

RPPFL: Robust and Privacy-Preserving Federated Learning via Trusted Execution Environments

1st Xiaolei Zhang

School of Computer Science
Peking University
Beijing, China
xl_zhang@pku.edu.cn

2nd Zhaoyu Chen

School of Software and Microelectronics
Peking University
Beijing, China
chenzhaoyu@stu.pku.edu.cn

3rd Guangpu Chen

School of Software and Microelectronics
Peking University
Beijing, China
chengguangpu@stu.pku.edu.cn

4th Xinyu Feng

School of Software and Microelectronics
Peking University
Beijing, China
xyf@pku.edu.cn

5th Qingni Shen[†]

School of Software and Microelectronics
Peking University
Beijing, China
qingnishen@ss.pku.edu.cn

6th Zhonghai Wu[†]

School of Software and Microelectronics
Peking University
Beijing, China
wuzh@pku.edu.cn

Abstract—Federated Learning (FL) is a distributed framework that enables multi-participant collaborative model training without the need for data sharing. Despite its advantages, FL is vulnerable to poisoning and inference attacks, which compromise model accuracy and data privacy. Trusted execution environments (TEEs) offer a potential solution by providing a secure and isolated execution space to address these security and privacy concerns in FL. However, existing TEE-based FL schemes often suffer from reduced training speed and compromised model accuracy. To mitigate these issues, we propose a robust and privacy-preserving framework for federated learning (RPPFL) that leverages TEE and pseudorandom masking. In our approach, a trusted local model is trained on a secure subset of local data within the client-side TEE, which is then used for anomaly detection to resist poisoning attacks. Additionally, we employ pseudorandom masking to obfuscate local updates and global parameters. Experimental results indicate that RPPFL effectively counters both poisoning and inference attacks, with only a minimal decrease in training speed and no adverse impact on model accuracy. Compared to full-TEE approaches, our method improved local training efficiency by $10\times$, with less than a 9% loss in model performance under poisoning attacks.

Index Terms—Federated Learning, Trusted Execution Environments, Data Obfuscation, Robustness, Privacy Preservation.

I. INTRODUCTION

Federated learning (FL) [1]–[3] is a distributed machine learning framework that enables participants to collaboratively train models without the need to share their data. This characteristic is particularly crucial for fields with stringent data privacy requirements, such as medicine [4], [5] and finance [6], [7]. Despite its advantages, numerous studies have shown that FL still encounters significant security and privacy challenges. In terms of security, attackers can compromise individual participants and carry out data or model poisoning attacks, which adversely affect the model’s performance [8]–[14]. Regarding privacy, research has indicated that private

information about participants’ raw data can be inferred by intercepting local updates or global parameters [15]–[19]. Consequently, ensuring security and preserving privacy in FL remain critical issues that need to be addressed.

Recently, privacy-preserving Federated Learning (FL) frameworks based on Trusted Execution Environments (TEEs) have emerged as promising solutions [20]–[24]. These frameworks typically deploy TEEs on participants’ devices to conduct local training within the secure environment, thereby preventing data or local updates from being compromised. Additionally, they employ differential privacy (DP) [22], [23] or public key cryptography (PKC) [20], [21], [24] to protect local updates and global parameters, ensuring that participants’ raw data cannot be inferred. While these schemes effectively address both security and privacy concerns in FL, they are not without drawbacks. Specifically, conducting complete local training within a TEE significantly reduces training speed, making it one or two orders of magnitude slower than execution on a GPU. This slowdown is particularly pronounced in the training of complex models. Furthermore, during the aggregation phase, schemes based on the combination of TEE and PKC incur additional communication and computational overhead. In contrast, schemes based on TEE and DP introduce noise that adversely affects model performance, including accuracy and convergence speed.

To address the aforementioned issues, we propose a novel framework that combines Trusted Execution Environment (TEE) and pseudorandom masking to enhance poisoning attack resistance and privacy preservation during model training in FL, while maintaining model performance and training speed. Our framework is composed of two primary components: TEE-based anomaly detection and privacy-preserving aggregation. Unlike previous TEE-based schemes, our approach ensures training speed by conducting the training using the full local dataset outside the TEE. In the TEE-based anomaly detection component, the TEE maintains a trusted

This work was supported by the National Key R&D Program of China under Grant No. 2022YFB2703301. [†] denotes the corresponding authors.

local model trained on a secure subset of the local data, which is then used to detect anomalies in the local model trained outside the TEE. For privacy-preserving aggregation, we employ carefully designed pseudorandom masks to obfuscate local updates and global parameters, thereby preventing the inference of private information from participants' data. Importantly, these masks do not affect the aggregation results and introduce no additional aggregation overhead.

In this paper, we propose a Robust and Privacy-Preserving framework for Federated Learning (RPPFL). To resist client-side poisoning attacks without significantly degrading training speed, we utilize a Trusted Execution Environment (TEE) to collect a secure subset of local data before initiating local training. Within the TEE, we train a trusted local model using this subset, which is subsequently employed for anomaly detection. Additionally, to protect the privacy of participants' data, we implement a data obfuscation method based on pseudorandom masking. This method effectively degrades inference attacks on local updates and global parameters to the level of random guessing. Our contributions can be summarized as follows:

- We propose a new framework named RPPFL to address security and privacy challenges in FL. This framework introduces a method that seamlessly integrates computations both within and outside the Enclave, effectively balancing efficiency and robustness during FL training. By facilitating efficient local updates and secure parameter aggregation, our approach ensures a streamlined FL process.
- To enhance the security of FL, we propose a novel threat model that addresses three critical aspects: local data security, parameter aggregation security, and model robustness. To operationalize this threat model, we incorporate trusted execution environments and zero-sharing-based pseudorandom masking techniques. These innovations collectively enable the development of a secure and efficient federated learning framework, ensuring resistance to potential attacks while maintaining high performance.
- We conducted extensive experiments to evaluate the performance and efficiency of the proposed framework. The results demonstrated that, compared to full-TEE privacy-preserving FL approaches, our method achieved a $10\times$ improvement in local training efficiency. Additionally, the model performance loss was less than 9% when subjected to poisoning attacks.

II. PRELIMINARIES

Federated learning. Federated Learning enables collaborative model training by aggregating local updates from multiple participants. For participant i , it uses local data x_i to train local model parameters θ_i (local update) and send it to the central server. The central server aggregates these local updates to generate global model parameters $\bar{\theta}$ (global parameters) as $\bar{\theta} = \sum \frac{m_i}{m} \theta_i$ where m_i is the weight of θ_i and $m = \sum m_i$.

Software Guard eXtensions. Software Guard eXtensions (SGX) provides protected isolated memory spaces (enclaves) for applications. The enclave prevents any other process from accessing this memory. SGX supports remote attestation to

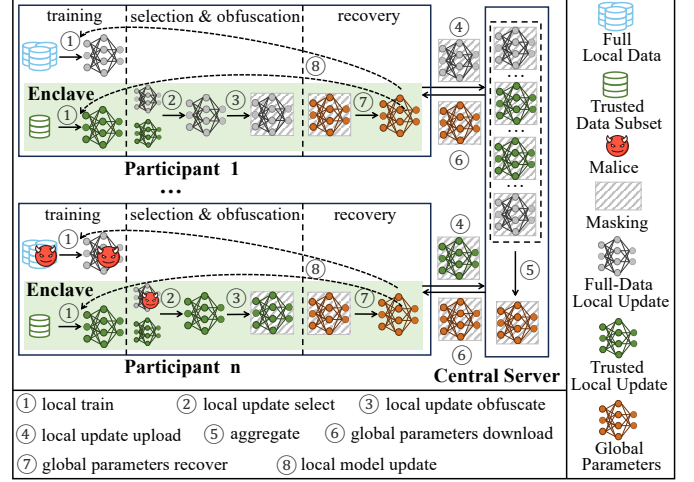


Fig. 1: The overview of RPPFL.

prove the program is running correctly in the enclave. For simplicity, we follow the formal model of SGX presented in [25] where each enclave instance produces an attestation key pair, which is used to indicate the enclave's identity.

System and threat model. There are 2 types of parties in our FL framework: (1) *Participant*. There are multiple participants equipped with SGX in our framework, each participant performs local training and uploads the local update to the central server in each training round. (2) *Central Server*. The central server aggregates the participants' local updates and returns the global parameters to them at the end of the training round. The threat model in this paper is defined as follows: (1) The participants may act maliciously (either voluntarily or be compromised by attackers), and they launch data or local poisoning attacks to impact model performance. (2) The central server is semi-honest, it follows the protocol of FL but attempts to infer private information about users' raw data from local updates or the aggregated global parameters.

III. METHOD

Our scheme is structured into two primary phases: Initialization and Federated Learning (FL). The Initialization phase, which sets the stage for the subsequent FL phase, encompasses four critical steps: enclave attestation, initial parameter synchronization, secure data subset collection, and mask negotiation. As illustrated in Fig. 1, a complete FL round comprises eight steps, which can be categorized into four distinct parts: training, selection and obfuscation, aggregation, and recovery and update. In the following sections, we will provide a detailed explanation of each step.

Initialization. This phase is executed only once.

(1) **Enclave attestation.** The central server verifies the enclaves and the code in them through remote attestation.

(2) **Initial parameter synchronization.** The central server distributes the initial parameters θ_0 to the verified enclaves. The latter initializes the local model and synchronizes it with the program outside the enclave.

(3) **Secure data subset collection.** After initialization, the enclave randomly collects a subset of local data D as the

secure data subset D' . Different sizes of D' will strike a tradeoff between model performance and training speed. We will discuss the selection of its size in our experiments.

(4) Mask negotiation. Building upon the literature [25], we design a pseudorandom mask that can obfuscate both local updates and global parameters. Assume a total order on verified enclaves $\mathbf{e} = \{e_1, \dots, e_i, \dots, e_n\}$ where e_i refers to an enclave and n is the number of verified enclaves. All enclaves share a common value k_0 and a pseudorandom function F . Each e_i maintains a personal value k_i and shares it with e_{i-1} . In the r -th FL round, the mask of e_i is $\alpha_i^r = F_{k_i}(r) - F_{k_{i+1}}(r) + F_{k_0}(r)$. Since local updates are aggregated based on the weight of training data, e_i multiplies the mask by a weighting factor. The masked local update is denoted as $[[\theta_i^r]] = \theta_i^r + \frac{\alpha_i^r}{m_i}$ where θ_i^r refers to the local update and m_i is the training data size for θ_i^r . The aggregation process is $\sum \frac{m_i}{m} [[\theta_i^r]] = \sum \frac{m_i}{m} \theta_i^r + \sum \frac{\alpha_i^r}{m}$ where $m = \sum m_i$. Since $\sum \alpha_i^r = nF_{k_0}(r)$, the aggregated result is $\sum \frac{m_i}{m} \theta_i^r + \frac{nF_{k_0}(r)}{m}$. It is clear that the global parameters $\sum \frac{m_i}{m} \theta_i^r$ is obfuscated by the random value $\frac{nF_{k_0}(r)}{m}$. To obtain the global parameters, the enclave only needs to eliminate $\frac{nF_{k_0}(r)}{m}$. This value can be computed by all enclaves. Compared to existing schemes, we protect the privacy of both local updates and global parameters without introducing noise and extra aggregation overhead.

Our proposed FL framework. There are multiple learning rounds in our FL framework. We use the r -th round as an example to illustrate the detailed process.

(1) Training. For participant i , local training using the full local dataset D_i and the global parameters $\bar{\theta}^{r-1}$ is conducted outside the enclave to obtain full-data local update $\theta_{i,F}^r$. Concurrently, training is performed inside the enclave using the secure subset D'_i and $\bar{\theta}^{r-1}$ to generate trusted local update $\theta_{i,T}^r$. As the training within the enclave is protected by SGX, it is secure from poisoning attacks. Consequently, only the training outside the enclave is susceptible to such attacks, making $\theta_{i,F}^r$ potentially compromised.

(2) Selection and obfuscation. After local training, we prefer to upload the full-data local update $\theta_{i,F}^r$ because it contains more features of local data. However, considering the potential risk of poisoning, we use the trusted local update $\theta_{i,T}^r$ to verify it. Specifically, the enclave calculates the Euclidean distance $d_{(F-T)}$ between $\theta_{i,F}^r$ and $\theta_{i,T}^r$, defined as $d_{(F-T)} = \|\theta_{i,F}^r - \theta_{i,T}^r\|_2$. We then specify a constant ξ to serve as a threshold. If $d_{(F-T)} \leq \xi$, it indicates that $\theta_{i,F}^r$ has not been poisoned, and the enclave selects $\theta_{i,F}^r$ for uploading. Otherwise, the enclave selects $\theta_{i,T}^r$. The local update uploaded by participant i in the r -th round is denoted as θ_i^r . The selection of the threshold ξ will be discussed in our analysis and experiments.

To safeguard the private information of local data, the enclave obfuscates the local update θ_i^r by adding a carefully designed pseudorandom mask weighted by the size of the training data. This process is mathematically represented as $[[\theta_i^r]] = \theta_i^r + \frac{1}{m_i} \cdot \alpha_i^r$, where $[[\theta_i^r]]$ denotes the masked local update, m_i refers to the weight of θ_i^r , and α_i^r is the pseudorandom

mask. The value of m_i is determined by the weights of θ_i^r : if θ_i^r is $\theta_{i,F}^r$, then $m_i = |D_i|$; otherwise, $m_i = |D'_i|$. The mask α_i^r is known only to the enclave and varies across different learning rounds, thereby preventing attackers from inferring any useful information from the masked local updates. Finally, the enclave sends $([[\theta_i^r]], m_i)$ to the server.

(3) Aggregation. The central server compiles the masked local updates into:

$$\tau^r = \{([[\theta_1^r]], m_1), \dots, ([[\theta_i^r]], m_i), \dots, ([[\theta_n^r]], m_n)\},$$

where n refers to the total number of participants. It then computes their weighted average to generate the masked global parameters $[[\bar{\theta}^r]]$ as:

$$[[\bar{\theta}^r]] = \sum_{i=1}^n \frac{m_i}{m} [[\theta_i^r]] = \sum_{i=1}^n \frac{m_i}{m} \theta_i^r + \frac{nF_{k_0}(r)}{m},$$

where $m = \sum_{i=1}^n m_i$, and n is the number of participants. $\sum_{i=1}^n \frac{m_i}{m} \theta_i^r$ refers to the global parameters $\bar{\theta}^r$. The global parameter $\sum_{i=1}^n \frac{m_i}{m} \theta_i^r$ is evidently obscured by the random value $\frac{nF_{k_0}(r)}{m}$. Finally, the central server distributes $([[\bar{\theta}^r]], m)$ to enclaves.

(4) Recovery and update. Upon receiving $([[\bar{\theta}^r]], m)$, the enclave eliminates the mask from $[[\bar{\theta}^r]]$ to obtain the global parameters $\bar{\theta}^r$ as follows:

$$[[\bar{\theta}^r]] - \frac{nF_{k_0}(r)}{m} = \sum_{i=1}^n \frac{m_i}{m} \theta_i^r + \frac{nF_{k_0}(r)}{m} - \frac{nF_{k_0}(r)}{m} = \bar{\theta}^r.$$

The enclave updates the local model using $\bar{\theta}^r$ and synchronizes it with the external program.

Analysis. In our study, we explored the impact of varying the size of the secure data subset on model training and accuracy. A larger secure data subset captures more local data features, thereby ensuring that a greater number of features participate in model training, even when the full-data local update outside the enclave is compromised. However, this advantage comes at the cost of increased local training time. On the other hand, a smaller secure data subset reduces local training time but includes fewer local data features. Through our experiments, we assessed the accuracy and training time associated with different sizes of the secure data subset across various datasets, ultimately recommending an optimal size.

Additionally, we investigated the parameter ξ and its role in sensitivity to data poisoning. A smaller ξ enhances sensitivity to poisoning, allowing for the rejection of a higher number of poisoned full-data local updates. However, this heightened sensitivity may also result in the rejection of more correct updates. Conversely, a larger ξ decreases sensitivity to poisoning, which could lead to the acceptance of more poisoned updates. Our experiments evaluated the optimal size of ξ and tested its resistance to poisoning attacks of varying intensities, providing recommendations based on our findings.

IV. EXPERIMENTS

Experimental setup. The experiments for our scheme were conducted using image classification tasks across various datasets, including MNIST [26], CIFAR-10 [27], breast cancer

classification, OCT disease classification [28], and medical signal classification [29]. Our scheme was implemented in C++11 utilizing the Occlum platform [30], with the size of the trusted data subset set to 10% of the full dataset. All FL frameworks were implemented in PyTorch 1.10, involving 10 clients, and executed on an Ubuntu 20.04 LTS server equipped with a 2.40GHz Intel Xeon Silver 4314 CPU, an NVIDIA GeForce RTX 3090 Ti GPU, and 64GB of RAM.

TABLE I: The comparison of model accuracy encompasses several schemes. In DP-based FL, Gaussian noise is employed, with the privacy parameter set to $\epsilon = 30$.

Schemes	MNIST	CIFAR-10	Breast Cancer	OCT Disease	Medical Signal
Plain	0.9058	0.3118	0.7531	0.9042	0.8332
DP [22]	0.8392	0.3065	0.6955	0.8342	0.7641
FHE [24]	<u>0.8996</u>	<u>0.3265</u>	<u>0.7361</u>	<u>0.8754</u>	<u>0.8081</u>
Ours	0.9045	0.3140	0.7534	0.9051	0.8312

Performance. The accuracy results of various schemes in the absence of poisoning attacks are detailed in Table I. Notably, our scheme demonstrates equivalent accuracy to the plain FL scheme, which lacks any security or privacy protections, across all datasets. This finding underscores that our approach successfully enhances security guarantees and privacy protection without compromising model accuracy.

Training time. Fig. 2 presents a comparative analysis of the training times for different methods across various datasets. The Full-TEE method [20] conducts complete local training within TEE, whereas the Full-GPU method performs complete local training on GPUs outside TEE. The results indicate that while our method incurs a marginally higher training time compared to the Full-GPU method, it achieves a substantial speedup relative to the Full-TEE method. This demonstrates that our approach strikes a balance between security benefits of TEE and the computational efficiency of GPU-based training.

Selection of the secure data subset. The accuracy results and training times of our scheme on different sizes of training datasets are illustrated in Fig. 3 and Fig. 4, respectively, which indicate that when the training data constitutes 10% of the full dataset, the accuracy loss is less than 9% across all tested datasets. Beyond this 10% threshold, the improvement in accuracy becomes progressively slower as the size of the training data increases. Conversely, the training time exhibits a linear relationship with the size of the training data, where the training time for the full dataset is ten times that of the 10% subset. Therefore, we recommend selecting a trusted subset size of 10% of the full dataset, as this configuration achieves the optimal balance between accuracy and training time.

Selection of Euclidean distance threshold. Fig. 5 illustrates the Euclidean distances between local updates trained on varying sizes of data subsets and the local update trained on the full dataset. It reveals that, across different datasets, when training data size is reduced to 10%, the Euclidean distances from the full dataset remain consistent and do not exceed 0.006. Based on this observation, to maximize the likelihood

of selecting local updates derived from the full dataset, we recommend setting the threshold parameter ξ to 0.006.

Resistance to poisoning attacks. We quantify the severity of poisoning attacks by measuring the Euclidean distance d between the poisoned local updates and the benign local updates trained using the full dataset. Table II presents the performance of methods in resisting poisoning attacks. Specifically, when $\xi = 0.006$, it implies that half of the poisoned local updates with $d \leq 0.012$ will be incorrectly classified as benign, thereby affecting the overall accuracy. However, the impact on accuracy is minimal due to the relatively low severity of these poisoning attacks. Conversely, poisoned local updates with $d > 0.012$ are rejected. The results indicate that for $d > 0.012$, the accuracy of our scheme is comparable to that achieved by training with 10% of the full dataset. In contrast, there is a slight decrease in accuracy when $d \leq 0.012$.

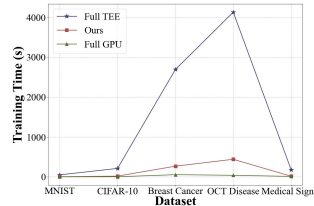


Fig. 2: Training time.

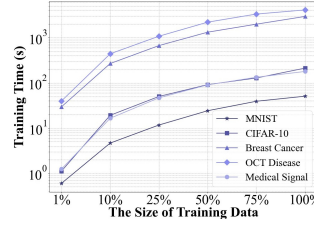


Fig. 4: Training time vs. data size.

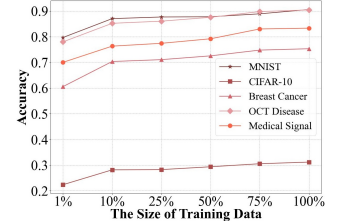


Fig. 3: Accuracy results.

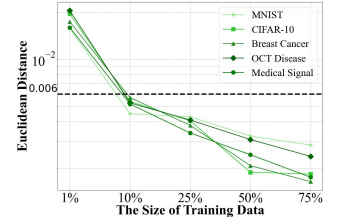


Fig. 5: Euclidean distances.

TABLE II: Accuracy results across various methods under different levels of poisoning.

Poisoning Levels	Methods	MNIST	CIFAR-10	Breast Cancer	OCT Disease	Medical Signal
$d \leq 0.012$	Full-TEE	0.9058	0.3118	0.7531	0.9042	0.8332
	Full-CPU	0.8267	0.2672	0.6648	0.8215	0.7216
	Ours	<u>0.8614</u>	<u>0.2801</u>	<u>0.6954</u>	<u>0.8412</u>	<u>0.7529</u>
$d > 0.012$	Full-TEE	0.9047	0.3123	0.7544	0.8993	0.8348
	Full-CPU	0.4675	0.1455	0.3932	0.4383	0.4176
	Ours	<u>0.8703</u>	<u>0.2866</u>	<u>0.7034</u>	<u>0.8521</u>	<u>0.7632</u>

V. CONCLUSION

In this paper, we propose a novel framework, RPPFL, designed to address both poisoning and inference attacks in FL. To effectively resist poisoning attacks, we utilize a trusted local model maintained within TEE to perform anomaly detection, achieving this with only a marginal increase in training time. Additionally, we obfuscate local updates and global parameters through pseudorandom masking, thereby preventing the inference of private information from participants' raw data. To validate the feasibility and effectiveness of RPPFL, we conducted a series of comparative experiments, the results of which demonstrate its robustness and efficiency.

REFERENCES

- [1] Yuhao Shen, Wei Xi, Yunyun Cai, Yuwei Fan, He Yang, and Jizhong Zhao, "Multi-objective federated learning: Balancing global performance and individual fairness," *Future Gener. Comput. Syst.*, vol. 162, pp. 107468, 2025.
- [2] Jiaqi Wu, Ruihan Zheng, Jiade Jiang, Zijian Tian, Wei Chen, Zehua Wang, Fei Richard Yu, and Victor C. M. Leung, "A lightweight small object detection method based on multilayer coordination federated intelligence for coal mine iomt," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 20072–20087, 2024.
- [3] Louis Mullie, Jonathan Afilalo, Patrick M. Archambault, Rima Bouchakri, Kip Brown, David L. Buckeridge, Yiorgos Alexandros Cavayas, Alexis F. Turgeon, Denis Martineau, François Lamontagne, Martine Lebrasseur, Renald Lemieux, Jeffrey Li, Michaël Sauthier, Pascal St-Onge, An Tang, William Witteman, and Michael Chassé, "CODA: an open-source platform for federated analysis and machine learning on distributed healthcare data," *J. Am. Medical Informatics Assoc.*, vol. 31, no. 3, pp. 651–665, 2024.
- [4] Bjarne Pfitzner, Nico Steckhan, and Bert Arnrich, "Federated learning in a medical context: A systematic literature review," *ACM Trans. Internet Techn.*, vol. 21, no. 2, pp. 50:1–50:31, 2021.
- [5] Ashish Rauniyar, Desta Haileselassie Hagos, Debesh Jha, Jan Erik Håkegård, Ulas Bagci, Danda B. Rawat, and Vladimir Vlassov, "Federated learning for medical applications: A taxonomy, current trends, challenges, and future research directions," *IEEE Internet Things J.*, vol. 11, no. 5, pp. 7374–7398, 2024.
- [6] Zhongyi Wang, Jin Xiao, Lu Wang, and Jianrong Yao, "A novel federated learning approach with knowledge transfer for credit scoring," *Decis. Support Syst.*, vol. 177, pp. 114084, 2024.
- [7] Bin Gu, An Xu, Zhouyuan Huo, Cheng Deng, and Heng Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 11, pp. 6103–6115, 2022.
- [8] Le Sun, Jing Tian, and Ghulam Muhammad, "Fedkcc: Personalized federated learning with robustness against model poisoning attacks in the metaverse for consumer health," *IEEE Trans. Consum. Electron.*, 2024.
- [9] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020, Srdjan Capkun and Franziska Roesner, Eds. 2020*, pp. 1605–1622, USENIX Association.
- [10] Fan Zhang, Hui Huang, Zhixiong Chen, and Zhenjie Huang, "Robust and privacy-preserving federated learning with distributed additive encryption against poisoning attacks," *Comput. Networks*, vol. 245, pp. 110383, 2024.
- [11] Xuan Li, Naiyu Wang, Shuai Yuan, and Zhitao Guan, "Fedimp: Parameter importance-based model poisoning attack against federated learning system," *Comput. Secur.*, vol. 144, pp. 103936, 2024.
- [12] Yiran Li, Shibin Zhang, Yan Chang, Guowen Xu, and Hongwei Li, "Privacy-preserving and poisoning-defending federated learning in fog computing," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 5063–5077, 2024.
- [13] Han Yang, Dongbing Gu, and Jianhua He, "A robust and efficient federated learning algorithm against adaptive model poisoning attacks," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16289–16302, 2024.
- [14] Xue-Yang Li, Xue Yang, Zhengchun Zhou, and Rongxing Lu, "Efficiently achieving privacy preservation and poisoning attack resistance in federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 4358–4373, 2024.
- [15] Matteo Pennisi, Federica Proietto Salanitri, Giovanni Bellitto, Bruno Casella, Marco Aldinucci, Simone Palazzo, and Concetto Spampinato, "FedER: Federated learning through experience replay and privacy-preserving data synthesis," *Comput. Vis. Image Underst.*, vol. 238, pp. 103882, 2024.
- [16] Jingwei Liu, Yating Li, Mengjiao Zhao, Lei Liu, and Neeraj Kumar, "Epfl: Enhancing privacy and fairness in federated learning for distributed e-healthcare data sharing services," *IEEE Transactions on Dependable and Secure Computing*, 2024.
- [17] Rihab Saidi, Tarek Moulahi, Suliman Aladhadh, and Salah Zidi, "Advancing federated learning: Optimizing model accuracy through privacy-conscious data sharing," in *25th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2024, Perth, Australia, June 4-7, 2024*, 2025, pp. 64–69, IEEE.
- [18] Syed Muhammad Salman Bukhari, Muhammad Hamza Zafar, Mohamad Abou Houran, Syed Kumayl Raza Moosavi, Majad Mansoor, Muhammad Muaaz, and Filippo Sanfilippo, "Secure and privacy-preserving intrusion detection in wireless sensor networks: Federated learning with scnn-bi-lstm for enhanced reliability," *Ad Hoc Networks*, vol. 155, pp. 103407, 2024.
- [19] Wei Yang, Yuan Yang, Yingjie Xi, Hailong Zhang, and Wei Xiang, "FLCP: federated learning framework with communication-efficient and privacy-preserving," *Appl. Intell.*, vol. 54, no. 9-10, pp. 6816–6835, 2024.
- [20] Yu Chen, Fang Luo, Tong Li, Tao Xiang, Zheli Liu, and Jin Li, "A training-integrity privacy-preserving federated learning scheme with trusted execution environment," *Inf. Sci.*, vol. 522, pp. 69–79, 2020.
- [21] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis, "PPFL: privacy-preserving federated learning with trusted execution environments," in *MobiSys '21: The 19th Annual International Conference on Mobile Systems, Applications, and Services, Virtual Event, Wisconsin, USA, 24 June - 2 July, 2021*, Suman Banerjee, Luca Mottola, and Xia Zhou, Eds. 2021, pp. 94–108, ACM.
- [22] Fan Mo and Hamed Haddadi, "Efficient and private federated learning using tee," in *Proc. EuroSys Conf., Dresden, Germany*, 2019.
- [23] Arup Mondal, Yash More, Ruthu Hulikal Rooparagunath, and Debayan Gupta, "Poster: FLATEE: federated learning across trusted execution environments," in *IEEE European Symposium on Security and Privacy, EuroS&P 2021, Vienna, Austria, September 6-10, 2021*, 2021, pp. 707–709, IEEE.
- [24] Fengwei Wang, Hui Zhu, Xingdong Liu, Yandong Zheng, Hui Li, and Jiafeng Hua, "Achieving federated logistic regression training towards model confidentiality with semi-honest TEE," *Inf. Sci.*, vol. 679, pp. 121115, 2024.
- [25] Toshinori Araki, Jun Furukawa, Yehuda Lindell, Ariel Nof, and Kazuma Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 805–817.
- [26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [27] Yehya Abouelnaga, Ola S. Ali, Hager Rady, and Mohamed Moustafa, "CIFAR-10: knn-based ensemble of classifiers," *CoRR*, vol. abs/1611.04905, 2016.
- [28] Daniel S Kermay, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al., "Identifying medical diagnoses and treatable diseases by image-based deep learning," *cell*, vol. 172, no. 5, pp. 1122–1131, 2018.
- [29] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Chang Peng, and H Eugene Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
- [30] Youren Shen, Hongliang Tian, Yu Chen, Kang Chen, Runji Wang, Yi Xu, Yubin Xia, and Shoumeng Yan, "Occlum: Secure and efficient multitasking inside a single enclave of intel SGX," in *ASPLOS '20: Architectural Support for Programming Languages and Operating Systems, Lausanne, Switzerland, March 16-20, 2020*, James R. Larus, Luis Ceze, and Karin Strauss, Eds. 2020, pp. 955–970, ACM.