BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 74(6), 2026, Article number: 1620
DOI: 10.24425/bpasts.2026.1620

An approach utilizing frequency analysis to counteract poisoning attacks in Federated Learning

Ramamurthy ANUSUYA [1]* **and D. Karthika RENUKA** [2]

[1] TIFAC-CORE in Cyber Security, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Coimbatore Campus, TN, India
[2] Department of Information Technology, PSG College of Technology, Coimbatore, TN, India

Abstract. Machine learning requires diverse training datasets from multiple clients for improved performance. However, sharing datasets is often a legal and privacy issue across countries and organizations. Federated Learning (FL) is a machine learning framework allowing individual clients to train datasets locally and share only the weight updates to a central server where the updates are aggregated. FL addresses the security and privacy issues concerned with data-sharing; however, it is vulnerable to poisoning attacks where a malicious client can purposefully alter the model updates. Even a smallest input deviation can exploit the system leading to misclassification. In this study, we propose a lightweight defense mechanism for mitigating poisoning attacks in Federated Learning (FL) systems. Our approach involves transforming model weights into the frequency domain to identify core frequency components containing sufficient model weight information. Additionally, we employ a model filtering algorithm to predict poisoning attacks based on the output of the frequency analysis method. This enables effective filtering of malicious updates during local training on client devices. Our proposed defense mechanism enhances the security and integrity of FL systems against adversarial attack ensuring secure model aggregation.

Keywords: Adversarial Attacks; Federated Learning; Secure Machine Learning.

1. INTRODUCTION

Federated Learning (FL) [1] has expanded its reach across various domains, including the medical and industrial equipment maintenance sectors. In FL environment multiple clients participate to collaboratively execute a machine learning or deep learning model in the client device. The data remain on the client device, which enhances the security and privacy of their data. Though an FL environment enhances the security of distributed clients, it is prone to adversarial attacks. Adversarial attacks are a type of cyberattack that exploits the vulnerabilities of machine learning models. Adversaries can create adversarial examples, which are inputs to machine learning models that are designed to cause the model to make a mistake. Adversarial examples can be created for a variety of machine learning tasks, including image classification, object detection, and natural language processing. Adversarial attacks significantly impact the reliability of the FL model by manipulating updates and inducing misclassifications. Conventional techniques for constructing resilient Federated Learning models often do not offer practical defense mechanisms against adversarial examples. In this work, we propose a lightweight and privacy-preserving framework for detecting poisoning attacks by analyzing the frequency domain of model updates. The proposed work transforms model updates into frequency domain using Discrete Cosine Transform (DCT). The dominant frequency components, defined as the DCT coefficients with the highest magnitude, are selected as representative features of the model update. These features help in the anomaly detection to identify potentially malicious client updates. Unlike the existing works which focus on gradient similarity or the entire frequency components, our work emphasizes a compact and computationally efficient representation by using only dominant frequency components, enabling scalable detection. The main contributions of this work to mitigate adversarial attacks in Federated Learning include:

- A frequency-domain feature representation of model updates using Discrete Cosine Transform (DCT), enabling privacy-preserving analysis without accessing raw data.
- A dominant frequency extraction strategy that reduces high-dimensional model updates into a compact feature space while preserving discriminative characteristics of benign and malicious updates.
- A comparative evaluation of classical anomaly detection methods (K-Means, One-Class SVM, and Isolation Forest) applied to frequency-based features.
- An empirical analysis demonstrating that frequency-domain patterns of model updates exhibit distinguishable signatures under poisoning attacks, supporting the feasibility of lightweight detection mechanisms in FL environments.

Though the techniques like DCT and anomaly detection algorithms are well established, the novelty of this work lies primarily in the feature representation and filtering pipeline in FL rather

*e-mail: ranusuya@cb.amrita.edu

Manuscript submitted 2025-12-03, revised 2026-03-30, initially accepted for publication 2026-05-13, published in September 2026.

© 2026 The Author(s). This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/)

than in the aggregation or detection algorithms themselves. The proposed work introduces a light-weight frequency based filtering mechanism operating on model updates, thus offering an efficient defense mechanism for poisoning attacks in FL. While similarity-based aggregation methods like FoolsGold work on pairwise gradient similarity which may be sensitive to collusion, our work analyzes frequency pattern and avoids dependency on inter-client comparisons.

2. LITERATURE REVIEW

Fung *et al.* [2] developed FoolsGold, which considers that benign models differ considerably from poisoned models and targets rigorous non-IID instances. FoolsGold determines pairwise similarities and applies weights during aggregation to reduce the effect of extremely similar models and avoid poisoning attacks. The work by [3] discusses the danger to FL caused by a particular type of backddor attack called Cerberus Poisoning. Shen *et al.* [4] designed Auror to filter out-of-distribution parameters and protect FL against malicious upgrades; yet, it has a high computational overhead and is vulnerable to numerous backdoors. A model poisoning attack (MPA) was identified by Bagdasaryan *et al.* [5]. [6] also proposes mitigation for backdoor attacks in FL. In order to avoid detection and backdoor the joint model, individuals can limit update norms in this attack. Baruch *et al.* [7] demonstrated the effectiveness of the LIE attack, which works within population variance and modifies a number of factors to avoid countermeasures. The research [8] proposes a two-stage privacy attack method focusing on Transformer-based language models featuring a peculiar Pooler layer and suggests a two-stage attack that exploits vulnerabilities in this particular module.

Empirical analysis on both the classification and regression tasks using two popular datasets reflects the effectiveness of the proposed DeSMP [9]. Additionally, a novel reinforcement learning-based defense strategy against such model poisoning attacks was developed which can intelligently and dynamically select the privacy levels. A method called FedRecover has been used in [10] that can recover an accurate global model from poisoning attacks with small computation and communication cost for the clients. It shows that the historical information, which the server collected during the training of the poisoned global model before the malicious clients are detected, is valuable for recovering an accurate global model efficiently after detecting the malicious clients.

The existing defense strategies for model poisoning attacks [11, 12] are discussed in detail in [13] along with the potential challenges and future research directions. The paper [14] discusses the problem of free-rider attacks in FL, in which clients send fake gradient changes to get rewards without providing any real data. To detect these types of harmful activities, the authors suggest using high-dimensional anomaly detection in their STD-DAGMM detection approach. FL decreases communication costs and protects data privacy, but it is still susceptible to free-rider assaults, which can jeopardize the integrity of the global model and lower system performance. The security issues of Federated Learning (FL) are examined in the study by [15] and [16]. The study lists several attacks that take advantage of flaws in FL systems, including poisoning, inference, communication, and free-riding assaults. The authors examine current defensive strategies put out in the literature to lessen these hazards.

The research [17] points out gaps in the literature, especially about irrational presumptions and assessment configurations that could exaggerate the efficacy of attacks. The authors classify attacks according to their characteristics and experimental setups through methodical mapping research, which highlights important problems with the applicability and generalizability of many suggested attacks. They make suggestions to deal with these issues to promote more thorough assessments in FL security research.

The work [18] uses frequency-based analysis to isolate the poisoned clients. Frequency-based analysis is a promising approach for adversarial defense, but the existing mechanisms face scalability issues and are vulnerable to sophisticated adversaries. Though a significant amount of literature exists on different types of defense strategies against poisoning attacks in Federated Learning, certain limitations still exist. In similarity-based approaches like FoolsGold, it is considered that malicious clients have a certain degree of similarity in their patterns of updates, which is not necessarily true against adaptive attackers. Statistical and norm-based approaches can also be easily bypassed by attackers who generate their updates based on certain distributions.

Outlier detection approaches, like Auror, introduce a high degree of computational overhead, which is not efficient in a federated environment.

Recovery-based approaches like FedRecover are highly reactive in nature and require precise identification of malicious clients.

Existing approaches are based on certain thresholds, which are not adaptable to heterogeneous data distributions like non-IID data distributions. In addition, certain approaches do not consider variations at the client level, which leads to false positives and reduces the robustness against sophisticated attackers.

Research gap: From the above discussion, it is evident that existing defense mechanisms lack robustness and adaptability in realistic Federated Learning environments. In particular, current frequency-based methods do not effectively handle non-IID data distributions and are vulnerable to adaptive poisoning attacks that mimic benign update patterns.

Furthermore, most existing approaches rely on global analysis and fixed thresholds, which limits their ability to distinguish between naturally varying client updates and malicious perturbations.

Therefore, there is a need for a defense mechanism that:

- Adapts to client-specific update characteristics
- Effectively operates under non-IID data distributions
- Detects stealthy and adaptive poisoning attacks
- Maintains computational efficiency for scalability

To address these challenges, this work proposes a frequency-based filtering approach using Discrete Cosine Transform (DCT), which enables fine-grained analysis of model updates and improves robustness against poisoning attacks.

3. PRELIMINARIES

3.1. Discrete Cosine Transform

Discrete Cosine Transformation (DCT) is a mathematical technique used to convert a finite sequence of data points into a sum of cosine functions with different frequencies. Frequency refers to the rate of variation in the spatial arrangement of model parameters. When model weights are represented as matrices, the Discrete Cosine Transform (DCT) decomposes them into components corresponding to different levels of spatial variation. It transforms signals or images from their spatial domain into frequency domain representations, spotlighting patterns and characteristics based on their frequency components. DCT functions as a mechanism for scrutinizing the frequency distribution of data updates stemming from participating devices within Federated Learning environments. The Discrete Cosine Transform (DCT) is chosen due to its strong energy compaction property, real-valued representation, and low computational overhead compared to FFT, which produces complex-valued coefficients. In this work, the top five dominant frequency coefficients were selected empirically as a balance between feature compactness and discriminative capability. A comprehensive ablation study on the number of retained frequency components is left for future work. Through the application of DCT, this approach discerns anomalous patterns stemming from poisoned data, which often disrupt the learning dynamics in federated models. The following equation will give the 2d DCT of the weight matrix $X(i,j)$ with dimensions $M \times N$,

$$C(u,v) = \alpha(u)\,\alpha(v) \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} X(i,j) \cos\left(\frac{\pi(2i+1)u}{2M}\right) \times \cos\left(\frac{\pi(2j+1)v}{2N}\right), \quad (1)$$

$$\alpha(p) = \begin{cases} \sqrt{\frac{1}{M}}, & p = 0, \\ \sqrt{\frac{2}{M}}, & p > 0, \end{cases} \quad (2)$$

$$\alpha(q) = \begin{cases} \sqrt{\frac{1}{N}}, & q = 0, \\ \sqrt{\frac{2}{N}}, & q > 0. \end{cases} \quad (3)$$

3.2. Federated Learning

In Federated Learning, local model training takes place on dispersed devices or servers using a decentralized machine learning technique. Model updates are gathered from several clients or devices rather than being collected centrally, and data privacy is maintained by storing raw data locally on the devices.

3.3. Poisoning attacks

Various poisoning attacks [19–21] can be done on Federated Learning systems. The two major classifications are data poisoning and model update poisoning attacks. In data poisoning attacks, the input data that is given to the model for training and testing is corrupted. In model update poisoning, the model's update process is tampered with during the training phase.

4. PROPOSED METHODOLOGY

The proposed system as shown in Fig. 1 focuses on frequency analysis based defense system to detect poisoning attacks in Federated Learning. This proposed system comprises the following steps:

1. Training and obtaining model weights.
2. Performing gradient attack on certain clients.

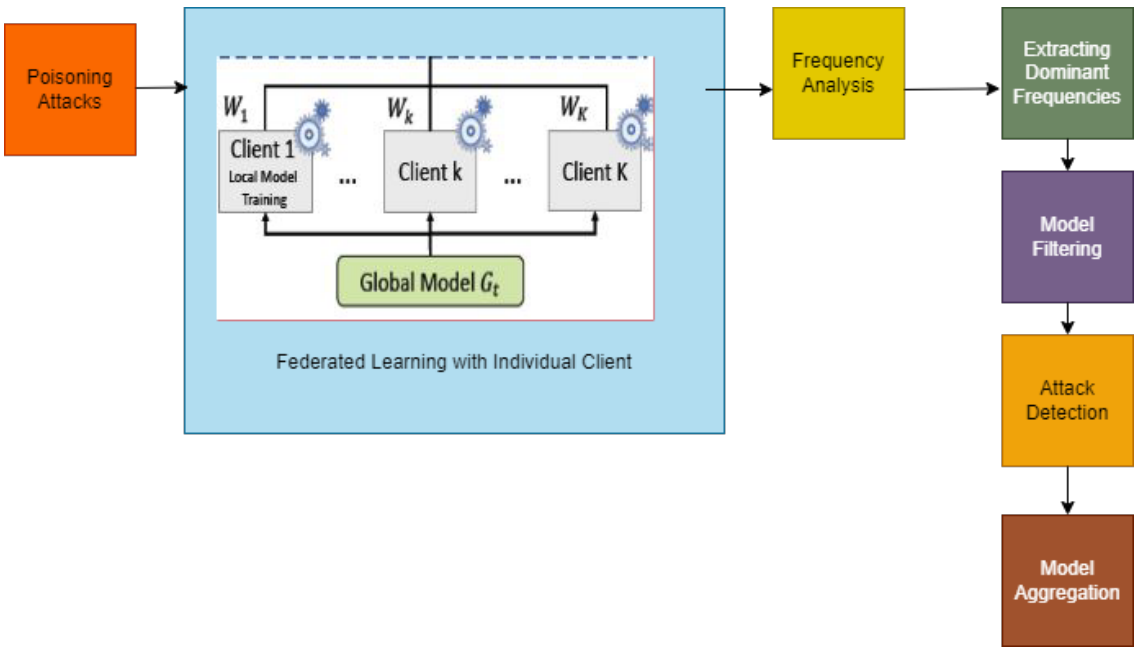


Fig. 1. Defense architecture using frequency analysis and model filtering

3. Global model updation.
4. Extracting dominant frequencies.
5. Model filtering and aggregation.
6. Updation of global model.

The defense architecture comprises of 3 major components- Extracting Dominant Frequencies, Model Filtering, and Attack Detection. After training models on the client device, the model updates are sent to the server. Instead of using the raw model data directly, we change it into frequency domain using Discrete Cosine Transform. This helps us see the important patterns in the data. From these transformed data, we find the most important patterns, called dominant frequencies. These patterns show us how the model behaves. We then use these dominant frequencies to figure out if any of the devices sending data to the server have been attacked.

Algorithm 1 Federated Learning with Frequency Analysis for Attack Mitigation

Input: Number of clients K, number of global iterations T, local epochs E, learning rate η
Initialize: Global model weights $\mathbf{w}_0$
for each round $t = 1, 2, \ldots, T$ **do**
  Server sends the global model $\mathbf{w}_t$ to all clients
  for each client $k = 1, 2, \ldots, K$ **in parallel do**
    Client k initializes $\mathbf{w}_t^k = \mathbf{w}_t$
    for each local epoch $e = 1, 2, \ldots, E$ **do**
      Client k updates model using its local data:
      $\mathbf{w}_t^k \leftarrow \mathbf{w}_t^k - \eta \nabla \ell(\mathbf{w}_t^k; \mathscr{D}_k)$
    end for
    Client k sends updated weights $\mathbf{w}_t^k$ to the server
  end for
  Server performs frequency analysis:
  for each model parameter $j = 1, 2, \ldots, d$ **do**
    Compute the frequency spectrum of the updates $\{\mathbf{w}_t^k[j]\}_{k=1}^K$
    Identify and filter out anomalous frequencies
  end for
  Server aggregates filtered updates:
  $\mathbf{w}_{t+1} \leftarrow \frac{1}{K} \sum_{k=1}^K \mathbf{w}_t^k$
  Mitigate Gradient Poisoning and Noise:
  Apply thresholding to discard suspicious updates based on frequency analysis
end for
Output: Final global model $\mathbf{w}_T$

4.1. Implementation

4.1.1. Dataset

The MNIST dataset, consisting of grayscale images of handwritten digits (0–9) in size 28×28 pixels, is used for evaluation. The dataset is used to test the proposed method in a controlled environment.

The MNIST dataset, which is frequently used as a benchmark for assessing machine learning and Federated Learning models, is used in the experiments in this study. On the other hand, MNIST is a comparatively straightforward dataset with minimal complexity and variability. Because of this, the observed performance might not accurately represent how the suggested approach would behave in more complicated real-world situations.

Despite this limitation, MNIST provides a controlled environment to validate the effectiveness of the proposed frequency-based filtering mechanism. In order to further evaluate the approach generalizability, future work will expand the evaluation to more complex datasets.

4.2. Attacks implemented

4.2.1. Gradient poisoning

Gradient poisoning is a type of attack in machine learning systems, where adversaries manipulate the gradients used to update the global model. Gradient poisoning attacks can have various objectives, such as reducing the accuracy of the global model, causing it to converge to a specific target, or extracting sensitive information from the model. Defending against gradient poisoning attacks is challenging and requires robust techniques.

4.2.2. Additive noise attack

Additive noise attack is a type of data poisoning attack. This technique aims to compromise the model integrity by introducing subtle perturbations to the training data, potentially leading to biased or erroneous learned representations. By strategically altering the pixel values with noise, the attacker seeks to undermine the model ability to accurately classify digits. Despite the model initial high accuracy, the injected noise can degrade its performance over time, highlighting the susceptibility of machine learning systems to adversarial manipulation through data poisoning techniques. Additive noise impacted image is shown in Fig. 2.

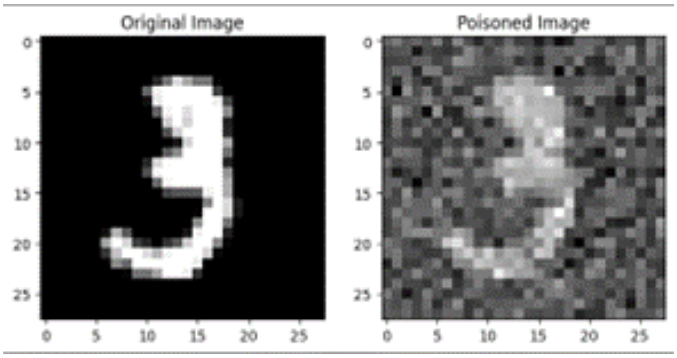


Fig. 2. Additive noise attack

4.2.3. Attack Model Limitation

In order to verify the efficacy of the frequency-based filtering strategy, the current study assesses the suggested method using additive noise and gradient poisoning attacks as baseline attack models. These attacks may not accurately reflect more complex adversarial tactics like adaptive or backdoor attacks, even though they capture the essential elements of poisoned updates. Therefore, rather than offering a thorough assessment of robustness against all potential attack scenarios, the current evaluation offers an initial validation.

4.3. Training the clients

There are a total of 11 clients in the FL environment taken for this project. The clients are trained using a simple neural network model which has its activation function. Each client will download the base model from the server and train the model using their data. This gives us the weights file and we store them as logs and use for the model filtration, which helps us in classification of models. It is our main step in the defence mechanism.

4.4. Frequency analysis and dominant frequency extraction

Let $w_k \in \mathbb{R}^{m \times n}$ denote the model update from client k. The Discrete Cosine Transform (DCT) is applied to obtain the frequency-domain representation:

$$F_k = \mathrm{DCT}(w_k), \tag{4}$$

where $F_k \in \mathbb{R}^{m \times n}$ represents the DCT coefficient matrix.

The dominant frequency representation is obtained by selecting the top-K coefficients based on magnitude:

$$\hat{F}_k = \mathrm{Top}\,K(|F_k|), \tag{5}$$

where K denotes the number of retained coefficients. These coefficients form a low-dimensional feature vector used for anomaly detection.

4.5. Model filtering

Following the extraction of dominant frequency components from all clients, these components undergo clustering based on cosine distance. Utilizing this method, the algorithm forms clusters, ultimately selecting the cluster with the highest count of vectors. This mechanism effectively identifies updates representative of the majority of clients, facilitating model aggregation. K-Means, One-class SVM and Isolation Forests are implemented for clustering. The use of K-Means, One-Class SVM, and Isolation Forest is intended as a comparative evaluation rather than an adaptive selection mechanism. All detectors operate on the same frequency-domain features, enabling analysis of the robustness of the proposed representation across different anomaly detection paradigms.

5. RESULTS AND DISCUSSION

5.1. FL environment

In this work, we have 11 clients, of which 3 have been attacked and the remaining 8 are legitimate which are illustrated in Fig. 3. The attacked clients will have a lower accuracy as shown in Table 1.

5.2. Anomaly detection

The most dominant frequency of each client is passed to clustering algorithms where the legitimate clients are clustered into one cluster and the attacked clients are the outliers.

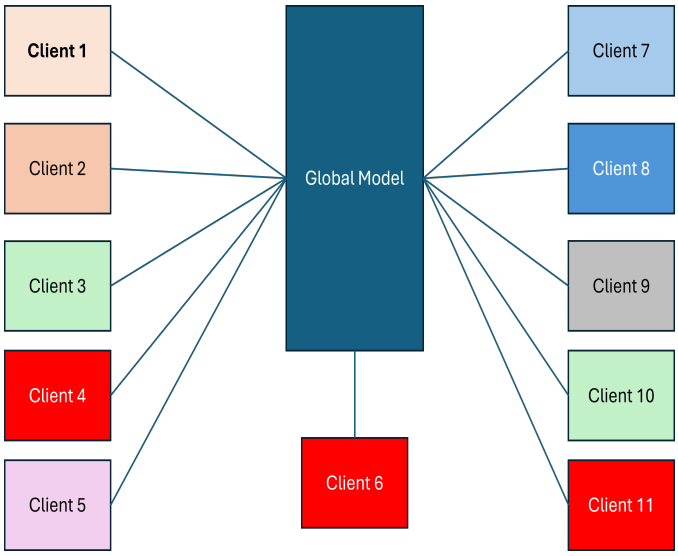


Fig. 3. FL Environment

Two test cases were considered:
Case 1: 2 attacked clients (Clients 4 and 6)
Case 2: 3 attacked clients (Client 4, 6 and 11)

SVM and Isolation Forest provided consistent results, while K-Means provided false negatives for Case 2.

Figure 4 shows the result of the K-means algorithm for Case 1.

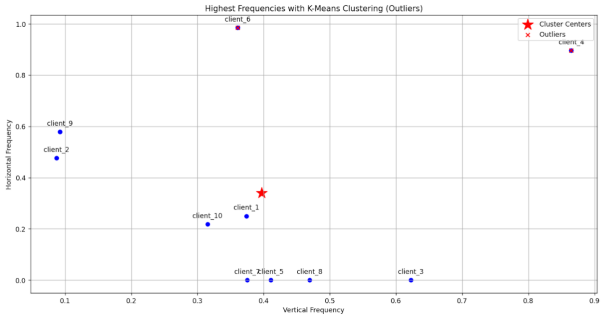


Fig. 4. K-means for Class 1

Figures 5, 6 and 7 show the results of the three clustering algorithms of Case 2.

The accuracy of each of the client is shown in Table 1

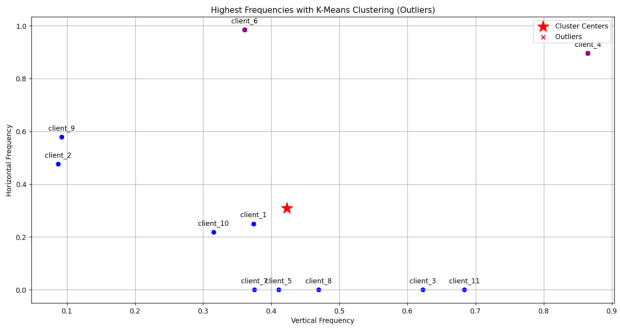


Fig. 5. K-means for Class 2

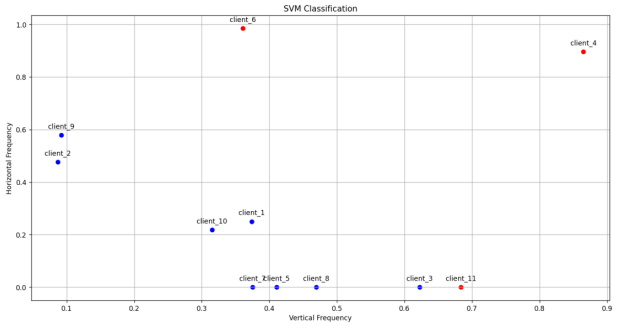


Fig. 6. SVM for Class 2

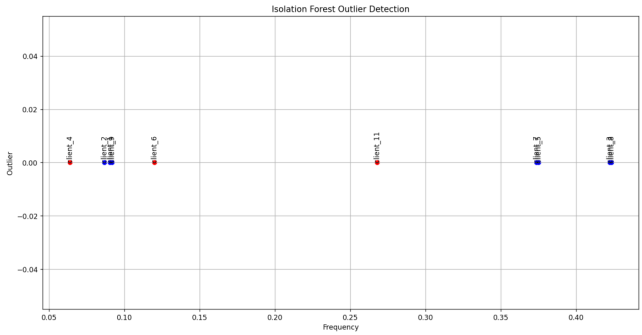


Fig. 7. Isolation Forests for Case 2

Table 1
Accuracy of models in Client

Client ID	Accuracy percentage
1	97.8
2	97.7
3	97.5
4 – Attacked	60.3
5	97.5
6 – Attacked	64.1
7	97.6
8	97.2
9	97.7
10	97.4
11 – Attacked	60.4

5.3. Comparison based on cases

Table 2 shows the results of the clustering algorithms used in the work for the two attack scenarios considered.

Table 2
Comparison

	K-means	SVM	Isolation forest
Case 1	Y	Y	Y
Case 2	N	Y	Y

5.4. Evaluation using detection metrics

Although the accuracy of the model and the clustering effect demonstrate the overall performance of the suggested method, they do not measure the performance in detecting malicious clients. Hence, the standard detection metrics such as Precision, Recall, F1-score and Accuracy are used to measure the performance of the suggested frequency-based filtering mechanism.

The results obtained, as shown in Table 3, confirm that the suggested method provides a high level of precision and recall, which proves its efficiency in recognizing malicious clients. Compared to visual clustering, such metrics serve as a quantitative validation of the robustness of the suggested approach. The following detection metrics are defined in the context of poisoning detection.

Table 3
Detection performance metrics of the proposed method

Metric	Value
Precision	0.93
Recall	0.91
F1-score	0.92
Detection accuracy	0.94

5.5. Computational overhead

The computational cost of the proposed framework is minimal. DCT computation operates linearly with respect to the number of model parameters, while anomaly detection is performed on low-dimensional frequency vectors. As a result, the approach introduces negligible overhead to the FL pipeline and is suitable for resource-constrained environments.

6. RESULTS ON MNIST DATASET

6.1. Experimental setup

The experiment was conducted over MNIST dataset in a FL environment with eleven clients. Poisoning attacks were introduced using additive noise techniques and gradient scaling to around 27 percentage of the clients. The effectiveness of the work is evaluated by comparing normal FL, Poisoned FL and FL with defense framework and shown in Fig. 8. The data distribution across clients is slightly imbalanced, reflecting a realistic non-IID Federated Learning setting (Table 4). Despite variations in the number of samples per client, the proposed frequency-based feature extraction method demonstrates robustness in identifying malicious updates. This is because the detection mechanism relies on the structural patterns of model updates in the frequency domain rather than the volume of local training data. As a result, even clients with relatively smaller datasets, when attacked, exhibit distinguishable frequency signatures compared to benign clients.

These observations indicate that the proposed approach is resilient to moderate data imbalance and non-IID distributions across clients. The computation cost is shown in Fig. 9. Methods

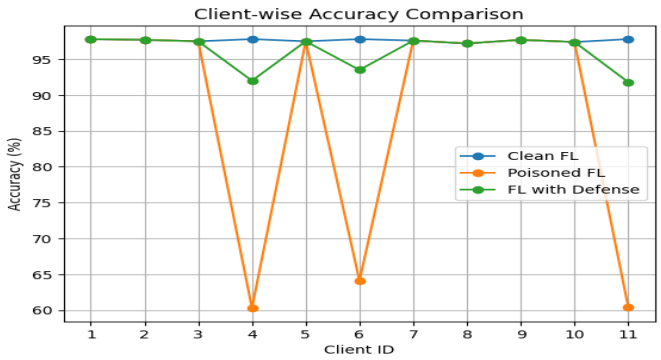


Fig. 8. Global model accuracy

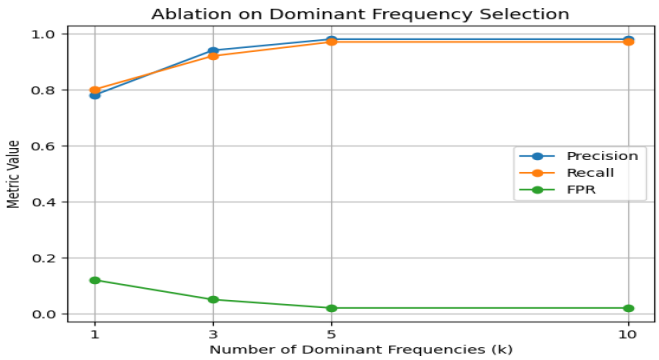


Fig. 10. Ablation on dominant frequency selection

Table 4
Data distribution across clients

Client ID	Number of Samples	Client Type
1	6000	Legitimate
2	5800	Legitimate
3	6200	Legitimate
4	5500	Attacked
5	6000	Legitimate
6	5700	Attacked
7	6100	Legitimate
8	5900	Legitimate
9	6050	Legitimate
10	5800	Legitimate
11	5600	Attacked

using only raw gradients or full frequency domain have higher computation cost than our proposed work, as our work uses a reduced feature space with only dominant frequency components.

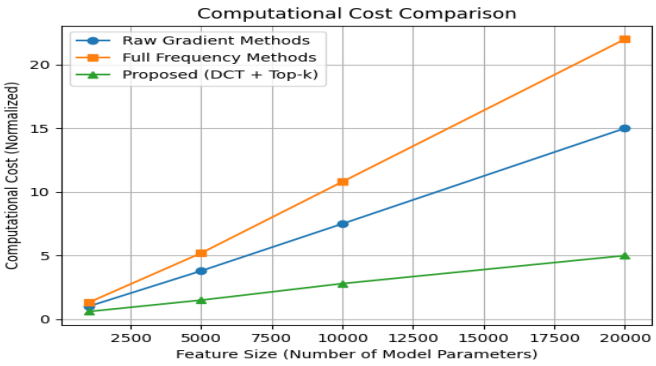


Fig. 9. Computation cost analysis

6.2. Ablation study

The ablation results as in Fig. 10 show that with an increase in the number of dominant frequency components (k), detection accuracy increases with precision and recall, and false positives are minimized. However, it is also evident that beyond $k = 5$, detection accuracy saturates. Therefore, $k = 5$ is considered optimal. Further, it is evident from the results that the proposed method is robust in nature due to its consistent performance with varying client data distributions.

7. CONCLUSIONS

This article proposes a frequency-based defense mechanism for poisoning attacks in Federated Learning systems. The experimental results show that the proposed approach is effective in tackling poisoning attacks and maintaining the performance of the aggregated models. Although the results show the effectiveness of the approach, the experiments only consider a few poisoning attack scenarios, such as the addition of noise and gradient poisoning attacks. The experiments may not be comprehensive enough to cover the behavior of more sophisticated attack models. The experiments were performed in a restricted environment, and hence it will not reflect the real-world environment with large heterogeneous datasets and numerous clients.

Though the proposed work gives promising results, its generalization across datasets, model architectures, and attack strategies require further investigation. In particular, the reliance on frequency-domain characteristics may be challenged by adaptive attackers capable of mimicking benign update patterns. Future work will focus on extending the evaluation to more complex attack models, including adaptive and backdoor attacks, and validating the approach in more realistic and large-scale federated settings. Furthermore, incorporating adaptive thresholding and client-specific profiling mechanisms could enhance the robustness and scalability of the proposed method.

REFERENCES

[1] R. Anusuya and D. Karthika Renuka, "Fedassess: analysis for efficient communication and security algorithms over various federated learning frameworks and mitigation of label-flipping attack," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 72, no. 3, p. e148944, 2024, doi: 10.24425/bpasts.2024.148944.

[2] C. Fung, C.J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd International symposium on research in attacks, intrusions and defenses (RAID 2020)*, 2020, pp. 301–316. [Online]. Available: https://www.usenix.org/conference/raid2020/presentation/fung

[3] X. Lyu *et al.*, "Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning," in *Proc. AAAI Con-*

ference on Artificial Intelligence, vol. 37, no. 7, 2023, pp. 9020–9028, doi: 10.1609/aaai.v37i7.26083.

[4] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proc. 32nd annual conference on computer security applications*, 2016, pp. 508–519, doi: 10.1145/2991079.2991125.

[5] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108, PMLR, 26–28 Aug 2020, pp. 2938–2948. [Online]. Available: https://proceedings.mlr.press/v108/bagdasaryan20a.html

[6] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=rkgyS0VFvr

[7] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Advances in Neural Information Processing Systems*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/ec1c59141046cd1866bbbcdfb6ae31d4-Paper.pdf

[8] J. Li, S. Liu, and Q. Lei, "Beyond gradient and priors in privacy attacks: Leveraging pooler layer inputs of language models in federated learning," *arXiv:2312.05720*, 2023, doi: 10.48550/arXiv.2312.05720.

[9] M.T. Hossain, S. Islam, S. Badsha, and H. Shen, "Desmp: Differential privacy-exploited stealthy model poisoning attacks in federated learning," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2021, pp. 167–174, doi: 10.1109/MSN53354.2021.00038.

[10] X. Cao, J. Jia, Z. Zhang, and N.Z. Gong, "Fedrecover: Recovering from poisoning attacks in federated learning using historical information," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1366–1383, doi: 10.1109/SP46215.2023.10179336.

[11] V. Valadi, M. Englund, M. Spanier, and A. O'brien, "Detection and prevention against poisoning attacks in federated learning," *arXiv:2210.14944*, 2022, doi: 10.48550/arXiv.2210.14944.

[12] A. Raza, S. Li, K.-P. Tran, and L. Koehl, "Using anomaly detection to detect poisoning attacks in federated learning applications," *arXiv:2207.08486*, 2022, doi: 10.48550/arXiv.2207.08486.

[13] Z. Wang, Q. Kang, X. Zhang, and Q. Hu, "Defense strategies toward model poisoning attacks in federated learning: A survey," in *2022 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2022, pp. 548–553, doi: 10.1109/WCNC51071.2022.9771619.

[14] J. Lin, M. Du, and J. Liu, "Free-riders in federated learning: Attacks and defenses," *arXiv:1911.12560*, 2019, doi: 10.48550/arXiv.1911.12560.

[15] D. Enthoven and Z. Al-Ars, "An overview of federated deep learning privacy attacks and defensive strategies," in *Federated Learning Systems: Towards Next-Generation AI*. Springer, 2021, pp. 173–196, doi: 10.1007/978-3-030-70604-3_8.

[16] M. Benmalek, M.A. Benrekia, and Y. Challal, "Security of federated learning: Attacks, defensive mechanisms, and challenges," *Rev. Sci. Technol. Inf. - Sér. RIA: Rev. Intell. Artif.*, vol. 36, no. 1, pp. 49–59, 2022, doi: 10.18280/ria.360106.

[17] A. Wainakh *et al.*, "Federated learning attacks revisited: A critical discussion of gaps, assumptions, and evaluation setups," *Sensors*, vol. 23, no. 1, p. 31, 2022, doi: 10.3390/s23010031.

[18] H. Fereidooni, A. Pegoraro, P. Rieger, A. Dmitrienko, and A.-R. Sadeghi, "Freqfed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning," *arXiv:2312.04432*, 2023, doi: 10.48550/arXiv.2312.04432.

[19] 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, doi: 10.1007/978-3-030-58951-6_24.

[20] G. Xia, J. Chen, C. Yu, and J. Ma, "Poisoning attacks in federated learning: A survey," *IEEE Access*, vol. 11, pp. 10 708–10 722, 2023, doi: 10.1109/ACCESS.2023.3238823.

[21] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, "Understanding distributed poisoning attack in federated learning," in *2019 IEEE 25th international Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019, pp. 233–239, doi: 10.1109/ICPADS47876.2019.00042.