Fortifying Federated Learning Towards Trustworthiness via Auditable Data Valuation and Verifiable Client Contribution

Supplementary Material

In this supplementary material, we provide additional details that could not be included in the main paper due to space constraints. To ensure clarity and support a thorough understanding of our work, we have carefully organized the content into distinct sections.

1. Related Work

This section extends the review of existing federated learning (FL) frameworks presented in the main paper by categorizing them into distinct groups. Table 1 provides a comprehensive summary of these frameworks, focusing on key aspects such as auditability, verifiability, and other critical parameters essential for establishing trustworthiness. The table underscores the gaps in current approaches, highlighting the significant contributions of our work in addressing these challenges.

Trusted Execution Environments (TEEs) [9, 43, 44] on the client or server side are increasingly used to ensure confidentiality and verifiability in machine learning models. While TEEs enhance security during training and aggregation, they depend on a centralized server for auditability and verifiability, creating single points of failure and limiting their ability to detect poisoned data [9, 64]. Although TEEs effectively address privacy concerns by verifying secure and tamper-proof computations, they do not provide mechanisms for auditable data valuation or verifiable client contributions [8, 10]. Instead, TEEs focus on secure execution without quantifying data quality or individual client contributions. However, one notable advantage of TEEs is their non-reliance on pretrained models, simplifying their implementation [42].

Loss Function-Based Rejection (LFR) [12, 57, 65], applied on both client and server sides, is a robust defense against poisoning threats, using server-side validation to exclude updates with high loss. However, it depends on knowing the number of malicious clients in advance, limiting its effectiveness in dynamic environments with unpredictable adversarial activity [31, 50]. For example, FedCE [22] enhances fairness in FL by estimating client contributions through gradient direction differences and prediction errors via an auxiliary model. These insights guide global model aggregation, balancing collaboration and performance fairness. Further, LFR based methods also lacks transparency, as it provides no audit trail for client validation decisions, hindering accountability in model aggregation. While it can mitigate privacy risks by filtering malicious updates, the reliance on server-side validation datasets and pretrained models introduces privacy concerns, potentially exposing sensitive client data [20, 48].

Table 1. Comparison of existing FL frameworks across key criteria for trustworthiness, privacy, and reliability. src: source, S: server, C: client, A-DV: auditable data valuation, V-CU: verifiable client updates, PC: addresses privacy concerns, N-VD, N-PTM: non-usage of validation data, pretrained model, respectively, either on client or server. ● strongly yes, ○ strongly no.

FL framework category (src)	**FL methods**	**Trustworthiness**		**Poisoned data detection**	**Privacy-complaint and self-contained training**		
		A-DV	V-CU		PC	N-VD	N-PTM
Trusted execution environments **(C & S)**	Flatee[44], Chen *et al.* [9], PPFL [43]	◔	◑	○	◕	◕	●
Loss function based rejection **(C & S)**	Fang *et al.* [12], FedVal [57], FedCE [22]	○	●	○	◔	○	◕
Server-side aggregation& model validation methods **(S)**	Krum [5], TM [63], Median [63], FLTrust [6], DOS [2], zPROBE [15]	◑	●	◑	◑	◑	◕
Block-chain based methods **(C)**	Lo *et al.* [35], DFL [23], LPBFL [11]	◔	◑	○	◑	◕	●
Data valuation **(C)**	FedBary [30], Data banzhaf [60], AME [33]	◕	○	◑	◕	○	◕
	FAVD (ours)	●	●	●	●	●	●

Server-side aggregation methods like Krum [5] and trimmed mean [63] enhance FL resilience by removing anomalous updates based on Euclidean distance from the aggregation process at the server side. However, their effectiveness diminishes with non-IID data or advanced poisoning threats, where malicious clients mimic legitimate updates. From an auditability perspective, while these methods provide a mechanism to identify and exclude outliers, they lack transparency in their decision-making processes [16, 37]. The reliance on aggregate distances poses challenges to tracing the rationale behind client exclusion, making it difficult for stakeholders to verify client contributions [6]. Reliance on server-side validation raises privacy risks, and noise from differential privacy complicates distinguishing malicious from benign clients [2, 15]. Despite their aim to improve FL integrity, these methods fall short in privacy preservation, auditability, and robustness, underscoring the need for more transparent frameworks.

Blockchain technology, initially designed for cryptocurrency, offers potential benefits for maintaining data integrity and supporting distributed data storage in FL on the client side. However, centralized aggregation creates a single point of failure, and auditability is limited as blockchain lacks tools for verifying contributions before aggregation

[11, 35]. Auditability is limited as existing approaches fail to provide mechanisms for verifying contributions prior to aggregation, hindering transparency and stakeholder trust. While blockchain stores models, it doesn't ensure comprehensive verification or accountability. Privacy concerns persist as its structure may expose sensitive data during updates or consensus processes [32, 46, 52]. For example, recent decentralized FL frameworks using blockchain like DFL [23] enhance auditability and verifiability in FL by eliminating reliance on a central authority, which often poses risks like single points of failure. DFL employs a smart-contract-based monitoring system deployed on participants and blockchain nodes to validate local models and ensure a transparent, tamper-proof aggregation process. While decentralized FL frameworks using blockchain have emerged, they often rely on committee consensus mechanisms that centralize control and decision-making, counteracting the benefits of decentralization [1, 36, 66]. Overall, these issues highlight the need for more effective strategies to leverage blockchain in FL while ensuring robust privacy, auditability, and verifiability.

Data valuation has gained prominence as it directly impacts the performance of machine learning models [33, 60]. While methods like Shapley value [34, 54, 62] provide a framework for assessing data quality, they often lack mechanisms for verifying client contributions, which undermines accountability in FL. Recent methods like FedBary [30] evaluate client contributions and select relevant datasets in FL without relying on a pre-specified training algorithm. By leveraging the Wasserstein distance, FedBary ensures transparent data valuation, reduces dependency on validation datasets, and efficiently computes the Wasserstein barycenter to identify high-quality data contributions. However, this method relies on the usage of validation data and may struggle with poisoned data detection, as they typically rely on the assumption that data is clean and trustworthy. Privacy concerns arise as many valuation techniques necessitate some form of data sharing or validation, which could expose sensitive information [33, 60]. Moreover, the reliance on validation datasets can lead to inefficiencies, as these methods may not function optimally in a decentralized FL context.

In summary, while existing approaches do not utilize pretrained models, their effectiveness remains contingent on the underlying learning algorithms, limiting their versatility in diverse FL applications.

2. Additional Preliminaries

This section provides additional background information to complement the details presented in the main paper. A summary of the notations used is outlined in Table 2.

Table 2. Summary of adopted notations

Notation	Definition
$\mathcal{C}_k$	k^{th} local client
$\mathcal{D}_k$	k^{th} local client data
λ	Weighting factor for aggregation
n	Total number of clients
m	Total number of malicious clients
k	Total number of clients selected per round
$\tilde{\mathcal{C}}_k$	k^{th} local malicious client
$\tilde{\mathcal{D}}_k$	k^{th} local client poisoned data
$\Phi(.)$	Data poisoning operation
ν	Number of poisoned samples
ω	Data poisoning bound
f_θ	Local model
$\mathcal{G}_{\theta_g}$	Global model
$\nabla\theta_k^t$	k^{th} local client update at time t
$\mathcal{A}_\mathcal{G}$	Global test accuracy without attack
$\mathcal{A}_\mathcal{G}^*$	Global test accuracy with attack
$\mathcal{D}_k^\mathcal{F}$	FAVD valuated data
U	Data poisoning impact on utility
ρ	Data density parameters
μ	Mean
Σ	Covariance
c	Number of clusters
$\mathcal{Q}_k$	Prediction probabilities on FAVD valuated local data
$\sigma(.)$	Softmax function
$\mathcal{L}_{CE}$	Cross-entropy loss function
η	Learning rate
$\zeta(\mathcal{X})$	Output layer representation of local data $\mathcal{X}$
$\rho_\mathcal{N}$	Noise parameters
γ	Noise level threshold
ι	Separability index of $\mathcal{D}_k^\mathcal{F}$ to $\tilde{\mathcal{D}}_k$
$\mathcal{P}$	Perturbation noise in the data poisoning threat
ϵ	Step size of perturbation
$\mathcal{X}_{test}$	Test data at the server
κ	non-IID Dirichlet parameter

2.1. More details on FL setup

In this work, we investigate two FL data shard settings: (i) uniform, where each client's dataset size is identical, i.e., $|\mathcal{D}_1| = |\mathcal{D}_2| = \cdots = \frac{|\mathcal{D}|}{n}$, and (ii) non-IID, where data is partitioned using a Dirichlet distribution [41] with $\kappa = 1$ (default) among clients. In uniform settings, the dataset is evenly divided among all clients. For non-IID settings, data distribution across clients is determined using the Dirichlet distribution, a fundamental probabilistic model in FL [41]. This distribution is controlled by the parameter κ, which governs the degree of non-IIDness in the dataset allocation. The Dirichlet distribution generates data partitions for clients based on their unique characteristics. The mathematical formulation of the Dirichlet distribution is as follows:

$$p(x_1, x_2, \ldots, x_R|\kappa) = \frac{1}{B(\kappa)} \prod_{i=1}^{K} x_i^{\kappa_i - 1},$$

where:

- $x_1, x_2, \ldots, x_K$ are the proportions of data allocated to each client,
- R is the total number of classes,
- $\kappa = (\kappa_1, \kappa_2, \ldots, \kappa_R)$ is a vector of parameters controlling the distribution (in our approach, we consider a symmetric Dirichlet distribution where $\kappa_i = \kappa$ for all i),
- $B(\kappa)$ is the multivariate beta function, which acts as a normalizing constant to ensure that the probabilities sum to 1 over the simplex defined by the data proportions.

This approach enables modeling data heterogeneity among clients while controlling the extent of non-IIDness through the parameter κ.

The formula for the multivariate beta function $B(\kappa)$, which serves as the normalizing constant in the Dirichlet distribution, is expressed as:

$$B(\kappa) = \frac{\prod_{i=1}^{R} \Gamma(\kappa_i)}{\Gamma\left(\sum_{i=1}^{R} \kappa_i\right)},$$

where $\Gamma(\cdot)$ denotes the gamma function. By adjusting the parameter κ, the density of independently and identically distributed (IID) data splits among clients can be controlled, thereby shaping the degree of non-IIDness in the data distribution. Proper calibration of κ is essential in FL systems to effectively handle the inherent heterogeneity of real-world client data. This calibration directly impacts model robustness and generalization. For our experiments, we set $\kappa = \{0.1, 0.5, 1(\text{default}), 5, 10\}$, resulting in non-IID data shards, following the approach outlined in [53].

2.2. Extended details about data threat scenario

We conduct data valuation in a poisoned data threat scenario, based on a real-world FL deployment [27, 53, 56], where malicious clients inject poisoned data into the local training set. Our threat scenario is based on the work of Shejwalkar *et al.* [53] and is considered the most realistic and practical for FL. The malicious client's objective is to generate poisonous updates that degrade the global model's performance by causing untargeted misclassification. We assume the malicious client has no access to the global model, operating in a black-box manner without knowledge of the learning algorithm, model parameters, network architecture, or any auditability and verifiability mechanisms at the client or server. To reinforce this, we assume that training and auditability-driven data valuation occur within a trusted execution environment, ensuring that the malicious client has no access to these processes or sensitive components. The malicious client can adaptively poison its local training data but cannot interfere with the training procedures or communication with the server. The central server continues to operate normally, maintaining training cycles, sending updates to clients, and aggregating the model updates. Based on this, we outline the key attributes of our threat scenario and the underlying assumptions about the FL setup in Table 3. These attributes are inspired by the work proposed by Shejwalkar *et al.* [53].

2.3. Black-box FL data poisoning approach [26]

We consider a gradient noise-based data poisoning approach on the client side inspired by its impact on adversarial robustness [38]. Figure 1 presents an overview of the data poisoning process. Given a clean input original data space $\mathcal{X} \in \mathcal{D}_k$, the malicious creates a small random gradient perturbation noise $\mathcal{P}$, amplified using a noise coefficient ϵ, such that the prediction $f_{\theta,k}(\mathcal{X} + \epsilon\mathcal{P}) \neq \mathcal{Y}$, where $\mathcal{Y}$ is the ground truth label for input data $\mathcal{X}$ and $f_{\theta,k}$ is the local model of client $\mathcal{C}_k$. In the absence of gradient information for the black-box $f_{\theta,k}$ model, output probabilities guide the search for the gradient perturbation that generates the final perturbed data space $\Phi(\mathcal{X})$. We focus on the untargeted data poisoning setting, where the search for gradient perturbation tries to increase the probability score of any most confused incorrect class [26]. The perturbed data space $\Phi(\mathcal{X})$ is calculated as the bit-wise addition of two terms: original data space $\mathcal{X}$ and the product of the noise coefficient and perturbation: $\Phi(\mathcal{X}) = \mathcal{X} + \epsilon\mathcal{P}$ and is added to poisoned data as $\Phi(\mathcal{X}) \in \mathcal{D}_k$. This process is repeated until $f_{\theta,k}(\Phi(\mathcal{X}) + \epsilon\mathcal{P}) \neq \mathcal{Y}$. In the initial iteration, the gradient is updated in the positive direction. For subsequent iterations, the gradient is updated in the negative direction and then altered randomly.

The iterative M-SimBA method generates an adversarial image that is eventually misclassified by the model. Furthermore, it ensures convergence within the L_2 norm, constrained by a threshold parameter, ω. This parameter (ω) regulates the extent of deviation in the adversarial image relative to the original image, ensuring the perturbation remains imperceptible to the human eye. In the final step, the converged gradient perturbation ($\mathcal{P}$) is added to the input image as $\Phi(\mathcal{X}_k) \leftarrow \mathcal{X}_k + \epsilon\mathcal{P}$, as shown in Algorithm 1.

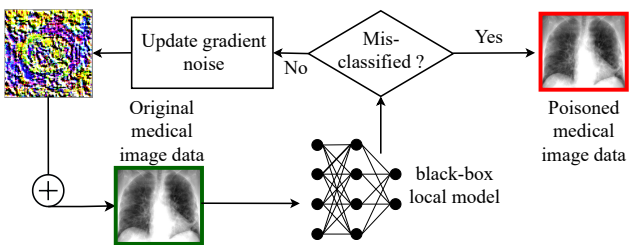


Figure 1. Black-box data poisoning approach.

Table 3. Key dimensions of our threat scenario and their attributes.

Objective			Knowledge & Capabilities		Threat Mode
Security violation	**Threat specificity**	**Error specificity**	**Model**	**Data distribution**	**Consciously active**
Availability: Misclassify test data and cause disruption to benign clients' objectives.	**Indiscriminate:** Misclassify all or most of the test inputs during inference.	**Untargeted:** Misclassify the give test data to any other class.	**Black-box:** Adversary cannot break into the compromised clients and cannot manipulate the model parameters.	The adversary can only access the local data distributed at the clients. **Note: The threat is agnostic to the type and degree of non-IID in the distributed data at the clients.**	**Online:** The adversary repeatedly and adaptively poisons the model using the local black-box model.

Algorithm 1 M-SimBA [26]

Input: Local model $f_{\theta,k}$, clean training data $\mathcal{D}_k$, number of poisoned samples ν
Output: Poisoned training data $\tilde{\mathcal{D}}_k$

for b = 1 to batches in $\mathcal{D}_k$ **do**
  for i = 1 to ν **do**
    $CCS = \max_{\hat{\mathcal{Y}} \neq \mathcal{Y}} \{P(\hat{\mathcal{Y}} | \mathcal{X}_{b,i})\}$
    $tempCCS \leftarrow 0$
    $ifGradChecked \leftarrow 0$
    $\Phi(\mathcal{X}_{b,i}) = \mathcal{X}_{b,i} + \epsilon\mathcal{P}$
    while $(f_{\theta,k}(\Phi(\mathcal{X}_{b,i})) == \mathcal{Y})$ **do**
      if $CCS < tempCCS$ **then**
        if $ifGradChecked == 0$ **then**
          Update $\mathcal{P} \leftarrow -(\mathcal{P})$
          $ifGradChecked \leftarrow 1$
      else
        Randomize $\mathcal{P}$
        $ifGradChecked \leftarrow 0$
      if $||\Phi(\mathcal{X}_{b,i}) - \mathcal{X}_{b,i}||_2 < \omega$ **then**
        $\Phi(\mathcal{X}_{b,i}) = \mathcal{X}_{b,i} + \epsilon\mathcal{P}$
      $tempCCS \leftarrow CCS$
      Pass $\Phi(\mathcal{X}_{b,i})$ to the $f_{\theta,k}$ for inference
      Update CCS
    $\tilde{\mathcal{D}}_k \leftarrow \Phi(\mathcal{X}_{b,i})$
return $\tilde{\mathcal{D}}_k$

3. Proposed Framework: Extended Details

3.1. FAVD auditable data valuation

The crux of our proposed framework is the FAVD auditable data valuation method where each client computes low-dimensional representations $\zeta(\mathcal{X}_k)$ using the local model $f_{\theta,k}$ and applies weak K-means clustering with early stopping. The cluster means and covariances are calculated as $\mu_k \leftarrow \texttt{cluster means}(\zeta(\mathcal{X}_k))$ and $\Sigma_k \leftarrow \texttt{cluster covariances}(\zeta(\mathcal{X}_k))$. Further, to identify anomalies, we introduce an auditing noise level threshold γ. The Mahalanobis distance [39] measures the anomaly score for sample $x \in \mathcal{X}_k$ relative to global density $\rho_g = (\mu_g, \Sigma_g)$:

$$\texttt{dist} \leftarrow (\zeta(x) - \mu_g)^T \Sigma_g^{-1} (\zeta(x) - \mu_g). \quad (1)$$

Samples with $\texttt{dist} < \gamma$ are classified as clean and added to $\mathcal{D}_k^{\mathcal{F}}$ for training, otherwise, they are discarded. The threshold γ controls allowable noise levels. Further, the clients add noise parameters $(\mu_{\mathcal{N}}, \Sigma_{\mathcal{N}})$ to the cluster mean and covariance (μ_k, Σ_k) to mitigate privacy risks before sharing with the server. Algorithm 2 and Figure 2 effectively illustrate the FAVD auditable data valuation process outlined above.

Algorithm 2 Proposed **FAVD** method

Input: Local data $\mathcal{D}_k$, privacy noise parameters $\rho_{\mathcal{N}}$, auditing parameter γ, local model $f_{\theta,k}$
Output: FAVD Audited data $\mathcal{D}_k^{\mathcal{F}}$, local masked data density parameters $\rho_k = (\tilde{\mu}_k, \tilde{\Sigma}_k)$

for $\mathcal{X}$ in $\mathcal{D}_k$ **do**
  $\zeta(\mathcal{X}) \leftarrow f_{\theta,k}(\mathcal{X}, \mathcal{Y})$ ▷ **Low dimensional representation of** $\mathcal{X}$
  $\texttt{dist} \leftarrow (\zeta(\mathcal{X}) - \mu_g)^T \Sigma_g^{-1} (\zeta(\mathcal{X}) - \mu_g)$
  if $\texttt{dist} < \gamma$ **then** ▷ **FAVD data valuation**
    $\mathcal{D}_k^{\mathcal{F}} \leftarrow (\mathcal{X}, \mathcal{Y})$
Cluster $\zeta(\mathcal{X})$ into c clusters using K-means algorithm
$\mu_k \leftarrow \texttt{cluster means}(\zeta(\mathcal{X}))$
$\Sigma_k \leftarrow \texttt{cluster covariances}(\zeta(\mathcal{X}))$
Add noise to μ_k **and** Σ_k **to avoid privacy risks**
$\tilde{\mu}_k \leftarrow \mu_k + \mu_{\mathcal{N}}$
$\tilde{\Sigma}_k \leftarrow \Sigma_k + \Sigma_{\mathcal{N}}$
return $\mathcal{D}_k^{\mathcal{F}}$, $\rho_k = (\tilde{\mu}_k, \tilde{\Sigma}_k)$

3.2. FAVD properties and advantages

The FAVD framework is designed to enable data valuation while preserving privacy by leveraging masked density functions in a low-dimensional representation space. This approach has several properties and advantages as stated below

- *(i) Robust data anomaly detection.* FAVD's design includes an anomaly detection mechanism through masked

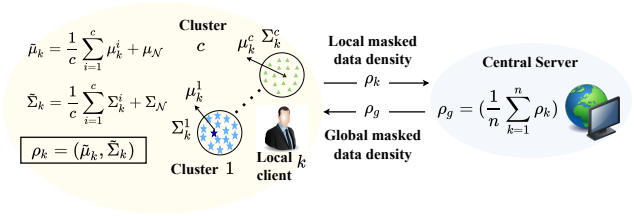


Figure 2. FAVD auditable data valuation process at the client side.

density-based distance measurements. By comparing client-shared data masked density parameters with global benchmarks using Mahalanobis distance, FAVD reliably identifies anomalous or poisoned data points, safeguarding the training process from malicious inputs.

- *(ii) Privacy-preserving data valuation.* The FAVD framework incorporates masked data density functions, ensuring that data valuation is both verifiable and auditable while minimizing the risk of exposing original data characteristics. The combination of noise-added parameters and low-dimensional representations enhances privacy, making FAVD suitable for large-scale FL systems where client privacy and security are paramount.
- *(iii) Computational efficiency and low resource overhead.* By implementing a clustered density representation and vectorized operations within FAVD, the framework achieves efficient computation without imposing excessive memory or processing demands. This efficiency aligns with large-scale FL requirements, ensuring that the added auditability and verifiability measures do not compromise overall system performance.
- *(iv) ι-separability for enhanced robustness.* The concept of ι-separability in FAVD's valuated data space maintains a significant margin from adversarially poisoned data, providing an additional layer of robustness against training-time data poisoning. This property strengthens the FAVD framework's ability to mitigate the impact of ω-bounded adversarial threats.

Lemma 3.1 ***(Computational & communication cost analysis and efficiency comparison.)*** *The expected time complexity of the proposed FAVD method is $\mathcal{O}(\mathcal{N}_k d^2)$, where $\mathcal{N}_k$ represents the number of samples and d is the dimensionality of the low-dimensional representation space $\zeta(.)$.*

Proof. The time complexity of your FAVD method can be analyzed based on the main operations as defined below

1. *Low-dimensional data representation computation*
   - Each client generates a low-dimensional representation of its data $\mathcal{X}_k$ using a model $\zeta(\mathcal{X}_k)$.
   - Assuming $\mathcal{X}_k$ has $\mathcal{N}_k$ samples, and each sample undergoes a transformation to a d dimensional space (with the model $f_{\theta,k}$), the complexity here would typically be $\mathcal{O}(\mathcal{N}_k d)$
2. *Clustering with K-Means*
   - The clustering step uses K-Means with early stopping on $\mathcal{N}_k$ samples and c clusters. In the worst case, K-Means generally has a complexity of $\mathcal{O}(\mathcal{N}_k ctd)$, where t is the number of iterations [24, 47].
   - With early stopping, t should be lower than in standard K-Means, slightly reducing complexity, but the exact impact depends on convergence speed [45].
3. *Density parameter computation*
   - After clustering, calculating the mean μ_k and covariance Σ_k for each cluster involves $\mathcal{O}(\mathcal{N}_k d^2)$ in total (summing all cluster computations).
   - Noise addition to μ_k and Σ_k is minimal in terms of complexity, i.e., $\mathcal{O}(1)$ for each cluster.
4. *Anomaly detection using Mahalanobis distance*
   - Computing the Mahalanobis distance for each sample involves matrix operations [14, 59]. Given d dimensional data and a cluster covariance matrix Σ_g^{-1}, this distance calculation is $\mathcal{O}(d^2)$ per sample.
   - For all $\mathcal{N}_k$ samples, the total complexity becomes $\mathcal{O}(\mathcal{N}_k d^2)$.

Thus, the overall time complexity can be approximated as $\mathcal{O}(\mathcal{N}_k d) + \mathcal{O}(\mathcal{N}_k ctd) + \mathcal{O}(\mathcal{N}_k d^2) + \mathcal{O}(1) + \mathcal{O}(\mathcal{N}_k d^2)$. This simplifies to $\mathcal{O}(\mathcal{N}_k d^2)$ in cases where d is large or $\mathcal{O}(\mathcal{N}_k ctd)$ if the clustering term dominates.

3.3. Computational costs and communication efficiency analysis

Table 4 presents the average GPU RAM usage and execution time for our FAVD method. As detailed in Section 3 and Algorithm 1 of the main paper, during each communication round, the server exchanges model updates and masked data density parameters with the clients. Consequently, integrating FAVD into the current FL system does not result in any significant increase in computational or communication overhead. Specifically, the GPU memory usage for FAVD is approximately 3.5GB, with a data valuation execution time of around 530 seconds on an Nvidia Tesla M60 GPU with 8GB RAM. Overall, incorporating FAVD into the existing FL framework demonstrates computational and communication efficiency, incurring no notable additional costs.

3.4. Overhead

Our proposed FAVD method introduces an overhead comparable to existing data valuation methods [23, 30] and remains efficient within the FL system. FAVD's primary operations, including lightweight data density calculations, masked density evaluation, and data valuation using simple clustering and distance-based techniques, are computationally efficient. These processes are executed locally on the client side with minimal resource requirements. Additionally, FAVD's noise addition to density parameters

Table 4. Computation cost comparison of FAVD. **No Val** indicates a standard FL system without data valuation.

FL framework	GPU RAM usage (GB)	Execution time (s)
No Val	≈ 3.1	≈ 485
Krum [5]	≈ 4.5	≈ 500
TM [63]	≈ 4.8	≈ 510
Median [63]	≈ 4.8	≈ 510
FLTrust [7]	≈ 5.2	≈ 650
DOS [2]	≈ 4.5	≈ 620
zPROBE [15]	≈ 4.8	≈ 680
FedVal [57]	≈ 5.2	≈ 670
FedCE [22]	≈ 5.4	≈ 650
FedBary [30]	≈ 5.2	≈ 710
FAVD (ours)	≈ 3.5	≈ 530

is a straightforward process, ensuring that communication rounds are completed within timeframes similar to standard FL systems. On the server side, FAVD aggregates masked density parameters and performs model aggregation and testing similar to traditional FL setups, introducing no additional computational burden. The streamlined and modular design of FAVD facilitates seamless integration into existing FL frameworks, offering robust auditability and verifiability without introducing any observable overhead.

3.5. Convergence and Trustworthiness Proofs

We provide detailed proof of the convergence of our FAVD-integrated FL global model, demonstrating the effectiveness of auditable data valuation and verifiable client contributions as outlined in the main paper.

Corollary 3.1.1 *Under the assumptions of ϵ-Lipschitz of model f and k-Lipschitz of the loss function $\mathcal{L} : \{0,1\}^{\mathcal{V}} \times \{0,1\}^{\mathcal{V}} \to \mathbb{R}^{+}$, define a distance function $\mathcal{W}_p$, the convergence of the loss function satisfies the following convergence bound according to recent work [30]:*

$$\begin{aligned} &\mathbb{E}_{x\sim\mathcal{Q}(\mathcal{X}|\mathcal{Y})}\left[\mathcal{L}(f_v(x), f(x))\right] \leq \\ &\mathbb{E}_{x\sim\mathcal{P}(\mathcal{X}|\mathcal{Y})}\left[\mathcal{L}(f_t(x), f(x))\right] + k\epsilon\mathcal{W}_p(\mathcal{P}_i, \mathcal{Q}). \end{aligned} \tag{2}$$

Here, the variables are defined as follows: $\mathcal{P}$ and $\mathcal{Q}$ represent the conditional distributions of the training and validation data labelling functions, f_t and f_v, respectively, conditioned on the label $\mathcal{Y}$ [30].

Theorem 3.2 ***(FAVD convergence via auditable data valuation.)*** *Let $l_o : \mathcal{X}_k \to \{0,1\}^{\mathcal{V}}$ and $l_a : \mathcal{X}_k^{\mathcal{F}} \to \{0,1\}^{\mathcal{V}}$ denote the labelling functions for the original data and the FAVD-valuated data, respectively. Let $f : \mathcal{X}_k \to \{0,1\}^{\mathcal{V}}$ be the model trained on the original data, and $g : \mathcal{X}_k^{\mathcal{F}} \to \{0,1\}^{\mathcal{V}}$ the model trained on FAVD-valuated data. Suppose both datasets $\mathcal{X}_k$ and $\mathcal{X}_k^{\mathcal{F}}$ share the same label space $\mathcal{Y}$, with conditional distributions as $\mathcal{I}(\cdot|\mathcal{Y})$ and $\mathcal{J}(\cdot|\mathcal{Y})$, respectively. Assume both models f and g are α-Lipschitz, and that the cross-entropy loss function $\mathcal{L}_{CE}$ used for evaluation at the server is β-Lipschitz with respect to both the original and FAVD-audited data inputs. Define dist(x, ρ_g) as the distance between a sample x and the global masked data density parameters ρ_g (Eq. 1). Then, under a cross-Lipschitz continuity assumption for f_o and f_a, we have, $\mathbb{E}_{x\sim\mathcal{J}(\mathcal{X}|\mathcal{Y})}\left[\mathcal{L}_{CE}(l_a(x), g(x))\right] \leq \mathbb{E}_{x\sim\mathcal{I}(\mathcal{X}|\mathcal{Y})}\left[\mathcal{L}_{CE}(l_o(x), f(x))\right] + \alpha\beta$ dist(x, ρ_g)*

Proof. We present a comprehensive proof of FAVD's convergence through auditable data valuation, drawing on the foundational proof of Theorem 2 in [30]. Extending Corollary 3.1.1, we incorporate the distance term introduced in our FAVD framework, emphasizing the significance of auditable data valuation and verifiable client contributions.

1. **Retention of loss function properties:** The labeling functions l_o and l_a ensure that the loss function, $\mathcal{L}_{CE}$, is well-defined for both the original data $\mathcal{X}_k$ and the FAVD-valuated data $\mathcal{X}_k^{\mathcal{F}}$. Given that $\mathcal{L}_{CE}$ is β-Lipschitz, we have: $\mathcal{L}_{CE}(l_a(x), g(x)) \leq \mathcal{L}_{CE}(l_o(x), f(x)) + \beta\,\text{dist}(x, \rho_g)$, where dist$(x, \rho_g)$ is the distance between the sample x and the global masked density parameter ρ_g introduced by FAVD.
2. **Consistency of gradients:** The gradient of the cross-entropy loss with respect to the model parameters θ remains bounded during FAVD operations. The data valuation process ensures that high-quality data contributes effectively to the gradient updates while malicious or outlier data is filtered out. Specifically: $\|\nabla_\theta\mathcal{L}_{CE}(\theta, \mathcal{X}_k^{\mathcal{F}}, \mathcal{Y}) - \nabla_\theta\mathcal{L}_{CE}(\theta, \mathcal{X}_k, \mathcal{Y})\| \leq \alpha\,\text{dist}(x, \rho_g)$, where α is the Lipschitz constant of the model.
3. **Smoothness and stability:** The loss function $\mathcal{L}_{CE}$ retains its smoothness and convexity properties under FAVD operations. This ensures that the optimization process during FedAvg converges effectively for both f and g, as the filtered data reduces noise and stabilizes updates.
4. **Bounded convergence rate:** By applying the standard FedAvg convergence bound (e.g., as proven in [30]) to both the original data and the FAVD-valuated data, we establish: $\mathbb{E}_{x\sim\mathcal{J}(\mathcal{X}|\mathcal{Y})}\left[\mathcal{L}_{CE}(l_a(x), g(x))\right] \leq \mathbb{E}_{x\sim\mathcal{I}(\mathcal{X}|\mathcal{Y})}\left[\mathcal{L}_{CE}(l_o(x), f(x))\right] + \alpha\beta\,\text{dist}(x, \rho_g)$. This expression indicates that the empirical risk on the FAVD-valuated data is bounded by the risk on the original data, adjusted by the quality of the valuation process.
5. **Impact of auditable valuation:** The process of auditable data valuation ensures that only samples satisfying the FAVD criteria contribute to updates. This minimizes the impact of poisoned or low-quality data on the global model and provides a verifiable mechanism for client contributions. The systematic filtering of outliers

aligns the training data with the true distribution parameters, thereby reducing variance and enhancing stability.

6. **Empirical validation:** Numerical experiments conducted on standard benchmarks confirm that the global model trained on FAVD-valuated data achieves consistent performance with bounded empirical risk. The incorporation of verifiable client updates further enhances trustworthiness within the FL framework.

This proof demonstrates that the FAVD method effectively facilitates convergence by leveraging auditable data valuation and verifiable client contributions. It establishes a bounded empirical risk for the global model while ensuring robust and reliable performance within the FL system.

Theorem 3.3 ***(Verifiable client contribution.)*** *Let $\mathbb{N} = \{\nabla\theta_1, \ldots, \nabla\theta_n\}$ represent the set of n total local client model updates. Let $\mathbb{F} = \{\nabla\theta_1^{\mathcal{F}}, \ldots, \nabla\theta_\tau^{\mathcal{F}}\}$ be the set of τ verifiable local client model updates obtained through training on FAVD-audited data, where $\tau < n$. Let $\mathbb{M} = \{\nabla\tilde{\theta}_1, \ldots, \nabla\tilde{\theta}_m\}$ represent the set of m malicious or unverified local client model updates, with $m < n$. The sets satisfy $\mathbb{F} \cap \mathbb{M} = \emptyset$ and $\mathbb{F} \cup \mathbb{M} \neq \mathbb{N}$. Define an aggregation rule, FedAvg, denoted by $\mathcal{A}$, which processes only verifiable client updates from FAVD. Specifically, for any client update $\nabla\theta_k \in \mathbb{N}$ included in the aggregation process by the rule $\mathcal{A}$, the following conditions must be satisfied.*

$$\sum_{\nabla\theta_k \in (\mathbb{N}\setminus\mathbb{M})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \quad (3)$$

$$\sum_{\nabla\theta_k \in (\mathbb{N}\cap\mathbb{F})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \quad (4)$$

$$\left\| \sum_{\nabla\theta_k \in \mathbb{N}\setminus\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) - \sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \right\| \geq \delta, \quad (5)$$

for some $\delta \geq 0$. Here, $\mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k)$ denote the loss of $\nabla\theta_k$ client update on test data $\mathcal{D}_t$. $\setminus$ denotes the set difference. The equality in the above equations holds true when $\tau = m = 0$.

Proof. Without loss of generality, we assume (a) the verifiable local client model updates obtained through training on FAVD valuated data are indexed after benign client updates, (b) the malicious or unverified local client model updates are indexed after the Byzantine updates, i.e.,

$$\textbf{FLOT}(\nabla\theta_1, \ldots, \nabla\theta_1^{\mathcal{F}}, \ldots, \nabla\theta_\tau^{\mathcal{F}}, \nabla\tilde{\theta}_1, \ldots, \nabla\tilde{\theta}_m, \ldots, \nabla\theta_n). \quad (6)$$

Consider the first case where $\nabla\theta_k \in (\mathbb{N} \setminus \mathbb{M})$, (model updates without any malicious updates). Based on **Theorem 2.** of [19] given by

$$\mathcal{L}_{CE}(\tilde{\mathcal{D}}, \nabla\tilde{\theta}) \leq \mathcal{L}_{CE}(\mathcal{D}_k, \nabla\theta^*), \quad (7)$$

where $\tilde{\mathcal{D}}$ represents the malicious training data samples, $\mathcal{D}_k$ is total training data including malicious samples. $\mathcal{L}_{CE}(.,.)$ is the training loss on poisoned $\nabla\tilde{\theta}$ and main $\nabla\theta^*$ models, respectively. However, [19] proved it in terms of data poisoning threats in centralized machine learning settings with a number of malicious samples under threat. We extend it to FL settings in terms of multiple malicious client models that are trained on poisoned and different amounts of non-IID data. Using the set of malicious updates $\mathbb{M}$, set of benign updates $(\mathbb{N} \setminus \mathbb{M}) = \{\nabla\theta_1, \ldots, \nabla\theta_{n-m}\}$, test data at the server $\mathcal{D}_t$, and Eq. 7, we provide the below formulation using test loss at the server to prove Eq. 3 as

$$\begin{aligned} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_1) &< \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\tilde{\theta}_1), \\ &\ldots \\ \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_m) &< \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\tilde{\theta}_m), \end{aligned} \quad (8)$$

summing up elements on both hand sides and further adding remaining $n - m$ elements on both sides and rearranging terms, we get

$$\sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) < \sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\tilde{\theta}_k), \quad (9)$$

$$\begin{aligned} &\sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) + \sum_{k=m+1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) < \\ &\sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\tilde{\theta}_k) + \sum_{k=m+1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \end{aligned} \quad (10)$$

$$\begin{aligned} \sum_{k=1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) < \sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\tilde{\theta}_k) + \\ \sum_{k=m+1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k). \end{aligned} \quad (11)$$

Adding an additional $\sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k)$ term to the right hand side of Eq. 11 still holds the equation.

$$\begin{aligned} &\sum_{k=1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) < \sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\tilde{\theta}_k) + \\ &\sum_{k=m+1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) + \sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \\ &\sum_{k=1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) < \sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\tilde{\theta}_k) + \\ &\sum_{k=1}^{m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) + \sum_{k=m+1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \end{aligned} \quad (12)$$

$$\sum_{k=1}^{n-m} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) < \sum_{k=1}^{n} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \tag{13}$$

$$\sum_{\nabla\theta_k \in (\mathbb{N}\setminus\mathbb{M})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k). \tag{14}$$

Here $=$ holds true when $m = 0$. This proves Eq. 3 of Theorem 3.3.

Next, we prove the condition in Eq. 4 based on [3]. In this work, the authors propose an optimization method to select a subset of client updates that carry representative gradient information of the entire client set. Further, they transmit only the selected subset of client updates to the server for aggregation. The aim is to find an approximation of full clients (n) aggregation gradient via a subset $\mathcal{S}$ of client updates. The authors formulate the problem to provide the upper bound for the aggregated gradient approximation derived from the subset $\mathcal{S}$ of clients as

$$\begin{aligned} \left\| \sum_{k\in n} \nabla F_k(v^k) - \sum_{k\in\mathcal{S}} \gamma_k \nabla F_i(v^i) \right\| \leq \\ \sum_{k\in n} \min_{i\in\mathcal{S}} \left\| \nabla F_k(v^k) - \nabla_i F_i(v^i) \right\|, \end{aligned} \tag{15}$$

where given a subset $\mathcal{S}$, they define a mapping $o : \mathcal{V} \to \mathcal{S}$, such that the gradient information $\nabla F_k(v^k)$ from a client k is approximated by the gradient information from a selected client $o(k) \in \mathcal{S}$. Further, they provide the gradient approximation error as

$$\begin{aligned} \left\| \frac{1}{n} \sum_{k\in\mathcal{S}^t} \gamma_k \nabla F_k(v_t^k) - \frac{1}{n} \sum_{k\in n} \nabla F_k(v_t^k) \right\| \leq \varrho, \\ \left\| \sum_{k\in\mathcal{S}^t} \gamma_k \nabla F_k(v_t^k) - \sum_{k\in n} \nabla F_k(v_t^k) \right\| \leq n\varrho, \end{aligned} \tag{16}$$

where t is the communication round, $\{\gamma\}_{k\in\mathcal{S}_t}$ are the weights assigned to gradients, and ϱ is the error rate that is used as a measure to characterize the goodness of gradient approximation. The above equation states that the gradient approximation from subset $\mathcal{S}$ of clients at communication round t is less than $n\varrho$ times full gradient aggregation from all clients. Furthermore, we extend this observation to test loss, demonstrating that a subset of client updates, verified through FAVD, are effectively trained on data evaluated and validated by the FAVD framework. It is given as

$$\begin{aligned} \left\| \sum_{k\in\mathcal{S}^t} \mathcal{L}_{CE}(\mathcal{D}_t, v_t^k) - \sum_{k\in n} \mathcal{L}_{CE}(\mathcal{D}_t, v_t^k) \right\| \leq n\varrho, \\ \left\| \sum_{\nabla\theta_k \in (\mathbb{N}\cap\mathbb{F})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) - \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \right\| \leq n\varrho. \end{aligned} \tag{17}$$

Here, $\mathbb{N} \cap \mathbb{F}$ denote the subset of τ verified clients updates obtained after training models using FAVD valuated data whose test loss is lower than that of remaining clients.

$$\begin{aligned} \sum_{\nabla\theta_k \in (\mathbb{N}\cap\mathbb{F})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq n\varrho \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \\ \sum_{\nabla\theta_k \in (\mathbb{N}\cap\mathbb{F})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k). \end{aligned} \tag{18}$$

Here $=$ holds true when $\tau = 0$. This proves Eq. 4 of Theorem 3.3. Combining Eq. 14 and Eq. 18 we get

$$\sum_{\nabla\theta_k \in (\mathbb{N}\setminus\mathbb{M})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \tag{19}$$

$$\sum_{\nabla\theta_k \in (\mathbb{N}\cap\mathbb{F})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \tag{20}$$

$$\begin{aligned} \sum_{\nabla\theta_k \in (\mathbb{N}\setminus\mathbb{M})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) + \sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \\ \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) + \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \end{aligned} \tag{21}$$

$$\begin{aligned} \sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) + \\ \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) - \sum_{\nabla\theta_k \in (\mathbb{N}\setminus\mathbb{M})} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \end{aligned} \tag{22}$$

$$\begin{aligned} \sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) + \\ \cancel{\sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k)} - \cancel{\sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k)} + \\ \sum_{\nabla\theta_k \in \mathbb{M}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \end{aligned} \tag{23}$$

$$\sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) + \sum_{\nabla\theta_k \in \mathbb{M}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \tag{24}$$

$$\sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \leq \sum_{\nabla\theta_k \in \mathbb{N}\setminus\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k), \tag{25}$$

$$\left\| \sum_{\nabla\theta_k \in \mathbb{N}\setminus\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \right\| \geq \left\| \sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \right\|, \tag{26}$$

$$\left\| \sum_{\nabla\theta_k \in \mathbb{N}\setminus\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) - \sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \right\| \geq 0, \tag{27}$$

generalizing,

$$\left\| \sum_{\nabla\theta_k \in \mathbb{N}\setminus\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) - \sum_{\nabla\theta_k \in \mathbb{N}\cap\mathbb{F}} \mathcal{L}_{CE}(\mathcal{D}_t, \nabla\theta_k) \right\| \geq \delta, \tag{28}$$

where $\delta \geq 0$. Consequently, Eq. 14, Eq. 18, and Eq. 28 collectively establish the validity of the conditions stated in Eq. 3, Eq. 4, and Eq. 5 of Therorem 3.3, respectively. These results demonstrate that our **FAVD**-integrated FL system effectively generates verifiable client updates through auditable data valuation.

4. More Details: Experiments and Ablation Study

4.1. Datasets and model architectures

We extensively evaluated our approach using five benchmark datasets, namely, Covid-chestxray [13], Camelyon17 [4], HAM10000 [55], CIFAR10 [25], and CIFAR100 [25].

- **Covid-chestxray [13].** The Covid-chestxray dataset, prepared by Feki *et al.* [13], consists of 108 chest X-ray images from 76 patients diagnosed with COVID-19 and 108 chest X-ray images from healthy patients. To augment the dataset, we applied geometric transformations such as rotation and zoom. Specifically, rotation was applied by randomly rotating the images by small degrees (with a rotation range of 10 degrees), while zoom augmentation was performed by zooming in or out within a small range (zoom range = 0.1). These augmentations expanded the dataset, increasing the number of training samples from 38 to 152 (76 for COVID-19 cases and 76 for normal cases) for each client, following the methodology outlined in [13]. We utilize the ResNet50 architecture [17], as recommended by Feki et al. [13], with four clients, including one malicious client. The output layer features vector ($\zeta \in \mathbb{R}^d$) dimension for ResNet50 is 2048.
- **Camelyon17 [4].** We utilized the publicly available Camelyon17 tumour dataset [4], which includes 450,000 histology images from five different hospitals, each with varying stain types. Each hospital was treated as a separate client, with images across clients displaying heterogeneous appearances, though they shared the same label distribution (normal and tumour tissues), as outlined in [21]. We adopted the DenseNet121 architecture [18] and implemented FL using five clients, as described in [21], incorporating one malicious client in the experimental setup. The output layer features vector ($\zeta \in \mathbb{R}^d$) dimension for DenseNet121 is 1024.
- **HAM10000 [55].** This is a skin lesion classification dataset with 10,015 images. We classified images of actinic keratoses (akiec), melanoma (mel), and basal cell carcinoma (bcc) as malignant, and images of benign keratosis (bkl), dermatofibroma (df), melanocytic nevi (nv), and vascular skin lesions (vasc) as benign. Given the inherent variability in skin lesion images, no additional data augmentation was applied, as noted in [61]. We utilized the data splits from [61] and applied the FL-Fixcaps model architecture across 2, 4, 8, 16, 32, and 64 clients. We also examined both single- and multi-client threat scenarios (5, 10, 20 clients) on this dataset. $d = 156$.
- **CIFAR10 [25] & CIFAR100 [25].** These are widely used benchmark datasets for classification, consisting of 60,000 samples across 10 and 100 distinct classes, respectively. For CIFAR-10, we employ the ResNet18 architecture with an input size of 224×224, while for CIFAR-100, we use ResNet50 with the same input size. In our experiments, we simulate FL with 100 clients for CIFAR-10, including 1, 10, and 50 malicious clients, and with 10,000 clients for CIFAR-100, considering 1, 50, and 100 malicious clients. The output layer feature dimension d for ResNet18 is 512, and for ResNet50, it is 2048.

FL-Fixcaps architecture. Capsule networks (CapsNets) [51] have demonstrated potential in capturing pose and spatial relationships of features, overcoming some limitations of conventional deep learning approaches in image classification. However, these networks often face challenges with complex images, such as dermatoscopic images, and do not adequately address critical data privacy concerns in the medical AI domain. To tackle these issues, we propose FL-Fixcaps, an enhanced capsule network designed for skin lesion classification within a distributed, decentralized FL

framework. While the Fixcaps architecture [28] has been explored in existing literature, to the best of our knowledge, no prior work has applied Fixcaps within the FL setting for skin lesion classification. Our approach fills this gap by leveraging FL with the Fixcaps architecture, offering a novel solution for distributed skin lesion classification while ensuring data privacy.

FL-Fixcaps builds upon traditional CapsNets by introducing a large-kernel convolution layer (31x31, empirically determined) at the input stage, significantly expanding the receptive field to capture more contextual information from dermoscopic images. This approach contrasts with the smaller kernels commonly used (3x3, 5x5, 7x7). The capsule layer in FL-Fixcaps is divided into primary and digit capsules, with the primary capsule utilizing group convolution to prevent underfitting, reduce computational complexity, and enhance classification accuracy. In this work, we present the FL-Fixcaps model architecture, which is employed for local model training in FL clients, using a seven-class skin lesion classification task. FL-Fixcaps achieves excellent performance on the HAM10000 dataset. Through extensive experimentation and empirical testing, we have determined that the combination of the large-kernel convolution layer, attention mechanism, and modified capsule structure in FL-Fixcaps produces optimal performance for skin lesion classification in a FL setting.

4.2. Implementation details

Our dataset is partitioned with 80% allocated for training and 20% for testing. The training data is distributed in two settings for FL: a uniform setting, where data is evenly split across clients, and a non-IID setting, where data is randomly distributed across client shards following a Dirichlet distribution with parameter $\kappa = 1$. We evaluate the system under three maliciousness levels 30%, 50%, and 100%, which is calculated as $\frac{\nu}{\mathcal{N}_k} \times 100$, representing the proportion of samples under threat to the total samples. Experiments are conducted over 150–500 global epochs, each comprising 5–10 local epochs using local data. The training employs a batch size of 64 and a learning rate $\eta = 0.01$. *A sample implementation code is included in the supplementary material, and the full code will be made publicly available upon acceptance.*

Rationale behind the choice of models and FL settings. For the respective datasets, we selected models that have demonstrated the best performance in prior studies, aligning with our primary focus on introducing a framework for auditable data valuation and verifiable client contributions. While transformer-based models could potentially enhance classification performance, we chose not to explore them to maintain focus on our proposed framework and avoid expanding the experimental space, which could detract from the main objectives. It is important to note that our FAVD framework is model-agnostic, as it relies on the output layer representations for comparisons, making it adaptable to a wide range of architectures. To demonstrate the robustness of FAVD, we conducted extensive experiments comparing different architectures, including our FL-Fixcaps model. Furthermore, we analyzed the scalability of the framework by increasing the number of clients to 10,000 and explored various settings, such as different numbers of clients, clients selected per round, malicious clients, maliciousness levels, and model architectures, offering a thorough evaluation of the proposed approach.

4.3. Software and hardware configuration

The experiments were conducted using Python version 3.6, leveraging frameworks such as PyTorch, Pandas, and NumPy. The implementation was designed to facilitate local model training on the client side and global model evaluation on the server side, utilizing an NVIDIA Tesla M60 GPU with 8GB of RAM.

4.4. Metrics

To evaluate FAVD, we propose two new metrics: Malicious Sample Detection Rate (MSDR) and Benign Misclassification Rate (BMR) for auditability, alongside the Client Contribution Consistency (CCC) score for verifiability.

(i) Malicious Sample Detection Rate (MSDR). It serves as a critical metric for evaluating the effectiveness of a framework in identifying and discarding malicious data samples in FL. It is given as

$$\text{MSDR} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{29}$$

where:

- **TP (true positives):** Number of malicious samples correctly identified and discarded.
- **FP (false positives):** Number of benign samples incorrectly classified as malicious.
- **FN (false negatives):** Number of malicious samples misclassified as benign.

(ii) Benign Misclassification Rate (BMR). It is a critical metric designed to evaluate the accuracy of a FL framework in identifying benign samples as non-malicious. It is given as

$$\text{BMR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \tag{30}$$

where:

- **TN (True Negatives):** Number of benign samples correctly evaluated as benign.

Here, higher values of MSDR and lower values of BMR are considered desirable.

(iii) Client Contribution Consistency (CCC) To ensure stable global model performance across communication rounds, the Client Contribution Consistency (CCC) metric

evaluates whether individual client contributions fall within a defined tolerance interval (TI), promoting verifiability and alignment with overall model updates. Contributions within the TI are considered consistent, while those outside might indicate potential outliers.

Metric formulation:

1. *Calculate the average benign model weights*

$$B_w^m = \frac{1}{\text{len}(B_w)} \sum B_w$$

2. *Compute the L2 distance of each client contribution c_w to the average:*

$$D_{c_w} = \|c_w - B_w^m\|_2 = \sum_{i=1}^{n} (c_{w_i} - B_{w_i}^m)^2$$

3. *Define the tolerance interval (TI):* The TI is defined as:

$$TI = \left[D_{B_w^m} - 2\sigma_{B_w},\, D_{B_w^m} + 2\sigma_{B_w} \right]$$

where:
- $D_{B_w^m}$ is the mean distance of benign weights to B_w^m.
- σ_{B_w} is the standard deviation of these distances.

4. The CCC score measures how consistently each client's contribution falls within the tolerance interval:

$$\text{CCC} = \begin{cases} 1, & \text{if } D_{c_w} \in TI \\ \frac{\text{TI}_{\text{upper}} - D_{c_w}}{\text{TI}_{\text{upper}} - \text{TI}_{\text{lower}}}, & \text{if } D_{c_w} < \text{TI}_{\text{lower}} \\ 0, & \text{if } D_{c_w} > \text{TI}_{\text{upper}} \end{cases}$$

where:

$$\text{TI}_{\text{lower}} = D_{B_w^m} - 2\sigma_{B_w},$$
$$\text{TI}_{\text{upper}} = D_{B_w^m} + 2\sigma_{B_w}.$$

Interpretation:

- **CCC = 1:** The client's contribution lies within the tolerance interval, indicating it is consistent with benign contributions and supports stable global model performance.
- **CCC between 0 and 1:** The client's contribution is close to the tolerance interval but may require adjustment to align with other clients.
- **CCC = 0:** The client's contribution is outside the acceptable range, potentially affecting the stability and consistency of the global model.

4.5. Baselines

We evaluate the effectiveness of our proposed FAVD method against the below baselines based on their relevance and applicability in the chosen problem statement in FL.

- **Krum [5]:** Krum selects one local model update that is representative of a majority of client models. We set $x = 2$ for the Covid-chestxray and Camelyon datasets and $x = 3$ for other datasets to handle the malicious clients in our experimentation.
- **TM [63]:** Trimmed mean (TM) aggregates each dimension of input updates separately and sorts the values along the i^{th}-dimension. Then, it removes x largest and smallest values of that dimension and computes the average of the rest. We consider the suggested configuration of $x = 1$ for the Covid-chestxray and Camelyon datasets and $x = 5$ for other datasets to handle the malicious clients in our experimentation.
- **Median [63]:** The median aggregates each dimension of input updates separately and sorts the values of the i^{th}-dimension. Then, it takes the median as the global model's i^{th} parameter.
- **FLTrust [7]:** In this method, the server trains an auxiliary model using a root dataset and computes trust scores for clients based on the similarity of their weight updates to the server model. The server then updates the global model by taking a weighted average of the client models, with the weights proportional to their trust scores.
- **DOS [2]:** Distance-based Outlier Suppression (DOS) is an aggregation rule for FL designed to mitigate byzantine failures by suppressing malicious client updates. DOS uses Copula-based Outlier Detection (COPOD) to compute outlier scores for local parameter updates, which are then normalized to derive weighted averages, ensuring resilient global model updates without requiring hyperparameter tuning, even under data heterogeneity.
- **zPROBE [15]:** This is a privacy-preserving framework for Byzantine-resilient FL that detects and removes malicious updates using rank-based statistical bounds derived in zero-knowledge proofs. By employing randomized clustering, zPROBE enhances scalability while maintaining the privacy of user updates and ensuring robust model aggregation.
- **FedVal [57]:** This is a novel server-side validation method for FL that uses a score function to assess client updates, enabling optimal aggregation without compromising privacy. It ensures robustness against poisoning threats and reduces group bias, promoting fairness while maintaining differential privacy.
- **FedCE [22]:** It is a FL method that simultaneously optimizes collaboration fairness and performance fairness by estimating client contributions in gradient and data spaces. It uses gradient direction differences and prediction errors on client data to determine aggregation weights for the global model, promoting fairness and model quality.
- **FedBary [30]:** It is a privacy-preserving method for evaluating client contributions in FL using Wasserstein distance to compute a transparent data valuation. It identifies relevant datasets without relying on validation data or a pre-specified training algorithm, ensuring fairness and efficiency in FL tasks.

In addition, we compare our FL-Fixcaps method against

the below baselines.

- **MOON [29]:** Model-Contrastive Federated Learning (MOON) is a framework that improves FL by leveraging model-level contrastive learning to align model representations and correct local training. It effectively addresses data heterogeneity, achieving high performance on image datasets with deep learning models.
- **MoE [49]:** The Mixture of Experts (MoE) approach combines outputs from a generalist public model and private user-specific models in a privacy-preserving FL framework. This method enables personalized adaptation while leveraging the strengths of both general and private models.
- **SplitNN [58]:** This is a distributed deep learning method that enables collaborative model training among health entities without sharing raw data or model details. It supports practical settings such as multi-modality data sharing, multi-task collaboration, and learning without label sharing.
- **CusFL [61]:** This is a FL approach that enables each client to train a personalized model based on a federated global model aggregated from client-specific models. By using a federated feature extractor for alignment and guiding private model training, CusFL ensures tailored performance while leveraging collaborative learning.

4.6. Performance comparison of FL-Fixcaps under benign setting

Table 5 compares our FAVD method with the FL-Fixcaps architecture to other baselines on the HAM10000 dataset. Our architecture outperforms all client configurations, with improvements of ≈7 to 20%, due to Fixcaps' large convolution layers that capture nuanced features of skin lesions. This improvement can be attributed to the large convolutions in FL-Fixcaps, which effectively capture the dynamic feature representation of skin lesions in a distributed setup. However, applying the same architecture to the other two medical datasets yielded less improvement, likely due to limited data and their binary classification nature.

4.7. FAVD analysis with varying degrees of non-IID

From Table 6, it is evident that the global test accuracy ($A_{\mathcal{G}}$) improves significantly with increasing non-IID parameter (κ), as more uniform data distributions across clients ($\kappa = 10$) lead to higher performance. For instance, the Camelyon17 dataset shows an increase in $A_{\mathcal{G}}$ from 84.24% ($\kappa = 0.1$) to 92.10% ($\kappa = 10$) under benign settings, while CIFAR-10 and CIFAR-100 achieve $A_{\mathcal{G}}$ values of 88.24% and 77.51%, respectively, for $\kappa = 10$. We selected these three datasets based on the data availability and brevity. Further, In the presence of threats for the CIFAR10 dataset, Figure 3 illustrates that FAVD demonstrates remarkable robustness, particularly under multi-client threat scenarios ($m = 50$), where it consistently outperforms other FL methods by mitigating accuracy degradation caused by malicious contributions. Under single-client threats, FAVD sustains lower U compared to state-of-the-art methods such as FedVal and FedCE, effectively isolating malicious contributions. Furthermore, FAVD achieves stable performance under benign settings even with up to $n = 10,000$ clients, as shown for CIFAR-10 and CIFAR-100 in Table 6, effectively handling scalability. However, extreme non-IID scenarios ($\kappa = 0.1$) still pose challenges, with accuracy reductions of ≈ 8-10% compared to more uniform settings ($\kappa = 10$), indicating the need for continued enhancements in robust auditability and verifiability strategies for highly non-IID data distributions.

Table 5. Comparison of global test accuracy ($\mathcal{A}_{\mathcal{G}}\%$) ↑ of FAVD & FL-FixCaps vs. other methods on the HAM10000 dataset under uniform distribution and no data poisoning threat. **Bold** and result marks the best and second-best results, respectively.

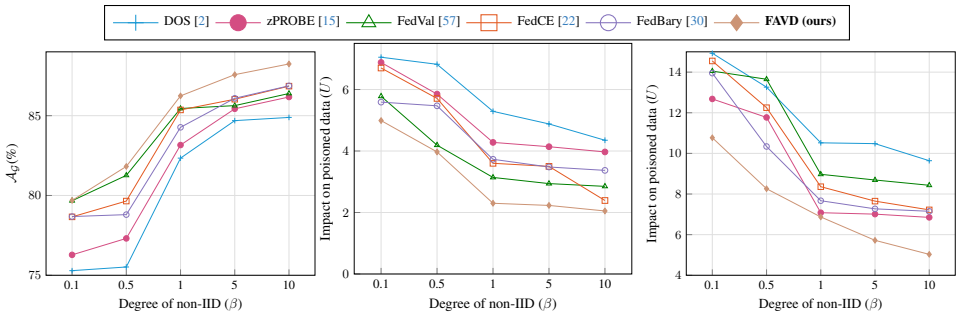
No. of clients	FedAvg [40]	MOON [29]	MoE [49]	SplitNN [58]	CusFL [61]	FL-Fixcaps (ours)
2	76.6±1.8	72.4±0.7	69.5±1.3	70.0±1.0	77.7±1.4	**84.78**±0.8
4	60.5±3.0	61.0±1.3	62.1±2.5	57.7±1.5	64.4±1.2	**83.73**±1.4
8	59.8±2.0	59.1±1.8	58.4±2.2	54.6±1.0	62.8±2.0	**81.81**±1.7
16	57.4±3.0	56.0±2.0	55.2±0.8	52.0±1.2	60.7±0.8	**80.29**±0.7
32	54.4±1.7	53.2±1.7	54.6±2.3	50.2±0.3	59.6±0.3	**79.89**±1.4
64	54.0±1.0	51.9±0.4	54.9±1.4	50.2±0.2	60.4±0.9	**75.48**±0.4

Table 6. Global test accuracy ($\mathcal{A}_{\mathcal{G}}\%$) ↑ comparison of FAVD for three datasets under non-IID and no threat settings.

nonIID (κ) ↓	Camelyon17 [4]	CIFAR10 [25]	CIFAR100 [25]
FL setting →	$n = 5$, $k = 5$	$n = 100$, $k = 70$	$n = 10^3$, $k = 500$
0.1	84.24±0.79	79.69±0.34	66.35±1.19
0.5	87.54±0.11	81.82±1.81	68.74±0.16
1 (default)	91.38±1.86	86.24±1.22	76.38±1.83
5	91.56±1.69	87.57±0.83	76.82±0.14
10	92.10±1.43	88.24±0.67	77.15±1.92

4.8. Additional results of auditable data valuation under threat settings

Table 7 presents a detailed comparison of various FL frameworks, including FAVD, FedCE, FedBary, and No Val, under a 30% malicious client scenario using the M-SimBA threat model. Across all datasets, FAVD consistently demonstrates superior performance in mitigating the impact of poisoned data ($U \downarrow$). For instance, in the Covid-chestxray dataset, FAVD achieves the lowest U of 2.47, outperforming both FedCE (3.51) and FedBary (3.50). Similar trends are observed for Camelyon17 and HAM10000 datasets, where FAVD achieves U values of 2.93 and 1.12, respectively, highlighting its robust capability to detect and



(a) No threat (b) Single-client threat (c) Multi-client ($m = 50$) threat

Figure 3. Comparison of global test accuracy ($\mathcal{A}_\mathcal{G}$, %) (↑) and the impact of poisoned data, quantified as $U = \mathcal{A}_\mathcal{G} - \mathcal{A}^*_\mathcal{G}$ (↓), across various FL frameworks. Results are presented for no-threat, single-client, and multi-client threat scenarios under non-IID data distributions (Dirichlet κ) on the CIFAR-10 dataset, with 50% maliciousness settings included for brevity.

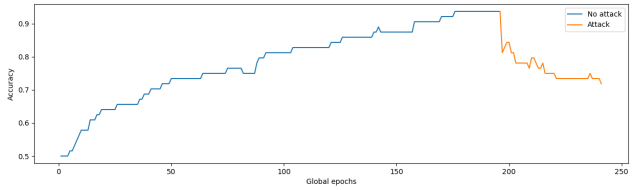
Table 7. Additional results on auditable data valuation comparison of FL Frameworks: Impact of poisoned data (U) ↓ in the presence of the M-SimBA threat, with a 30% maliciousness under uniform and non-IID (Dirichlet parameter ($\kappa = 1$) data partitioning across five datasets with different client numbers n, clients selected per round k, and malicious number of clients m. **No Val** indicates a standard FL system without data valuation, and **Bold** and result highlight the best and second best results, respectively.

Dataset	n	k	m	No Val	FedCE [22]	FedBary [30]	FAVD (ours)
Covid-chestxray [13]	4	4	1	31.92±1.01	21.1±1.86	20.89±1.12	**18.37**±0.48
Camelyon17 [4]	5	5	1	14.94±0.55	3.86±0.51	3.51±0.94	**2.47**±1.63
HAM10000 [55]	2	2	1	27.14±1.06	3.5±0.58	2.92±0.52	**2.93**±1.48
	4	4	1	20.27±0.10	3.97±1.53	2.15±0.47	**1.12**±1.16
	8	8	1	12.84±1.14	1.71±1.90	1.09±1.49	**0.34**±0.29
	16	16	5	17.21±1.02	2.31±1.37	2.43±0.18	**2.21**±0.92
	32	32	10	26.12±1.89	3.28±0.64	2.7±0.61	**1.12**±1.21
	64	64	20	37.48±0.51	7.85±1.98	6.38±1.09	**5.36**±0.23
CIFAR10 [25]	100	40	1	39.26±0.17	1.89±1.79	1.37±1.27	**0.43**±1.74
			10	58.37±0.44	5.02±0.72	5.28±1.03	**4.84**±1.41
		70	50, 30% mal	61.38±1.71	7.42±1.99	6.92±1.13	**5.68**±1.36
			50, 100% mal	70.88±1.28	13.9±1.18	12.85±0.87	**10.97**±1.76
CIFAR100[25]	10000	100	1	20.42±0.41	1.28±0.91	1.11±0.46	**0.29**±1.61
			50	59.27±1.84	4.95±0.54	3.61±0.34	**2.88**±1.41
		500	100	63.14±1.41	6.97±0.76	7.52±1.15	**6.37**±1.22

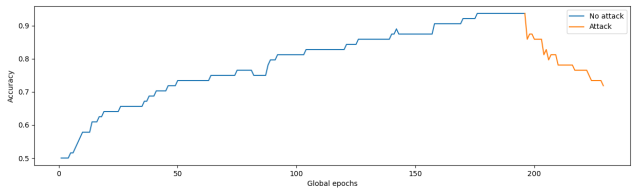
neutralize malicious contributions. In larger-scale datasets such as CIFAR-10 and CIFAR-100, FAVD significantly reduces the impact of poisoned data compared to baseline methods, particularly in extreme settings like 100% maliciousness or large client pools ($n = 10,000$). For CIFAR-10 with 50 malicious clients, FAVD achieves $U = 5.68$, which is markedly lower than FedCE (6.92) and FedBary (7.52). These results validate FAVD's effectiveness in maintaining auditable and verifiable data valuations, making it a reliable choice for FL under adversarial threat conditions.

4.9. Impact of ϵ on M-SimBA data poisoning threat effectiveness

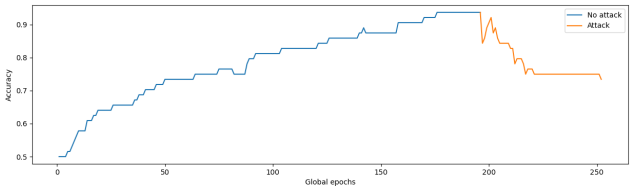
We analyzed the global test accuracy ($\mathcal{A}_\mathcal{G}$) under no-threat and single-client threat settings for the Covid-chestxray dataset, focusing on the impact of varying ϵ values. Our findings reveal that an ϵ value of 0.031 resulted in optimal threat accuracy from the adversary's perspective, as shown in Figure 4. Interestingly, both increasing and decreasing ϵ from this value led to reduced threat effectiveness. Lower ϵ values produced perturbations too subtle to significantly impact the model, while higher values made threats more detectable or caused severe performance degradation that could alert defenders. In contrast, the no-threat scenario maintained consistently high accuracy across all ϵ values, serving as a baseline. This non-linear relationship between ϵ and threat success highlights the delicate balance adversaries must strike and underscores the importance of adaptive defense mechanisms in FL systems for medical imaging. These insights contribute to our understanding of threat



(a) $\epsilon = 0.0031$



(b) $\epsilon = 0.031$



(c) $\epsilon = 0.31$

Figure 4. Global test accuracy under no threat and single-client threat settings for the Covid-chestxray dataset with varying ϵ values. Optimal threat accuracy is achieved with $\epsilon = 0.031$ while increasing or decreasing ϵ does not lead to improved threat accuracy.

dynamics and inform the development of more resilient security strategies for FL in sensitive domains.

4.10. Parameter sensitivity analysis of FAVD under threat settings

Figure 5 presents the sensitivity analysis of FAVD performance with respect to various parameters under a 50% maliciousness setting across multiple datasets. The analysis evaluates the impact of the noise level threshold (γ), the number of clusters (c), the cluster mean noise ($\mu_{\mathcal{N}}$), and the cluster covariance noise ($\Sigma_{\mathcal{N}}$) on the impact of poisoned data ($U \downarrow$). Figure 5a shows that increasing the noise level threshold (γ) leads to a steady decline in U, highlighting that lower thresholds $\gamma = 500$ allow FAVD to better filter malicious contributions, particularly for CIFAR-10 and CIFAR-100 datasets. Figure 5b demonstrates that U increases with an increase in the number of clusters (c) across all datasets, indicating that finer-grained clustering improves the ability to identify and discard poisoned data. Figures 5c and 5d examine the effects of noise parameters ($\mu_{\mathcal{N}}$ and $\Sigma_{\mathcal{N}}$) on U. Results indicate that FAVD maintains robust performance across varying noise conditions, with minimal variation in U for most datasets. Notably, the Covid-chestxray dataset exhibits a higher sensitivity to noise parameters compared to others, reflecting its inherent data characteristics. Overall, these findings emphasize FAVD's adaptability and resilience to parameter variations, reinforcing its reliability in adversarial scenarios. Further, we have selected the optimal parameter values, based on their performance in the sensitivity analysis, for all subsequent experiments.

4.11. Discussion: Enhancing FAVD to address label flipping threats

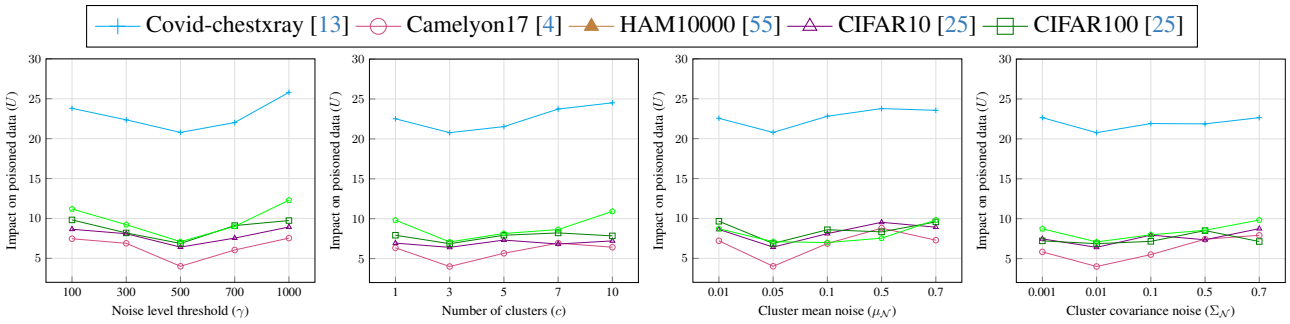
The FAVD method, as primarily designed, is effective in mitigating data poisoning threats where adversaries inject gradient noise into input images. It achieves robust, auditable data valuation by analyzing data contributions, identifying outliers based on their deviation from global masked data density parameters, and discarding noisy or malicious samples. However, in the case of label-flipping threats such as DPA-SLF [53] and DPA-DLF [53], where the adversary manipulates data by flipping the labels of input samples instead of altering the inputs themselves, FAVD can be adapted to maintain its robustness and support auditable data valuation and verifiable client contributions. To handle label-flipping threats, FAVD can incorporate additional mechanisms that evaluate the consistency between input features and their assigned labels. Specifically, as part of future work, FAVD could be extended with a feature-label consistency analysis to evaluate semantic alignment between features and their labels, flagging inconsistent samples. Further, cross-client label validation could detect label inconsistencies across clients, while statistical label auditing would identify anomalies in label distributions compared to global trends. Ensemble predictions for validation could enhance robustness by flagging labels inconsistent with federated predictions, and an auditable label verification pipeline could systematically inspect flagged samples for potential manipulation. These enhancements align with FAVD's goals of auditable data valuation and verifiable client contributions. However, adapting FAVD for label-flipping threats remains outside the scope of the current work and is left for our future exploration.

References

[1] Mansoor Ali, Hadis Karimipour, and Muhammad Tariq. Integration of blockchain and federated learning for internet of things: Recent advances and future challenges. *Computers & Security*, 108:102355, 2021. 2

[2] Naif Alkhunaizi, Dmitry Kamzolov, Martin Takáč, and Karthik Nandakumar. Suppressing poisoning attacks on federated learning for medical imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 673–683. Springer, 2022. 1, 6, 11, 13

[3] Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. Diverse client selection for federated learning via submodular maximization.



(a) γ vs. U (b) c vs. U (c) $\mu_{\mathcal{N}}$ vs. U (d) $\Sigma_{\mathcal{N}}$ vs. U

Figure 5. Sensitivity analysis of parameters on FAVD performance under threat across all datasets. The analysis examines the effect of the noise level threshold (γ), the number of clusters (c), and the noise parameters ($\mu_{\mathcal{N}}$, $\Sigma_{\mathcal{N}}$). The results follow the same settings as Table 4 in the main paper, with 50% maliciousness.

In *International Conference on Learning Representations*, 2021. 8

[4] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018. 9, 12, 13, 15

[5] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 6, 11

[6] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020. 1

[7] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021. 6, 11

[8] Yihao Cao, Jianbiao Zhang, Yaru Zhao, Pengchong Su, and Haoxiang Huang. Srfl: A secure & robust federated learning framework for iot with trusted execution environments. *Expert Systems with Applications*, 239:122410, 2024. 1

[9] Yu Chen, Fang Luo, Tong Li, Tao Xiang, Zheli Liu, and Jin Li. A training-integrity privacy-preserving federated learning scheme with trusted execution environment. *Information Sciences*, 522:69–79, 2020. 1

[10] Shifu Dong, Deze Zeng, Lin Gu, and Song Guo. Offloading federated learning task to edge computing with trust execution environment. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 491–496. IEEE, 2020. 1

[11] Mochan Fan, Kailai Ji, Zhaofeng Zhang, Hongfang Yu, and Gang Sun. Lightweight privacy and security computing for blockchained federated learning in iot. *IEEE Internet of Things Journal*, 10(18):16048–16060, 2023. 1, 2

[12] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1623–1640, 2020. 1

[13] Ines Feki, Sourour Ammar, Yousri Kessentini, and Khan Muhammad. Federated learning for covid-19 screening from chest x-ray images. *Applied Soft Computing*, 106:107330, 2021. 9, 13, 15

[14] Guillermo Gallego, Carlos Cuevas, Raul Mohedano, and Narciso Garcia. On the mahalanobis distance classification criterion for multidimensional normal distributions. *IEEE Transactions on Signal Processing*, 61(17):4387–4396, 2013. 5

[15] Zahra Ghodsi, Mojan Javaheripi, Nojan Sheybani, Xinqiao Zhang, Ke Huang, and Farinaz Koushanfar. zprobe: Zero peek robustness checks for federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4860–4870, 2023. 1, 6, 11, 13

[16] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018. 1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 9

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 9

[19] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018. 7

[20] Najeeb Moharram Jebreel and Josep Domingo-Ferrer. Fl-defender: Combating targeted attacks in federated learning. *Knowledge-Based Systems*, 260:110178, 2023. 1

[21] Meirui Jiang, Zirui Wang, and Qi Dou. Harmofl: Harmonizing local and global drifts in federated learning on heteroge-

neous medical images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1087–1095, 2022. 9

[22] Meirui Jiang, Holger R Roth, Wenqi Li, Dong Yang, Can Zhao, Vishwesh Nath, Daguang Xu, Qi Dou, and Ziyue Xu. Fair federated medical image segmentation via client contribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16302–16311, 2023. 1, 6, 11, 13

[23] Aditya Pribadi Kalapaaking, Ibrahim Khalil, Xun Yi, Kwok-Yan Lam, Guang-Bin Huang, and Ning Wang. Auditable and verifiable federated learning based on blockchain-enabled decentralization. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2024. 1, 2, 5

[24] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine Piatko, Ruth Silverman, and Angela Y Wu. The analysis of a simple k-means clustering algorithm. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 100–109, 2000. 5

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. pages 32–33, 2009. 9, 12, 13, 15

[26] K Naveen Kumar, C Vishnu, Reshmi Mitra, and C Krishna Mohan. Black-box adversarial attacks in autonomous vehicle technology. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7. IEEE, 2020. 3, 4

[27] K Naveen Kumar, C Krishna Mohan, and Linga Reddy Cenkeramaddi. The impact of adversarial attacks on federated learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[28] Zhangli Lan, Songbai Cai, Xu He, and Xinpeng Wen. Fixcaps: An improved capsules network for diagnosis of skin cancer. *IEEE Access*, 10:76261–76267, 2022. 10

[29] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021. 12

[30] Wenqian Li, Shuran Fu, Fengrui Zhang, and Yan Pang. Data valuation and detections in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12027–12036, 2024. 1, 2, 5, 6, 11, 13

[31] Xingyu Li, Zhe Qu, Shangqing Zhao, Bo Tang, Zhuo Lu, and Yao Liu. Lomar: A local defense against poisoning attack on federated learning. *IEEE Transactions on Dependable and Secure Computing*, 2021. 1

[32] Yuzheng Li, Chuan Chen, Nan Liu, Huawei Huang, Zibin Zheng, and Qiang Yan. A blockchain-based decentralized federated learning framework with committee consensus. *IEEE Network*, 35(1):234–241, 2020. 2

[33] Jinkun Lin, Anqi Zhang, Mathias Lécuyer, Jinyang Li, Aurojit Panda, and Siddhartha Sen. Measuring the effect of training data on deep learning predictions via randomized experiments. In *International Conference on Machine Learning*, pages 13468–13504. PMLR, 2022. 1, 2

[34] Zelei Liu, Yuanyuan Chen, Han Yu, Yang Liu, and Lizhen Cui. Gtg-shapley: Efficient and accurate participant contribution evaluation in federated learning. *ACM Transactions on intelligent Systems and Technology (TIST)*, 13(4):1–21, 2022. 2

[35] Sin Kit Lo, Yue Liu, Qinghua Lu, Chen Wang, Xiwei Xu, Hye-Young Paik, and Liming Zhu. Toward trustworthy ai: Blockchain-based architecture design for accountability and fairness of federated learning systems. *IEEE Internet of Things Journal*, 10(4):3276–3284, 2022. 1, 2

[36] Chuan Ma, Jun Li, Long Shi, Ming Ding, Taotao Wang, Zhu Han, and H Vincent Poor. When federated learning meets blockchain: A new distributed learning paradigm. *IEEE Computational Intelligence Magazine*, 17(3):26–33, 2022. 2

[37] Zhuoran Ma, Jianfeng Ma, Yinbin Miao, Yingjiu Li, and Robert H Deng. Shieldfl: Mitigating model poisoning attacks in privacy-preserving federated learning. *IEEE Transactions on Information Forensics and Security*, 17:1639–1654, 2022. 1

[38] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 3

[39] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, 80:S1–S7, 2018. 4

[40] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017. 12

[41] Thomas Minka. Estimating a dirichlet distribution, 2000. 2

[42] Fan Mo, Ali Shahin Shamsabadi, Kleomenis Katevas, Soteris Demetriou, Ilias Leontiadis, Andrea Cavallaro, and Hamed Haddadi. Darknetz: towards model privacy at the edge using trusted execution environments. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 161–174, 2020. 1

[43] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. Ppfl: privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, pages 94–108, 2021. 1

[44] Arup Mondal, Yash More, Ruthu Hulikal Rooparaghunath, and Debayan Gupta. Poster: Flatee: Federated learning across trusted execution environments. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 707–709. IEEE, 2021. 1

[45] D Napoleon and P Ganga Lakshmi. An efficient k-means clustering algorithm for reducing time complexity using uniform distribution data points. In *Trendz in information sciences & computing (TISC2010)*, pages 42–45. IEEE, 2010. 5

[46] Dinh C Nguyen, Ming Ding, Quoc-Viet Pham, Pubudu N Pathirana, Long Bao Le, Aruna Seneviratne, Jun Li, Dusit Niyato, and H Vincent Poor. Federated learning meets blockchain in edge computing: Opportunities and challenges. *IEEE Internet of Things Journal*, 8(16):12806–12825, 2021. 2

[47] Malay K Pakhira. A linear time-complexity k-means algorithm using cluster shifting. In *2014 international conference on computational intelligence and communication networks*, pages 1047–1051. IEEE, 2014. 5

[48] Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, and Prateek Mittal. Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *International Conference on Artificial Intelligence and Statistics*, pages 7587–7624. PMLR, 2022. 1

[49] Daniel Peterson, Pallika Kanani, and Virendra J Marathe. Private federated learning with domain adaptation. *arXiv preprint arXiv:1912.06733*, 2019. 12

[50] Nuria Rodríguez-Barroso, Eugenio Martínez-Cámara, M Victoria Luzón, and Francisco Herrera. Dynamic defense against byzantine poisoning attacks in federated learning. *Future Generation Computer Systems*, 133:1–9, 2022. 1

[51] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017. 9

[52] Muhammad Shayan, Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Biscotti: A blockchain system for private and secure federated learning. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1513–1525, 2020. 2

[53] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, 2022. 3, 14

[54] Qiheng Sun, Xiang Li, Jiayao Zhang, Li Xiong, Weiran Liu, Jinfei Liu, Zhan Qin, and Kui Ren. Shapleyfl: Robust federated learning based on shapley value. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2096–2108, 2023. 2

[55] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. 9, 13, 15

[56] Dmitrii Usynin, Alexander Ziller, Marcus Makowski, Rickmer Braren, Daniel Rueckert, Ben Glocker, Georgios Kaissis, and Jonathan Passerat-Palmbach. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. *Nature Machine Intelligence*, 3(9): 749–758, 2021. 3

[57] Viktor Valadi, Xinchi Qiu, Pedro Porto Buarque De Gusmão, Nicholas D Lane, and Mina Alibeigi. {FedVal}: Different good or different bad in federated learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 6365–6380, 2023. 1, 6, 11, 13

[58] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018. 12

[59] Nakul Verma and Kristin Branson. Sample complexity of learning mahalanobis distance metrics. *Advances in neural information processing systems*, 28, 2015. 5

[60] Jiachen T Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pages 6388–6421. PMLR, 2023. 1, 2

[61] Jeffry Wicaksana, Zengqiang Yan, Xin Yang, Yang Liu, Lixin Fan, and Kwang-Ting Cheng. Customized federated learning for multi-source decentralized medical image classification. *IEEE Journal of Biomedical and Health Informatics*, 26(11):5596–5607, 2022. 9, 12

[62] Chengyi Yang, Jia Liu, Hao Sun, Tongzhi Li, and Zengxiang Li. Wtdp-shapley: Efficient and effective incentive mechanism in federated learning for intelligent safety inspection. *IEEE Transactions on Big Data*, 2022. 2

[63] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. 1, 6, 11

[64] Yuhui Zhang, Zhiwei Wang, Jiangfeng Cao, Rui Hou, and Dan Meng. Shufflefl: Gradient-preserving federated learning using trusted execution environment. In *Proceedings of the 18th ACM international conference on computing frontiers*, pages 161–168, 2021. 1

[65] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2545–2555, 2022. 1

[66] Yang Zhao, Jun Zhao, Linshan Jiang, Rui Tan, Dusit Niyato, Zengxiang Li, Lingjuan Lyu, and Yingbo Liu. Privacy-preserving blockchain-based federated learning for iot devices. *IEEE Internet of Things Journal*, 8(3):1817–1829, 2020. 2