PPBR: Privacy-Preserving and Byzantine-Robust Edge-Assisted Hierarchical Federated Learning in Mobile Networks

Yuanyuan He, *Member, IEEE*, Jie Zhang, *Member, IEEE*, Peng Yang, *Member, IEEE*, Zhe Sun, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Edge-assisted Hierarchical Federated Learning (EHFL) accelerates global model training across mobile devices by hierarchically aggregating models. However, EHFL encounters critical challenges such as privacy risks for local and edge-level models, vulnerability to collusive Byzantine attacks, and issues with model diversity and heterogeneity due to Non-Independent and Identically Distributed (Non-IID) data. In this paper, we propose PPBR, a novel hybrid scheme that subtly integrates Condensed Local Differential Privacy (CLDP) and Packed Linearly Homomorphic Encryption (PLHE) to achieve strong privacy protection and resilience against various Byzantine attacks in Non-IID data scenarios. Specifically, PPBR clusters the sign statistics of local models and clips the norms of edge-level momenta to filter anomalous models and mitigate Byzantine faults while retaining diverse models coming from Non-IID data. To enhance privacy protection with acceptable accuracy loss, the sign tuples of local models are perturbed with CLDP guarantees, and the momenta of edge-level models are encrypted under PLHE. Meanwhile, PPBR enhances privacy in single-edge-server and single-cloud-server aggregations by using random perturbations, secret sharing, and PLHE. In addition to safeguarding privacy with accommodating abrupt dropouts of mobile devices and edge servers, the aggregations effectively mitigate the adverse effects of Non-IID data under advanced Byzantine attacks. Theoretical analysis and comprehensive experiments validate PPBR's strong privacy guarantees and resilience to various Byzantine attacks under Non-IID data.**

***Index Terms*—EHFL, privacy preservation, Byzantine resilience, Non-IID.**

Received 11 February 2025; revised 13 July 2025; accepted 19 August 2025. Date of publication 26 August 2025; date of current version 9 January 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62472186, in part by Hubei Provincial Natural Science Foundation of China under Grant 2024AFD413, in part by the Natural Science Foundation of Wuhan under Grant 2024040801020213, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011492. Recommended for acceptance by S. Wang. *(Corresponding author: Peng Yang.)*

Yuanyuan He and Jie Zhang are with the Hubei Key Laboratory of Distributed System Security, Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yuanyuan_cse@hust.edu.cn; zhangjie001@hust.edu.cn).

Peng Yang is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yangpeng@hust.edu.cn).

Zhe Sun is with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China (e-mail: sunzhe@gzhu.edu.cn).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/TMC.2025.3602911

I. Introduction

As VARIOUS mobile devices explosively increase, a massive amount of diverse and personal data is generated and aggregated per second at the network edge. These data power a multitude of Machine Learning (ML) applications and services, such as object detection [1], speech recognition [2], disease diagnosis [3], and recommendation systems [4], raising widespread privacy concerns regarding the potential leakage of raw mobile device data [5] to remote cloud servers during the ML model training process.

To protect sensitive raw data, Federated Learning (FL) has been widely adopted because it enables distributed training of a global ML model [6]. In each round, mobile devices, acting as clients, use their local datasets to refine the models and upload the local updates to the cloud server. The cloud server then updates the global model by aggregating the received updates, without directly accessing the sensitive raw data of clients. The communication traffic between the mobile devices and cloud server increases significantly with the proliferation of devices. To alleviate the load on the backhaul network bandwidth, researchers introduced mobile edge computing and proposed Edge-assisted Hierarchical FL (EHFL) [7] as shown in Fig. 1. In EHFL, edge servers periodically aggregate clients' updated local models to obtain intermediate (edge-level) models, known as edge aggregation [7]. The cloud server then aggregates the edge-level models to produce the global model, known as global aggregation. Thus, EHFL effectively accelerates FL across mobile devices by reducing the need for data transfers from edge servers to the remote cloud.

Nevertheless, EHFL still faces the following challenges: (1) Mobile devices (clients) can be easily corrupted to perform Byzantine attacks by disobeying the protocols or sending arbitrary messages, which disturb the model aggregations [8]. In EHFL, Byzantine clients are able to collude and optimize their attacks across multiple rounds to effectively conceal themselves within the edge aggregations, which aggregate diverse local models derived from Non-Independent and Identically Distributed (Non-IID) data across various mobile devices. (2) Edge servers can accurately infer the distribution of clients'

1536-1233 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

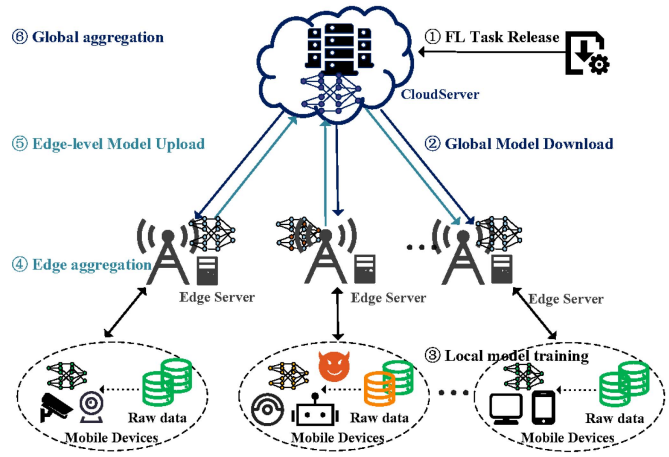


Fig. 1. An illustration of the EHFL architecture.

training data [9] based on the observed local models. Similarly, the cloud server is able to reconstruct the information associated with the private training data of clients within the service domain of an edge server.

Existing Byzantine-robust solutions primarily rely on analyzing model gradients [10] to detect and filter out Byzantine clients, or using auxiliary labeled datasets [11], [12] to verify the correctness of received model parameters to identify Byzantine clients. In a realistic EHFL system, mobile devices usually differ, resulting in Non-IID training data [13], [14]. Non-IID data aggravates the divergence between local models from different devices [13], blurring the distinction between Byzantine clients and benign ones. Many strategies further introduce bucketing [15], gradient splitting [16], clustering [17], gradient history analyzing [18], and data redundancy [19] techniques to resist Byzantine attacks in certain Non-IID scenarios. However, the solutions for both IID and Non-IID cases mostly rely on the premise that edge and cloud servers can access true local and edge-level models in hierarchical aggregations. It offers only limited privacy guarantees, since edge and cloud servers can use the observed models to launch gradient inversion attacks, recovering sensitive training data of mobile devices.

Existing privacy-preserving methods for EHFL primarily rely on differential privacy (DP) [20], [21], [22], [23] and cryptographic techniques [6], [24]. Their core idea is to prevent edge and cloud servers from obtaining precise updates of the local models. This is obviously opposite to the essential prerequisite of the Byzantine-robust EHFL. It is nontrivial to address this issue, since the DP noises randomly disturb the local models of Byzantine and benign clients, further blurring the boundary between them [25] which is already obfuscated by Non-IID data; and the used cryptographic primitives only support basic operations (e.g., averaging) on encrypted local models with the acceptable computing cost [26], making it impractical to perform much more complex and time-consuming operations (e.g., comparison, sorting, similarity calculation and filtering [27]) over ciphertexts that are necessary in the most Byzantine-robust methods.

To address this challenge, many works have been proposed for FL, e.g., LSFL [28], SEAR [26], DisBezant [29], APFed [30], ShieldFL [14], providing the robustness to specific Byzantine attacks. These solutions for FL can be straightforwardly extended to EHFL by executing themselves twice in the hierarchical aggregations, including the edge aggregation and global aggregation processes. However, some critical issues arise: (1) Byzantine clients in EHFL are capable of colluding and strategically optimizing their attacks over multiple rounds. These advanced Byzantine models are difficultly detected from the distance and cosine-similarity perspectives in most existing solutions. (2) The FL-based extensions usually presume prior knowledge of edge-level models to identify the malicious ones contaminated by Byzantine attacks during the global aggregation. Different from in FL, edge-level models in EHFL require heightened protection against edge servers, since an edge server could use its edge-level model to infer sensitive information related to the private training data of its clients.

In this paper, we propose PPBR, a novel hybrid scheme for privacy-preserving and Byzantine-robust EHFL on Non-IID data. PPBR consists of a privacy-enhanced dual-layer anomalous models filter based on the sign-statistics clustering and the norms clipping, and privacy-enhanced hierarchical model aggregations. During the filtering process to address issue (1), the sign tuples of local models are perturbed with Condensed Local Differential Privacy (CLDP) [31] guarantees before clustering, and the norms of momenta of edge-level models are screened and clipped under Packed Linearly Homomorphic Encryption (PLHE) [32] to mitigate advanced Byzantine faults with strong privacy protection and acceptable accuracy loss. During the hierarchical aggregations to address issue (2), the local and edge-level models are protected using random perturbations and PLHE. In PPBR, CLDP reduces the overall noise impact on clustering results and ensures a generalized Local DP (LDP) by perturbing a condensed representation, such as a distance function, of the sign tuples rather than each dimension individually. PLHE encrypts multiple values into a single ciphertext and supports performing linear operations (such as addition and scalar multiplication) on these ciphertexts, enabling privacy enhancement during aggregations as well as norms screening and clipping. PPBR has the following distinguished features:

- The dual-layer filter mitigates various potential Byzantine faults under the privacy protection. The average-based hierarchical aggregations 'mix' the remaining momenta of local models under encryption. Both fully consider the benign but diverse models coming from Non-IID training data. They enable PPBR to achieve high accuracy in EHFL on Non-IID data under advanced Byzantine attacks, while ensuring that the privacy protection of both local and edge-level models is not compromised.
- During filtering out malicious mobile devices, the sign statistics of local models are clustered with CLDP guarantees to ensure privacy while maintaining strong robustness without additional communication costs. To further enhance the robustness and privacy, the cloud server and edge servers filter

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

out anomalous edge-level momenta by performing norm screening and clipping operations under the PLHE, without accessing the true momenta.

• PPBR protects local models by using secret sharing and random perturbations to generate noisy edge-level momenta during single-edge-server aggregation. It then aggregates the clipped versions of these noisy momenta and removes all noise to produce the global model, protected by secret sharing and PLHE with a single cloud server. Thus, PPBR shields the sensitive data of mobile devices from collusive inference attacks during aggregation, while handling the sudden dropouts of mobile devices and edge servers.

The remainder of this paper is organized as follows. We overview the related works in Section II, followed by the problem statement in Section III. Then, we propose PPBR in Section IV and analyze its privacy in Section V. Finally, we evaluate its performance in Section VI and draw the conclusion in Section VII.

II. Related Works

EHFL [7] integrates mobile edge computing techniques to accelerate FL across mobile devices by hierarchically aggregating local models and edge-level models. Nevertheless, EHFL faces security and privacy challenges coming from Byzantine mobile devices (clients) and local models sharing. Byzantine threats are considered as the worst case among various data and model update poisoning attacks which degrade or break the EHFL model training phase, since Byzantine clients can generate arbitrary outputs and collude to amplify their detrimental effects [35]. Besides, edge and cloud servers are able to recover the distribution of clients' sensitive training data based on the observed local models [9] and edge-level models by gradient inversion attacks. The true labels of the training data can even be inferred by analyzing the gradient signs of local models [37].

Current Byzantine-robust solutions for FL and EHFL mostly use statistics-based robust aggregation rules to estimate model parameters, such as the median and trimmed mean [38], Krum [39], Geomed [40], DnC [36], Fedinv [41], norm clipping [18], [33], or their combinations. Their efficacy is based on majority-vote strategy, assuming that the proportion of Byzantine clients is less than 50%. To resist the proportion of Byzantine clients exceeding 50%, edge or cloud servers are assumed to have auxiliary labeled datasets [11], [12] to verify the benign nature of received model parameters in many solutions. Servers usually struggle to obtain additional benign data that sufficiently mimics the global data distribution in practice. Meanwhile, they mostly exhibit a substantial decline in the accuracy when the training data is Non-IID, since Non-IID data amplifies the update differences between benign clients and weakens the server's ability to distinguish Byzantine clients. To address this issue, some works introduce bucketing [15], gradient splitting [16], clustering [17], gradient history analyzing [18], and data redundancy [19] techniques into Byzantine-robust aggregation [41], improving the robustness to some Byzantine attacks in few Non-IID data settings. However, the edge servers require access to the clients' local models, and the cloud server observes edge-level models in nearly all of these methods, providing a weak privacy guarantee.

To enhance privacy preservation, existing schemes mainly rely on cryptography and DP. [6] utilizes double masking and secret sharing to achieve secure aggregation across devices in FL. Inspired by this, [42] improves the efficiency by limiting the number of neighboring nodes. Numerous studies introduce Homomorphic Encryption (HE) into FL in cross-silo scenarios, resulting in significant computational and communication overhead. To improve the efficiency, many researchers have further combined quantization [12] and packed encryption techniques [43], [44]. For example, PLHE [32] can pack multiple messages into a single ciphertext and supports linearly homomorphic operations such as addition and scalar multiplication on ciphertexts. Meanwhile, LDP noise is generated and added into the parameters of local models to protect clients' private data [20], [21], and the shuffled DP model [45] is further used to improve the accuracy. The underlying principle of these encryption-based and DP-based methods is to prevent edge servers from accessing the true updates of the local models in EHFL. However, this directly conflicts with the prerequisite of Byzantine-robust methods for FL and EHFL on Non-IID data. Hence, it is nontrivial to ensure both privacy and Byzantine robustness in FL and EHFL, particularly with Non-IID data [25].

Many researchers are exploring solutions for FL to mitigate the technical challenges inherent in this contradiction, as shown in Table I. LSFL [28] uses secret sharing and employs non-colluding dual servers to achieve Byzantine robustness and privacy-preservation in FL. SEAR [26] relies on a trusted execution environment to significantly improve efficiency. To support FL on Non-IID data, many works [14], [30] perform the Byzantine-robust aggregation based on cosine similarity under fully HE, which is not well-suited to resource-constrained mobile devices. DisBezant [29] is proposed as a suitable solution for mobile devices, leveraging trustworthiness and secure two-party computation. Although they are effective against specific Byzantine attacks, including Gaussian attacks [34], sign flipping attacks [33], and label flipping attacks [35], as well as typical targeted and untargeted attacks, they usually struggle to identify Byzantine clients that have colluded and optimized their attacks based on collected messages across multiple rounds. Notably, malicious gradients from these advanced Byzantine clients such as ALIE attackers [36] and MinMax attackers [36] are difficultly detected from the distance and cosine-similarity perspectives. Moreover, compared to FL, the existing literature is sparse regarding privacy-preserving and Byzantine-robust EHFL. To extend the schemes from FL to EHFL, a straightforward approach is to execute their aggregation algorithms in both the edge aggregation and global aggregation processes. However, it introduces a critical issue: edge and cloud servers must identify edge-level models contaminated by Byzantine attacks without prior knowledge of the edge-level models, which is essential for the privacy-preservation of EHFL.

Therefore, it remains a significant technical challenge in EHFL on Non-IID data to protect both local models and edge-level models related to mobile devices' sensitive data, while

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON BETWEEN PPBR AND PREVIOUS WORKS

Schemes	Techniques	Privacy Guarantee	Server	Byzantine Robustness	Non-IID Robustness
LSFL [28]	Secret sharing, k-nearest weight selection	✓	Dual	Sign flipping attack [33]	✓
SEAR [26]	Trusted execution environment (Intel SGX), coordinate-wise median	✓	Single	Gaussian attack [34]	✓, weak
DisBezant [29]	Random perturbation, credibility measurement	✓	Dual	Gaussian attack [34]	✓
APFed [30]	Linear Homomorphic Encryption (LHE), cross-cluster-similarity-based filtering	✓	Single	Label flipping attack [35], sign flipping attack [33]	✓
ShieldFL [14]	Two-trapdoor HE, cosine-similarity-based filtering	✓	Single	Label flipping attack [35]	✓
PPBR	CLDP, PLHE, sign-statistics clustering, clipping	✓	Single	Label flipping attack [35], sign flipping attack [33], Gaussian attack [34], ALIE attack [36], IPM attack [10], and MinMax attack [36]	✓

enhancing robustness to advanced Byzantine attacks. To address this challenge, we propose PPBR, a novel hybrid solution based on CLDP and PLHE.

III. PROBLEM STATEMENT

A. System Model

The EHFL system consists of moving clients (mobile devices, e.g., smart vehicles, unpiloted aerial vehicles, and portable medical terminals), edge servers (which are installed at road side units, micro base stations, and macro base stations) near the clients, and the remote cloud server. The cloud server A collects messages from the edge servers $\{A_i\}_{i=1}^{l}$. The randomly selected clients in the edge server A_i's coverage area are denoted by $U_i = \{u_{i,j}\}_{j=1}^{n_i}$ with $|U_i| = n_i$. The entities collaboratively train a shared global ML model in EHFL for tasks such as content caching, data sharing, and collaborative analysis. The tasks can be expressed mathematically as the following distributed optimization problem: $\min_{\vec{w}\in\mathbb{W}^m} = \frac{1}{n}\sum_{(i,j)=(1,1)}^{(l,n_i)} \mathbb{E}_{\xi_{i,j}\sim\mathcal{D}_{i,j}}[\ell(\vec{w},\xi_{i,j})]$, where $n = \sum_{i=1}^{l} n_i$ is the total number of clients, ℓ is the local loss function over the training data sampled from $u_{i,j}$'s local dataset $\mathcal{D}_{i,j}$ and the given global model parameters $\vec{w}$. Here, the distribution of $\mathcal{D}_{i,j}$ may differ from that of $\mathcal{D}_{i',j'}$ in Non-IID data setting.

A typical EHFL scheme comprises the following steps:

1) *FL Task Release:* Upon receiving a FL task, the cloud server A initiates the global model $\vec{w}_a^{(0)}$.
2) *Global Model Download:* The edge server A_i downloads the global model $\vec{w}_a^{(r)} \in \mathbb{W}^m$ from the cloud server and distributes it to the client $u_{i,j}$ within its range during iteration round r.
3) *Local model training:* Every client $u_{i,j}$ performs local Stochastic Gradient Descent (SGD) method with respect to the global model and updates the local model: $\vec{w}_{i,j}^{(r+1)} = \vec{w}_a^{(r)} - \eta \bigtriangledown \ell\left(\vec{w}_a^{(r)}, \mathcal{D}_{i,j}\right)$, where η is the local learning rate.
4) *Edge aggregation:* The masked or noisy versions of the updated local models are periodically sent to the nearby edge server A_i for the protection of sensitive local models. A_i aggregates the received updates from clients in U_i.
5) *Edge-level Model Upload:* Each A_i sends its edge-level aggregation result $\vec{w}_i^{(r+1)}$ to the cloud server A.
6) *Global aggregation:* A updates the global model by aggregating the collected edge-level aggregation results, and gets $\vec{w}_a^{(r+1)} \in \mathbb{W}^m$ that will be sent to the selected clients through each A_i for the subsequent round.

Steps 2-6 repeat until the stop condition is satisfied, in accordance with the basic workflow of PPBR.

B. Threat Model

In our threat model, the sensitive information encompasses the training data and local model updates of each client, and the edge-level models which are related to local models in every training round of EHFL.

The cloud server, most edge servers and most clients are semi-honest [6]: they follow the protocol honestly and try to infer private information about some targeted clients (e.g., the distribution of the clients' training data [9], [26]) based on the received messages during the aggregation, but do not exhibit active malicious behaviors.

Some clients and edge servers are Byzantine attackers, who deviate from the protocol to corrupt the federated model by sending out arbitrary updates in all rounds. The fraction of Byzantine clients (edge servers) is set to be $\beta < 0.5$. Furthermore, advanced Byzantine clients (edge servers) have the ability to collude together to maximize their detrimental impact on the aggregation results and the coverage process. For instance, they may substitute the corresponding dimensions of their updated models with similar values.

The semi-honest clients, Byzantine clients, semi-honest edge servers, or Byzantine edge servers may be collusive inference attackers, who collude to infer private information about other clients. Without loss of generality, assume that there are t_i' corrupt parties in edge server A_i's coverage area, and there are t' corrupt edge servers in any round.

An arbitrary fraction of semi-honest clients and Byzantine clients may drop out during the protocol execution in EHFL due to the mobility of clients (i.e., mobile devices).

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

C. Design Goal

PPBR aims to protect clients' privacy and mitigate Byzantine failures in EHFL on Non-IID data in mobile networks.

Privacy preservation: Any party (i.e., client, edge server or the cloud server) inside the EHFL system cannot learn and infer the targeted clients' real parameters of local models based on the received messages during the execution of our PPBR protocol. Any coalition described in our threat model cannot infer information about the sensitive local models of other clients. Meanwhile, PPBR protocol should support client dropouts in the mobile network.

Byzantine resilience: PPBR should achieve robustness against Byzantine clients described in our threat model in Non-IID data setting with high accuracy. Various Byzantine attacks are considered as follows.

• Label flipping attack [18], [35]. Each Byzantine client alters the training data by flipping the label of each sample.

• Sign flipping attack [33]. Each Byzantine client strives to maximize the loss through gradient ascent instead of the normal gradient descent approach. Specifically, the sign of the gradient is flipped during the local updating step.

• Gaussian attack [35]. Each Byzantine client generates a random noise using a distribution like Gaussian and submits it as the local model update.

• ALIE attack [36]. To introduce undetectable perturbations, each Byzantine attacker capitalizes on the variance observed in benign updates and injects noise within a predefined range. For every coordinate $i \in [d]$, the attackers calculate the mean μ_i and standard deviation σ_i based on benign updates. Subsequently, they collusively assign malicious updates with values falling within the interval $(\mu_i - z^{max}\sigma_i, \mu_i + z^{max}\sigma_i)$, where z varies from 0 to 1 and is typically obtained from the Cumulative Standard Normal Function. The ith malicious update is $\Delta^t_{k,i} \leftarrow \mu_i - z^{max}\mu_i$.

• IPM attack [10]. Each Byzantine client strives to identify the negative inner product between the true mean of updates and the aggregation result to prevent loss from descending. If $\beta_i n$ Byzantine clients in edge server A_i's coverage area know the mean of benign updates, they perform an IPM attack by setting their gradients to be $\bigtriangledown\ell_1 = \bigtriangledown\ell_2 = \cdots = \bigtriangledown\ell_{\beta_i n} = -\frac{\xi}{n_i - \beta_i n}\sum_{j=\beta_i n+1}^{n_i} \bigtriangledown\ell_j$, where ξ serves as a positive factor that regulates the magnitude of the malicious updates.

• MinMax attack [36]. Each Byzantine client aims to position the malicious updates near the cluster of benign updates. MinMax attackers adjust the scaling of z^{max} to ensure that the maximum distance between malicious updates and any benign update is limited to the maximum distance observed between any two benign updates.

Non-IID resilience: PPBR should provide high accuracy without sacrificing its privacy protection and Byzantine resilience, even if the training data from distributed clients is Non-IID in EHFL.

D. Technical Primitives

Secure single-server aggregation: We use SA, a variant of the secure single-server aggregation protocol [6] based on Shamir Secret Sharing $Shamir(t, n, x)$ [46], κ-secure key agreement S [24] and PseudoRandom Generator PRG [46] to finish the edge/global aggregation with high efficiency [6], while supporting client/edge server dropouts in PPBR. Algorithm 1 presents the details of the SA: $\vec{m} \leftarrow SA(\{\vec{m}_i\}_{C_i \in U}, t)$.

Algorithm 1: $\vec{m} \leftarrow SA\left(\{\vec{m}_i\}_{C_i \in U}, t\right)$.

Input: Each client C_i's parameters corresponding to local model $\{\vec{m}_i \in \mathbb{W}^m\}_{C_i \in U}$, and the Shamir threshold t

Output: The aggregated result $\vec{m}$

Each client C_i:

1: Generate key pair $(sk_i; pk_i)$, sample b_i, and obtain $pk_{i'}$ of client $u_{i'}$ through the aggregation server;

2: $\vec{v}_{i,i'} \leftarrow PRG(S(sk_i; pk_{i'}))$, $\vec{x}_i \leftarrow PRG(b_i)$, $\{\vec{h}_{i'}\}_{u_{i'} \in U_i, u_{i'} \neq u_i} \leftarrow Shamir(t, n, \vec{x}_i)$;

3: Send $E_{pk_{i'}}(\vec{h}_{i,i'})$ to each $C_{i'}$ and obtain $C_{i'}$'s $\vec{h}_{i',i}$ through the aggregation server;

4: Mask the parameters by calculating $\vec{z}_i = \vec{m}_i + \vec{x}_i - \sum_{0<i'<i, C_{i'} \in U} \vec{v}_{i,i'} + \sum_{i<i', C_{i'} \in U} \vec{v}_{i',i}$;

5: Send $\vec{z}_i$ to the aggregation server;

The aggregation server:

6: Form a subset of dropped clients $U^{(d)}$ and a subset of surviving clients $U^{(s)}$;

7: **if** $|U(s)| < t$ **then**

8: Abort;

9: **else**

10: Send $U^{(d)}$ to each surviving client $C_i \in U^{(s)}$;

11: Cooperate with its surviving clients to recover $\vec{x}_i$ and obtain $\vec{\mathbf{V}}_i = \sum_{0<i'<i, C_{i'} \in U^{(d)}} \vec{v}_{i,i'}$, $\vec{\mathbf{V}'}_i = \sum_{i'<i, C_{i'} \in U^{(d)}} \vec{v}_{i,i'}$ for each $C_i \in U^{(s)}$;

12: Unmask and aggregate clients' momenta via $\vec{m} = \frac{\sum_{C_i \in U^{(s)}} (\vec{z}_i - \vec{x}_i + \vec{\mathbf{V}}_i - \vec{\mathbf{V}'}_i)}{|U^{(s)}|} = \frac{\sum_{C_i \in U^{(s)}} \vec{m}_i}{|U^{(s)}|}$;

13: **end**

1) *Initialize:* Each client C_i generates $\{(sk_i; pk_i), b_i\}$ and exchanges public key pk_i with others. C_i uses S and PRG to compute share $\overrightarrow{v}_{i,i'}$ locally for each $C_{i'} \in U$; and splits $\overrightarrow{x}_i \leftarrow PRG(b_i)$ into n shares with threshold t via Shamir secret sharing protocol; and sends the encrypted $\overrightarrow{h}_{i,i'}$ to $C_{i'}$ and obtains $\overrightarrow{h}_{i',i}$ through the aggregation server.
2) *Masking:* Each client C_i executes line 4 of Algorithm 1 to mask the parameters corresponding to her local model.
3) *Unmasking:* The aggregation server counts the received masked model parameters to form the set $U^{(s)}$, which consists of surviving clients. It aborts this round if $|U^{(s)}| < t$; and otherwise, it recovers $\overrightarrow{x}_i$ and obtains $\{\vec{\mathbf{V}}_i, \vec{\mathbf{V}'}_i\}$ related to the client dropouts for each surviving client to unmask the average of model parameters on lines 10-12, Algorithm 1.

Condensed LDP (CLDP): CLDP [31] is a generalization of traditional LDP [20], [21]. Unlike traditional LDP, CLDP additionally relies on a specific distance function $d(\vec{s}, \vec{s'})$ to

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

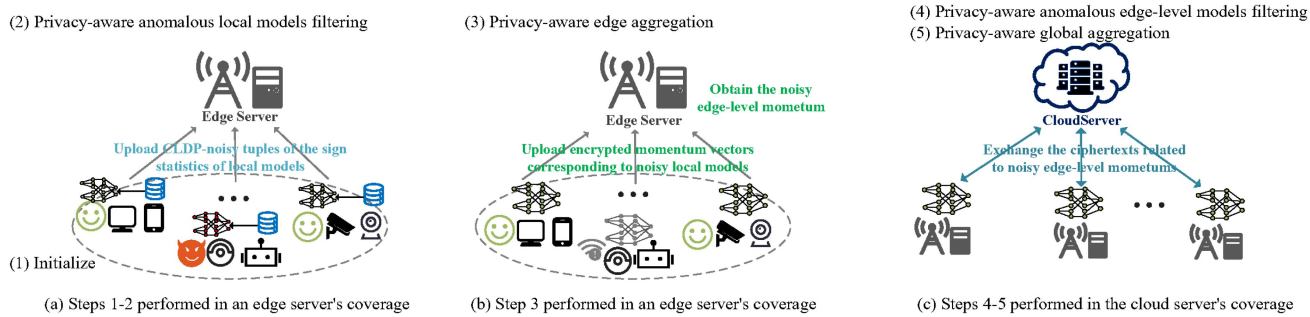


Fig. 2. Overview of PPBR in each round.

generalize the level of privacy leakage, incorporating the relationship between different inputs.

Definition 1: ϵ_χ-Condensed Local differential privacy (ϵ_χ-CLDP). Given set $\mathcal{X}$ containing all possible parameters of local models, a randomized algorithm $\mathcal{M}: \mathcal{X} \to \mathcal{R}$ follows the probability density function $f_{\vec{s}}: \mathcal{R} \to \mathbb{R}_{\geq 0}$ for $\forall \vec{s} \in \mathcal{X}$, where $\mathcal{R}$ denotes all possible outputs of algorithm $\mathcal{M}$. $\mathcal{M}$ achieves ϵ_χ-CLDP if and only if for any $\vec{s}, \vec{s'} \in \mathcal{X}$ and $\vec{y} \in \mathcal{R}$,

$$f(\vec{y}|\vec{s}) \leq e^{\epsilon_\chi d(\vec{s},\vec{s'})} f(\vec{y}|\vec{s'}), \tag{1}$$

where $d(\vec{s}, \vec{s'})$ is a distance function.

We set $d(\vec{s}, \vec{s'}) = \|\vec{s} - \vec{s'}\|_2$ in PPBR for EHFL.

PLHE: A PLHE [43], [44] enables the packing of multiple messages into a single ciphertext and supports component-wise homomorphic computation in a Single Instruction Multiple Data (SIMD) manner [47] to take advantage of parallelism. PLHE$=(KeyGen; Enc; Dec; Eval)$:

1) $KeyGen(1^k) \to (pk, sk)$;
2) $Enc(pk, \vec{m}) \to [\vec{m}], \vec{m} \in \mathbb{Z}^t$;
3) $Dec(sk, [\vec{m}]) \to \vec{m}$;
4) Evaluation algorithm

$Eval(pk, [\vec{m}_1], [\vec{m}_2], \ldots, f) \to [f(\vec{m}_1], [\vec{m}_2)]$, where f can be constructed using homomorphic addition (Add) and multiplication ($Mult$). For ciphertexts $[\vec{m}_1]$ and $[\vec{m}_2]$, $Add([\vec{m}_1], [\vec{m}_2])$ outputs a ciphertext $[\vec{m}_1 + \vec{m}_2]$, which encrypts the elementwise sum of $\vec{m}_1$ and $\vec{m}_2$, and $Mult([\vec{m}_1], [\vec{m}_2])$ outputs a ciphertext $[\vec{m}_1 \cdot \vec{m}_2]$; and for ciphertext $[\vec{m}]$ and plaintext $\vec{s}$, $Mult([\vec{m}], \vec{s})$ outputs $[\vec{m} \cdot \vec{s}]$.

We only require scalar multiplication between a ciphertext and a plaintext in PPBR. In PPBR, the Brakerski-Fan-Vercauteren (BFV) scheme [32] is used as a PLHE primitive. Compared to traditional HE schemes, it employs SIMD-based batch processing and optimized polynomial operations [32], [47], which effectively reduce computational complexity and communication overhead, thereby significantly enhancing overall efficiency.

IV. Design of PPBR

A. Overview

As shown in Fig. 2, PPBR protocol consists of 5 steps.

1) Key public parameters are initialized.
2) Clients in each edge server A_i's coverage area construct sign statistics of the extracted coordinates of local models and perturb them with CLDP guarantees, which maintain high accuracy in the trimmed k-means clustering of the noisy sign statistics. Each A_i filters out anomalous local models based on the privacy-preserving clustering result.
3) Each A_i executes the SA protocol with the noisy momenta of local models from its remaining trusted clients to securely calculate their average momenta as the edge-level aggregation results with supporting client dropouts.
4) The cloud server A filters out outliers by screening the norms of the remaining edge-level aggregation momenta without knowing the true momenta due to PLHE. The outliers are detected as Byzantine behaviors.
5) A clips the norms of the remaining noisy edge-level momenta under PLHE, then securely aggregates them by performing SA, subsequently removes all random noises in the aggregation results to obtain the true mean momentum, and finally recovers the corresponding global model.

In PPBR, the anomalous local and edge models filtering, which are based on the sign-statistics clustering and the norms screening and clipping, as well as dual averaging aggregations in steps 2-5 are designed to offer strong robustness against Byzantine attacks in Non-IID data settings; and the CLDP perturbation in step 2(b), and the SA and PLHE operations in steps 3-5 are carefully crafted to enhance privacy protection without sacrificing the robustness.

Table II presents the key notation used in PPBR.

B. Detailed PPBR

The design of PPBR aims to securely eliminate anomalous models and perform hierarchical model aggregation in each round r of the EHFL system.

1) Initialize: The model aggregation begins with the inputs of the cloud server, the selected edge servers, and its selected clients. Specifically, the cloud server A sets and broadcasts the privacy budget ϵ_χ, parameter λ, distance thresholds L and L', threshold t related to edge server dropouts in global aggregation, parameters for computing momenta φ and ζ, and a parameter for screening norms θ. A stores the global model update from the last round $\vec{w}_a^{(r)}$. Each edge server A_i selects the clients in its coverage to form set U_i and sets threshold t_i related to client

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

TABLE II
NOTATIONS

Notations	Descriptions
A	The cloud server
$\{A_i\}_{i=1}^{l}$	The set of edge servers
η	The learning rate for local training
r	The r-th training round
$U_i = \{u_{i,j}\}_{j=1}^{n_i}$	The clients selected by A_i in its coverage area
n	The total number of clients, i.e., $n = \sum_{i=1}^{l} n_i$
$\mathcal{D}_{i,j}$	$u_{i,j}$'s local dataset
∇, ℓ	The gradient operator, and the loss function
β	The proportion of Byzantine clients
$\vec{w}_a^r$, $\vec{m}_a^r$	The global model update in the round $r-1$ and the corresponding momentum
$\vec{w}_i^{(r+1)}$, $\vec{m}_i^{(r+1)}$	The edge-level model update in the round r and the corresponding momentum
$\vec{w}_{i,j}^{(r+1)}$, $\vec{m}_{i,j}^{(r+1)}$	$u_{i,j}$'s local model update in the round r and the corresponding momentum
$\vec{m'}_{i,j}^{(r+1)}$, $\vec{m'}_i^{(r+1)}$	The noisy momenta
$U_i^{(T)}$	The set of remaining clients after step 2
$\mathcal{A}^{(T)}$	The set of remaining edge servers after step 4
t_i/t	The threshold for the number of online clients/edge servers during the edge/global aggregation
t_i'/t'	The number of colluding clients during A_i's/A's edge/global aggregation
ϵ_χ	The privacy budget used in CLDP-based clustering
λ, L, L'	The parameter λ, and distance thresholds L and L' for CLDP-based clustering
$(sk_{i,j}, pk_{i,j})$	$u_{i,j}$'s private-and-public-key pair used in SA
M	The median of all received $\vec{m}_i^{(r+1)}$ before screening norms in the round r
$\vec{m''}_a^{(r+1)}$, $\vec{m}_a^{(r+1)}$	The average of the noisy momenta and the real average during the global aggregation in the round r
$\vec{s}_{i,j}^{(r+1)}$	$u_{i,j}$'s tuple of sign statistics
$\vec{y}_{i,j}^{(r+1)}$	The noisy tuple of $\vec{s}_{i,j}^{(r+1)}$
$\vec{y}_r^i$	The masked vector of C_i's local model
$\mathcal{C}_i$	The clustering result obtained by A_i
φ, ζ, θ	The parameters for transforming a model update to the corresponding momentum and screening norms

dropouts in its edge aggregation. Each client $u_{i,j}$ inputs her local model $\vec{w}_{i,j}^{(r+1)} \in \mathbb{W}^m$. Meanwhile, A selects model coordinates to form subset G, where $|G| = \iota$, and broadcasts $\vec{w}_a^{(r)}$, G, and its public key pk_A in PLHE to each edge server A_i and its selected clients in U_i.

2) Privacy-Aware Anomalous Local Models Filtering: Each edge server A_i clusters the sign statistics of the selected clients with CLDP guarantees by executing Algorithm 2.

a) *Generating sign vectors:* Each client $u_{i,j}$ generates its sign-statistics tuple as features based on the random coordinates extracted by A on line 1 of Algorithm 2. The sign-statistics tuple consists of the proportions of positive, negative, and zero signs [17], with a dimensionality of 3.

b) *Perturbing with CLDP:* Each $u_{i,j}$ perturbs the sign tuple $\vec{s}_{i,j}^{(r+1)}$ on line 2 to obtain $\vec{y}_{i,j}^{(r+1)}$ with the distribution density function in the d-dimensional space $\mathcal{S}_{L'} = \{\vec{y}_{i,j}^{(r+1)} | \|\vec{y}_{i,j}^{(r+1)} - \vec{s}_{i,j}^{(r+1)}\|_\infty \le L'\}$ in (2), where L and L' are pre-set distance thresholds, and $\mathcal{B}_L = \{\vec{y}_{i,j}^{(r+1)} | \|\vec{y}_{i,j}^{(r+1)} - \vec{s}_{i,j}^{(r+1)}\|_2 \le L\}$.

$$f\left(\vec{y}_{i,j}^{(r+1)}\right) | \vec{s}_{i,j}^{(r+1)}\Big) = \lambda e^{-\epsilon_\chi \min\left\{\|\vec{y}_{i,j}^{(r+1)} - \vec{s}_{i,j}^{(r+1)}\|_2, L\right\}}$$

Algorithm 2: CLDP-Based Clustering, *CC*.

Input: Set U_i of the selected clients in each edge server A_i's coverage, local models $\vec{w}_{i,j}^{(r+1)} \in \mathbb{W}^m$ of all clients $u_{i,j} \in U_i$, privacy budget ϵ_χ, parameter λ, and distance thresholds L and L', and set G of selected model coordinates;
Output: Clusters $\mathcal{C}_i = \{C_1, C_2, \ldots, C_k\}$;
Each client $u_{i,j} \in U_i$**:**
1: Compute sign tuple $\vec{s}_{i,j}^{(r+1)}$ based on the sign statistics on selected coordinates in G, $|\vec{s}_{i,j}^{(r+1)}| = d = 3$;
2: Sample $\vec{y}_{i,j}^{(r+1)}$ from the distribution density function in (2) within the d-dimensional space $\mathcal{S}_{L'}$;
3: Send $\vec{y}_{i,j}^{(r+1)}$ to edge server A_i;
Edge server A_i**:**
4: Perform trimmed k-means clustering with received $\{\vec{y}_{i,j}^{(r+1)}\}_{j=1}^{n_i}$ and obtain clusters $\mathcal{C}_i = \{C_1, C_2, \ldots, C_k\}$, each of which is the corresponding client set.

$$= \begin{cases} \lambda e^{-\epsilon_\chi \left\|\vec{y}_{i,j}^{(r+1)} - \vec{s}_{i,j}^{(r+1)}\right\|_2}, & \vec{y}_{i,j}^{(r+1)} \in \mathcal{B}_L; \\ \lambda e^{-\epsilon_\chi L}, & \vec{y}_{i,j}^{(r+1)} \in \mathcal{S}_{L'} - \mathcal{B}_{L'}; \\ 0, & \vec{y}_{i,j}^{(r+1)} \notin \mathcal{S}_{L'}. \end{cases} \quad (2)$$

λ is a constant such that $\int \cdots \int_{\mathcal{S}_{L'}} f\left(\vec{y}_{i,j}^{(r+1)}\right) d\vec{y}_{i,j}^{(r+1)} = 1$, i.e.,

$$\lambda = \left(B_L + e^{-\epsilon_\chi L}\left(2^d L'^d - V_L\right)\right)^{-1} \quad [48],$$

where $V_L = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})+1} L^d$ denotes the volume of $\mathcal{B}_L$, and $B_L = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})+1} L^d e^{\epsilon_\chi(c_L - L)}$ denotes the integral of $f\left(\vec{y}_{i,j}^{(r+1)}\right)$ in $\mathcal{B}_L$, $c_L \in (0, L)$. Eq. (2) can be proven to satisfy CLDP by perturbing the entire vector $\vec{s}_{i,j}^{(r+1)}$ as a whole, rather than perturbing each dimension independently. This approach avoids the need to divide the privacy budget between different dimensions. Additionally, (2) uses bounded perturbation, which helps maintain high clustering accuracy. Client $u_{i,j}$ reports the noisy sign tuple $\vec{y}_{i,j}^{(r+1)}$ to A_i on line 3.

c) *Trimmed k-means clustering:* Edge server A_i performs a trimmed k-means clustering method [49] on the noisy sign tuples to obtain the set of clusters on line 4, Algorithm 2, i.e., $\mathcal{C}_i = \{C_1, C_2, \ldots, C_k\}$.

Each A_i can filter out anomalous local models based on the output $\mathcal{C}_i$ of Algorithm 2. A_i initializes $U_i^{(T)} = \emptyset$ and chooses the clusters having top-b sizes from $\mathcal{C}_i$ to build the trusted clients set $U_i^{(T)}$ with size $|U_i^{(T)}| = n_i'$, where the b is pre-set by A_i.

In this filtering, sign-statistics tuples serve as lightweight feature representations that preserve gradient directionality. In practice, benign clients often produce gradients with similar directions and are thus more likely to be grouped into larger clusters. Conversely, malicious or anomalous clients typically yield gradients with divergent or adversarial directions, making their sign tuples less likely to belong to large, cohesive clusters.

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3: Privacy-Aware Edge Aggregation, *EA*.

Input: Edge server A_i's public key pk_{A_i} and the cloud server A's public key pk_A in PLHE; Set $U_i^{(T)}$ of the remaining clients in A_i's coverage; Local models $\vec{w}_{i,j}^{(r+1)} \in \mathbb{W}^m$ of all clients $u_{i,j} \in U_i^{(T)}$; The threshold $\{t_i\}$ related to client dropouts in A_i's edge aggregation, parameters for computing momenta φ;

Output: Noisy edge-level momentum $\vec{m'}_i^{(r+1)}$ and ciphertexts $\left\{\left[2\vec{m'}_i^{(r+1)}\right]_{A_i}, \left[||\vec{m'}_i^{(r+1)}||_2^2\right]_{A_i}, \left[\vec{x}_i^{(r+1)}\right]_A\right\}$;

1: Each $u_{i,j} \in U_i^{(T)}$ generates a secret random vector $\vec{x}_{i,j}^{(r+1)} \in \mathbb{Z}^m$ and calculates the momentum $\vec{m}_{i,j}^{(r+1)} = \varphi\vec{m}_{i,j}^{(r)} + (1-\varphi)\vec{w}_{i,j}^{(r+1)}$ and its noisy vector $\vec{m'}_{i,j}^{(r+1)} = \vec{m}_{i,j}^{(r+1)} - \vec{x}_{i,j}^{(r+1)}$; and uses pk_A to calculate $[\vec{x}_{i,j}^{(r+1)}]_A$ and sends it to A_i;

2: A_i works as a bucket to securely average the perturbed momenta of its clients with supporting client dropouts, $\vec{m'}_i^{(r+1)} \leftarrow SA\left(\left\{\vec{m'}_{i,j}^{(r+1)}\right\}_{u_{i,j}\in U(T)_i}, t_i\right)$;

3: A_i calculates $\left[\vec{x}_i^{(r+1)}\right]_A \leftarrow Add\left(\left\{[\vec{x}_{i,j}^{(r+1)}]_A | u_{i,j} \in U_i^{(T)}\right\}\right)$ and sends $\left\{\left[2\vec{m'}_i^{(r+1)}\right]_{A_i}, \left[||\vec{m'}_i^{(r+1)}||_2^2\right]_{A_i}, \left[\vec{x}_i^{(r+1)}\right]_A\right\}$ to A.

Algorithm 4: Privacy-Aware Anomalous Edge-Level Models Filtering, *EF*.

Input: Edge server A_i's public key pk_{A_i} and the cloud server A's public key pk_A in PLHE; outputs of Algorithm 3 $\left\{\left[2\vec{m'}_i^{(r+1)}\right]_{A_i}, \left[||\vec{m'}_i^{(r+1)}||_2^2\right]_{A_i}, \left[\vec{x}_i^{(r+1)}\right]_A\right\}$ on the A for edge server A_i; parameters for screening norms θ;

Output: Set $\mathcal{A}^{(T)}$ of the surviving edge servers and norms $\{M, \{||\vec{m}_i^{(r+1)}||_2\}_{A_i\in\mathcal{A}^{(T)}}\}$ on A; $(x'^{(r+1)}_i)^2||\vec{m}_i^{(r+1)}||_2^2$ on A_i;

1: A initializes $\mathcal{A}^{(T)} = \emptyset$;

2: A decrypts $[\vec{x}_i^{(r+1)}]_A$ to obtain $\vec{x}_i^{(r+1)}$ and calculates

$$[||\vec{m}_i^{(r+1)}||_2^2]_{A_i} = Add\left\{\left[\left|\left|\vec{m'}_i^{(r+1)}\right|\right|_2^2\right]_{A_i}, \left[(\vec{x}_i^{(r+1)})_2^2\right]_{A_i}, Mult\left(\left[2\vec{m'}_i^{(r+1)}\right]_{A_i}, \vec{x}_i^{(r+1)}\right)\right\};$$

and uses $Mult$ with random $x'^{(r+1)}_i \in \mathbb{Z}$ to obtain $[(x'^{(r+1)}_i)^2||\vec{m}_i^{(r+1)}||_2^2]_{A_i}$ and sends it to each A_i;

3: Each A_i decrypts to get $\left(x'^{(r+1)}_i\right)^2||\vec{m}_i^{(r+1)}||_2^2$ and sends $\left[\left(x'^{(r+1)}_i\right)^2||\vec{m}_i^{(r+1)}||_2^2\right]_A$ back to A;

4: A puts each surviving A_i in $\mathcal{A}^{(T)}$;

5: A obtains the norm $||\vec{m}_i^{(r+1)}||_2$ of each $A_i \in \mathcal{A}^{(T)}$ and calculates the median M of all these norms;

6: A deletes A_i from $\mathcal{A}^{(T)}$ unless it meets $\|\vec{m}_i^{(r+1)}\|_2 \in [M-\theta, M+\theta]$, where θ is an adjustable threshold to ensure $|\mathcal{A}^{(T)}| = l' = \min\{(1-\beta)l, l_s\}$ for known β or $\min\{0.5l, l_s\}$ for unknown β, and l_s is the count of surviving edge servers.

Thus, by selecting clients from the largest clusters for the edge aggregation, A_i can effectively filter out anomalous local model updates and enhance the robustness of the resulting edge-level model. For example, the simplest case occurs when $b = 1$, $k = 2$ [17]. In this case, each edge server splits the clients' sign tuples into two clusters and selects the cluster with the highest number of clients for edge aggregation. This strategy emphasizes consensus in gradient directionality while discarding potential outliers.

3) Privacy-Aware Edge Aggregation: Each edge server A_i performs Algorithm 3 with its remaining clients $u_{i,j} \in U_i^{(T)}$ to securely aggregate their noisy momenta with supporting client dropouts. In Algorithm 3, each client calculates the momentum of her local model and adds noise to it to obtain $\vec{m'}_{i,j}^{(r+1)}$ on line 1. Then, A_i works as a bucket and executes the SA algorithm with each of its trusted clients to securely get the average $\vec{m'}_i^{(r+1)}$ of their noisy momenta, accommodating potential client dropouts on line 2. SA is based on secret sharing and PRG with high efficiency and accuracy as presented in Algorithm 1. SA can be replaced by other efficient secure model aggregation method based on LDP [20], [21], secret sharing [46], and etc. All surviving clients' $[\vec{x}_{i,j}^{(r+1)}]_A$ are summed to obtain $[\vec{x}_i^{(r+1)}]_A$, and the messages related to $\vec{m'}_i^{(r+1)}$ are encrypted on line 3, and A_i sends the ciphertexts to A.

4) Privacy-Aware Anomalous Edge-Level Models Filtering: The cloud server A performs Algorithm 4 with its surviving edge servers to achieve norm screening with privacy preservation. First, A calculates the noisy norm square of each A_i's edge-level momentum encrypted with pk_{A_i} in PLHE on line 2, Algorithm 4. Then, A uses the secret $x'^{(r+1)}_i$ to obtain the norm of noisy momentum $\vec{m}_i^{(r+1)}$ for each surviving edge server without knowing the actual $\vec{m}_i^{(r+1)}$, and calculates the median M of all these norms on lines 3-5. Finally, A filters out Byzantine behaviors by performing norm-screening on line 6 to obtain a set $\mathcal{A}^{(T)}$ comprising trusted surviving edge servers. Its size is $|\mathcal{A}^{(T)}| = l' = \min\{(1-\beta)l, l_s\}$ if A knows β, or $|\mathcal{A}^{(T)}| = \min\{0.5\,l, l_s\}$ if A does not know β, where l_s denotes the number of surviving edge servers. After executing Algorithm 4, A obtains set $\mathcal{A}^{(T)}$ of the surviving edge servers in its coverage, norms $\{||\vec{m}_i^{(r+1)}||_2\}_{A_i\in\mathcal{A}^{(T)}}$, and their median M, and each A_i obtains the noisy norm $(x'^{(r+1)}_i)^2||\vec{m}_i^{(r+1)}||_2^2$ corresponding to the edge-level model.

5) Privacy-Aware Global Aggregation: The cloud server A performs Algorithm 5 with its remaining edge servers in $\mathcal{A}^{(T)}$ to securely obtain the updated global model $\vec{w}_a^{(r+1)} \in \mathbb{W}^m$. A sets

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

Algorithm 5: Privacy-Aware Global Aggregation, *GA*.

Input: Outputs of EF on A and each surviving A_i; the threshold t related to edge server dropouts in global aggregation, parameters for computing momenta ζ;

Output: The updated global model $\vec{w}_a^{(r+1)} \in \mathbb{W}^m$;

1: For each A_i who satisfies $||\vec{m}_i^{(r+1)}||_2 \geq M$, A sends $[x_i'^{(r+1)}M]_{A_i}$ back and sets $x_i''^{(r+1)} = \frac{M}{||\vec{m}_i^{(r+1)}||_2}$, and for the other A_i in $\mathcal{A}^{(T)}$, A sets $x_i''^{(r+1)} = 1$.

2: Each A_i who receives the $[x_i'^{(r+1)}M]_{A_i}$ decrypts it and calculates $\vec{m}''^{(r+1)}_i = \frac{x_i'^{(r+1)}M}{x_i'^{(r+1)}||\vec{m}_i^{(r+1)}||_2}\vec{m}'^{(r+1)}_i$; and other A_i sets $\vec{m}''^{(r+1)}_i = \vec{m}'^{(r+1)}_i$;

3: A securely averages the momentum vectors of edge servers in the updated $\mathcal{A}^{(T)}$ with supporting edge server dropouts, $\vec{m}''^{(r+1)}_a \leftarrow SA(\{\vec{m}''^{(r+1)}_i\}_{A_i \in \mathcal{A}^{(T)}}, t)$;

4: A calculates $\vec{m}_a^{(r+1)} = \vec{m}''^{(r+1)}_a + \sum_{A_i \in \mathcal{A}^{(T)}} x_i''^{(r+1)}\vec{x}_i^{(r+1)}$;

5: A recovers the global model $\vec{w}_a^{(r+1)} \leftarrow \vec{w}_a^{(r)} - \zeta\vec{m}_a^{(r+1)}$.

$x_i''^{(r+1)}$ and sends its ciphertext to each remaining edge server A_i whose norm exceeds the median on line 1 to prepare for norm-clipping. Based on the careful design of $[x_i'^{(r+1)}M]_{A_i}$, each edge server A_i clips the norm of its noisy momentum to obtain $\vec{m}''^{(r+1)}_i$ on line 2. Specifically, A_i trims the true momentum hidden in $\vec{m}'^{(r+1)}_i$, ensuring that the norm of the trimmed true momentum is less than or equal to the median M. This operation also reduces the norm of the noise $\vec{x}_i^{(r+1)}$ added to $\vec{m}'^{(r+1)}_i$, effectively removing a portion of the noise. A executes the SA algorithm with its trusted edge servers in $\mathcal{A}^{(T)}$ to securely obtain the average of all trimmed momenta $\{\vec{m''}i_i^{(r+1)} | A_i \in \mathcal{A}^{(T)}\}$ on lines 3, and further moves all remaining noises to obtain the real average momentum $\vec{m}_a^{(r+1)}$ on line 4. A finally recovers the corresponding global model as the updates in this round, i.e., $\vec{w}_a^{(r+1)}$.

Note that PPBR is specially proposed for Byzantine-robust EHFL on Non-IID data. The dual filtering steps in PPBR inherently select gradients from the majority of clients that exhibit greater consistency by clustering their sign statistics and screening and clipping the norms, to fully consider the valuable but diverse models incurred by Non-IID training data. Thus, the subsequent averaging aggregation steps hierarchically 'mix' all the remaining models at both the edge and cloud levels and produce a series of momenta characterized by a lower variance than the original local models. As the aggregation rounds progress, PPBR could continuously increase the consistency among the gradients of remaining momenta of local and edge models. It allows PPBR to enhance the accuracy and ensure its convergence to the optimum [15], resulting in the strong robustness to Non-IID data. Meanwhile, the increased consistency of the remaining momenta makes any attempt by Byzantine attackers to sabotage the training process easily stand out, especially at the end of the training, which provides the strong Byzantine resilience.

V. Privacy Analysis

We begin by analyzing the privacy of the anomalous local models filtering step, in which only the sign-statistic tuples are perturbed before clustering in Algorithm 2 to provide privacy protection. Thus, we discuss its privacy in Lemma 1.

Lemma 1: Algorithm 2 achieves ϵ_χ-CLDP defined in (1).

Lemma 1 ensures that it is difficult for any attacker within the EHFL system to infer the true sign statistics of targeted clients from the perturbed tuples, thus preventing the inference of local models [37].

Then, we analyze the privacy of the edge aggregation step, i.e., Algorithm 3, in Lemma 2.

Lemma 2: In Algorithm 3, the real parameters of clients' local models are secure against the coalition of $t_i' < n_i' - 1$ parties for each edge server A_i, and the A_i's edge aggregation result can be obtained with less than $n_i' - t_i$ dropped out clients.

Lemma 2 shows that Algorithm 3 can effectively protect local models against collusive inference attacks and accommodate partial client dropouts during the edge aggregation process.

Subsequently, for the edge-level models filtering step, we discuss the operations of screening the norms of edge-level momenta, which primarily rely on PLHE and the random noise added during edge aggregation to protect the edge momentum. Thus, we conclude that the privacy of the edge model filtering step is ensured by the semantic security of PLHE, as demonstrated in Lemma 3.

Lemma 3: Algorithm 4 securely detects and filters out the anomalous edge-level models, if PLHE is semantically secure.

Finally, we analyze the privacy of the global aggregation step, which is based on a similar principle as Lemma 2, resulting in Lemma 4.

Lemma 4: In Algorithm 5, the real parameters of edge-level models are secure against the coalition consisting of $t' < l' - 1$ parties for the cloud server A, and the global aggregation result can be obtained with less than $l' - t$ edge servers dropping out.

To improve readability and logic, the detailed proofs of Lemmas 1–4 are provided in the Appendix, available online. Based on Lemmas 1–4, we can derive Theorem 1.

Theorem 1: PPBR achieves the privacy protection goal described in Section III-C.

Proof: PPBR implements a dual-layer filter for anomalous models via Algorithm 2 and Algorithm 4, and performs hierarchical model aggregations through Algorithms 3 and 5. Thus, based on the privacy analysis of Algorithms 2–5 in Lemmas 1–4, we can conclude that PPBR achieves the privacy preservation goal described in Section III-C. □

Theorem 1 demonstrates that CLDP, SA, and PLHE are essential for PPBR to achieve its privacy preservation goal. As shown in Lemma 1, CLDP enhances privacy protection during the anomalous local models filtering step; SA protocol plays a pivotal role in preserving privacy during the edge aggregation discussed in Lemma 2 and the global aggregation discussed in Lemma 4; and as demonstrated in Lemma 3, PLHE boosts

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

privacy protection during the anomalous edge models filtering step. If any one of these techniques is removed, specific privacy vulnerabilities arise: without CLDP, the information about the updated local models of clients could be inferred from the sign tuples during clustering; without SA, the edge or cloud servers might reconstruct local or edge momenta during aggregations; and without PLHE, the cloud sever could access sensitive edge-level models during the filtering process. The integration of PLHE, SA, and CLDP in PPBR significantly strengthens privacy protection throughout the entire Byzantine-robust EHFL process.

Additionally, we further discuss the impact of the CLDP perturbation in Algorithm 2 on clustering performance by analyzing the distance properties between the real and perturbed centroids in Theorem 2.

Theorem 2: In Algorithm 2, for any $C_\kappa \in \mathcal{C}_i$ and each random variable $\overrightarrow{y}_{i,j}^{(r+1)} \in C_\kappa$, supposing that the real centroid $\overrightarrow{s}_\kappa = \frac{1}{|C_\kappa|}\sum_{C_\kappa} \overrightarrow{s}_{i,j}^{(r+1)}$ and the perturbed centroid $\overrightarrow{y}_\kappa = \frac{1}{|C_\kappa|}\sum_{C_\kappa} \overrightarrow{y}_{i,j}^{(r+1)}$, $\overrightarrow{y}_\kappa$ is unbiased, i.e.,

$$E(\overrightarrow{y}_\kappa) = \overrightarrow{s}_\kappa,$$

and the expectation of its MSE (Mean Squared Error) satisfies

$$E(\|\overrightarrow{y}_\kappa - \overrightarrow{s}_\kappa\|_2^2) \le \frac{3L^2(e^{\epsilon_\chi c_L}-1)}{5|C_\kappa|(e^{\epsilon_\chi c_L}+1)} + \frac{L^2}{|C_\kappa|}.$$

Proof: For any $C_\kappa \in \mathcal{C}$ and $\overrightarrow{s}_\kappa$, we have

$$\begin{aligned}
&E(\overrightarrow{y}_{i,j}^{(r+1)}) - \overrightarrow{s}_{i,j}^{(r+1)} \\
&= \int\cdots\int_{\mathcal{S}_{L'}} (\overrightarrow{y}_{i,j}^{(r+1)} - \overrightarrow{s}_{i,j}^{(r+1)}) f(\overrightarrow{y}_{i,j}^{(r+1)}) d\overrightarrow{y}_{i,j}^{(r+1)} \\
&= \int\cdots\int_{\mathcal{B}_L} (\overrightarrow{y}_{i,j}^{(r+1)} - \overrightarrow{s}_{i,j}^{(r+1)}) \lambda e^{-\epsilon_\chi \|\overrightarrow{y}_{i,j}^{(r+1)} - \overrightarrow{s}_{i,j}^{(r+1)}\|_2} d\overrightarrow{y}_{i,j}^{(r+1)} \\
&+ \int\cdots\int_{\mathcal{S}_{L'}-\mathcal{B}_L} (\overrightarrow{y}_{i,j}^{(r+1)} - \overrightarrow{s}_{i,j}^{(r+1)}) \lambda e^{-\epsilon_\chi L} d\overrightarrow{y}_{i,j}^{(r+1)}. \qquad (3)
\end{aligned}$$

In (3), $\mathcal{B}_L$ and $\mathcal{S}_{L'} - \mathcal{B}_L$ are symmetrical under all rotations about center $\overrightarrow{s}_{i,j}^{(r+1)}$, and $(\overrightarrow{y}_{i,j}^{(r+1)} - \overrightarrow{s}_{i,j}^{(r+1)})\cdot \lambda e^{-\epsilon_\chi \|\overrightarrow{y}_{i,j}^{(r+1)} - \overrightarrow{s}_{i,j}^{(r+1)}\|_2}$ and $(\overrightarrow{y}_{i,j}^{(r+1)} - \overrightarrow{s}_{i,j}^{(r+1)})\lambda e^{-\epsilon_\chi L}$ are symmetrical about center $\overrightarrow{s}_{i,j}^{(r+1)}$ as well. Thus, we can expect $E(\overrightarrow{y}_{i,j}^{(r+1)}) - \overrightarrow{s}_{i,j}^{(r+1)}$ to be 0, i.e., $E(\overrightarrow{y}_{i,j}^{(r+1)}) = \overrightarrow{s}_{i,j}^{(r+1)}$ that equals to $E(\overrightarrow{y}_{i,j}^{(r+1)})[\omega] = \overrightarrow{s}_{i,j}^{(r+1)}[\omega], \omega \in [1,d]$, and have

$$E(\overrightarrow{y}_\kappa) = \frac{1}{|C_\kappa|}\sum_{C_\kappa} E(\overrightarrow{y}_{i,j}^{(r+1)}) = \frac{1}{|C_\kappa|}\sum_{C_\kappa} \overrightarrow{s}_{i,j}^{(r+1)} = \overrightarrow{s}_\kappa.$$

Then, we consider

$$E\left(\|\overrightarrow{y}_\kappa - \overrightarrow{s}_\kappa\|_2^2\right) = \frac{\sum_{\omega\in[1,d]}\sum_{C_\kappa} E((\overrightarrow{y}_{i,j}^{(r+1)}[\omega] - \overrightarrow{s}_{i,j}^{(r+1)}[\omega])^2)}{|C_\kappa|^2}.$$

Since

$$\begin{aligned}
&E((\overrightarrow{y}_{i,j}^{(r+1)}[\omega] - \overrightarrow{s}_{i,j}^{(r+1)}[\omega])^2) \\
&= \int\cdots\int_{\mathcal{S}_{L'}} (\overrightarrow{y}_{i,j}^{(r+1)}[\omega] - \overrightarrow{s}_{i,j}^{(r+1)}[\omega])^2 f(\overrightarrow{y}_{i,j}^{(r+1)}) d\overrightarrow{y}_{i,j}^{(r+1)} \\
&= \int\cdots\int_{\mathcal{B}_L} (\overrightarrow{y}_{i,j}^{(r+1)}[\omega] - \overrightarrow{s}_{i,j}^{(r+1)}[\omega])^2 g(\overrightarrow{y}_{i,j}^{(r+1)}) d\overrightarrow{y}_{i,j}^{(r+1)} \\
&+ \int\cdots\int_{\mathcal{S}_{L'}} (\overrightarrow{y}_{i,j}^{(r+1)}[\omega] - \overrightarrow{s}_{i,j}^{(r+1)}[\omega])^2 \lambda e^{-\epsilon_\chi L} d\overrightarrow{y}_{i,j}^{(r+1)} \\
&- \int\cdots\int_{\mathcal{B}_L} (\overrightarrow{y}_{i,j}^{(r+1)}[\omega] - \overrightarrow{s}_{i,j}^{(r+1)}[\omega])^2 \lambda e^{-\epsilon_\chi L} d\overrightarrow{y}_{i,j}^{(r+1)} \\
&\le \frac{L^2(e^{\epsilon_\chi c_L}-1)}{(d+2)(e^{\epsilon_\chi c_L}+1)} + \frac{L^2}{3}
\end{aligned}$$

according to Lemma 4 in [48], where $g(\overrightarrow{y}_{i,j}^{(r+1)}) = \lambda e^{-\epsilon_\chi \|\overrightarrow{y}_{i,j}^{(r+1)} - \overrightarrow{s}_{i,j}^{(r+1)}\|_2}$ and $d = 3$, we have

$$\begin{aligned}
E(\|\overrightarrow{y}_\kappa - \overrightarrow{s}_\kappa\|_2^2) &\le \frac{dL^2(e^{\epsilon_\chi c_L}-1)}{(d+2)|C_\kappa|(e^{\epsilon_\chi c_L}+1)} + \frac{dL^2}{3|C_\kappa|} \\
&= \frac{3L^2(e^{\epsilon_\chi c_L}-1)}{5|C_\kappa|(e^{\epsilon_\chi c_L}+1)} + \frac{L^2}{|C_\kappa|}.
\end{aligned}$$

□

TABLE III
DATASETS AND TRAINING HYPERPARAMETERS

Dataset	Model	Clients	LR	Epochs	Steps	Batch size
MNIST (IID)	MLP	240	0.01	600	20 (FedAvg)	32
MNIST (Non-IID)	MLP	240	0.01	2000	1 (FedSGD)	32
F-MNIST (IID)	CNN	240	0.01	600	20 (FedAvg)	64
F-MNIST (Non-IID)	CNN	240	0.01	2000	1 (FedSGD)	64

VI. PERFORMANCE EVALUATION

We evaluate the robustness of PPBR scheme under data heterogeneity settings and Byzantine attacks, and discuss its overhead and convergence through extensive simulations.

A. Experiment Setup

The proposed PPBR is implemented in Python using the PyTorch machine learning framework. PLHE in PPBR is realized via TenSEAL[1], a library designed for HE operations on tensors, built on top of Microsoft SEAL. All simulations are performed on a Linux server equipped with dual Intel Xeon Silver 4210R CPUs (2.40 GHz), a RTX 3090 GPU, and 512 GB of RAM.

Datasets and models: As shown in Table III, a Multi-Layer Perceptron (MLP) [50] model with two hidden layers is trained on the MNIST dataset [35], and a Convolutional Neural Network (CNN) [50] model, consisting of 3 convolutional layers and 2 fully connected layers, is trained on the Fashion-MNIST (F-MNIST) dataset [35]. Both datasets contain 60,000 training samples and 10,000 test samples. Each sample is a grayscale image with dimensions of 28×28. By default, we set the batch size to 32 for the MNIST dataset and 64 for the F-MNIST dataset.

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
RESILIENCE OF PPBR AND BASELINES FOR EFHL ON NON-IID DATA TO BYZANTINE ATTACKS

Dataset (model)	**Model aggregation algorithms**	**Byzantine attacks** with $\beta = 30\%$							
		No Attack(IID)	Label flipping	Gaussian	Sign flipping	ALIE	IPM ($\xi = 0.1$)	IPM ($\xi = 100$)	MinMax
MNIST (MLP)	Mean (FedAvg)	**95.66**	94.99	17.55	92.02	93.02	94.53	11.35	59.75
	Median	91.63	84.61	90.34	86.00	88.41	62.40	29.16	65.25
	TrimmedMean	95.65	88.12	90.91	88.07	86.27	73.56	29.68	61.12
	GeoMed	95.25	95.30	46.78	91.80	70.93	29.41	86.67	85.59
	Multi-Krum	92.60	89.33	89.31	76.64	87.18	80.17	89.31	88.91
	DnC	95.56	95.21	95.27	91.99	91.17	93.58	95.10	95.01
	CenterClipping	95.64	95.04	49.04	91.92	87.37	94.49	84.57	59.51
	PPBR	95.47	**95.82**	**95.37**	**93.70**	**95.37**	**95.35**	**95.37**	**95.37**
F-MNIST (CNN)	Mean (FedAvg)	**88.13**	87.53	10.00	83.78	75.10	86.52	10.00	29.61
	Median	87.47	81.86	56.51	79.97	33.05	64.71	45.56	48.62
	TrimmedMean	88.08	82.01	58.47	79.63	66.26	78.22	39.97	44.02
	GeoMed	88.03	87.74	61.31	84.09	32.62	64.88	77.73	57.19
	Multi-Krum	86.89	82.63	83.14	77.08	81.24	76.21	82.43	83.06
	DnC	88.10	87.51	87.39	84.13	73.89	85.24	86.88	87.08
	CenterClipping	88.09	87.74	27.77	84.21	76.75	87.11	6.01	81.09
	PPBR	88.00	**87.89**	**87.97**	**87.53**	**87.77**	**87.33**	**87.83**	**87.76**

Stochastic Gradient Descent (SGD) is used as the optimizer in all experiments, with a global learning rate of 1.0 and a local learning rate of 0.01. For IID and Non-IID data, we used 200 rounds of FedAvg [38] and 2000 rounds of FedSGD, respectively. Each client locally performed 20 SGD steps before uploading the updates to the server.

Data heterogeneity. We consider an FL system with 240 clients in both IID and Non-IID data settings, using the training hyperparameters listed in Table III. In IID data setting, each client randomly samples data of the same size from the dataset. Non-IID data are simulated using the Dirichlet distribution $\boldsymbol{p}_l \sim Dir_K(\alpha)$ [13], where a smaller α indicates a stronger divergence from the IID data.

Byzantine attacks: We consider the Byzantine attacks described in Section III-C, including label flipping attack, sign flipping attack, Gaussian attack, ALIE attack, IPM attack and MinMax attack.

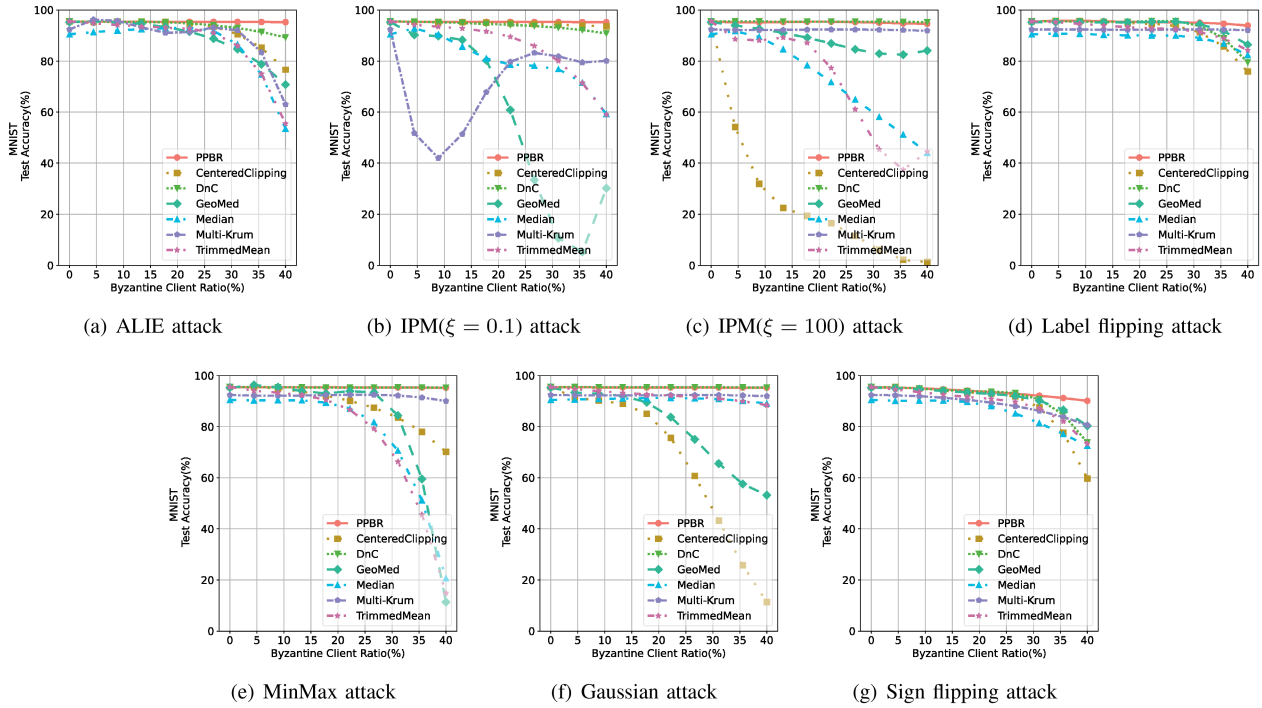
Baselines: We select several widely recognized robust model aggregation algorithms that are closely related to PPBR as baselines, including Median [38], TrimmedMean [38], GeoMed [40], Multi-Krum [39], DnC [36], and CenteredClipping [18]. Note that these baselines are implemented under various Byzantine attacks in data heterogeneity settings without LDP guarantees. Thus, their accuracy is not affected by LDP noise.

Key parameter setting and evaluation metric: The parameters $k = 2$ and $b = 1$ are set, following [17], to quickly filter out anomalous local models based on clustering results. In line with the discussion on CLDP in [48], we set $l = 8$, $L = 6$, $L' = 7$ to help limit the accuracy loss brought by the CLDP perturbation. $\varphi = 0.9$ and $\zeta = 0.0125$ [15] are selected based on the convergence analysis for the bucketing aggregations in [15], which are closely related to the dual averaging aggregations used in our PPBR scheme. For the adversarial setting, we use $\beta = 0.3$ and $\alpha = 0.5$ as default values, indicating a Byzantine ratio of 30% and a moderate Non-IID level of 0.5in the EHFL system. In the following experiments, all parameters are set to the above defaults unless stated otherwise. All the final results are average values over 10-20 runs. We choose AES and Diffie-Hellman protocols as the PRG and key agreement protocols for the SA in Algorithm 1. Meanwhile, we use the BFV scheme for PLHE with the number of SIMD slots $n = 2048$. As for the cryptographic tools, all the parameters are set to a 128-bit security level. We employ accuracy as our evaluation metric to assess the performance of robust EHFL, which defined as the ratio of the samples correctly predicted by the final model to the total number of testing samples. The ratio of the selected benign clients is used to measure the effectiveness of the CLDP-guaranteed clustering during the privacy-aware anomalous local models filtering.

B. Robustness to Byzantine Attacks

Table IV shows PPBR's robustness to Byzantine attacks in the Non-IID data setting of $\alpha = 0.5$ and $\beta = 30\%$, since its accuracy values under all considered Byzantine attacks are the closet to the theoretical optimums 95.66 and 88.13 on MNIST and F-MNIST, respectively. The theoretical optimums are set to be the accuracy of the model aggregated by Mean [38] in IID data setting without Byzantine adversaries. First, DnC and PPBR usually provide stronger robustness to Byzantine attacks than others do, especially under Gaussian, ALIE, IPM and MinMax attacks. Besides the dimension statistics of local models considered in others, DnC analyzes the similarity of models, and PPBR clusters their sign statistics. Although PPBR theoretically has its own accuracy losses due to the CLDP noises, PPBR provides higher accuracy than Dnc under all attacks especially ALIE attack on F-MNIST dataset. The main reason is that PPBR neatly filters out a great part of anomalous local models from malicious Byzantine mobile devices based on the sign-statistics clustering

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.



(a) ALIE attack (b) IPM($\xi = 0.1$) attack (c) IPM($\xi = 100$) attack (d) Label flipping attack

(e) MinMax attack (f) Gaussian attack (g) Sign flipping attack

Fig. 3. Resilience of PPBR and baselines for EHFL on Non-IID data (generated from MNIST dataset with $\alpha = 0.1$) to Byzantine attacks.

and norms screening and clipping. Thus, its robustness is not negatively impacted by the CLDP perturbations. Furthermore, the robustness of CenterClipping is limited and cannot withstand Gaussian, IPM ($\xi = 100$) and MinMax attacks. Finally, PPBR stands out as it achieves Byzantine robustness without requiring knowledge of the fraction of Byzantine clients, while also providing strong privacy guarantees. Note that PPBR under label flipping attacks in the Non-IID setting even outperforms FedAvg in Table IV. This is because label flipping attacks typically produce gradients with distinctive patterns and large deviations, which PPBR can effectively suppress and mitigate through sign-statistics clustering and norm clipping. By filtering these anomalies, PPBR steers updates toward benign models, curbs overfitting, and achieves the "unexpected positive effect."

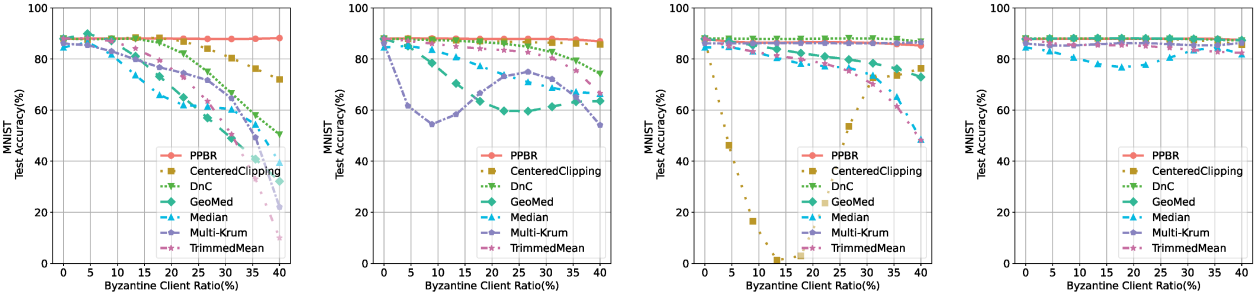
As shown in Figs. 3 and 4, the impact of Byzantine attacks generally becomes more prominent as the proportion of Byzantine clients β increases, resulting in lower accuracy. Among all the attacks considered, ALIE, IPM ($\xi = 100$) and MinMax attacks have a significant impact on all robust aggregation methods except for PPBR, as the proportion of Byzantine clients increases. Meanwhile, TrimmedMean, GeoMed and CenteredClipping show the weaker robustness to IPM, Gaussian, and ALIE attacks as more Byzantine clients existed in EHFL. For example, when β is 40%, the accuracy of CenteredClipping, a gradient clipping-based robust aggregation method, plummets to approximately 10% under Gaussian attacks. This indicates that the Gaussian attacks can bypass CenteredClipping method, essentially reducing its output to the level of random guessing. Different from them, PPBR demonstrates robust resilience against all considered Byzantine attacks, since PPBR experiences the least decrease in accuracy with increasing β, and PPBR maintains the highest accuracy under all attacks. All robust aggregation methods show a certain degree of accuracy degradation in Non-IID data settings compared to the IID data setting, even without any attacks. Among them, Median and Multi-Krum experience relatively larger accuracy drops of 3.60% and 2.06%, primarily due to data heterogeneity.

C. Robustness to Non-IID Data

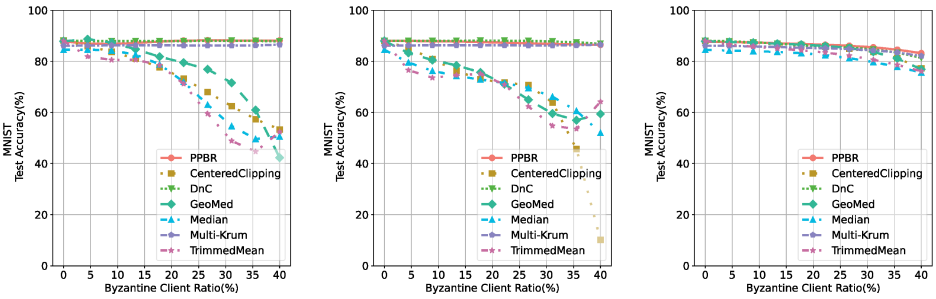
Figs. 5 and 6 show the impact of Non-IID data on the robustness of aggregation methods to Byzantine attacks. Here, ALIE, sign flipping, and MinMax attacks are chosen for illustrative purposes based on their strong attack abilities presented in Figs. 3 and 4. As described in Section VI-A, α is the parameter indicating the divergence from IID data, and a smaller α corresponds to a stronger divergence.

First, under Byzantine attacks, the accuracy of all robust aggregation approaches but PPBR obviously decreases with decreasing α on MNIST and F-MNIST datasets. Furthermore, metric-based and clipping-based methods such as Multi-Krum, DnC and CenteredClipping experience a marked reduction in the accuracy in the Non-IID setting of $\alpha = 0.1$ on MNIST or F-MNIST datasets, since a great divergence from IID data brings a big deviation of benign updates that is a damage to the robust schemes based on metrics of clients' updates. Finally, PPBR maintains high accuracy across diverse Non-IID data settings in the presence of all considered Byzantine attacks, presenting strong Non-IID resilience. Since its average-based hierarchical aggregations thoroughly mix the data from all the remaining clients after the sign-statistics filtering and norms screening filtering, PPBR can consider the valuable updates of

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

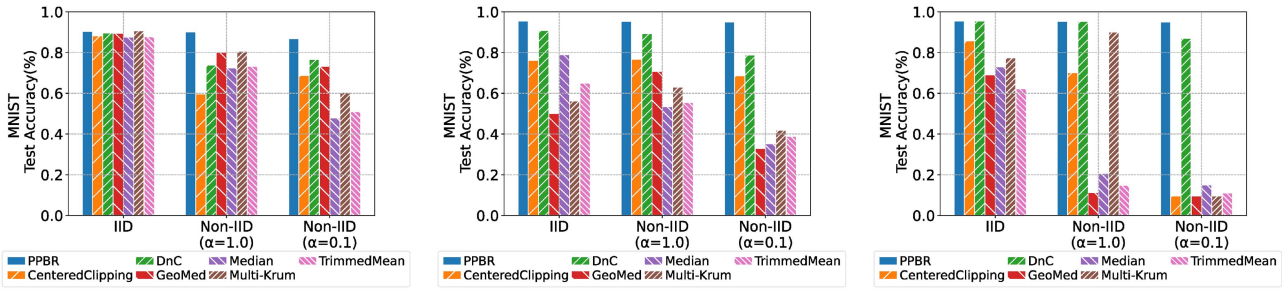


(a) ALIE attack (b) IPM($\xi = 0.1$) attack (c) IPM($\xi = 100$) attack (d) Label flipping attack



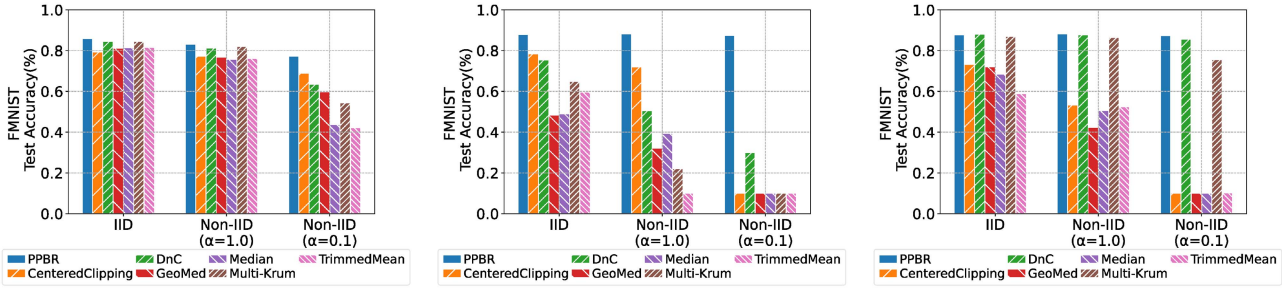
(e) MinMax attack (f) Gaussian attack (g) Sign flipping attack

Fig. 4. Resilience of PPBR and baselines for EHFL on Non-IID data (generated from F-MNIST dataset with $\alpha = 0.1$) to Byzantine attacks.



(a) Sign flipping attack (b) ALIE attack (c) MinMax attack

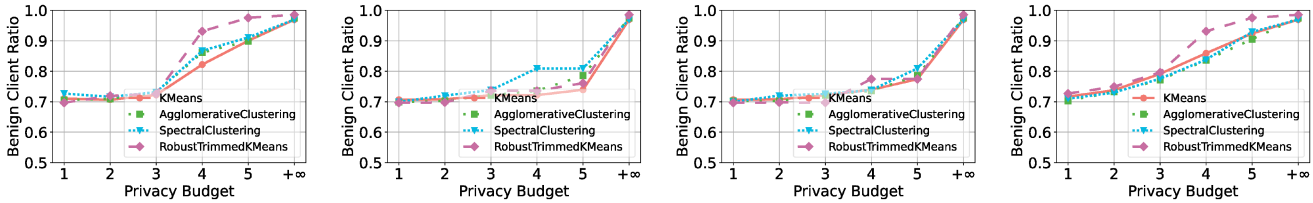
Fig. 5. Resilience of PPBR and baselines across IID and Non-IID data ($\alpha = 1.0, 0.1$) generated from MNIST dataset to Byzantine attacks with $\beta = 40\%$.



(a) Sign flipping attack (b) ALIE attack (c) MinMax attack

Fig. 6. Resilience of PPBR and baselines across IID and Non-IID data ($\alpha = 1.0, 0.1$) generated from F-MNIST dataset to Byzantine attacks with $\beta = 40\%$.

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.



(a) RR (b) GRR (c) OLH (d) PPBR

Fig. 7. Benign client ratio of different clustering algorithms under different LDP perturbation mechanisms, FMNIST dataset.

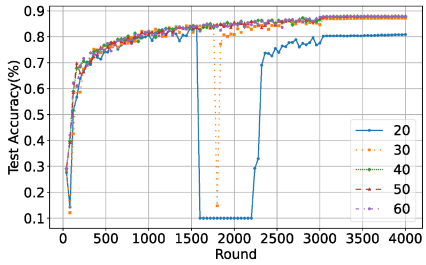
the models on non-representative local data brought by Non-IID and mitigate the negative effect of Non-IID on its accuracy.

D. Impact of CLDP on Robustness

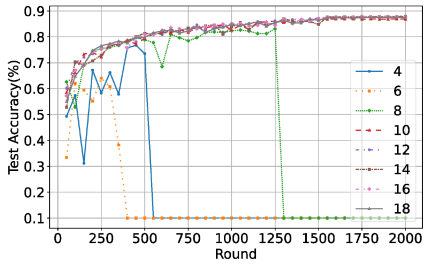
Considering that CLDP is used in PPBR to strongly protect clients' private data with accuracy loss, we discuss impacts of CLDP on robustness of PPBR.

Effectiveness of the trimmed k-means clustering with CLDP guarantee: We compare the CLDP-guaranteed clustering method (designed in Algorithm 2) with several combinations of other typical clustering algorithms and LDP mechanisms, namely, k-means clustering, agglomerative clustering, and spectral clustering, as well as RR [51], GRR [52], and OLH [53]. For consistency, we translate the privacy budget ϵ of these LDP mechanisms into ϵ_χ under the CLDP concept. Fig. 7 shows that Algorithm 2 has a higher ratio of the selected benign clients than others in all privacy budget settings. Although the performance of the combination of trimmed k-means clustering with RR is close to Algorithm 2, Algorithm 2 can more accurately filter out malicious models when the privacy protection strength is particularly high, e.g., $\epsilon_\chi \geq 3$. For instance, more than 80% of the benign models are selected to participate in the aggregations with achieving 3-CLDP in Algorithm 2. Its superior performance primarily stems from the design of CLDP perturbation and the implementation of the robust trimmed k-means clustering techniques.

Impact of LDP parameters on the robustness: In PPBR, the parameters related to CLDP are the privacy budget ϵ_χ and the number n_i of clients in the edge server's coverage. We discuss their impact on the average accuracy of PPBR against ALIE and MinMax attacks with $\beta = 40\%$ in Fig. 8. It is obvious that as the bucket n_i and privacy budget ϵ_χ increase, the accuracy significantly increases, presenting strong robustness to the Byzantine attacks. Meanwhile, the accuracy converges rapidly to a high value if both parameters are large enough. Specifically, the average accuracy nearly reaches the best, when the privacy budget and the number of clients held by each edge server are greater than 10 and 40, respectively. Fig. 9 further shows the impact of LDP privacy budgets on the robustness across diverse Non-IID data. As the privacy budget ϵ_χ increases, PPBR consistently improves its accuracy at all levels of data heterogeneity (controlled by α) in different datasets. Here, a smaller α indicates a higher degree of data heterogeneity, meaning that the data distribution among clients is more diverse and



(a) PPBR with different n_i under ALIE attack



(b) PPBR with different ϵ_χ under MinMax attack

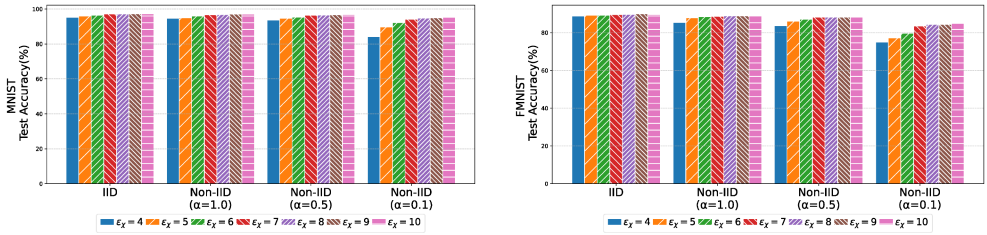
Fig. 8. Impact of privacy parameters on the Robustness of PPBR to Byzantine attacks with $\beta = 40\%$, F-MNIST dataset.

the data scale may vary significantly across clients compared to the IID data. Under high levels of data heterogeneity (i.e., small α), PPBR still achieves high model accuracy when the privacy budget satisfies $\epsilon_\chi \geq 4$. Although PPBR experiences a slight decrease in accuracy at higher levels of data heterogeneity settings (i.e., smaller α), its accuracy is still higher than 85% with $\epsilon_\chi \geq 7$. This demonstrates PPBR's notable adaptability and consistent robustness against Non-IID data under sign flipping attacks. The primary reason for this robust performance is that the filtering and aggregation steps in PPBR effectively offset the accuracy loss caused by CLDP noise in all considered complexities and scales of Non-IID datasets, especially when appropriate privacy budgets are applied.

E. Overhead of Privacy-Preserving Techniques

Considering that PLHE, random perturbations, CLDP-based clustering, and SA in PPBR strongly protect clients' private data but add overhead, we discuss the associated overhead of these privacy-preserving techniques.

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.



(a) PPBR with different ϵ_χ and α on MNIST

(b) PPBR with different ϵ_χ and α on F-MNIST

Fig. 9. Impact of ϵ_χ on the robustness of PPBR across IID and diverse Non-IID data ($\alpha = 1.0, 0.5, 0.1$) generated from MNIST and F-MNIST dataset to sign flipping attacks with $\beta = 40\%$.

TABLE V
THE KEY OPERATIONS IN PLHE IN EACH ROUND OF PERFORMING PPBR

Operations in PLHE	Cloud server A (assuming l surviving edge servers)	Edge server A_i	Client $u_{i,j}$
Enc	l	3	1
Dec	$2l$	2	1
Add	$2l$	$n_i - 1$	0
$Mult$	$2l$	0	0

TABLE VI
THE AVERAGE TIME (S) CONSUMED ON OPERATIONS IN PLHE IN EACH ROUND OF PERFORMING PPBR

Dataset	Traffic (MB)	Client $u_{i,j}$	Edge server A_i	Cloud server ($l = 6$)	Cloud server ($l = 10$)
MNIST	36.39	0.1200	0.5053	2.3668	3.9447
F-MNIST	133.22	0.4296	1.8140	8.4840	14.1399

The overhead of PLHE and random perturbation operations: In PPBR, edge and cloud servers are tasked with execution of PLHE operations in hierarchical aggregations to ensure the strong protection of clients' private data. Each client with a mobile device only needs to select a random vector for perturbation, encrypt it, and send it. Thus, we outline the computational complexity of key operations in PLHE in Table V and analyze the time costs of the edge and cloud servers and the corresponding traffic to perform these operations in Table VI. For each client, the computation time per round of PPBR in EHFL is less than 0.5 s. The communication overhead for PLHE ciphertexts per round is 36.39 MB in MNIST and 133.22 MB in F-MNIST, respectively. These overheads do not present substantial transmission pressure, especially considering that the corresponding transmission times in a 5 G network are only 0.6 s and 2.1 s, respectively. Both tables indicate that edge servers experience only a slight increase in communication and computation time to perform PLHE operations. For the cloud server, the computational time for PLHE operations increases linearly with the number of edge servers. However, this computational overhead is deemed acceptable. For example, the additional computation time required for PLHE operations in each round by the cloud server does not exceed 10 s with $l = 6$ edge servers. The reason is that PPBR carefully deploys the operations of encryption, decryption, addition under ciphertexts, and multiplication between a ciphertext and a plaintext within PLHE to edge and cloud servers in each aggregation round without significant efficiency loss.

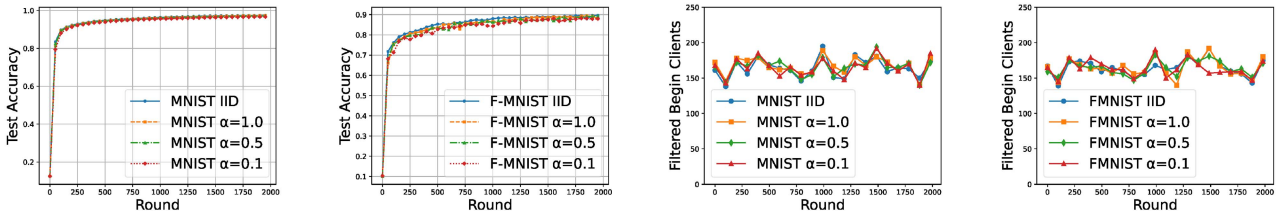
TABLE VII
THE OVERHEAD OF CLDP-BASED CLUSTERING (CC) AND SA OPERATIONS ON EACH ENTITY (CLIENT $u_{i,j}$, EDGE SERVER A_i, OR THE CLOUD SERVER A) IN EACH ROUND OF PERFORMING PPBR

Operation	Entity	Computation	Communication
CC	$u_{i,j}$	$O(d)$	$O(d)$
	A_i	$O(In_ikd)$	$O(n_id)$
SA	$u_{i,j}$	$O(n_i'^2 + n_i'm)$	$O(n_i' + m)$
	A_i	$O(n_i'^2m + l'^2 + l'm)$	$O(n_i'^2 + n_i'm + m + l')$
	A	$O(l'^2m)$	$O(l'^2 + l'm)$

The overhead of CLDP-based clustering (CC) operation: In Algorithm 2, each client generates a perturbed sign-tuple of dimensionality d and sends it to the corresponding edge server while performing the CC operation. Thus, as shown in Table VII, the computational and communication complexity of CC on the client side are both $O(d)$. Since $d = 3$ for a sign tuple in PPBR, this overhead is particularly manageable for clients using mobile devices. Edge server A_i collects the perturbed sign-tuples from n_i clients and executes the trimmed k-means clustering over I iterations. Hence, the computational complexity on the edge server side is $O(In_ikd)$, while its communication complexity remains $O(n_id)$. Here, $k = 2$ is sufficient to select the larger cluster's tuples, which represent the majority of reliable gradient directions.

The overhead of SA: The SA described in Algorithm 1 is executed once during both the edge and global aggregation steps. Since n_i' clients participate in A_i's edge aggregation, and each client's local model update corresponds to a momentum of dimension m, the computational complexity of running SA once on the client side is $O(n_i'^2 + n_i'm)$. It is mainly consumed by the operations involved in $Shamir(t_i, n_i', \cdot)$, secure key agreement S and PRG. The communication cost on the client side is $O(n_i' + m)$, which involves sending and receiving public keys, sending and receiving secret shares, sending masked inputs, and sending secret shares of keys/masks. For edge server A_i during the execution of SA, its computational cost is $O(n_i'^2\, m)$, which includes reconstructing keys and masks and generating PRG outputs. A_i's communication cost is $O(n_i'^2 + n_i'm)$, which involves sending model updates, mediating pairwise communications, and receiving masked inputs. The overhead of performing SA once in edge aggregation increases with the growth of n_i' and

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.



(a) Iterations (I) on MNIST (b) Iterations (I) on F-MNIST (c) Filtered clients on MNIST (d) Filtered clients on F-MNIST

Fig. 10. Impact of Non-IID data on convergence iterations and the number of benign clients retained after filtering in PPBR under ALIE attacks.

m, but it remains acceptable for clients primarily using mobile devices and edge servers. As detailed in [6], simulations on a Linux workstation with an Intel Xeon CPU E5-1650 v3 (3.50 GHz) and 32 GB of RAM show that when $n_i' = 500$ and $m = 100k$, the total execution time of SA for each client is 13.159 s, the communication volume is 0.95 MB, and the edge server's execution time is about 14.670 s without dropouts and 62.239 s with 10% dropouts. Similarly, l' edge servers and the cloud server execute SA in global aggregation. Thus, the computational and communication costs per edge server during global aggregation are $O(l'^2 + l'm)$ and $O(l' + m)$, respectively. The computational and communication costs of the cloud server are $O(l'^2\, m)$ and $O(l'^2 + l'm)$, respectively.

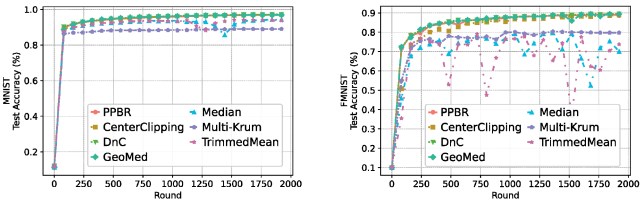
F. Impact of Non-IID Data on Overhead

Since clients and edge servers in PPBR perform privacy-preserving hierarchical filtering and secure aggregation in parallel in each round, and considering the complexity analysis in Table VII, it can be concluded that Non-IID data mainly affects the overhead of PPBR by influencing the number of convergence iterations and the number of trusted clients participating in secure aggregation after filtering. Thus, we discuss the number of iterations (I) required for PPBR to achieve a given accuracy and the number of benign clients ($\sum_i n_i$) retained after filtering in each round under different data heterogeneity settings ($\alpha = 0.1, 0.5, 1.0$) in Fig. 10.

As shown in Fig. 10, for EHFL on the MNIST and F-MNIST datasets, the convergence speed of PPBR and the total number of benign clients retained after filtering in each round are similar under each Non-IID data setting compared to those under the IID data setting. For example, in the setting of $\alpha = 0.5$, PPBR requires approximately 5 additional rounds on MNIST and 200 additional rounds on F-MNIST to achieve the same accuracy as under the IID data setting. Meanwhile, the number of clients retained after filtering in each round under each Non-IID data setting exhibits random fluctuations around the value observed in the IID data setting. Based on the overhead analysis per round presented in Tables V–VII and the preceding overhead analysis, it is evident that although the Non-IID data increase the overhead of PPBR, the impact remains within manageable limits.

G. Convergence Speed Analysis

Fig. 10(a) and (b) show that PPBR is able to converge rapidly to high accuracy with consistently similar convergence speed across all Non-IID data settings under ALIE attacks. Fig. 11 compares the convergence behavior of PPBR and several baselines in a representative setting with $\alpha = 1.0$ and label-flipping attackers ($\beta = 0.3$), showing that PPBR converges at a comparable or faster rate while consistently achieving the highest accuracy. This high convergence speed and accuracy come from PPBR's dual-layer anomalous model filtering steps, which effectively remove malicious models, and its hierarchical model aggregation steps, which preserve the diversity of benign models induced by Non-IID data. Importantly, these robustness-enhancing steps do not compromise training efficiency, demonstrating that PPBR scheme can maintain high accuracy and fast convergence in practical EHFL systems.



(a) MNIST (b) F-MNIST

Fig. 11. Convergence behavior of PPBR and baselines for EHFL on Non-IID data ($\alpha = 1.0$) to label flipping attacks.

VII. Conclusion

In this paper, we have proposed PPBR, a new hybrid approach to achieve resilience to various Byzantine attacks in Non-IID data settings for EHFL in mobile networks, while protecting mobile devices' private information about local models and training data with a single cloud server. Specifically, PPBR skillfully blends CLDP into its enhanced anomalous local models filtering by clustering the sign statistics of local models and performs norms screening and clipping under random perturbation and PLHE to filter out anomalous edge-level models with strong privacy guarantees in EHFL. Furthermore, PPBR protects the private information about local models and edge-level models from being learned by the colluding mobile devices/edge servers based on random perturbing, secret sharing and PLHE techniques, and supports their abrupt dropouts during the average-based hierarchical aggregations. Finally, PPBR provides strong privacy protection for EHFL in mobile networks without sacrificing its robustness to simple and advanced Byzantine attacks,

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

including label flipping, sign flipping attack, Gaussian, ALIE, IPM, and MinMax attacks in Non-IID data settings. PPBR has high accuracy, since its enhanced dual-layer filter can detect a great part of Byzantine mobile devices and remain the information about the benign but diverse models coming from Non-IID training data, and its edge and global aggregations fully 'mix' the remaining momenta of models, making the increased accuracy sufficient to compensate for the losses incurred by CLDP noises. In the future, we can further enhance the efficiency of privacy-preserving and Byzantine-robust EHFL on Non-IID data without privacy and accuracy losses.

REFERENCES

[1] L. Dong, Z. Yang, X. Cai, Y. Zhao, Q. Ma, and X. Miao, "WAVE: Edge-device cooperated real-time object detection for open-air applications," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 4347–4357, Jul. 2023.

[2] X. Sun et al., "EarSSR: Silent speech recognition via earphones," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8493–8507, Aug. 2024.

[3] H. Rao, M. Zeng, X. Zhao, and C. Miao, "A survey of artificial intelligence in gait-based neurodegenerative disease diagnosis," *Neurocomputing*, vol. 626, 2025, Art. no. 129533.

[4] Z. Zhao et al., "Recommender systems in the era of large language models (LLMs)," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 6889–6907, Nov. 2024.

[5] Y. He et al., "RSAM: Byzantine-robust and secure model aggregation in federated learning for Internet of Vehicles using private approximate median," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 6714–6726, May 2024.

[6] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[7] K. Wang, Q. He, F. Chen, H. Jin, and Y. Yang, "Fededge: Accelerating edge-assisted federated learning," in *Proc. Int. Conf. World Wide Web*, 2023, pp. 2895–2904.

[8] J. Xue, S. Sun, M. Liu, Q. Li, and K. Xu, "Enhancing federated learning robustness using locally benignity-assessable Bayesian dropout," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 2464–2479, 2025.

[9] N. Tabassum, K.-H. Chow, X. Wang, W. Zhang, and Y. Wu, "On the efficiency of privacy attacks in federated learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2024, pp. 4226–4235.

[10] S. Li, E. C.-H. Ngai, and T. Voigt, "An experimental study of byzantine-robust aggregation schemes in federated learning," *IEEE Trans. Big Data*, vol. 10, no. 6, pp. 975–988, Dec. 2024.

[11] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–18.

[12] M. Hao, H. Li, G. Xu, H. Chen, and T. Zhang, "Efficient, private and robust federated learning," in *Proc. 37th Annu. Comput. Secur. Appl. Conf.*, 2021, pp. 45–60.

[13] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *Proc. Int. Conf. Data Eng.*, 2022, pp. 965–978.

[14] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "ShieldFL: Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1639–1654, 2022.

[15] S. P. Karimireddy, L. He, and M. Jaggi, "Byzantine-robust learning on heterogeneous datasets via bucketing," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–13.

[16] Y. Liu, C. Chen, L. Lyu, F. Wu, S. Wu, and G. Chen, "Byzantine-robust learning on heterogeneous data via gradient splitting," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 21404–21425.

[17] J. Xu, S. Huang, L. Song, and T. Lan, "Byzantine-robust federated learning through collaborative malicious gradient filtering," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst.*, 2022, pp. 1223–1235.

[18] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for Byzantine robust optimization," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 5311–5319.

[19] S. Rajput, H. Wang, Z. Charles, and D. Papailiopoulos, "Detox: A redundancy-based framework for faster and more robust gradient aggregation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 10320–10330.

[20] X. Lin, J. Wu, J. Li, C. Sang, S. Hu, and M. J. Deen, "Heterogeneous differential-private federated learning: Trading privacy for utility truthfully," *IEEE Trans. Mobile Comput.*, vol. 20, no. 6, pp. 5113–5129, Nov./Dec. 2023.

[21] R. Chen, C. Huang, X. Qin, N. Ma, M. Pan, and X. Shen, "Energy efficient and differentially private federated learning via a piggyback approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2698–2711, Apr. 2024.

[22] Y. He, X. Tan, J. Ni, L. T. Yang, and X. Deng, "Differentially private set intersection for asymmetrical ID alignment," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 3479–3494, 2022.

[23] Y. He, M. Wang, X. Deng, P. Yang, Q. Xue, and L. T. Yang, "Personalized local differential privacy for multi-dimensional range queries over mobile user data," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2025.3568511.

[24] K. Mandal and G. Gong, "PrivFL: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks," in *Proc. ACM SIGSAC Conf. Cloud Comput. Secur. Workshop*, 2022, pp. 57–68.

[25] H. Zhu and Q. Ling, "Bridging differential privacy and Byzantine-robustness via model aggregation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 2427–2433.

[26] L. Zhao, J. Jiang, B. Feng, Q. Wang, C. Shen, and Q. Li, "SEAR: Secure and efficient aggregation for Byzantine-robust federated learning," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 5, pp. 3329–3342, Sep./Oct. 2022.

[27] C. Huang, D. Liu, A. Yang, R. Lu, and X. Shen, "PPRP: Preserving location privacy for range-based positioning in mobile networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 9451–9468, Oct. 2024.

[28] Z. Zhang et al., "LSFL: A lightweight and secure federated learning scheme for edge computing," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 365–379, 2023.

[29] X. Ma, Q. Jiang, M. Shojafar, M. Alazab, S. Kumar, and S. Kumari, "DisBezant: Secure and robust federated learning against Byzantine attack in IoT-enabled MTS," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2492–2502, Feb. 2023.

[30] X. Chen, H. Yu, X. Jia, and X. Yu, "APFed: Anti-poisoning attacks in privacy-preserving heterogeneous federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 5749–5761, 2023.

[31] M. E. Gursoy, A. Tamersoy, S. Truex, W. Wei, and L. Liu, "Secure and utility-aware data collection with condensed local differential privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2365–2378, Sep./Oct. 2021.

[32] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," *IACR Cryptol. ePrint Arch.*, 2012, Art. no. 144.

[33] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1544–1551.

[34] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 691–706.

[35] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. USENIX Secur.*, 2020, pp. 1605–1622.

[36] V. Shejwalkar and A. Houmansadr, "Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2021, pp. 1–18.

[37] Z. Wang et al., "Breaking secure aggregation: Label leakage from aggregated gradients in federated learning," in *Proc. IEEE Conf. Comput. Commun.*, 2024, pp. 151–160.

[38] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.

[39] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[40] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," in *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 1–25, 2017.

[41] B. Zhao, P. Sun, T. Wang, and K. Jiang, "Fedinv: Byzantine-robust federated learning by inversing local model updates," in *Proc. Conf. Assoc. Adv. Artif. Intell.*, 2022, pp. 9171–9179.

[42] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, "Secure single-server aggregation with (poly)logarithmic overhead," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2020, pp. 1253–1269.

[43] Z. Zuo, N. Su, B. Li, and T. Zhang, "Pack: Towards communication-efficient homomorphic encryption in federated learning," in *Proc. ACM Symp. Cloud Comput.*, 2024, pp. 470–486.

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.

[44] Q. Zhang, C. Xin, and H. Wu, "GALA: Greedy computation for linear algebra in privacy-preserved neural networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2021, pp. 1–16.
[45] R. Liu, Y. Cao, H. Chen, R. Guo, and M. Yoshikawa, "Flame: Differentially private federated learning in the shuffle model," in *Proc. Conf. Assoc. Adv. Artif. Intell.*, 2021, pp. 8688–8696.
[46] P. Xu, M. Hu, T. Chen, W. Wang, and H. Jin, "LaF: Lattice-based and communication-efficient federated learning," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2483–2496, 2022.
[47] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Des., Codes Cryptogr.*, vol. 71, no. 1, pp. 57–81, 2014.
[48] M. Yang, I. Tjuawinata, and K. Y. Lam, "K-means clustering with local dχ-privacy for privacy-preserving data analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 2524–2537, 2022.
[49] O. Dorabiala, J. Kutz, and A. Aravkin, "Robust trimmed K-means," *Pattern Recognit. Lett.*, vol. 161, pp. 9–16, 2021.
[50] C. Xie, O. Koyejo, and I. Gupta, "Generalized Byzantine-tolerant SGD," 2018, *arXiv: 1802.10116*.
[51] L. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 638–649.
[52] T. Christofides, "A generalized randomized response technique," *Metrika*, vol. 57, no. 2, pp. 9–16, 2003.
[53] N. Wang et al., "Collecting and analyzing multidimensional data with local differential privacy," in *Proc. Int. Conf. Data Eng.*, 2019, pp. 638–649.

Yuanyuan He (Member, IEEE) received the PhD degree in computer system architecture from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, in 2019. She is currently an associated professor with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. Her research interests are privacy computing and data security in mobile networks, with current focus on Federated Learning, AI on edge, and mobile generative AI.

Jie Zhang (Member, IEEE) received the MS degree from the School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China. His main research interests include federated learning and privacy preservation.

Peng Yang (Member, IEEE) received the BE degree in communication engineering and the PhD degree in information and communication engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2013 and 2018, respectively. He was with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, as a Postdoctoral Fellow from 2018 to 2019. Since 2020, he has been an associate professor with the School of Electronic Information and Communications, HUST. His current research focuses on mobile edge computing, video analytics, virtual reality, and mobile generative AI.

Zhe Sun (Member, IEEE) received the BS degree from the Dalian University of Technology, in 2010, followed by the MS degree from the University of Science and Technology of China, in 2012, and the PhD degree from the University of Chinese Academy of Sciences, in 2019. He is an associate professor with the Cyberspace Institute of Advanced Technology, Guangzhou University, in China. He has authored more than 50 publications in international conferences and journals, focusing on privacy-preserving mechanisms for multiparty data sharing and deep learning.

Xuemin (Sherman) Shen (Fellow, IEEE) is an University professor with the University of Waterloo, Canada. His research interests include network resource management, wireless network security, AI for networks, and future communication systems. Professor Shen is the Technical Program Committee Chair for IEEE Globecom'24, Globecom'16, IEEE Infocom14, and IEEE VTC'10 Fall. He was the editor-in-chief of the *IEEE Internet of Things Journal*, *IEEE Network*, and *IET Communications*. Professor Shen served as the 2022-2023 President of IEEE Communications Society. He is a Fellow of the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada, a Foreign Member of Chinese Academy of Engineering, and an International Fellow of the Engineering Academy of Japan.

Authorized licensed use limited to: University of Waterloo. Downloaded on February 18,2026 at 00:05:26 UTC from IEEE Xplore. Restrictions apply.