Autoencoder for Detecting Malicious Updates in Differentially Private Federated Learning

Lucia Alonso[1] and Mina Alishahi[2]
[1]*Informatics Institute, University of Amsterdam, The Netherlands*
[2]*Department of Computer Science, Open Universiteit, The Netherlands*

Keywords: Federated Learning, Differential Privacy, Autoencoder, Anomaly Detection.

Abstract: Differentially Private Federated Learning (DP-FL) is a novel machine learning paradigm that integrates federated learning with the principles of differential privacy. In DP-FL, a global model is trained across decentralized devices or servers, each holding local data samples, without the need to exchange raw data. This approach ensures data privacy by adding noise to the model updates before aggregation, thus preventing any individual contributor's data from being compromised. However, ensuring the integrity of the model updates from these contributors is paramount. This research explores the application of autoencoders as a means to detect anomalous or fraudulent updates from contributors in DP-FL. By leveraging the reconstruction errors generated by autoencoders, this study assesses their effectiveness in identifying anomalies while also discussing potential limitations of this approach.

1 INTRODUCTION

Federated learning has emerged as a critical paradigm in contemporary machine learning, particularly in scenarios where data privacy and security are paramount concerns (Li et al., 2020a). With the proliferation of Internet of Things (IoT) devices and edge computing, federated learning enables collaborative model training across distributed entities without centralizing sensitive data (Alishahi et al., 2022). However, traditional federated learning approaches may still pose privacy risks, as individual data contributors' information could be susceptible to inference attacks. To address this challenge, the integration of differential privacy into federated learning has garnered significant attention (Wei et al., 2020), (Li et al., 2020b) (Lopuhaä-Zwakenberg et al., 2021). By augmenting federated learning with differential privacy, organizations can enhance data privacy protections, ensuring that individual contributors' data remains confidential even during model training (Fathalizadeh et al., 2024). This incorporation of federated learning and differential privacy not only strengthens privacy guarantees but also fosters trust and collaboration among participating entities, making it a vital tool for modern data-driven applications (Yang et al., 2023).

While Differentially Private Federated Learning (DP-FL) offers robust privacy guarantees, ensuring the integrity of model updates from contributing devices or servers is essential for maintaining the efficacy and reliability of the trained global model. Anomalies or misbehavior in the contributions of individual entities can compromise the integrity of the learning process and undermine the overall performance of the federated learning system. Therefore, there is a growing need to develop mechanisms capable of detecting and mitigating such anomalies in DP-FL settings.

This paper explores the use of autoencoders, a type of artificial neural network, as a potential solution for detecting anomalous contributions from individual devices or servers in DP-FL (Bank et al., 2023). Autoencoders are known for their ability to learn efficient representations of data and reconstruct input samples with minimal error (An and Cho, 2015) (Yan et al., 2023). By leveraging the reconstruction errors generated by autoencoders, we aim to identify and flag anomalous model updates, thus safeguarding the integrity of DP-FL .

In this study, we investigate the effectiveness of autoencoders as anomaly-detecting mechanisms in DP-FL and examine their potential limitations. Ultimately, our goal is to contribute to the development of robust and privacy-preserving federated learning sys-

Alonso, L. and Alishahi, M.
Autoencoder for Detecting Malicious Updates in Differentially Private Federated Learning.
DOI: 10.5220/0012766700003767

In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 467-474
ISBN: 978-989-758-709-2; ISSN: 2184-7711

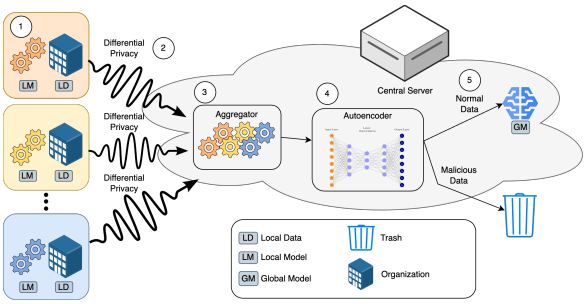


Figure 1: Scheme displaying an overview of the federated learning with differential privacy process.

tems capable of operating securely in real-world settings.

2 BACKGROUND

This section presents the preliminary concepts employed in this study.

2.1 Federated Learning

Federated deep learning is a particular form of federated learning in which the goal is training a deep learning model in decentralized setting, and it can be formally defined as follows (Li et al., 2020a). Let $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N$ denote the local datasets available at each of the N participating devices or servers. Each dataset $\mathcal{D}_i$ contains a set of data samples (or one sample in individual setting) $\{(x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{in_i}, y_{in_i})\}$, where x_{ij} represents the input data and y_{ij} represents the corresponding label. The goal of federated learning is to learn a global model θ by aggregating the local updates from each device. The global model is updated iteratively using the following formula:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{1}{N} \sum_{i=1}^{N} \nabla \ell_i(\theta_t)$$

where θ_t represents the global model parameters at iteration t; η is the learning rate; $\nabla \ell_i(\theta_t)$ is the gradient of the loss function ℓ_i with respect to the model parameters θ_t computed using the local dataset $\mathcal{D}_i$. This process repeats until convergence, with each device contributing its local gradient to update the global model without sharing its raw data.

2.2 Differential Privacy (DP)

Differential privacy aims to protect the privacy of individuals' data by ensuring that the presence or absence of any single individual's data does not significantly affect the outcome of a computation or analysis (Dwork, 2008). Formally, let $\mathcal{D}_1$ and $\mathcal{D}_2$ be neighboring datasets that differ by at most one individual's data, and let $\mathcal{A}$ represent a randomized algorithm that operates on datasets. A randomized algorithm $\mathcal{A}$ satisfies ε-differential privacy if, for all possible outputs S in the algorithm's output space:

$$\Pr[\mathcal{A}(\mathcal{D}_1) = S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(\mathcal{D}_2) = S]$$

where $\varepsilon > 0$ is a parameter that controls the privacy guarantee. Smaller values of ε correspond to stronger privacy guarantees.

2.3 Differentially Private Federated Learning

Differentially Private Federated Learning aims to train a global model while preserving the privacy of individual data contributors by adding DP noise to the model updates before aggregation (Geyer et al., 2017). Formally, let $D = \{D_1, D_2, ..., D_N\}$ denote the set of N decentralized data sources, each holding a local dataset D_i. Let M represent the global machine learning model to be trained. At each iteration of the federated learning process, a subset of data sources is selected to participate in model training. Denote the selected subset as $S \subseteq D$, where $|S| = k$. The local models trained on the data sources in S are denoted as $M_1, M_2, ..., M_k$. To ensure differential privacy, each local model M_i is trained using a differentially private algorithm, which adds carefully calibrated noise to the model updates. Let $\mathcal{M}$ denote the set of all possible global models, and let $\mathcal{D}$ denote the set of all possible datasets that could have been used to train M. The privacy guarantee ensures that for any two datasets $D_i, D_j \in \mathcal{D}$ that differ in a single element, the distribution of the global model M learned from D_i is close to the distribution of the global model learned from D_j. The objective of Differentially Private Federated Learning is to find the global model M that minimizes the empirical risk over the union of all local datasets while satisfying a given privacy constraint ε. This can be expressed as:

$$\min_{M \in \mathcal{M}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(M, D_i)$$

subject to the constraint that for any pair of adjacent datasets $D_i, D_j \in \mathcal{D}$ and any measurable set $S \subseteq \mathcal{M}$, the following differential privacy condition holds:

$$\Pr[M \in S] \leq e^{\varepsilon} \times \Pr[M \in S']$$

2.4 Autoencoders

An autoencoder is a type of artificial neural network that learns to encode input data into a lower-dimensional representation and then decode it back to

its original form. It consists of an encoder network, which compresses the input data into a latent representation, and a decoder network, which reconstructs the original data from the latent representation. The goal of an autoencoder is to minimize the reconstruction error, thereby learning an efficient representation of the input data (Bank et al., 2023).

Formally, let $X \in \mathbb{R}^d$ represent the input data, and let $Z \in \mathbb{R}^m$ represent the latent representation (also known as the encoding) obtained by the encoder function $f_{\text{enc}} : \mathbb{R}^d \to \mathbb{R}^m$. Similarly, let $\hat{X} \in \mathbb{R}^d$ represent the reconstructed input data obtained by the decoder function $f_{\text{dec}} : \mathbb{R}^m \to \mathbb{R}^d$. The autoencoder aims to learn a representation of the input data such that the reconstructed output closely matches the original input. This is achieved by minimizing a reconstruction loss function $\mathcal{L}(X, \hat{X})$, typically the mean squared error (MSE) or cross entropy, between the input data X and its reconstruction $\hat{X}$. The optimization problem for training the autoencoder can be expressed as:

$$\min_{f_{\text{enc}}, f_{\text{dec}}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(X_i, f_{\text{dec}}(f_{\text{enc}}(X_i)))$$

where N is the number of training samples; X_i is the i-th training sample; f_{enc} and f_{dec} are the encoder and decoder functions, respectively. The encoder function maps the input data to a lower-dimensional latent space, while the decoder function reconstructs the input data from the latent representation.

3 FRAMEWORK

This section presents our methodology and the evaluation metrics.

3.1 Methodology

The primary objective of this study is to assess the efficacy of autoencoders in identifying anomalies within differentially private federated learning frameworks. As depicted in Figure 1, our methodology follows the upcoming steps:

(1) Initiation: The process commences with the central server transmitting the initial model parameters to each individual client. Subsequently, each client configures its local model using the provided parameters and its own dataset.

(2) Differential Privacy Integration: Each client augments their locally trained model and compute the associated gradients. The clients then add DP noise to their parameters, ensuring the privacy of their contributed data.

Algorithm 1: Autoencoders for labeling input data.

```
Run data through autoencoder A
α ← the limit of ε for each metric
threshold_acc, threshold_loss ← threshold for accuracy, threshold for loss
while ε ≤ α do                ▷ Working with accuracy for low values of ε
    n = array of batches of size y
    for each batch in n do
        acc_batch = accuracy(batch)
        if acc_batch ≤ accuracy_original − threshold_acc then
            batch contains an anomaly and discard
        end if
    end for
end while
while ε > x do                ▷ Working with loss for high values of ε
    dict = images submitted by client
    for each image in dict do
        loss_img = cross_entropy(image)
        if loss_img ≥ loss_original + threshold_loss then
            image is an anomaly and discard
        end if
    end for
end while
```

(3) Data Aggregation: The aggregator receives and integrates the augmented data from all clients before forwarding it to the autoencoder for analysis and discarding malicious updates.

(4) Anomaly Detection: Next, the autoencoder carefully examines the combined data, identifying any irregularities or suspicious inputs.

(5) Decision and Model Update: Identified malicious data is discarded, while the remainder is utilized in refining the global model.

Throughout this study, we delve into the efficacy of autoencoders in pinpointing anomalies within the steps 4 and 5 of this architecture. Before initiating the federated learning environment, the dataset undergoes processing via an autoencoder. Subsequently, the baseline values for loss, denoted as "loss_original," and accuracy, referred to as "accuracy_original," are recorded. Algorithm 1 summarizes the process of labeling data using autoencoder in our framework. The algorithm's objective is to identify any potentially malicious data and eliminate it before it impacts the global model. The first three lines of the pseudocode are universal for all ε values and handle parameter initialization. Afterward, the code branches into two blocks. The first block is suitable for environments with low ε values, where anomaly detection relies on accuracy. In contrast, the second block is tailored for higher ε values, where the autoencoder's re-

construction errors are minimized, enabling anomaly detection through loss.

3.2 Evaluation Metrics

We measure the performance of our methodology using two metrics, namely cross-entropy and accuracy defined as follows.

Cross-Entropy: also known as log loss, is a metric used to quantify the difference between two probability distributions. Formally,

$$H(y,\hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i))$$

where y is the true label vector, $\hat{y}$ is the predicted probability vector, and N is the number of samples. From now on, for the sake of simplicity, we use the term "Loss" instead of "Cross-entropy loss". In this study, the overall loss of a dataset is determined by summing up the reconstruction errors generated by the autoencoder for each individual sample, where the lower loss is better outcome.

Accuracy: represents what proportion of a set is correctly represented compared to the original value.

$$Accuracy = \frac{\text{correct labels}}{\text{the total number of records}} \times 100$$

In this study, we evaluate the performance of two key components using accuracy metrics. Firstly, we assess the accuracy of the autoencoder in reconstructing images, where higher accuracy indicates greater similarity between input and output images. Secondly, we measure the performance of the classifier in identifying anomalies, represented by the percentage of true anomalies correctly detected among all anomalous images. In both cases, achieving higher accuracy values reflects improved method performance. Following the establishment of specific accuracy and loss thresholds, data classification is conducted to evaluate the efficacy of anomaly detection using these metrics. The outcomes are then presented through a confusion matrix, providing insights into the effectiveness of autoencoders for anomaly detection in DP-FL.

4 RESULTS

This section presents the key findings of our research. We begin by analyzing the accuracy and loss metrics generated by the model, followed by an examination of the model's sensitivity. Through these analyses, we aim to assess the effectiveness of autoencoders in detecting anomalies in DP-FL and provide insights into their performance characteristics.

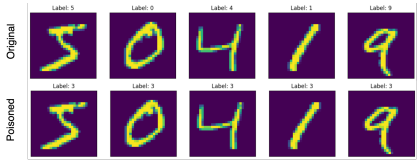


Figure 2: Comparison between unmodified images (on the top row) vs poisoned images (on the bottom row).

4.1 Experimental Set-Up

Here we present the dataset and environment employed in our work.

Dataset: The MNIST dataset consists of 70.000 images containing handwritten digits. There are 10 distinct digits ranging from 0-9. Each image is described with 28 × 28 pixels. Figure 2 presents a comparison between unaltered images (top row) and poisoned images (bottom row). Despite the imperceptible changes to the human eye, subtle alterations to certain pixels have been inserted. This meticulous manipulation of noise within the images leads the deep learning model to misclassify the images, as depicted atop each image.

Environment: This research builds upon the foundation laid by Wenzhuo Yang's code[1], which provided a federated learning framework with differential privacy. Our work extends this framework by incorporating methods for conducting poison attacks and evaluating model performance under such adversarial scenarios. These additions were tailored to meet the specific requirements of our study, enhancing the versatility and applicability of the original codebase. Our implementation code can be found here [2].

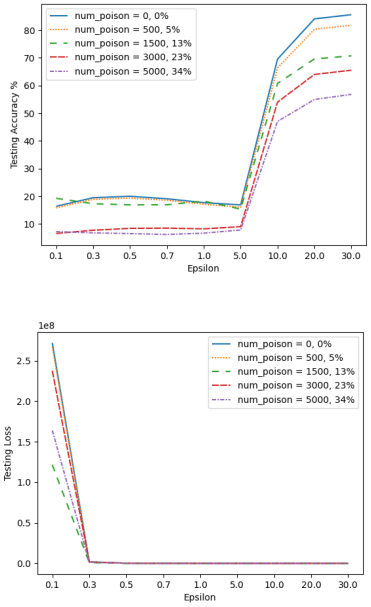
4.2 Accuracy and Loss

The model's accuracy across various epsilon values has been illustrated in Figure 3(a). Each line in the graph corresponds to a different quantity of poisoned images within the testing set. Initially, the model operates without any poisoned images, after which varying quantities of poisoned images are introduced, as indicated in the graph legend. Additionally, the percentage of poisoned images within the testing set is provided alongside the corresponding quantity.

The graph reveals a notable peak point where the accuracy experiences a significant increase. In this dataset, the model's accuracy undergoes a remarkable improvement when $\varepsilon \geq 5$. This phenomenon can be attributed to the noise adjustment facilitated

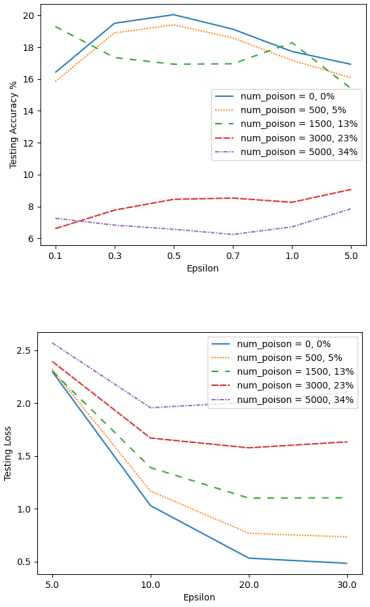
[1]https://github.com/wenzhu23333/Differential-Privacy-Based-Federated-Learning

[2]https://github.com/lucialonso/Federated-Learning-Differential-Privacy-RP2



(b)

Figure 3: Autoencoder results in terms of (a) Accuracy over epsilon from $0.1 \leq \varepsilon \leq 30$ (b) Loss values over epsilon when $0.1 \leq \varepsilon \leq 30$.



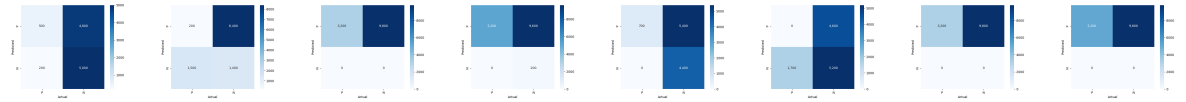
(b)

Figure 4: Autoencoder results in terms of (a) Accuracy over epsilon for $0.1 \leq \varepsilon \leq 5$ (b) Loss over epsilon for $5 \leq \varepsilon \leq 30$.

by differential privacy, which becomes more relaxed with higher privacy budgets. Consequently, the model gradually converges to a notable accuracy level beyond a certain threshold of privacy. Additionally, the graph highlights minimal variance in the model's accuracy within the range of $0.1 \leq \varepsilon \leq 5$. During this interval, the presence of considerable noise and reconstruction errors leads to less accurate predictions. Furthermore, a distinct decrease in accuracy is observed when the proportion of poisoned images exceeds 23% of the test set, particularly evident within the range of $0.1 \leq \varepsilon \leq 5$. In machine learning, employing loss to identify anomalies is favored over accuracy due to its heightened sensitivity to subtle alterations. However, as depicted in Figure 3(b), the model's loss exhibits significant fluctuations for lower values of epsilon. The magnitude of loss for smaller epsilon values is substantial, to the extent that data corresponding to higher epsilon values becomes indiscernible.

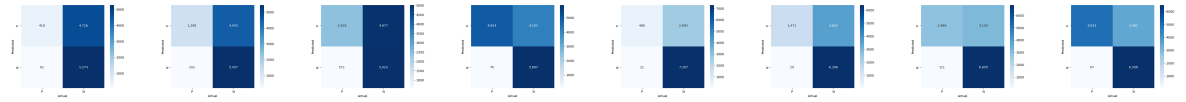
4.3 Optimal Thresholds

Determining whether to utilize accuracy or loss to establish the threshold for identifying anomalies presents a formidable challenge. This study aims to detect anomalies characterized by an increase in loss of approximately 0.02, akin to methodologies employed in traditional federated learning. However, relying solely on loss as a metric proves inadequate for discerning anomalies in scenarios featuring small epsilon values, where the presence of excessive noise obscures the distinction between reconstruction errors, anomalies-induced loss, and noise stemming from differential privacy. Therefore, based on an analysis of model performance with a specific dataset, this research advocates for a combined approach leveraging both metrics to effectively detect anomalies. Specifically, accuracy is proposed as a metric for detecting anomalies at lower epsilon values, while loss is recommended for higher epsilon values. Figure 4 (a) is a scaled-up graph of the accuracy values when $0.1 \leq \varepsilon \leq 5$. The figure shows that when 5% of the test set is poisoned, a reduction in accuracy is noticeable. In this case, the accuracy of the model with unaltered data is 16,4 and 15,8 when 5% of the data is an anomaly. When more poisoned images are introduced the decrease in accuracy is more significant. There is however an irregularity when 13% of the data in the test set is poisoned. As an increasing quantity of modified data is added to the test set, a corresponding decline in accuracy would be anticipated. In Figure 4(b) the trend that the loss value takes for different amounts of poisoned images and epsilon values can be distinctly seen. As expected, the unaltered data yielded the lowest loss values while the more modified images are added to the test set, the higher the loss value gets. For reference, the loss value at the lowest epsilon of this graph ($\varepsilon = 5$), when only unaltered data has been evaluated, is 2.30. In contrast,



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5: For (a,b,c,d) $\varepsilon = 0.1$, and the testing set contains 500, 1500, 3000, and 5000 poisoned images respectively. For (e,f,g,h) $\varepsilon = 1$, and the testing set contains 500, 1500, 3000, and 5000 poisoned images respectively.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6: For (a,b,c,d) $\varepsilon = 5$, and the testing set contains 500, 1500, 3000, and 5000 poisoned images respectively. For (e,f,g,h) $\varepsilon = 30$, and the testing set contains 500, 1500, 3000, and 5000 poisoned images respectively.

when 5% of data in the test set is altered, the loss value is 2.32. As epsilon increases, the interval between loss values for different quantities of anomalies and the baseline widens.

4.4 Efficacy Testing

In this section, we assess the effectiveness of autoencoders in detecting anomalies in DP-FL, following the threshold determination outlined in the previous section. Figure 5 illustrates the classification process based on the autoencoder's accuracy as the threshold metric. When the autoencoder's accuracy decreases by 0.2, data is classified as anomalous and discarded accordingly. Among the various threshold values considered, 0.2 emerged as the most effective in correctly identifying anomalies. Since accuracy is calculated over batches rather than individual images, the testing set is evaluated in batch sizes of 200, with each batch assessed independently. If the classifier identifies a batch as potentially containing an anomaly, the entire batch is discarded. Through experimentation, a batch size of 200 was determined to be optimal.

The confusion matrix provides insight into the classification outcomes: the top left corner denotes correctly classified anomalies, the top right corner indicates misclassified anomalies, the bottom left corner represents misclassified unaltered data, and the bottom right corner signifies correctly classified unaltered data. With epsilon set to 0.1 and 500 poisoned images included in the testing set, all poisoned images are accurately classified, while 4800 unaltered images are erroneously labeled as anomalies and discarded. At epsilon equal to 1, all images containing anomalies are correctly identified, but 5400 original images are discarded. Notably, the least data loss occurs when 500 poisoned images are added to the test set. Conversely, with higher volumes of poisoned images (3000 and 5000), almost all original data is lost. As seen in Table 1, after examining all the results for $0.1 \leq \varepsilon \leq 1$ and all the different amounts of poisoned data, the proposed method has correctly detected 77.7% of anomalies overall. However, 83.6% of unaltered data has been lost overall due to misclassification as seen in Table 2.

Table 1: The classifier's precision at detecting anomalies when using accuracy. The percentage of poisoned images found in each case is shown.

ε \ # poison	500	1500	3000	5000
0.1	100 %	13.3 %	100 %	100 %
0.3	100 %	13.3 %	100 %	100 %
0.5	100 %	13.3 %	100 %	100 %
0.7	100 %	13.3 %	100 %	100 %
1	100 %	0 %	100 %	100 %
Average	100 %	10.6 %	100 %	100 %

Table 2: The percentage of unaltered images lost due to a misclassification when using accuracy.

ε \ # poison	500	1500	3000	5000
0.1	48 %	84 %	100 %	98 %
0.3	52 %	98 %	100 %	100 %
0.5	48 %	98 %	100 %	100 %
0.7	52 %	94 %	100 %	100 %
1	54 %	46 %	100 %	100 %
Average	50.8 %	84 %	100 %	99.6 %

Figure 6 represents how data would be classified when using the autoencoder's loss as a deciding metric and setting the threshold to 0.03. Any images with a loss value 0.03 higher than the loss of the model, when only unaltered data is processed, will be classified as an anomaly. Out of different values tested, an increase of 0.03 in the loss value proved to yield the most precise classifications. The improvement of the algorithm can be observed in Figure 6 where less original data is lost. In Figure 5, when 3000 or 5000 poisoned images are added to the test set, almost all original data is discarded and lost. In contrast, at higher values of epsilon, as shown in Figure 6, when 3000

Table 3: The classifier's precision at detecting anomalies when using loss. The percentage of poisoned images found in each case is shown.

ε \ # poison	500	1500	3000	5000
5	83.6 %	86.5 %	80.9 %	98.5 %
10	99.4 %	98.1 %	99.7 %	99.0 %
20	98.4 %	98.5 %	97.8 %	99.4 %
30	97.6 %	98.1 %	96.3 %	98.7 %
Average	94.8 %	95.3 %	93.7 %	98.9 %

or 5000 poisoned images are introduced, less original data is lost. So, in this specific case, with this specific dataset, the algorithm performs more efficiently at higher epsilon values ($5 \leq \varepsilon \leq 30$).

As seen in Table 3, after examining all the results for $5 \leq \varepsilon \leq 30$ and all the different amounts of poisoned data, the classifier, overall, has correctly detected 95.7% of anomalies and 41.0% of unaltered data has been lost due to misclassification as shown in Table 4.

4.5 Discussion

Based on the findings presented, autoencoders emerge as a robust tool for anomaly detection in DP-FL settings. Accuracy proves effective for identifying anomalies at lower epsilon values, yet the precision of autoencoder-based reconstructions is compromised by significant noise levels. Consequently, a substantial portion (approximately 84%) of legitimate data is lost when epsilon is less than 5. Conversely, loss serves as a reliable metric for anomaly detection at higher epsilon values ($\varepsilon \geq 5$), correctly identifying and discarding 95.7% of anomalies. However, both approaches entail the risk of discarding genuine data erroneously classified as anomalous. Thus, when integrating these methods into the pipeline of a federated learning environment with differential privacy, careful consideration must be given to balancing the trade-off between anomaly detection and data preservation. It's essential to acknowledge that the method proposed in this study does not achieve 100% anomaly detection, implying that certain malicious data may bypass the global model's defenses. The considerable noise introduced by differential privacy renders the autoencoder unreliable until epsilon reaches 5 in this context. Therefore, deploying the proposed method at higher epsilon values is advisable to optimize anomaly detection while minimizing the loss of genuine data. This approach ensures the best balance between detection efficacy and privacy preservation, particularly when the method is most effective, thereby guaranteeing a lower level of privacy.

Table 4: The percentage of unaltered images lost due to a misclassification when using loss.

ε \ # poison	500	1500	3000	5000
5	47.3 %	45.4 %	49.8 %	41.0 %
10	40.9 %	61.0 %	55.1 %	38.7 %
20	29.3 %	40.7 %	38.1 %	39.4 %
30	26.9 %	36.0 %	31.5 %	34.9 %
Average	36.1 %	45.8 %	43.6 %	38.5 %

5 RELATED WORKS

Given the pivotal role of AI in modern life, the detection of malicious updates and adversarial instances has garnered significant attention. Consequently, a wealth of research efforts has been directed towards analyzing these types of attacks and defense mechanisms (Cina et al., 2023) . Although much of the existing research focuses on designing inherently robust models against security and privacy attacks (Rosenberg et al., 2021), fewer efforts address the specific challenge of detecting malicious updates in decentralized settings, particularly in federated learning environments. In (Zhang et al., 2022), FLDetector is introduced to tackle this issue by identifying malicious clients. The core insight is that in model poisoning attacks, the model updates from a client across multiple iterations exhibit inconsistency. Thus, FLDetector detects potentially malicious clients by examining the consistency of their model updates. In (Zhao et al., 2022), a poisoning defense mechanism is proposed to detect and mitigate poisoning attacks in federated learning by utilizing generative adversarial networks to generate auditing data during the training process and identifies adversaries by auditing their model accuracy. On the other hand, Fed-ANIDS (Idrissi et al., 2023) leverages autoencoders within a federated learning framework for anomaly detection in distributed networks. However, it primarily focuses on detecting anomalies rather than specifically targeting malicious updates within federated learning. While autoencoders have demonstrated their effectiveness in anomaly detection across various domains, their application in detecting malicious updates or misbehavior in federated learning remains relatively limited. Schram et al. (Schram et al., 2022) propose a novel iteration of DP-Fed-Avg GAN, which integrates denoising techniques, specifically autoencoders, to alleviate the typical loss in accuracy encountered when applying both differential privacy and federated learning to GANs. The closest work to ours is the Fedcvae framework proposed in (Gu and Yang, 2021), which focuses on detecting and excluding malicious or misleading information in federated networks. Fedcvae effectively identifies and removes

malicious model updates from client contributions in federated settings.Our research builds upon this foundation by specifically investigating the effectiveness of autoencoders in detecting malicious updates in differentially private federated learning settings. To the best of our knowledge, our work represents the first attempt to systematically evaluate and quantify the performance of autoencoders in this context, thereby advancing our understanding of their role in ensuring the security and reliability of differentially private federated learning systems.

6 CONCLUSION AND FUTURE DIRECTIONS

This paper delves into the potential of autoencoders, renowned for their data representation and reconstruction capabilities, as a solution for identifying anomalous updates in differentially private federated learning (DP-FL). Through empirical analysis, we assessed autoencoders' efficacy, addressing associated challenges to enhance differentially private federated learning's integrity in practical scenarios. Future directions for this work encompass exploring other attacks beyond malicious updates, such as adversarial learning approaches. Additionally, robustness analysis is crucial, requiring evaluation under diverse scenarios and datasets to assess its generalization performance under varying levels of noise and data distribution.

REFERENCES

Alishahi, M., Moghtadaiee, V., and Navidan, H. (2022). Add noise to remove noise: Local differential privacy for feature selection. *Computers & Security*, 123:102934.

An, J. and Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18.

Bank, D., Koenigstein, N., and Giryes, R. (2023). *Autoencoders*, pages 353–374. Springer International Publishing, Cham.

Cina, A. E., Grosse, K., Demontis, A., Vascon, S., Zellinger, W., Moser, B. A., Oprea, A., Biggio, B., Pelillo, M., and Roli, F. (2023). Wild Patterns Reloaded: A Survey of Machine Learning Security against Training Data Poisoning. *ACM Computing Surveys*, 55(13s):294:1–294:39.

Dwork, C. (2008). Differential privacy: A survey of results. In *Theory and Applications of Models of Computation TAMC*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer Verlag.

Fathalizadeh, A., Moghtadaiee, V., and Alishahi, M. (2024). Indoor geo-indistinguishability: Adopting differential privacy for indoor location data protection. *IEEE Transactions on Emerging Topics in Computing*, 12(1):293–306.

Geyer, R. C., Klein, T., and Nabi, M. (2017). Differentially private federated learning: A client level perspective. *CoRR*, abs/1712.07557.

Gu, Z. and Yang, Y. (2021). Detecting Malicious Model Updates from Federated Learning on Conditional Variational Autoencoder. In *Parallel and Distributed Processing Symposium (IPDPS)*, pages 671–680.

Idrissi, M. J., Alami, H., El Mahdaouy, A., El Mekki, A., Oualil, S., Yartaoui, Z., and Berrada, I. (2023). Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems. *Expert Systems with Applications*, 234:121000.

Li, L., Fan, Y., Tse, M., and Lin, K.-Y. (2020a). A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854.

Li, Y., Chang, T.-H., and Chi, C.-Y. (2020b). Secure federated averaging algorithm with differential privacy. In *IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

Lopuhaä-Zwakenberg, M., Alishahi, M., Kivits, J., Klarenbeek, J., van der Velde, G., and Zannone, N. (2021). Comparing classifiers' performance under differential privacy. In *Conference on Security and Cryptography, SECRYPT*, pages 50–61. SCITEPRESS.

Rosenberg, I., Shabtai, A., Elovici, Y., and Rokach, L. (2021). Adversarial machine learning attacks and defense methods in the cyber security domain. *ACM Comput. Surv.*, 54(5).

Schram, G., Wang, R., and Liang, K. (2022). Using autoencoders on differentially private federated learning gans.

Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., Jin, S., Quek, T. Q. S., and Vincent Poor, H. (2020). Federated learning with differential privacy: Algorithms and performance analysis. 15:3454–3469. Conference Name: IEEE Transactions on Information Forensics and Security.

Yan, S., Shao, H., Xiao, Y., Liu, B., and Wan, J. (2023). Hybrid robust convolutional autoencoder for unsupervised anomaly detection of machine tools under noises. *Robotics and Computer-Integrated Manufacturing*, 79:102441.

Yang, Y., Hui, B., Yuan, H., Gong, N., and Cao, Y. (2023). {PrivateFL}: Accurate, differentially private federated learning via personalized data transformation. In *USENIX Security Symposium*, pages 1595–1612.

Zhang, Z., Cao, X., Jia, J., and Gong, N. Z. (2022). Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD.

Zhao, Y., Chen, J., Zhang, J., Wu, D., Blumenstein, M., and Yu, S. (2022). Detecting and mitigating poisoning attacks in federated learning using generative adversarial networks. *Concurrency and Computation: Practice and Experience*, 34(7):e5906.