation stage for the subsequent application of trust-based exclusion policies.

B. Client Removal Based on Low Trust

Following the warm-up rounds, starting from the fourth round, we introduced a policy to remove the client with the lowest trust score from the aggregation process. This proactive approach aimed to mitigate the influence of potentially untrustworthy clients on the integrity of the global model. By systematically excluding clients with low trust values, we sought to safeguard the quality and reliability of the FL process.

C. Dynamic Trust Threshold for Client Exclusion

In subsequent rounds beyond the warm-up phase, we employed a dynamic trust threshold to determine whether a client needs to be excluded from the aggregation process. Clients whose trust values fell below the predefined threshold, set at 0.15 in our experiment, were automatically eliminated from participating in model aggregation. This threshold-based approach allowed for adaptive client management, ensuring that only reliable and trustworthy clients contributed to the collaborative learning process.

D. Evaluation and Result Analysis

Upon completing the FL process, we conducted a comprehensive analysis of the outcomes and performance metrics. Specifically, we plotted three graphs to visualize the results obtained through the Reputation and Trust strategy:

- **Reputation value of each client for each subsequent round.** This graph depicts the evolution of reputation

scores for individual clients over the course of the FL process, providing insights into their relative consistency and similarity among each other. Notably, clients 11 and 12 (adversarial clients) began with relatively high reputation values close to 0.5, but experienced a sharp decline below the threshold within a few rounds. Conversely, client 4 initially possessed a reputation value of 0 but notably improved its standing within just three rounds of the FL process (Figure 3(a)).

- **Trust Value of each client for each subsequent round.** The trust graph offers a compelling narrative, particularly showcasing clients 11 and 12 with consistently low trust values from the initial rounds. Conversely, other clients demonstrated an ability to improve their trust values over subsequent rounds, contrasting with the persistent low trust observed in clients 11 and 12. Notably, the Reputation and Trust strategy implemented in round 4 led to the removal of client 12 due to its lowest trust value, followed by the exclusion of client 11 in round 6 as its trust value fell below the threshold. Despite fluctuations, other clients managed to maintain their trust values above the threshold, ensuring their retention within the cluster (Figure 3(b)).
- **Accuracy of individual client for each subsequent round.** Initially, all clients exhibited accuracies ranging from 40 to 50%. Remarkably, the majority of clients, barring adversarial clients, demonstrated a steady improvement in accuracy over the span of 10 rounds. Notably, the decline in accuracy observed in clients 11 and 12 can be attributed to their reception of global updates from the server, albeit updating these parameters with poisoned data, thereby adversely affecting performance on validation data devoid of such contamination (Figure 3(c)).
- **Average accuracy across all clients for each subsequent round.** Finally, we analyzed the average accuracy of clients' local ML models across different rounds, demonstrating the impact of trust-based client exclusion on the overall performance and convergence of the FL system (Figure 3(d)).

Overall, the implementation of Reputation and Trust techniques yielded valuable insights into client behavior and facilitated the effective management of participant contributions in the FL process.

VII. CONCLUSION

In this paper, we proposed and developed a novel approach to identify adversarial attacks in FL by incorporating our Reputation and Trust techniques into the aggregation procedure. Without compromising the confidentiality of each client's local data, our approach enabled us to calculate and quantify trust towards each of the clients. The developed trust evaluation calculus takes into account the past local model updates' evaluations that each local client has submitted for aggregation. This makes the trust assessment process more comprehensive and reliable since it considers the contributions of each client in the previous rounds. We verify our approach experimentally by applying the strategy to a real-world ITS computer vision-based traffic sign classification application. Our results demonstrated that our Reputation and Trust-equipped FL aggregation could successfully identify clients that compromised by adversarial attacks against the training data and eliminate them from the aggregation, enhancing FL security and robustness to adversarial actions. In addition, our approach does not impose any extra communication burdens on the FL cyberinfrastructure as it employs only the local model updates that are an ordinary FL round communication component, which open avenues for its complementary employment to other FL security improvement methods and techniques.

REFERENCES

[1] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the gdpr perspective," *Computers & Security*, vol. 110, p. 102402, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404821002261

[2] A. Khan, Y. Li, X. Wang, S. Haroon, H. Ali, Y. Cheng, A. R. Butt, and A. Anwar, "Towards cost-effective and resource-aware aggregation at edge for federated learning," in *2023 IEEE International Conference on Big Data (BigData)*. Los Alamitos, CA, USA: IEEE Computer Society, dec 2023, pp. 690–699. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/BigData59044.2023.10386691

[3] H. Jeong, H. Son, S. Lee, J. Hyun, and T.-M. Chung, "Fedcc: Robust federated learning against model poisoning attacks," 2022.

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

[5] S. Chuprov, M. Memon, and L. Reznik, "Federated learning with trust evaluation for industrial applications," in *2023 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2023, pp. 347–348.

[6] S. Chuprov, I. Viksnin, I. Kim, E. Marinenkov, M. Usova, E. Lazarev, T. Melnikov, and D. Zakoldaev, "Reputation and trust approach for security and safety assurance in intersection management system," vol. 12, no. 23. MDPI, 2019, p. 4527.

[7] S. Otoum, N. Guizani, and H. Mouftah, "On the feasibility of split learning, transfer learning and federated learning for preserving security in its systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7462–7470, 2023.

[8] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2022.

[9] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2023.

[10] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

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

[12] Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu, "Fedmgda+: Federated learning meets multi-objective optimization," *arXiv preprint arXiv:2006.11489*, 2020.

[13] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5132–5143.

[14] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *arXiv preprint arXiv:2002.06440*, 2020.

[15] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020.

[16] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020.
[17] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farhad, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," 2019.
[18] F. Sattler, K.-R. Mu¨ller, T. Wiegand, and W. Samek, "On the byzantine robustness of clustered federated learning," 2020.
[19] L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning: Attacks and defenses," 2022.
[20] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusma˜o, and N. D. Lane, "Flower: A friendly federated learning research framework," 2022.
[21] S. Chuprov, K. M. Bhatt, and L. Reznik, "Federated learning for robust computer vision in intelligent transportation systems," in *2023 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2023, pp. 26–27.