# Trust-Based Anomaly Detection in Federated Edge Learning

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

2024 IEEE World AI IoT Congress (AIIoT) | 979-8-3503-8780-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIIoT61789.2024.10578967

***Abstract*—We present a novel approach for anomaly detection in a decentralized federated learning setting for edge units. We propose quantifiable metrics of Reputation and Trust that allow us to detect training anomalies on the local edge units during the learning rounds. Our approach can be combined with any aggregation method used on the server and does not impact the performance of the aggregation algorithm. Moreover, our approach allows to perform an audit of the training process of the participating edge units across training rounds based on our proposed metrics. We verify our approach in two distinct use cases: financial applications with the objective to detect anomalous transactions, and Intelligent Transportation System supposed to classify the input images. Our results confirm that our approach is capable of detecting training anomalies and even improving the effectiveness of the learning process if the anomalous edge units are excluded from the training process.**

***Index Terms*—Edge AI, IoT, Federated learning, trust, anomaly detection**

## I. Introduction

EDGE computing refers to a distributed computing paradigm where the computations are performed closer to the data sources, i.e. on the *edge*. With the recent developments in network infrastructure, Internet of Things (IoT) computational capabilities, and AI algorithms, Edge AI is becoming a viable alternative to conventional AI, where the model is trained on a centralized server. Several decentralized learning approaches have already been developed, including *Federated Learning* (FL) [1]. In classical FL, the training process is split between communication rounds. Each communication round a client performs several training iterations for a local model and then communicates the weights of the model to the aggregation server, which uses some aggregation scheme to produce a global model, which is then again distributed between clients for further training. The benefits of FL, which are (1) ability to train on real world data and (2) keeping the data private, are especially relevant in Edge AI. For instance, consider an ATM machine that performs transaction anomaly detection on the spot using a deep neural network (DNN) model. Such scenario would require privacy and confidentiality of the data on the local ATM machine, so FL can be employed as a way to train a local detection model. Another Edge AI use-case where FL can be employed beneficially [2] is an intelligent transportation system (ITS).

Despite its benefits, there are several ways the FL process can be abused, which can lead to unreliable models at each edge device. For instance, the local edge units sending their updates to the aggregation server might already be compromised. The adversary can then apply *data inference* attacks to derive sensitive information [3] from the updated global model received from the aggregation server. Such attacks can ultimately compromise the privacy of the local unit's data [4] solely by exploiting the model's gradient updates. Besides private data leakage, the adversary might interfere with the training process by performing *data poisoning* attacks on the edge client and sending malicious gradient updates to the aggregation server [5]. If the aggregation technique used by the server is not *byzantine-robust* [6], the whole model is compromised, and any inference results by this model cannot be trusted. In this work, we focus on the latter issue and derive our method to help prevent *data poisoning attacks*.

While many aggregation schemes have been proposed since the inception of FL, including [7]–[9], there is no consensus as to whether which one approach can facilitate a robust FL process in the real world. Indeed, some of the approaches proposed in literature outperform others in a particular setting under certain assumptions, but no approach is universal. Ultimately, it is up to the implementer to decide which aggregation technique to use. In this paper, we introduce a novel trust-based anomaly detection approach in federated edge units. The motivation behind our proposed solution is to develop a framework for finding anomalies in the federated learning process without impacting a particular aggregation algorithm that is used, while also allowing for an auditable FL process in a potentially malicious setting. Our approach proposes a novel trust and reputation calculus, which allows us to quantify the amount of trust an aggregation server can have in its edge units and identify anomalous and potentially malicious edge units based on the **model updates** the local edge units send to the aggregation server. Our approach does not interfere with any aggregation methods used, and in fact can be combined with any aggregation method chosen by the application author. The application stakeholders have the additional benefit of tracing the aggregation history and inspecting the behavior of anomalous units over the history of the training process based on the proposed indicators, while the aggregation server can also potentially exclude any anomalous edge units from participating in the current FL process in order to prevent further training degradation. We summarize our Trust-based

This work was supported in part by the National Science Foundation, USA, award #2321652

979-8-3503-8780-3/24/$31.00 ©2024 IEEE

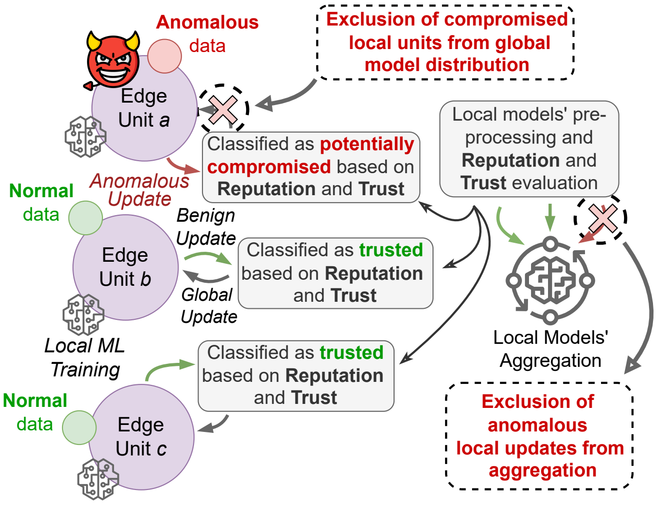


Fig. 1. Trust-based anomaly detection in Edge Federated Learning. The local models' aggregation is preceded by evaluating Trust and Reputation. Anomalous units can then be excluded from the aggregation process.

anomaly detection approach in Figure 1.

Many FL publications employ well-curated, balanced, and standardized datasets, such as MNIST [10], CIFAR-10 [11], and others. To evaluate the viability of our approach, we study its performance in two real-world Edge AI use-cases. In the first use case, we consider the case of anomalous financial transactions, where multiple edge (which could be ATM machines or banking branches) units are potentially compromised by an adversary. In this case the adversary's goal would be to adjust the model that detects anomalous transactions in such a way so that the fraudulent transactions are not considered anomalous anymore. We evaluate how our approach helps to improve the detection of transaction anomalies under malicious settings. In the second use case, we evaluate the performance of an image classification model in ITS in the traffic sign classification scenario. We provide an audit of the reputation of each edge unit. Additionally, in this scenario we evaluate the performance of our model, which uses FedAvg [1] alongside our anomaly detection method to guide the aggregation process, against a model that simply used the FedAvg aggregation. To simulate the data anomalies in both experiments we employ two distinct types of label flipping attacks. In the first type, the adversary maliciously modifies the labels of the unit by inverting them, i.e., by changing the anomalous transaction label to the benign one, and vice versa. The second type incorporates changing all the local data labels on the unit to the specific ones. We demonstrate that our approach allows to improve the performance of the FL process, regardless of the aggregation method, and enables auditing the aggregation process by inspecting the historical trust and reputation indicators, thus making the entire learning process more explainable.

## II. Related Work

Recently, Trust-driven approaches gained attention in enhancing the learning effectiveness and security of FL , e.g., to improve security and safety in ITSs [12]. However, the Trust and Trustworthiness definitions in the literature may vary, and there is no consensus on how to better approach, gauge, and decide if the FL unit is trusted. Below, we review the Trust in FL applications from various perspectives and discuss our Reputation and Trust evaluation.

Approaches that employ the concept of Trust to optimize communication in FL are mainly focused on establishing and maintaining trustworthy interactions between the participating nodes in a decentralized setting. These approaches recognize the importance of secure and efficient communication while preserving data privacy and integrity. For example, Gholami *et al.* [13] leverage the concept of Trust to evaluate the trustworthiness of the network nodes in a decentralized FL setting. To address the problem of adversaries sending malicious updates in mobile network environments, Kang *et al.* [14] propose to use Reputation as a fair indicator to select network units participating in the aggregation.

Cao *et al.* [7] propose FLTrust method, which uses trust bootstrapping technique to defend against Byzantine attacks in the FL environment.

Rjoub *et al.* [15] propose DDQN-Trust, a trust-based double deep Q-network reinforcement learning algorithm for IoT devices, which involves monitoring their CPU and RAM consumption in order to optimize the local units selection in FL. To address the privacy and security caveats in conventional FL, various approaches that leverage cryptographic methods and frameworks have been introduced to prevent data and model poisoning attacks, and mitigate their consequences, such as EIFFeL [16], RoFL [17], and ELSA [18]. Blockchain is another technology that also relies on the concept of Trust and is deemed as a perspective solution to address security, reliability, traceability, and other challenges faced by the FL systems [19]–[23].

As one can see, trust in FL can reflect the reliability, credibility, and even security of the edge units. While many approaches consider trust as a concept with no quantification, which is usually based on satisfying some requirements, for example, if the updates can be verified using cryptographic proofs or blockchain, we proceed in another direction and introduce the methods and calculus that allow to quantify the trust and reputation of local edge units.

## III. Trust Evaluation in Federated Edge Units

Our method relies on clustering the models trained over local data provided by the edge units. Anomalous local edge units might generate data with distribution patterns that strongly deviate from the majority of the edge units, and this will be reflected in the model parameters that will be communicated to the aggregation server for the global model. We assume that the majority of the edge units are not compromised and thus a shift in the edge unit's model parameters can be an indication of anomalous data patterns potentially caused by an attack.

To preserve the **local data privacy**, we utilize models trained over the local data and analyze their *trainable* parameters distribution. Clustering in the space of trainable parameters is a universal approach that allows us to analyze any learning model's behavior without restrictions on what specific task the model is aimed to perform. Another possibility is to cluster based on the model's residuals, however, that would possibly require additional communication between units and modification of the conventional FL flow. Clustering based on trainable parameters does not require any additional communication or modifications to the FL flow and preserves local data privacy (assuming the local units are not compromised). Before the aggregation, we cluster the received local models based on the *k-means* clustering method in their trainable parameter space, and calculate the distance from the cluster's center to each model. Then, based on the clustering results, we calculate our Reputation and Trust indicators that we employ for detecting and discarding compromised local units from the aggregation and further communication. Below we formalize our Reputation and Trust evaluation approach.

### A. Reputation and Trust Calculus

Following our previous efforts in [24], [12], [25], we define two basic metrics: *Reputation (R)* and *Trust*. *Reputation* captures the historical information of the differences between the model submitted by a given edge unit and all other edge units. However, *Reputation* alone is not robust to outliers and can be manipulated. That is why we introduce *Trust* as a function of *Reputation* that is more robust. The value of $R$ is calculated based on the normalized Euclidean distance $d$ from the cluster center of the model parameters submitted for aggregation by the local edge units. At the first communication round $t_0$ the value of $R$ is initialized as (1):

$$R_{t_0}^{(i)} = 1 - d_{t_0}^{(i)} \quad R_{t_0}^{(i)}, d^{(i)} \in [0, 1] \tag{1}$$

where $R_{t_0}^{(i)}$ and $d_{t_0}^{(i)}$ refers to the initial reputation value and distance to the cluster center of edge unit $i$ respectively. The value of $R$ is updated in each aggregation round according to the following (2):

$$R_t^{(i)} = \begin{cases} R_{t-1}^{(i)} + d_t^{(i)} - \frac{R_{t-1}^{(i)}}{t}, & \text{if } d_t^{(i)} \leq \alpha, \\ R_{t-1}^{(i)} + d_t^{(i)} - e^{1-d(\frac{R_{t-1}^{(i)}}{t})}, & \text{if } d_t^{(i)} > \alpha. \end{cases} \tag{2}$$

$\alpha$ is a specified threshold, which dictates how sensitive the *Reputation* indicator is to outliers and should be chosen empirically. In our applications we use $\alpha = 0.5$. Notice that such formulation allows to penalize local edge units heavily for anomalous models and requires them to consistently provide models that fall within the normal pattern to build up their *Reputation*. To cap the values of $R$ between 0 and 1, we also employ the following rule (3):

$$R_t^{(i)} := \begin{cases} 1, & \text{if } R_t^{(i)} \geq 1, \\ 0, & \text{if } R_t^{(i)} \leq 0, \\ R_t^{(i)}, & otherwise. \end{cases} \tag{3}$$

Based on $Reputation$, the $Trust$ indicator is calculated, which is a function of $Reputation$ that regulates how the change in $Reputation$ affects the $Trust$ an aggregation server can have in an edge unit. Additionally, it is possible to exclude a local unit with a low level of $Trust$ from following communication rounds to avoid the malicious manipulations on the global model. We formalize the definition of $Trust$ as follows (4):

$$\begin{aligned} Trust_t^{(i)} = {} & \sqrt{(R_t^{(i)})^2 + (d_t^{(i)})^2} - \\ & - \sqrt{(1 - (R_t^{(i)})^2 + (1 - d_t^{(i)})^2}, \end{aligned} \tag{4}$$

$$Trust_t^{(i)} \in [0, 1]$$

To bound the values of $Trust$ between 0 and 1, we use the same rule as for reputation:

$$Trust_t^{(i)} := \begin{cases} 1, & \text{if } Trust_t^{(i)} \geq 1, \\ 0, & \text{if } Trust_t^{(i)} \leq 0, \\ Trust_t^{(i)}, & otherwise. \end{cases} \tag{5}$$

Below we discuss the advantages our Trust and Reputation evaluation-driven learning anomaly detection offers:

1) Historical tracking: this feature allows accumulating and tracing the changes in Trust towards local units over time in the FL system. The retrospective information on the quality of models the edge units provide for aggregation is employed for Trust evaluation, enabling the identification of anomalous (potentially untrustworthy) units. This feature might be highly useful and employed, for example, for extensive security analytics and audit [26].

2) Aggregation-method agnostic: as mentioned previously, our anomaly detection approach works with any aggregation method that is used in a decentralized learning setting and does not interfere with the aggregation process. As such, the convergence rate of the selected aggregation algorithm will stay the same, however, the security of the learning process can be enhanced with addition of our anomaly detection approach.

3) Trust-based filtering: instead of selecting local units for aggregation [14], it is possible to decide which units should be discarded from aggregation based on Trust towards them. We develop calculus that quantifies Reputation and Trust indicators, and allows to evaluate them in a specified numerical range. Our calculus allows flexibility and personalization as the FL system user is able establish a specified level of Trust, based on the thresholds $\alpha$ and $\beta$, deemed acceptable for their application and context requirements. Untrusted local units can be excluded from further communication, preventing them

from receiving further global updates, which enhances privacy and security of FL.

4) Unsupervised clustering: our approach does not require any prior knowledge on the training data and its distribution. It solely relies on the models' updates sent by the local units to the aggregation unit. These updates are clustered in an unsupervised manner, and the Reputation and Trust are calculated based on these clustering results. Hence, our approach does not need access to ground truth or the local data distribution in advance for Trust estimation. In addition, the employment of unsupervised clustering requires only the model updates as an input, which makes the approach adaptable to many data types and FL applications.

To summarize, compared to previous approaches [7], [27], we believe that our approach allows for a more customizable and auditable FL process based on the values of $\alpha$ and $\beta$, and also does not require communicating a ground-truth model, which incurs a certain degree of privacy violation to the local clients' data.

## IV. Verification in Industrial Applications

To verify our novel Trust-based approach, we investigate two use cases that represent distinct industrial applications of Edge AI. First, we study the scenario of training the FL model on a dataset of financial transactions with a target to identify the anomalous ones (anomalous financial transactions scenario). Second, we examine the ITS image classifier supposed to categorize real traffic sign images (ITS traffic sign classification scenario). In both cases, we employ data poisoning attacks that affect some portion of the local units in the FL system. We assess how effective is our Trust evaluation approach in detecting the compromised local units and how it influences the performance of the trained model. Below we describe each of these use cases.

### A. Anomalous Financial Transactions Identification Case

We employ the industrial dataset provided by SWIFT[1] and used in U.S. PETs Prize Challenge[2] hosted by NIST and NSF, which incorporates about 4 million records on financial transactions performed over a 30 days interval. As the original dataset is highly imbalanced with almost 95% of the data being benign transactions, we utilize the SMOTE library [28] to avoid oversampling. For our FL training, we divide our data into 10 cohorts based on the records pertaining to a particular bank origin. Each of these cohorts has a roughly equal number of records and is composed of similar attributes. To select the appropriate basic ML topology, we compare various models based on their performance achieved after a centralized ML training: Deep Neural Network (DNN) and a Random Forest (RF) classifier. The performances demonstrated by these two ML architectures are given in Figure 2(a). From these results, one can see that DNN is able to achieve about 74% AUC in contrast to ≈63% showed by RF on a test set. Hence, we select DNN for the further FL empirical study. The best DNN topology, based on the the Area under the ROC Curve (AUC) metric, incorporates four dense layers: the first layer has 200 neurons, takes an input shape of 9 features, and employs the Rectified Linear Unit (ReLU) activation function; the next three layers have 100, 50, and 25 neurons, respectively, all employing ReLU as well. We also employ a dropout rate of 0.5, which results in preventing model's overfitting. The last layer has a single neuron with the Sigmoid activation function, which is typically used for binary classification problems.

TABLE I
Reputation And Trust values calculated for each local unit after the first local training round

| $d$ **from cluster center** | **Client ID** | $R$ | $Trust$ |
|---|---|---|---|
| 0 | Unit 9 | 1 | 1 |
| 0.139 | Unit 7 | 0.860 | 1 |
| 0.184 | Unit 6 | 0.815 | 0.893 |
| 0.289 | Unit 5 | 0.710 | 0.596 |
| 0.296 | Unit 8 | 0.703 | 0.576 |
| 0.325 | Unit 1 | 0.674 | 0.494 |
| 0.444 | Unit 4 | 0.555 | 0.156 |
| 0.461 | Unit 10 | 0.536 | 0.109 |
| 0.866 | **Unit 3** | **0.133** | **0** |
| 1 | **Unit 2** | **0** | **0** |

To recreate data poisoning attack scenario, we modify the original SWIFT data collection with label flipping attacks on two of the local units. Specifically, we explore two types of label flipping attacks. On unit 2, we invert labels in the local training data, i.e., we change the labels pertaining to anomalous transactions to benign ones and vice versa. On unit 3, we change labels of all the transactions in the stored local training data to anomalous ones. We employ these two types of malicious manipulations to augment local data on unit 2 and unit 3 respectively, while leaving the other clients' data intact. We assume that the attackers have full control over the local data and models of these clients, but not over the communication or aggregation process.

Initially, the constructed DNN model is distributed across 10 distinct local edge units from the aggregation server via encrypted socket communications. After the model is distributed, the local training process is initiated by each unit. The model is trained for 100 consequent epochs. To calculate the Reputation and Trust indicators, we employed k-means clustering of the model parameters submitted for the aggregation. We found the major cluster's center and calculated the Euclidean distances $d$ to each of the models, normalized $d$ in the range between 0 and 1, and employed the clustering results to initialize the Reputation indicator $R$ for each client. The initial Reputation was calculated according to (1), which means that the farther the model lies from the cluster center, the lower Reputation it receives. Then, the Trust indicator was calculated based on $R$, as described in sec. III. The Reputation and Trust indicators were updated in each aggregation round.

We evaluated the Reputation and Trust indicators for the units submitted models for aggregation after the first local

[1] https://www.swift.com/

[2] https://www.drivendata.org/competitions/98/nist-federated-learning-1/page/522/

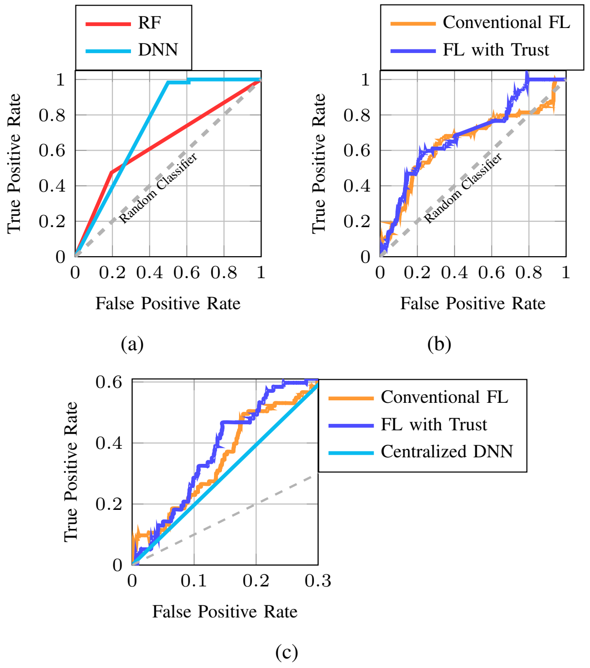


Fig. 2. Performance comparison of various models based on the ROC curve generated on a testing dataset: (a) – DNN vs RF trained in a centralized manner; (b) – DNN trained in a FL manner with and without the proposed Reputation and Trust-based indicators; (c) – performance of the advanced and conventional FL model vs conventional ML model within the most important in practical applications False Positive Rate region

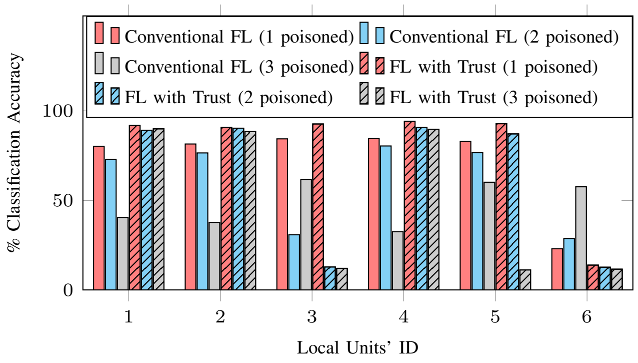


Fig. 3. Average performance demonstrated by both models, trained in a conventional FL manner and using our anomaly detection for exclusion in aggregation, and tested over the local data on each of the local units. The results are presented for the cases with the presence of 1 (unit 6), 2 (units 3 and 6), and 3 (units 3, 5, and 6) compromised local units

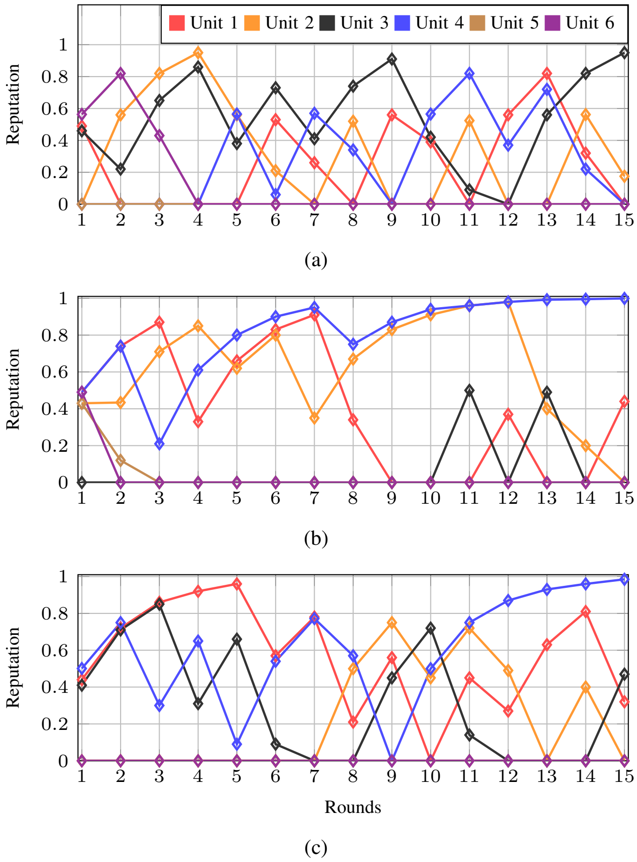
training round. As can be seen from Table I, units 2 and 3 received the lowest Reputation and Trust values even after the first training round, which indicates that they provided models lying out of the major distribution. The small differences in Trust values between the units 4, 10, 3, and 2 correspond to the retrospective nature of the Trust indicator – it requires to accumulate some historical data during several aggregation rounds to reflect the changes in the Reputation more accurate. In this particular case, just after the first training round, it is more reasonable to use Reputation as the decision-making criteria to detect compromised local units. Our approach is capable of detecting the compromised units even before aggregating the local updates, which means that, based on the Reputation and Trust values, they can be excluded from the further aggregation procedure and from the global model distribution. Discarding from the aggregation procedure will prevent the influence of harmful updates on the global model, and preventing compromised local units from receiving the global model will enhance the security and privacy of FL.

After discarding two units with the lowest Reputation, we continue the FL training process under normal conditions without data poisoning attacks. The local updates are aggregated using the FedAvg [1] to produce a global model. First, we evaluated the model trained in a conventional FL manner [29], without employing the Reputation and Trust indicators. Orange line in Figure 2(b) represents the performance demonstrated by this model on a test set. According to the results obtained, FL model demonstrated AUC of around 0.72, which was comparable to one achieved in the centralized learning case. Blue line in Figure 2(b) represents the performance results for the FL process that employs the proposed Reputation and Trust mechanisms with edge unit exclusion. In Figure 2(c) one can see the scaled version of these results over the more practical False Positive Rate values interval. The produced model evaluated over the testing data was able to achieve AUC of 0.77, which outperformed both models trained in centralized and conventional FL manners. Based on the results, the employed Reputation and Trust-based mechanisms demonstrated the FL model performance improvement.

### *B. Image Classification in Intelligent Transportation System Case*

In this use case, we employ a subset of real traffic images from Open Images V6 dataset [30], categorized into traffic and stop sign labels. As the ML model architecture, we employ a custom Convolutional Neural Network (CNN) consisting of 10 layers designed to be relatively small in scale. Our FL setup for this case incorporates six local units in each training round, and we also use the FedAvg aggregation strategy in each of the 15 training rounds. Initially, we randomly allocate images over 6 local edge units, each unit has around 120 images of each category. We conduct FL training using the conventional FL aggregation as well as FL training with the help of our Trust-based approach - to exclude local units identified as compromised from the aggregation. Our experiments include scenarios with one, two, and three local units compromised by the label-flipping attack. In this case, during the aggregation procedure, we remove two local units in each round with the lowest Reputation scores. In general, the application developers are free to choose any indicator between Reputation and Trust, and here we wanted to demonstrate that Reputation alone can be used for anomaly detection, although it may not be a very robust indicator.

Figure 3 represents average performance values demonstrated by the models trained over 15 rounds and tested over



(a)

(b)

(c)

Fig. 4. Reputation values for each local unit calculated in each consequent aggregation round for the scenario with: (a) – 1 compromised unit (unit 6 is compromised); (b) – 2 compromised units (units 3 and 6 are compromised); and (c) – 3 compromised units (units 3, 5, and 6 are compromised)

the data residing on each local unit in the end of each round. From these results, one can see that the ML model trained with our anomaly detection outperformed the conventional FL setup in terms of performance on each unit except ones that possess poisoned data, which can be observed in all investigated cases. The anomalous units were excluded from further training rounds, and this exhibited lower performance. Moreover, a relationship between the increase in the number of compromised units and the trained model's performance drop can be observed. The employment of our anomaly detection with unit exclusion allowed to achieve an average 14.8% performance growth while tested over the units possessing original non-compromised data with two compromised edge units; and in the case of three compromised units, the difference in performance became significant and achieved 130.8%. In all three cases, the performance of the ML model trained using edge unit exclusion based on anomaly detection experiences only marginal deviations while introducing additional compromised units.

Figures 4(a), 4(b), and 4(c) show results of the Reputation value changes for each of the local units over the 15 training rounds, which can be used for an audit of the learning process across communication rounds. From these results, one can observe that the Reputation values of all of the poisoned units either start at zero or eventually converge to zero after a number of training rounds. For instance, from Figure 4(a) we can see that unit 6 (the compromised one) demonstrates high Reputation score after the first training round, and it is not excluded from the aggregation in the second and third training rounds. However the Reputation value of this unit rapidly declines to zero after the third training round, and the further updates are excluded from the aggregation in all the remaining training rounds. In some cases, non-compromised local units also received low Reputation on par with the compromised ones, which can be considered as a false positive. The potential reason behind this is unbalanced local data distribution across the units. Our results demonstrate that, in some cases, our Trust-based anomaly detection may cause some false positives initially and requires some time for the correct identification of the units that possess compromised data. In other cases, our historic knowledge accumulation helps to correctly detect units that possess poisoned data at the early training stages.

## V. Conclusion

In this paper we proposed a method for Trust-based anomaly detection in edge units that participate in the federated learning process. We developed the calculus necessary to give a quantifiable definition of Trust and Reputation indicators in the FL setting based on the parameters that each edge unit submits to the aggregation server. We verified the validity of our approach in two industrial Edge AI use-cases. We demonstrated that our method allows for (1) historical tracking of the aggregation process and performing audits on the aggregation across training rounds, (2) anomaly detection independent from the aggregation method, (3) filtering of edge units based on the proposed indicators, (4) ground-truth-free learning anomaly detection.

## References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] D. M. Manias and A. Shami, "Making a case for federated learning in the internet of vehicles and intelligent transportation systems," *IEEE Network*, vol. 35, no. 3, pp. 88–94, 2021.

[3] A. Suri, P. Kanani, V. J. Marathe, and D. W. Peterson, "Subject membership inference attacks in federated learning," *arXiv preprint arXiv:2206.03317*, 2022.

[4] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, "When the curious abandon honesty: Federated learning is not private," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2023, pp. 175–199.

[5] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *European Symposium on Research in Computer Security*. Springer, 2020, pp. 480–501.

[6] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the works of leslie lamport*, 2019, pp. 203–226.

[7] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," *arXiv preprint arXiv:2012.13995*, 2020.

[8] 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.

[9] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International conference on machine learning*. PMLR, 2021, pp. 6357–6368.

[10] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[11] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)," accessed on July 30, 2023. [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[12] S. Chuprov, I. Viksnin, I. Kim, L. Reznik, 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)*. IEEE, 2020, pp. 1–8.

[13] A. Gholami, N. Torkzaban, and J. S. Baras, "Trusted decentralized federated learning," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 1–6.

[14] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[15] G. Rjoub, O. A. Wahab, J. Bentahar, and A. Bataineh, "Trust-driven reinforcement selection strategy for federated learning on iot devices," *Computing*, pp. 1–23, 2022.

[16] A. Roy Chowdhury, C. Guo, S. Jha, and L. van der Maaten, "Eiffel: Ensuring integrity for federated learning," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 2535–2549.

[17] H. Lycklama, L. Burkhalter, A. Viand, N. Küchler, and A. Hithnawi, "Rofl: Robustness of secure federated learning," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 453–476.

[18] M. Rathee, C. Shen, S. Wagh, and R. A. Popa, "Elsa: Secure aggregation for federated learning with malicious actors," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1961–1979.

[19] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for ai: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10 127–10 149, 2019.

[20] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2019.

[21] Y. Qu, M. P. Uddin, C. Gan, Y. Xiang, L. Gao, and J. Yearwood, "Blockchain-enabled federated learning: A survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–35, 2022.

[22] J. Zhu, J. Cao, D. Saxena, S. Jiang, and H. Ferradi, "Blockchain-empowered federated learning: Challenges, solutions, and future directions," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–31, 2023.

[23] X. Bao, C. Su, Y. Xiong, W. Huang, and Y. Hu, "Flchain: A blockchain for auditable federated learning with trust and incentive," in *2019 5th International Conference on Big Data Computing and Communications (BIGCOM)*. IEEE, 2019, pp. 151–159.

[24] 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," *Energies*, vol. 12, no. 23, p. 4527, 2019.

[25] S. Chuprov, I. Viksnin, I. Kim, T. Melnikov, L. Reznik, and I. Khokhlov, "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)*. IEEE, 2021, pp. 115–120.

[26] J. P. Pironti, "Five key considerations when applying a trust, but verify approach to information security and risk management," 2021, accessed on July 30, 2023. [Online]. Available: https://www.isaca.org/resources/news-and-trends/newsletters/atisaca/2021/volume-36/five-key-considerations-when-applying-a-trust

[27] R. Al Mallah, D. Lopez, G. Badu-Marfo, and B. Farooq, "Untargeted poisoning attack detection in federated learning via behavior attestation," *IEEE Access*, 2023.

[28] W. Satriaji and R. Kusumaningrum, "Effect of synthetic minority oversampling technique (smote), feature representation, and classification algorithm on imbalanced sentiment analysis," in *2018 2nd International Conference on Informatics and Computational Sciences (ICICoS)*, 2018, pp. 1–5.

[29] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[30] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.