Anomaly Detection in Federated Learning

A Comparative Analysis of TRIM and RONI

Görkem Orhan, José Ribeiro, Silas Pohl, Sofia Costa
Università di Bologna
Bologna, Italy
{gorkem.orhan, jose.araujoribeiro, silas.pohl, sofia.merinoferreira}@studio.unibo.it

Abstract

This report examines the vulnerabilities of Federated Learning (FL) systems to poisoning attacks and evaluates two anomaly detection methods, TRIM and RONI, in mitigating these threats. TRIM leverages an iterative trimming mechanism to filter anomalous updates, maintaining high model performance with minimal computational overhead. In contrast, RONI evaluates updates based on their impact on validation performance, which, while effective in some scenarios, is computationally expensive and inconsistent under high poisoning rates. The methods were tested on three datasets using various attack strategies, and results indicate that TRIM outperforms RONI in robustness, efficiency, and adaptability across datasets. These findings underscore the potential of TRIM as a scalable and reliable defense mechanism for FL systems, highlighting areas for improvement in anomaly detection methods and their application in adversarial settings.

I. Introduction

A. Background

In an era where privacy and data security are of great importance, Federated Learning (FL) has emerged as a transformative approach to decentralized machine learning (ML). Unlike traditional ML paradigms that rely on aggregating data into a centralized server, FL enables collaborative model training across multiple clients while keeping the data localized. Instead of sharing raw data, only model updates - such as gradients or weights - are exchanged, ensuring that sensitive information remains on the client devices [1] [2] [3]. FL also addresses challenges beyond privacy, including scalability and data heterogeneity. By distributing computation across multiple clients, FL scales well to large datasets. It also accommodates non-identical, independent data distributions across clients, reflecting real-world variability [1] [2] [3].

However, while FL reduces privacy risks, its decentralized nature introduces new vulnerabilities, particularly to adversarial threats. The decentralized architecture of FL makes it

particularly susceptible to adversarial attacks, including data poisoning, model poisoning, and backdoor attacks. In data poisoning, adversarial clients manipulate their local datasets by injecting carefully crafted or corrupted data points that degrade model accuracy or introduce systematic biases into the global model. Model poisoning attacks, on the other hand, involve adversarial clients submitting intentionally manipulated model updates - such as altered gradients or weights - during the training process. Finally, backdoor attacks represent a particularly insidious adversarial strategy, where the model is trained to behave correctly for most inputs but produce incorrect or malicious outputs for specific adversarial triggers. These triggers are designed so that the backdoor remains hidden under normal testing conditions but can be activated when adversarial inputs are presented [4]. These attacks exploit the fact that the central server cannot easily inspect or validate the contributions from individual clients [2] [3]. Unlike centralized ML systems, where outlier detection or anomaly filtering can be applied at the data level, FL requires methods to detect malicious contributions at the model update stage [4].

To address poisoning attacks, anomaly detection (AD) methods are critical for ensuring robust model training. TRIM and RONI are two prominent techniques designed to defend machine learning models from adversarial influence.

- **TRIM (Trimmed Loss Function):** A robust aggregation algorithm that filters out anomalous client updates by discarding outliers based on statistical metrics.
- **RONI (Reject on Negative Impact):** A validation-based approach that evaluates the impact of each update on a validation set and excludes those that degrade the global model's performance.

Both TRIM and RONI operate primarily on regression problems, where adversarial points can significantly impact the prediction of continuous values. However, the principles of these methods can also be extended to classification tasks, where anomalies are identified based on their misclassification impact [4]. These techniques are particularly valuable in FL settings, where adversarial devices can manipulate a fraction of the training data without direct oversight from the central server.

B. Objectives

This report aims to provide a comprehensive understanding of how poisoning attacks compromise federated learning systems and evaluates the effectiveness of two prominent AD methods, TRIM and RONI, in mitigating these threats. Specifically, the report seeks to achieve the following objectives:

1) **Examine TRIM and RONI approaches for anomaly detection:** Provide a detailed investigation into their methodologies, including their underlying principles, architectures, and the mechanisms they use to detect and mitigate poisoned data points.
2) **Evaluate the performance of TRIM and RONI:** Assess their effectiveness in reducing the impact of poisoning attacks using quantitative metrics. The evaluation is conducted based on existing experiments using datasets with varying characteristics to understand their generalizability and limitations.
3) **Compare TRIM and RONI:** Highlight the strengths, weaknesses, and operational differences between the two methods. This includes identifying conditions where one method outperforms the other and analyzing computational trade-offs.
4) **Provide insights for future research:** Discuss the findings and offer practical recommendations for improving anomaly detection in federated learning, particularly in adversarial settings.

C. Structure of the Report

The report is structured as follows:

- **Section II**. discusses the type of data and datasets that were used to evaluate the performance metrics of TRIM and RONI. The section highlights the relevance of these datasets to real-world poisoning scenarios.
- **Section III**. introduces TRIM, providing a detailed explanation of its approach, iterative architecture, and its ability to counter poisoning attacks by trimming high-residual data points. Additionally the performance metrics are presented.
- **Section IV**. focuses on RONI, outlining its methodology of rejecting data points based on their impact on model performance. Additionally the performance metrics are presented.
- **Section V**. contains a comparative analysis of TRIM and RONI, discussing their strengths, weaknesses, and performance differences.
- **Section VI**. concludes the report with key findings, summarizing the comparative results and offering insights for future research on anomaly detection in federated learning.

II. Datasets and Evaluation Setup

A. Description of Datasets

The datasets and evaluation metrics used in the following sections are based on the work of Jagielski et al. in their paper "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning" [4], where the TRIM AD method was first proposed. The metrics, including Mean Squared Error (MSE) and attack success rates, are drawn directly from their findings. To evaluate the effectiveness of TRIM and compare it to other anomaly detection techniques such as RONI, three real-world regression datasets were used. These datasets represent diverse application domains and include a mix of categorical and numerical features. Preprocessing steps, including normalization for numerical attributes and one-hot encoding for categorical variables, were applied to ensure consistency across experiments. Below is an overview of the datasets:

1) **Healthcare Dataset:** The healthcare dataset focuses on predicting the appropriate weekly dosage of the anticoagulant drug Warfarin for patients. This dataset includes a combination of demographic details, medical history, and genetic information, such as VKORC1 and CYP2C9 genotypes, as predictor variables. The response variable represents the Warfarin dosage in milligrams per week. Incorrect dosage predictions in this dataset carry serious risks, as deviations from the correct dose can result in harmful medical outcomes, such as blood clotting or excessive bleeding. After preprocessing, the dataset contains approximately 5,700 samples and 167 features [4].
2) **Loan Dataset:** The loan dataset originates from the Lending Club peer-to-peer lending platform and involves predicting loan interest rates based on various borrower and loan attributes. Predictor variables include loan-specific information (e.g., loan amount, repayment duration), borrower credit features (e.g., credit score, lines of credit), and demographic details. The original dataset contains nearly 900,000 records, with 75 features. After preprocessing, the number of features increases to 89. Due to its scale, a subset of 5,000 records was used for experimentation to maintain computational feasibility while still providing sufficient heterogeneity for anomaly detection evaluation [4].
3) **House Pricing Dataset:** The house pricing dataset involves predicting house sale prices based on a range of property characteristics. Predictor variables include numerical features such as square footage and number of rooms, as well as categorical attributes like location and building type. The dataset contains 1,460 samples and 275 features after preprocessing. [4].

B. Simulated Poisoning Attacks

To evaluate the performance and robustness of the anomaly detection methods, TRIM and RONI, simulated poisoning attacks were applied to each of the three datasets. These attacks were designed to reflect realistic adversarial strategies aimed at compromising regression models by injecting malicious data points into the training process. Two types of poisoning attacks were considered: statistical and optimization-based attacks [4]. Statistical attacks generate adversarial data points that mimic the statistical properties of the legitimate data, such as mean and covariance. These points are carefully crafted to blend seamlessly into the distribution of the training dataset, making them harder to detect as anomalies. By aligning with the overall data characteristics, statistical attacks exploit the inability of standard learning algorithms to distinguish adversarial inputs from clean data, thereby degrading model performance without being easily flagged [4]. In contrast, optimization-based attacks use advanced optimization techniques to maximize the impact of the adversarial points on the model's performance. These attacks directly target the model's objective function and manipulate the training process to introduce significant deviations in the model's predictions. Optimization-based approaches are particularly effective because they aim to identify and exploit the most vulnerable regions of the model's decision space, producing adversarial points that have a disproportionately high influence on the global model [4].

For both attack types, poisoning rates were systematically varied to examine the resilience of TRIM and RONI under different levels of adversarial influence. The poisoning rates ranged from 4% to 20% of the training data, simulating scenarios where adversarial clients control a small but potentially impactful fraction of the overall dataset [4].

C. Experimental Setup

For each dataset, the data was split into training, testing, and validation subsets using standard cross-validation techniques. Models were trained on poisoned training data and evaluated on clean test data to quantify the performance degradation caused by poisoning. The primary performance metric used throughout this report is MSE, which measures the prediction error of regression models. The experiments compare three key scenarios:

1) **Baseline MSE** The model's performance on clean, unpoisoned training data.
2) **Poisoned MSE:** The model's performance when trained on data containing adversarial points.
3) **Mitigated MSE:** The performance of models after applying TRIM or RONI [4].

III. TRIM

A. Approach and Architecture

The TRIM algorithm is a robust defense method designed to protect regression models against poisoning attacks. These attacks involve adversaries injecting malicious training points to manipulate model outcomes. Unlike traditional regression techniques that are vulnerable to such adversarial points, TRIM identifies and mitigates the influence of poisoned data while maintaining high performance on legitimate data [4].

TRIM operates through an iterative process designed to identify and exclude points with high residuals - the difference between the predicted and actual values - under the assumption that poisoned points deviate significantly from legitimate data distributions. The process can be broken down into the following steps:

1) **Initialization:** The algorithm begins with a regression model trained on all available data, including potentially poisoned points. This initial model provides baseline parameter estimates.
2) **Residual Calculation:** For each data point, the residual (the error between the model's prediction and the actual data) is calculated.
3) **Subset Selection:** TRIM selects a subset of training points with the smallest residuals, which are most likely to represent legitimate, unpoisoned data.
4) **Model Update:** The model is retrained using only the selected subset, refining its parameter estimates while ignoring high-residual (potentially poisoned) points.
5) **Iteration:** Steps 2-4 are repeated, with the model recalculating residuals and selecting a refined subset of training points in each iteration.
6) **Convergence:** The iterative process continues until the model parameters stabilize, and the loss function converges. At this point, the algorithm outputs the final regression model [4].

The iterative trimming approach allows TRIM to handle poisoned points even when they mimic legitimate data distributions, which traditional outlier detection methods often fail to address. TRIM avoids making assumptions about the underlying data distribution, enabling it to defend effectively against a wide range of poisoning attacks. TRIM includes formal guarantees about its convergence and the robustness of its parameter estimates. It ensures that, even under worst-case adversarial scenarios, the mean squared error (MSE) remains bounded and the model retains high accuracy on legitimate data [4].

B. Performance

The performance of TRIM was evaluated comprehensively through extensive experiments across multiple datasets and regression models. These evaluations focused on TRIM's ability to mitigate the effects of poisoning attacks and maintain model accuracy, robustness, and computational efficiency. The results highlight TRIM's significant advantages over traditional defenses (more on that in section V. and VI.) and its resilience under a variety of adversarial conditions [4].

TRIM consistently demonstrated superior resilience against poisoning attacks, achieving a median Mean Squared Error (MSE) increase of only 6.1% across all datasets and models. This is a substantial improvement compared to other defenses. Notably, TRIM was able to limit the maximum MSE increase to 27.2% in 80% of attack scenarios, and in certain cases, it even produced lower MSE than models trained on unpoisoned data. For example, on the healthcare dataset with a poisoning rate of 8%, TRIM reduced the MSE by 3.47% compared to the unpoisoned baseline, illustrating its ability to effectively utilize legitimate data while neutralizing poisoned points [4].

TRIM demonstrated strong computational efficiency, converging quickly even on high-dimensional datasets. For instance, on the house pricing dataset with 275 features, TRIM required an average of only 0.02 seconds to process and defend against poisoning attacks. This performance was significantly faster than other methods [4].

TRIM's robustness was demonstrated across all regression models and datasets, effectively handling both statistical and optimization-based poisoning attacks. On the healthcare dataset, TRIM significantly mitigated drastic changes in predicted Warfarin dosages, ensuring that dosage predictions remained close to their intended values. This was particularly important in preventing potentially harmful medical outcomes, where deviations in dosage could have serious consequences. On the loan dataset, TRIM successfully reduced the impact of highly non-linear poisoning attacks, which exploited the sparsity and heterogeneity of the data to maximize disruption. Despite the challenges posed by these attacks, TRIM maintained the model's performance and stability. Finally, on the house pricing dataset, TRIM proved resilient even when attackers leveraged the dataset's high-dimensional feature space to craft adversarial points. By isolating malicious contributions, TRIM preserved the accuracy of the predictions, showcasing its ability to operate effectively under complex conditions [4].

IV. RONI (Reject on Negative Impact)

A. Approach and Architecture

The Reject on Negative Impact (RONI) method is a validation-based anomaly detection technique designed to enhance the robustness of machine learning models, particularly within Federated Learning (FL) environments. Its primary function is to identify and exclude data points or client updates that adversely affect the model's performance [1].

RONI operates by evaluating the impact of individual data points or client updates on a model's performance. The process involves the following steps:

1) **Baseline Model Training:** Train the model using the existing dataset to establish a baseline performance metric [5].
2) **Impact Assessment:** For each new data point or client update, retrain the model by incorporating the new data and measure the performance on a validation set [5].
3) **Impact Evaluation:** Compare the new performance metric against the baseline. If the inclusion of the new data results in a performance degradation beyond a predefined threshold, the data is considered harmful [6].
4) **Data Rejection:** Exclude data points or client updates identified as harmful to maintain or improve the model's integrity [5].

The strength of RONI lies in its ability to directly validate updates against clean, trusted validation data, providing a principled mechanism to detect and exclude adversarial contributions. Unlike TRIM, which relies on residuals or statistical measures to identify outliers, RONI evaluates updates based on their actual impact on model performance. This makes it particularly effective in scenarios where adversarial points are subtle and designed to mimic the legitimate data distribution [4] [5]. However, RONI's reliance on a separate validation set introduces certain limitations. First, the availability and quality of clean validation data are critical for its effectiveness. If the validation set is small or not representative of the overall data distribution, RONI may fail to accurately identify harmful updates. Additionally, the process of evaluating each client update individually can be computationally intensive, particularly in large-scale federated learning scenarios with a high number of participants [5].

B. Performance

RONI's performance was evaluated across three datasets - the healthcare, loan, and house pricing datasets - under varying poisoning rates ranging from 4% to 20%. The results, measured using Mean Squared Error (MSE), demonstrated that while RONI offered improvements

over an undefended model, its effectiveness was inconsistent, particularly as the poisoning rate increased [4].

On the healthcare dataset, RONI showed significant performance degradation even at low poisoning rates. At 4% poisoning, the MSE was already elevated, nearing 0.10, and continued to rise as the poisoning intensity increased. At higher poisoning rates, such as 16% to 20%, RONI's MSE approached 0.20, indicating limited success in mitigating adversarial contributions in this critical dataset [4].

In the loan dataset, RONI performed relatively better, maintaining lower MSE values at modest attack intensities. At a poisoning rate of 4%, the MSE was close to 0.03, which represented a small deviation from the clean baseline. However, as the attack strength increased, the MSE gradually rose, reaching values near 0.04 at higher poisoning rates. While RONI demonstrated moderate resilience here, its performance was still far from optimal [4].

On the house pricing dataset, RONI initially performed well, maintaining an MSE around 0.015 at low poisoning rates. However, as the poisoning rate grew, the MSE steadily increased, reaching values above 0.025 under stronger adversarial attacks. The high-dimensional nature of the dataset made it particularly challenging for RONI to identify subtle adversarial updates effectively [4].

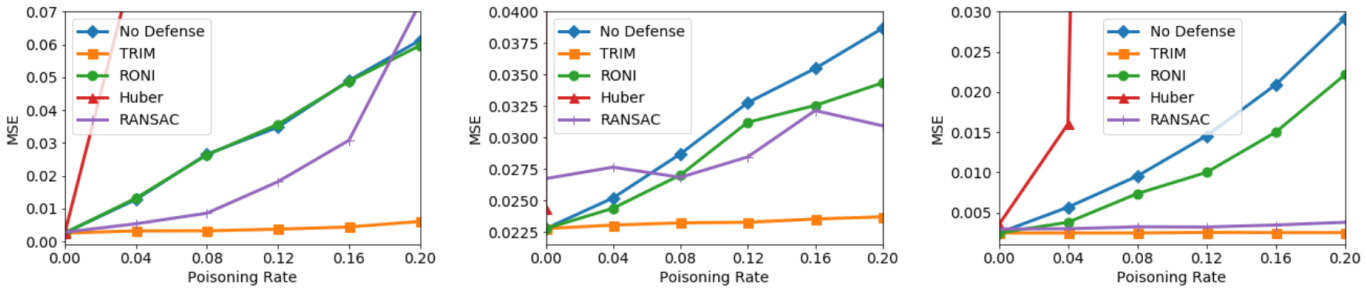
Overall, RONI provided measurable improvements compared to an undefended model but struggled to maintain consistent performance across all datasets and poisoning intensities. It showed moderate success on the loan dataset but underperformed on the healthcare and house pricing datasets, particularly when faced with higher poisoning rates or optimization-based attacks [4]. The effectiveness of RONI is contingent upon the quality of the validation set and the defined performance thresholds [5]. While it can effectively filter out harmful data, the method may be computationally intensive due to the need for retraining the model with each new data point or client update [7]. Additionally, setting appropriate thresholds is critical; overly stringent criteria may lead to the exclusion of beneficial data, whereas lenient thresholds might allow harmful data to influence the model [8] [9].

V. Comparative Analysis

A. Comparison Approach

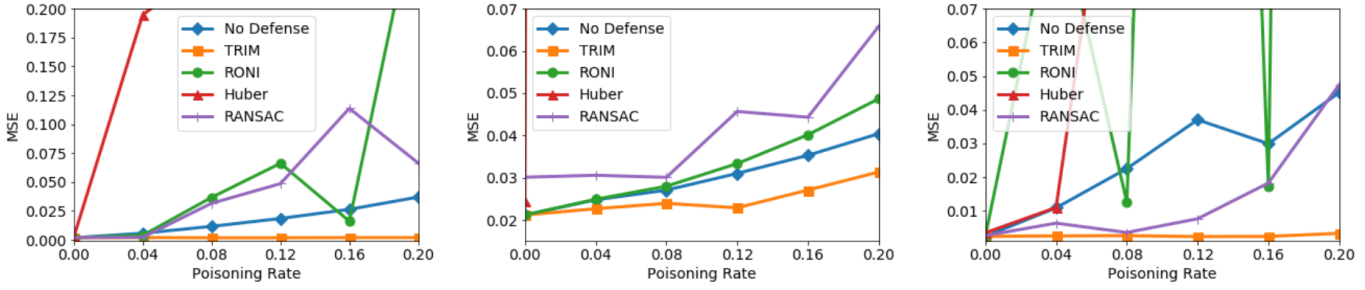
In this section, a comparative analysis of the two defense methods, TRIM and RONI, for mitigating poisoning attacks on regression models. Firstly, we present the performance graphs of each AD to evaluate their effectiveness, efficiency and robustness in defending poisoning attacks in the 3 datasets in comparison. The graphs inclue more ADs than TRIM and RONI, because the work from Jagielski et al. "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning" [4] contained a larger and more sophisticated comparison between more ADs. For our sake, we will exclusively focus on TRIM and RONI - so the orange and green graph. After that the advantages and disadvantages of TRIM and RONI are listed. To make it possible to draw a conclusion based on the comparative analysis.

B. Performance Comparison



(a) Health Care Dataset (b) Loan Dataset (c) House Price Dataset

Fig. 1. MSE of defenses on ridge. The graphs evaluate the performance of defenses (TRIM, RONI, Huber, RANSAC, and No Defense) across the three datasets.



(a) Health Care Dataset (b) Loan Dataset (c) House Price Dataset

Fig. 2. MSE of defenses on LASSO. The graphs evaluate the performance of defenses (TRIM, RONI, Huber, RANSAC, and No Defense) across the three datasets.

The results of Mean Squared Error(MSE) across the 3 datasets on the ridge and LASSO regression models highlight clear differences between the overall performance of TRIM and

RONI. TRIM had consistently better performance than all of the other approaches, with, in most cases, significantly lower MSE than other methods across all the datasets. It also has a consistent performance in all 3 datasets, in both regression models. This contrasts heavily with the performance of RONI which is inconsistent overall, and shows large variability in some cases (eg. House Price dataset) at different poisoning levels (although the reason for this poor performance could be because of the robust statistics methods used that are designed to remove/reduce the effect of outliers from the data, while RONI can only identify outliers with high impact on the trained models.); it also has many instances where its worse than the undefended models, which is demonstrated by the increase of the MSE by 8.06% against the undefended models. On the other hand, TRIM improves the MSE of RONI by a factor of 20.28 and, in some cases, achieves lower MSEs than those of unpoisoned models (by 3.47%) [4].

Regarding the robustness of the algorithms, we see a clear difference between the two approaches, as TRIM maintains a consistently low MSE with the increase of the poisoning rate, at median increase of 6.1% of MSE across all datasets and attacks, while only 20% of attacks cause more than 27.2% MSE increase, which is something that the RONI algorithm fails to do in all the 3 datasets in both regression models.

On top of that, TRIM is far more efficient than RONI, by taking an average of 0.02 seconds of running time, on the house and Health Care datasets, while RONI is considerably slower at an average of 14.80 seconds on the Health Care dataset and 15.69 seconds on the house dataset [4]. This can be attributed to the fact that RONI has to retrain the model for each new data point, which increases with computational overhead.

In summary, considering the performance graphs and results and the approach that each algorithm takes in tackling the attack scenarios, it is clearly noticeable that the TRIM algorithm outperforms RONI in all instances; also, considering its characteristics, TRIM appears to be more suitable for use in Federated Learning when it comes to adversarial scenarios: efficiency wise, FL systems demand defenses that scale effectively across a high number of clients, and it is a positive that TRIM's iterative trimming process converges quickly, imposing minimal computing overhead; TRIM demonstrates robustness at increased levels of poisoning, which is one of the main threats to an effective model training in FL, something that RONI fails to do; additionally, one key aspect that makes TRIM superior is its consistency in results across the 3 different datasets, meaning its versatile enough to adapt and detect the possible ways of poisoning each of the datasets.

C. Advantages and Disadvantages of TRIM and RONI

+	**TRIM**	-
High Robustness to Poisoning Attacks: TRIM excels at identifying and excluding poisoned data points, maintaining the model's performance even under high poisoning rates. Its iterative trimming mechanism allows it to adapt to a variety of adversarial attack strategies, including both statistical and optimization-based attacks [4].		**Dependence on Residual Thresholding:** TRIM relies on residual-based thresholding, which can be less effective if malicious points are carefully crafted to mimic legitimate data distributions and have low residuals [4].
General Applicability Across Domains: TRIM has demonstrated effectiveness across diverse domains, including healthcare, finance, and real estate, making it a versatile choice for anomaly detection in FL.		**Limited Applicability to Non-Linear Models:** While effective for linear regression, TRIM may require adaptations or extensions to handle more complex, non-linear models used in FL.
Computational Efficiency: Despite its iterative nature, TRIM converges quickly, even on high-dimensional datasets, ensuring minimal overhead during model training [4].		**Potential for Over-Filtering:** If legitimate points are erroneously identified as anomalies due to high residuals, TRIM may exclude valuable data, potentially reducing model accuracy.
No Assumptions on Data Distribution: Unlike many anomaly detection methods, TRIM does not require prior knowledge about the data distribution, enhancing its utility in heterogeneous FL environments [4].		**Scalability Challenges in Extremely Large Datasets:** Although efficient in most cases, TRIM's performance may degrade when processing extremely large datasets in resource-constrained environments [4].

+	**RONI**	-
Enhanced Model Integrity: By filtering out data that negatively impacts performance, RONI helps maintain the robustness of the model [1].		**Computational Overhead:** The necessity to retrain the model for each new data point or client update can be resource-intensive [5].
Threshold Sensitivity: Determining appropriate performance degradation thresholds is challenging and critical for the method's effectiveness [10].		**Potential for False Positives/Negatives:** Inaccurate threshold settings may result in the rejection of beneficial data or the acceptance of harmful data [7].

VI. Conclusion

This study explored the effectiveness of two anomaly detection methods, TRIM and RONI, in defending Federated Learning systems against poisoning attacks. TRIM demonstrated superior robustness, computational efficiency, and adaptability across all tested datasets and poisoning scenarios, maintaining consistent performance and even outperforming undefended models in certain cases. Its iterative trimming mechanism allowed for rapid convergence with minimal overhead, making it particularly suited for FL environments where scalability and resilience are critical.

In contrast, RONI, while offering a principled approach through validation-based anomaly detection, struggled with inconsistent performance and significant computational overhead. Its reliance on retraining models for each update limited its practicality in large-scale FL systems, particularly under higher poisoning rates and in datasets with high-dimensional features.

The comparative analysis revealed TRIM's clear advantages in mitigating adversarial contributions, maintaining low MSE, and operating effectively under diverse attack conditions. However, both methods face limitations, such as TRIM's potential for over-filtering legitimate data and RONI's sensitivity to validation thresholds. Future work should focus on addressing these challenges, including extending TRIM to non-linear models and improving RONI's computational efficiency. Furthermore, exploring hybrid approaches that combine the strengths of both methods may offer more comprehensive defenses against sophisticated adversarial attacks in FL systems.

These findings emphasize the importance of developing robust, scalable, and adaptive anomaly detection mechanisms to ensure the security and integrity of Federated Learning in increasingly adversarial environments.

References

[1] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 2, pp. 513–535, 2023. [Online]. Available: https://doi.org/10.1007/s13042-022-01647-y

[2] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X20329848

[3] N. Bouacida and P. Mohapatra, "Vulnerabilities in federated learning," *IEEE Access*, vol. 9, pp. 63 229–63 249, 2021.

[4] B. B. C. L. C. N.-R. B. L. Matthew Jagielski, Alina Oprea, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," *arXiv preprint arXiv:1804.00308*, 2018. [Online]. Available: https://arxiv.org/pdf/1804.00308

[5] M. Vucovich, A. Tarcar, P. Rebelo, N. Gade, R. Porwal, A. Rahman, C. Redino, K. Choi, D. Nandakumar, R. Schiller, E. Bowen, A. West, S. Bhattacharya, and B. Veeramani, "Anomaly detection via federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2210.06614

[6] M. Nardi, L. Valerio, and A. Passarella, "Anomaly detection through unsupervised federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2209.04184

[7] A. B. H. RAED ABDEL SATER, "Threshold sensitivity in federated models," *arXiv preprint arXiv:2010.10293*, 2020. [Online]. Available: https://arxiv.org/pdf/2010.10293

[8] P. Bhat, M. P. M M, and R. M. Pai, "Anomaly detection using federated learning: A performance based parameter aggregation approach," in *2023 3rd International Conference on Intelligent Technologies (CONIT)*, 2023, pp. 1–6.

[9] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," *Applied Sciences*, vol. 8, no. 12, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/12/2663

[10] C.-C. L. Tze-Qian Eng, Hsing-Kuo Pao, "Evaluating performance metrics for federated anomaly detection," *IEEE Machine Learning Journal*, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10386871