Whispers of Data: Unveiling Label Distributions in Federated Learning Through Virtual Client Simulation

Zhixuan Ma[1], Haichang Gao[*1], Junxiang Huang[1], and Ping Wang[1]

Xidian University, Xi'an Shannxi 710000, China mazhixuan@stu.xidian.edu.cn

Abstract. Federated Learning enables collaborative training of a global model across multiple geographically dispersed clients without the need for data sharing. However, it is susceptible to inference attacks, particularly label inference attacks.

Existing studies on label distribution inference exhibits sensitive to the specific settings of the victim client and typically underperforms under defensive strategies. In this study, we propose a novel label distribution inference attack that is stable and adaptable to various scenarios. Specifically, we estimate the size of the victim client's dataset and construct several virtual clients tailored to the victim client. We then quantify the temporal generalization of each class label for the virtual clients and utilize the variation in temporal generalization to train an inference model that predicts the label distribution proportions of the victim client. We validate our approach on multiple datasets, including MNIST, Fashion-MNIST, FER2013, and AG-News. The results demonstrate the superiority of our method compared to state-of-the-art techniques. Furthermore, our attack remains effective even under differential privacy defense mechanisms, underscoring its potential for real-world applications.

Keywords: Federated learning · Inference attack · Data privacy · Trust federated learning.

1 Introduction

The traditional centralized deep learning approach requires aggregating distributed data in a central computing center for training. During data transmission, there is a risk that user data is stolen, leaked, or misused. Federated learning (FL) [**DBLP:conf/esorics/ChoHYLBP24**, **chen2024federated**, **huang2024federated**, **miao2024rfed**], as a distributed machine learning framework, enables geographically isolated and resource-constrained participants to securely collaborate on model training. In federated training, the participants' local data remain on their devices and are not shared with other participants or the central server. The model is trained by exchanging model parameters and performing global aggregation. Consequently, federated learning has become a widely adopted distributed machine learning technique, supporting privacy-sensitive domains such as healthcare [**nguyen2022federated**, **li2021federated**], finance [**long2020federated**, **imteaj2022leveraging**], and wireless communications [**RJOUB2025113574**, **AZHARSHOKOUFEH2025113526**].

A critical question in federated learning (FL) is whether private data is truly secure and trustworthy [**DBLP:conf/esorics/MozaffariCH24**, **DBLP:conf/esorics/AsareBKY23**].

The model parameters uploaded by participants can inadvertently reveal characteristics of their local datasets [**10735243**, **DBLP:conf/esorics/GuepinMCM23**, **zhao2024loki**]. Label distribution inference attacks, which expose a participant's preferences, represent a significant threat to the security of FL systems. For instance, a malicious healthcare institution could infer the prevalence of a particular disease, allowing it to stockpile medications and manipulate prices. Similarly, a malicious retailer could infer the distribution of certain products, thereby deducing supply and demand dynamics, potentially disrupting the market and gaining an unfair competitive advantage.

The label distribution inference attack [**ayora2021profiling**, **anelli2021put**] originates from the Preference Profile Attack (PPA) [**PPA**], which was the first to propose inferring a participant's data preferences by observing the sensitivity of the model gradients uploaded by the participant. PPA identifies the top-k labels in which a target participant is most (or least) interested. Based on this framework, Raksha et al. [**ramakrishna2022inferring**] and LDIA [**gu2023ldia**] extend the approach to infer the entire label distribution by analyzing changes in the parameters of the model output layer of the target participant. However, the amplitude of these changes is influenced by factors such as the size of the victim client's dataset and the number of local training rounds, making it difficult for attackers to obtain prior knowledge of these parameters. Additionally, these methods are ineffective when differential privacy techniques are employed, as such techniques prevent an honest-but-curious server from accurately accessing the output layer parameters.

To eliminate sensitivity to changes in the model parameters, we leverage the temporal generalization of the model to train an inference model. This approach draws inspiration from the phenomenon of overfitting, where a model performs differently on training data compared to test data. This discrepancy occurs because the model tends to capture noise and details specific to the training data, thereby limiting its generalizability. Similarly, when the model is trained on labels with limited data, it often overfits by "memorizing" the features of these samples rather than learning their underlying patterns. The scarcity of diverse samples for these labels hinders the model's ability to generalize, resulting in suboptimal performance on underrepresented labels. In contrast, labels with abundant data provide a richer variety of samples, allowing the model to better capture their general characteristics and improve its overall generalization performance.

Based on the observation we proposed, we estimate the gradient information uploaded by the target client and design a virtual client environment that closely matches the target client's data size and distribution. By monitoring the temporal generalization performance of these virtual clients, we train an attack model to infer the local data distribution of the target client. Since inference defense strategies, such as differential privacy mechanisms, aim to balance model utility and privacy, our generalization performance-based inference attack is negligibly affected by these defense strategies. To achieve accurate label distribution inference, we introduce varying levels of noise into the virtual clients, effectively mitigating the impact of differential privacy mechanisms on the attack.

The main contributions of this paper are as follows.

– We achieve a more accurate and stable label distribution inference attack by estimating the size of the dataset.
– Based on the estimated dataset size, we construct a virtual client cluster that simulates the target client's behavior in every probable distribution, including both IID and various non-IID scenarios.
– We quantify the generalization of each label then use the temporal generalization data as input and the label distribution proportion as output to train a time-series-based attack model.
– We evaluate the proposed attack model on four datasets, and demonstrating its effectiveness even under differential privacy defenses.

Table 1: Comparison of three inference attacks.

Category	Focus	Attack Target	Potential Impact
Membership Inference Attack	Existence	Determine the existence of specific samples	Results in the disclosure of individual data users' private information.
Property Inference Attack	Sensitive Attributes	Extract specific sensitive attributes of training data	Causes the leakage of critical attribute information, such as gender or medical conditions.
Label Distribution Inference Attack	Dataset Composition	Reveal the label distribution across the entire dataset	Attacking a single participant: Reveals individual or institutional preferences. Attacking all participants: Exposes group traits, threatening collective privacy and fairness in decision.

2   Related Works

The label distribution inference attack aims to infer the label proportions in the training data of a target participant. By analyzing the model's outputs or statistics during the training process, the attacker seeks to determine the frequency or proportion of various class labels in the dataset. Zhou et al. first proposed the Preference Profiling Attack (PPA) [**PPA**], which infers the top k categories of greatest or least interest to the target client by capturing gradient sensitivity. Subsequently, Raksha et al. [**ramakrishna2022inferring**] and Gu et al. [**gu2023ldia**] extended PPA from inferring a limited number of categories to inferring the label distribution across all categories. Their method simulated training by constructing auxiliary datasets and calculated changes in the output layer parameters of the trained model to deduce the label distribution ratios of the target participant's training data. Dai et al. [**decaf**] also analyzed the gradient of the last fully connected layer, leveraging its class-specific neuron activation patterns to identify null classes and decompose non-null class distributions

via gradient bases constructed from auxiliary samples. However, these approaches has notable limitations, as the inference results are heavily influenced by the auxiliary data set. Changes in model parameters following training on auxiliary datasets of varying sizes are fixed, and an honest-but-curious server cannot a priori know the target participant's dataset information, resulting in inaccurate predictions. Additionally, requiring participants to upload their complete local model parameters without any restrictions is impractical. To enhance the security of participant models in federated learning, techniques such as differential privacy are often employed to protect uploaded information. However, these techniques also introduce noise, which contributes to the inaccuracy of existing methods in label proportion inference attacks.

Table 1 provides a comparative analysis of three inference attacks targeting federated learning. While all three methods exploit the data privacy of federated participants, they differ in their focus on data privacy, attack objectives, and potential impacts. The **membership inference attack** aims to determine whether a specific data sample is included in the training set [**DBLP:conf/esorics/HoCSL24**, **DBLP:journals/tdsc/ZhengL24**, **DBLP:journals/tdsc/Pichler0VP24**]. By analyzing the model's outputs, attackers can identify whether a user's sensitive data has been used in the training process. The success of this attack directly compromises individual privacy, exposing sensitive user information. In contrast, the **attribute inference attack** seeks to infer specific characteristics or attributes of the samples, such as gender or age [**DBLP:conf/uss/AnnamalaiGR24**, **10662889**]. Attackers analyze the model's prediction patterns in conjunction with auxiliary information to deduce sensitive attributes related to particular users. This type of attack not only undermines individual privacy but can also disproportionately affect certain societal groups, raising fairness concerns. Finally, the **label distribution inference attack** focuses on understanding the distribution of various labels within the training dataset. By examining changes in model parameter sensitivity, attackers infer the frequency and proportions of labels present in the dataset. This attack may disclose sensitive information about participant preferences or group characteristics, potentially impacting the fairness and reliability of the federated model.

3 Methodology

3.1 Threat Model

Victim Client: This study examines the privacy risks associated with the label distribution of a target client. The client owns a private dataset D, which is stored locally and is not shared with any third party. In a federated learning system, the central server distributes the current global model M_G to the client. The client then uses its local dataset D to train M_G, producing an updated local model M_L. This local model, or the corresponding gradient update G_L, is subsequently uploaded to the server. Beyond this exchange, clients neither share their models with other participants nor provide interfaces for third-party access.

Adversary Objective And Knowledge: We assume the attacker is an honest-but-curious server. The goal of the honest-but-curious server is to infer the label distribution of participating clients' training data by analyzing the uploaded model parameters or update gradients during the federated learning process. The server shares the same
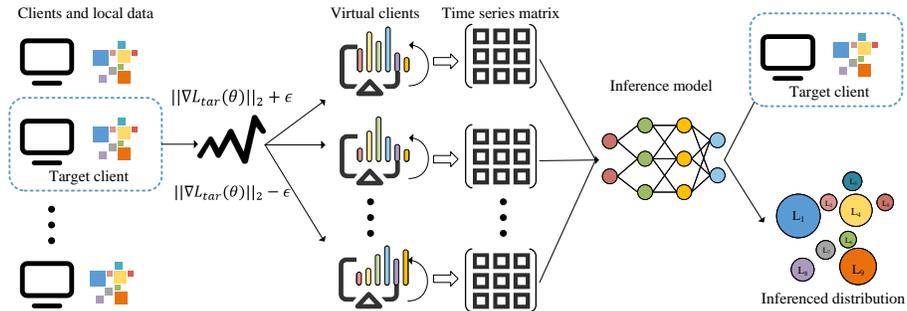
Fig. 1: The overview of the framework.

label space with clients but operates on different data spaces. The server has knowledge of the FL model's task and the class labels and can acquire auxiliary data corresponding to known labels from real-world sources. It cannot manipulate the FL aggregation process, modify the aggregation strategy, or access the local data of any client directly or indirectly. The server can only observe the model parameters $M_L^t, t \in \{1, 2, ...T\}$ uploaded by the victim client during each training iteration.

3.2 Overview

The proposed method is summarized in Figure 1. The honest-but-curious server aims to explore the data distribution privacy of participating clients without their awareness. The inference model is deployed on the server in two stages. The first stage estimates the size of the target participant's training dataset, and the second stage uses this estimation to construct virtual participants with a comparable dataset size. These virtual participants simulate federated training scenarios under various data distribution conditions. By collecting the temporal robustness accuracy changes during the training process of the virtual participants as input data and using their actual data distribution as labels, the server trains a inference model to probe the data distribution of the target participant.

3.3 Estimation Size of Dataset

In federated learning, clients are typically required to upload their dataset sizes to facilitate global convergence. However, to safeguard participant privacy, most federated learning strategies restrict clients from disclosing any details about their local datasets. Consequently, before inferring the label distribution of a target client, it is essential to first estimate the size of the client's dataset.

We found that the size of the training dataset plays a critical role in determining the quality and stability of gradient updates in machine learning models. As shown in Fig 2, we set the size of the client dataset to 2000(100%), 1600(80%), 1000(50%), 600(30%), and 200(10%), recording the gradient noise and gradient norm during its training. Larger datasets generally lead to more stable and reliable gradient estimates. With a larger dataset, each batch better represents the overall data distribution, reducing
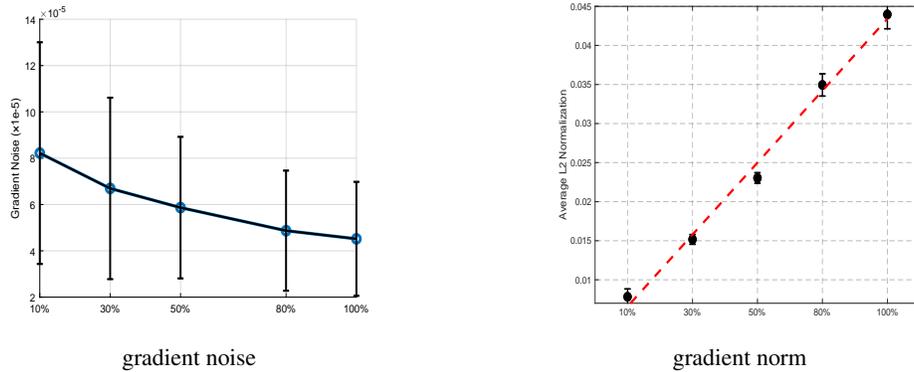
gradient noise

gradient norm

Fig. 2: The influence of dataset size on model gradient.

variance in gradient estimates. In contrast, smaller datasets result in noisier gradient estimates due to increased variance, as each sample has a larger influence on the gradient. Although the magnitude of the gradient may increase due to noisiness, the inconsistent directional alignment often leads to a smaller average L_2 norm ($||\nabla L(\theta)||_2$) resulting in oscillatory updates and slower convergence.

The relationship between dataset size and gradient variance can be quantified mathematically. For a gradient estimate $\hat{g}$ computed from a batch of size B, the variance is inversely proportional to B and can be expressed as $Var(\hat{g}) = \frac{\sigma^2}{B}$, where σ^2 represents the variance of individual sample gradients. Smaller datasets, characterized by higher variance, produce noisier gradient estimates, increasing the difficulty of achieving stable updates. According to the Central Limit Theorem, when the batch size B is sufficiently large, the mean gradient $\hat{g}$ computed from a batch approximates a normal distribution $\hat{g} \sim \mathcal{N}(\mu, \frac{\sigma^2}{B})$ where μ is the true gradient. For smaller datasets, the gradient estimates become noisier because of the higher variance introduced by fewer samples, which leads to more variability in the direction of the gradient. This results in unstable updates and difficulties in optimization. The noise in the gradient estimates can be described by ϵ, where $\epsilon \sim \mathcal{N}(0, \frac{\sigma^2}{n})$. This additional noise amplifies variability in gradient direction, resulting in a less stable optimization trajectory.

The Taylor expansion of the loss function L around a parameter θ further illustrates the impact of gradient stability on optimization:

$$\begin{aligned}
L(\theta + \Delta\theta) &\approx L(\theta) + \nabla L(\theta)^\top \Delta\theta \\
&\approx L(\theta) + \nabla L(\theta)^\top \Delta\theta + \frac{1}{2}\Delta\theta^\top H\theta
\end{aligned} \tag{1}$$

where $\nabla L(\theta)$ is the gradient, and H is the Hessian matrix. The L_2 norm of the gradient quantifies the sensitivity of the loss function to changes in model parameters. For small datasets, high variance and inconsistent gradient directions reduce the L_2 norm:

$$||\nabla L(\theta)||_2^2 = \sum_{i=1}^{d}(\nabla_i L(\theta))^2 \tag{2}$$

where $\nabla_i L(\theta)$ is the i-th component of the gradient. When $\nabla L(\theta) = \mu + \epsilon$, the expected value of the squared L_2 norm is:

$$\mathbb{E}[||\nabla L(\theta)||_2^2] = ||\mu||_2^2 + \mathbb{E}[||\epsilon||_2^2] \tag{3}$$

The noise ϵ introduces additional variance, leading to oscillatory updates and slower convergence. Furthermore, the Hessian matrix H, which encodes the second-order curvature information of the loss function, f, depends on the dataset size. For dataset of size n, H can be approximated as:

$$H \approx \frac{1}{n} \sum_{i=1}^{n} \nabla^2 L_i(\theta) \tag{4}$$

where $\nabla^2 L_i(\theta)$ represents the second-order partial derivatives for each sample. For smaller datasets, the estimate of the Hessian becomes less reliable, leading to a misrepresentation of the curvature and instability in optimization.

Finally, noisy gradients due to smaller datasets also lead to inconsistent directions during optimization. For a batch of size B, the average gradient is $\bar{g} = \frac{1}{B} \sum_{i=1}^{B} g_i$, and its L_2 norm is given by:

$$||\bar{g}||_2 = \sqrt{\sum_{j=1}^{d} \left(\frac{1}{B} \sum_{i=1}^{B} g_{ij} \right)^2} \tag{5}$$

where g_i represents the gradient for the i-th sample, and the sum is over the dimensions of the gradient vector. When individual gradients g_i vary significantly in direction, cancellation effects reduce the aggregated gradient norm $||\bar{g}||_2$, leading to smaller effective step sizes and slower convergence.

In summary, the size of the training dataset critically influences gradient stability during optimization. Larger datasets reduce variance, stabilize gradient directions, and yield more consistent L2 norms, facilitating more efficient and reliable optimization. Conversely, smaller datasets introduce noisier gradients, higher variance, and directional inconsistencies, which hinder convergence and reduce the effectiveness of the training process.

Based on the above principles, we infer the dataset size of the target client by constructing virtual clients and comparing their L_2 norms with that of the target client. We collect real-world data with the same labels as the federated task to construct an auxiliary dataset D_{aux}. The virtual client is initialized with the same structure and parameters as the global model provided in the federated setting, ensuring consistency with the target client. During training, we record the gradient norm $||\nabla L_v(\theta)||_2$ of the virtual client. We employ a binary search method to iteratively adjust the size of the virtual client's dataset until its gradient norm closely matches that of the target client, $||\nabla L_{tar}(\theta)||_2$. Specifically, we define a threshold ϵ to establish the acceptable range for the gradient norm, and the process continues until $||\nabla L_v(\theta)||_2$ satisfies the following condition:

$$||\nabla L_{tar}(\theta)||_2 - \epsilon < ||\nabla L_v(\theta)||_2 < ||\nabla L_{tar}(\theta)||_2 + \epsilon \tag{6}$$

Algorithm 1 Estimating Dataset Size of the target Client.

Input: Target client's gradient norm $||\nabla L_{\text{tar}}(\theta)||_2$, Auxiliary dataset D_{aux}, Global model parameters θ, Tolerance ϵ.

Output: Estimated dataset size of the virtual client.

1: Initialize virtual client's model parameters θ_v with θ
2: Set an initial guess for dataset size s_v
3: Define upper and lower bounds for the gradient norm: $\delta_1 = ||\nabla L_{\text{tar}}(\theta)||_2 - \epsilon$, $\delta_2 = ||\nabla L_{\text{tar}}(\theta)||_2 + \epsilon$
4: **while** $||\nabla L_v(\theta_v)||_2 < \delta_1$ or $||\nabla L_v(\theta_v)||_2 > \delta_2$ **do**
5:     Train the virtual client with dataset of size s_v
6:     Compute the gradient norm $||\nabla L_v(\theta_v)||_2$
7:     **if** $||\nabla L_v(\theta_v)||_2 < \delta_1$ **then**
8:         $s_v \leftarrow 2 \times s_v$
9:         Adjust the virtual client's dataset with D_{aux}
10:     **else if** $||\nabla L_v(\theta_v)||_2 > \delta_2$ **then**
11:         $s_v \leftarrow s_v/2$
12:         Adjust the virtual client's dataset with D_{aux}
13:     **end if**
14: **end while**
15: **return** s_v

3.4   Construct The Virtual Clients and Inference Model

To accurately simulate and train the inference model, we construct virtual clients under both IID and Non-IID scenarios. For the **IID scenario**, we ensure that each label is represented approximately equally across all virtual clients. The number of samples for each label is configured within the range $[C/s_v - \Delta, C/s_v + \Delta]$, where C is the total number of classes, and Δ represents the allowable fluctuation range. For the **Non-IID scenario**, we consider two types of imbalances: label quantity-based imbalance and distribution-based label imbalance. In label quantity-based imbalance, each virtual client randomly selects samples from a fixed subset of C_f classes. In distribution-based label imbalance, we employ a Dirichlet distribution to simulate uneven label distributions. The parameter α of the Dirichlet distribution controls the heterogeneity of the dataset. For each label distribution, we randomly sample from the auxiliary datasets to construct virtual client clusters P. By simulating federated training processes with these virtual clients and analyzing the changes in their generalization performance, we train the inference model to infer the label distribution of the target client.

The virtual client cluster P operates independently of the federated learning process, meaning it does not participate in the sampling or aggregation phases. These virtual clients update their models directly based on the results from global aggregation. It is important to note that the virtual clients rely solely on global aggregation outcomes, ensuring that their operations do not interfere with the actual federated learning system. The attacker uses these virtual clients to record their performance on the test set at the end of each training round. For each virtual client, a temporal matrix $\mathbb{R}^{E \times C}$ is constructed, where E represents the total number of training rounds. These temporal

---

Algorithm 2 Construct Virtual Clients and Train Inference Model

---

Initialize clusters of virtual clients P.

Input: Auxiliary dataset D_{aux}, Number of classes C, Fluctuation range Δ, Dirichlet parameter α, Global model parameters θ, Number of training rounds E.

Output: Record $\mathbb{R}^{E \times C}$.

1: **for** each virtual client in P **do**
2:    **if** IID scenario **then**
3:       $l \sim [C/s_v - \Delta, C/s_v + \Delta]$
4:    **end if**
5:    **if** Non-IID scenario **then**
6:       $l \sim Dir(\alpha)$ or select C_f classes
7:    **end if**
8:    Train the virtual clients $\hat{\theta}^{t+1} = \sum_{k=1}^{N} \frac{n_k}{n} \theta_k^t$
9:    Record $\mathbb{R}^{E \times C}$
10: **end for**

Inference the Label Distribution.

Input: $\mathbb{R}^{E \times C}$, W_h, b_h, v

Output: $l = \{l_1, l_2, \ldots, l_C\}$

1: $h_0, c_0 \leftarrow$ Initialize
2: **for** $t = 1$ to E **do**
3:    $h_t, c_t = \text{LSTM}(h_{t-1}, c_{t-1}, \mathbb{R}_{t,:})$
4: **end for**
5: $e_t = v^\top \tanh(W_h h_t + b_h), t = 1, \ldots, E$
6: $\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{E} \exp(e_k)}, t = 1, \ldots, E$
7: $c = \sum_{t=1}^{E} \alpha_t h_t$
8: $l = \text{FC}(c)$
9: **return** l

---

matrices capture the variations in the model's generalization performance across different labels throughout the training process of the virtual clients.

To effectively extract key information and capture temporal dependencies from sequence data, we designed an LSTM-based inference model. The temporal matrices are used as inputs, while the output represents the label distribution proportions of the virtual clients, serving as the training targets for the inference model. Additionally, a temporal attention layer is incorporated before the output layer to enhance the model's ability to capture dependencies within the input sequences. Specifically, the intermediate output of the inference model is $H = [h_1, h_2, \ldots, h_T]$, where h_t denotes the hidden state at time step t. The temporal attention layer generates a weighted summary vector c from these hidden states. To achieve this, we first compute an attention score e_t to quantify the importance of each hidden state $e_t = v^\top tanh(W_h h_t + b_h)$,

$$e_t = v^\top tanh(W_h h_t + b_h) \tag{7}$$

where W_h and b_h are learnable parameters, and v is a parameter vector that projects the LSTM output into a scalar score. The attention scores are then normalized using the

softmax function to generate weights $\alpha_t = \frac{exp(e_t)}{\sum_{i=1}^{T} exp(e_k)}$.

$$\alpha_t = \frac{exp(e_t)}{\sum_{i=1}^{T} exp(e_k)} \tag{8}$$

Finally, the context vector c is obtained by computing a weighted average of the hidden states h using the attention weights $c = \sum_{t=1}^{T} \alpha_t h_t$.

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{9}$$

This LSTM model, enhanced with a temporal attention mechanism, is well-suited for handling the temporal performance matrices $\mathbb{R}^{E \times C}$. It effectively identifies and focuses on changes that impact the generalization performance of the virtual clients.

For the selected target client, the honest-but-curious server continuously collects its uploaded model parameters or gradients throughout the federated training process. Simultaneously, the server evaluates the target client's performance by obtaining its test accuracy using a public dataset. After multiple rounds of data collection, the temporal test data from the target client is fed into the trained inference model, yielding the predicted label distribution proportions.

4 Experiments

4.1 Experimental Setup

The proposed label distribution inference attack were conducted using PyTorch and Python 3 on an NVIDIA GeForce RTX 3090 equipped with 4 GPUs. We conducted comparisons of our proposed method with the SOTA under four different distance metrics. These comparisons were carried out in both the IID as well as non-IID scenarios. For IID scenarios, we selected the MNIST [**MNIST**], Fashion-MNIST [**fashionMNIST**], and AG-News [**agnews**] datasets for experimentation. For non-IID scenarios, in addition to the aforementioned datasets, we incorporated the Fer-2013 [**fer**] dataset. We designed two non-IID situations to thoroughly evaluate the effectiveness of our method: one is label quantity-based imbalance, and the other is distribution-based label imbalance.

To emulate real-world data distributions, we constructed 1200 virtual clients under three different data distribution settings. The training group consisted of 900 virtual clients, while the testing group included 300 clients. Each virtual client was randomly assigned between 3,000 to 5,000 images to simulate varying levels of data availability. This setup allowed us to evaluate and compare the method's performance under different data distribution conditions in a controlled environment.

Evaluation metrics: We assessed the proximity between the predicted and actual class-label distributions of target clients using four distance metrics: Wasserstein distance, KL divergence, JS divergence, and L1 distance, for a comprehensive analysis from multiple perspectives.

The **Wasserstein distance** assesses the structural and positional discrepancies between two distributions, calculating the minimal effort needed to reshape the predicted distribution to match the actual distribution.

KL divergence acts as an asymmetric metric for quantifying discrepancies between two probability distributions.

JS divergence provides a symmetric, bounded method for evaluating the overall similarity between two distributions, highlighting their global resemblance.

L1 distance quantifies the total absolute differences across corresponding dimensions of two distribution vectors, emphasizing the point-to-point disparities without considering their overall shapes or sample positions.

We evaluate the effectiveness of the proposed label distribution inference attack in IID and two non-IID scenarios. In the IID scenario, we set the data volume for each label to fluctuate within the range of $[0.9 \times (N/C), 1.1 \times (N/C)]$, based on the total data volume N and the number of classes C. This setup reflects the possible natural variation in data across clients in federated learning while avoiding a scenario where label distributions are completely uniform, as inferring label distribution proportions is meaningless when labels are perfectly balanced. For the non-IID scenarios, we considered label quantity-based imbalance and distribution-based label imbalance. These conditions simulate common deviations in data distribution observed in the real world, where certain labels are overrepresented on some clients and relatively scarce on others.

4.2 Inference Performance in IID Scenario:

We conducted a comparative analysis of our technique against three SOTA approaches developed by Dai et al. (2024) [**decaf**], Raksha et al. (2022) [**ramakrishna2022inferring**] and Zhou et al. (2023) [**PPA**]. Specifically, in Zhou et al.'s method, the probabilities of predicting each class as the victim client's preferred label are normalized to estimate the proportions within the label distribution.

Table 2: Effectiveness of our model under different metrics in IID environment.

Method	MNIST				F-MNIST				AG-NEWS			
	Wass	KL	JS	L1	Wass	KL	JS	L1	Wass	KL	JS	L1
Dai et al.	0.0441	0.0010	0.0003	0.0372	0.0589	0.0015	0.0004	0.0444	0.0453	0.0016	0.0004	0.0493
Raksha et al.	0.2888	0.0733	0.0169	0.2871	0.3365	0.1257	0.0264	0.3441	0.1651	0.0223	0.0055	0.1771
Zhou et al.	2.9084	2.8086	0.3814	1.5061	2.7293	2.8857	0.3848	1.5267	1.0576	2.5025	0.2869	1.2998
Ours	**0.0730**	**0.0015**	**0.0003**	**0.0438**	**0.0496**	**0.0017**	**0.0004**	**0.0440**	**0.0502**	**0.0030**	**0.0007**	**0.0626**

As illustrated in the Table 2, the experimental results indicate that our model significantly outperforms three other advanced methods across all evaluated metrics. Specifically, our approach exhibited exceptional performance on the MNIST and Fashion-MNIST datasets, where the low Wasserstein and L1 distances underscore the effectiveness of our inference model. On the AG-News dataset, despite the increased complexity of unstructured text data, our model maintained superior performance, particularly reflected in the low values of KL divergence and JS divergence, indicating that our model matches the actual distributions closely.
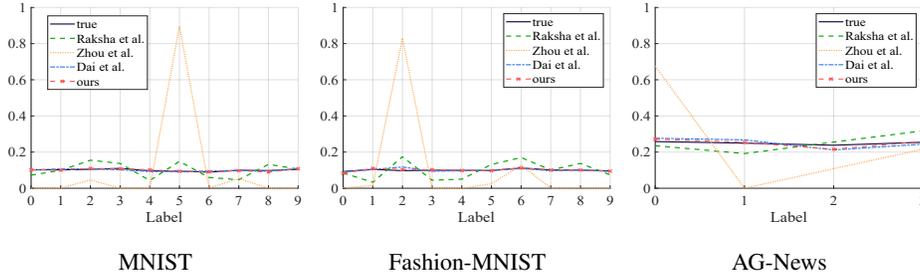
Fig. 3: Predicted distribution and true distribution in the case of IID.

We depicted the comparison between the predicted and actual label distributions as a line graph in Figure 3. The analysis reveals that our method closely approximates the true label distributions across all datasets, with particularly notable accuracy on the MNIST and Fashion-MNIST datasets, where the predicted distributions almost perfectly align with the actual distributions. In contrast, although the method by Dai et al. and Raksha et al. performs well on certain labels, it exhibits considerable overall variability and inconsistency. The approach proposed by Zhou et al. often results in predictions that deviate significantly from the actual distributions.

4.3   Inference Performance in Two Non-IID Scenario:

In federated learning, heterogeneity in data distribution presents a common and challenging issue. To verify the effectiveness of our proposed method within a federated learning environment, we conducted experiments in two non-IID scenarios: Label Quantity-Based Imbalance and Distribution-Based Label Imbalance.

Label quantity-based imbalance refers to significant disparities in the number of samples for each class label across different clients. In this scenario, each client possesses a varying number of samples for each class, while the total number of class labels C remains fixed. For datasets with a larger number of labels, such as MNIST, Fashion-MNIST, and Fer2013 we set C = 3. For the AG-News dataset, we set C = 2.
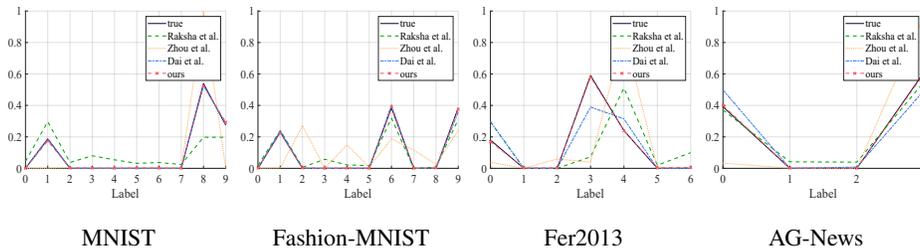

Fig. 4: Predicted distribution and true distribution in the case of Non-IID of label quantity-based imbalance.

The real label distribution proportions of the victim client compared to the predicted distributions are illustrated in the Figure 4. Our method exhibits significant advantages over the approaches by Dai et al., Raksha et al. and Zhou et al. across four datasets. Our model more accurately approximates the true distributions, particularly in scenarios where label distributions are highly concentrated or highly dispersed. On the MNIST and Fer2013 datasets, our model precisely captures the distribution of high-frequency labels, with the predicted and actual label distribution graphs nearly overlapping. Although the methods by Dai et al. and Raksha et al. can approximate the true distribution in some datasets, they tend to either overestimate or severely underestimate the proportions of certain labels in most cases. Overall, our model demonstrates extensive adaptability and robustness in complex label imbalances scenarios. This capability underscores its potential in practical federated learning applications, providing strong support and accurate analytical foundations for real-world applications with uneven label quality.

Distribution-based label imbalance concerns the feature distribution within each label, which may vary drastically across different clients. To simulate this data distribution, we employ a Dirichlet distribution to partition data across clients and adjust the heterogeneity level among clients by manipulating the Dirichlet parameter α. The smaller α values indicating high client data heterogeneity, while larger α values lead to a more uniform distribution. We set $\alpha = \{0.5, 1, 2\}$ for MNIST, F-MNIST, and FER2013, $\alpha = \{0.1, 0.4, 1\}$ for AG-NEWS.

Table 3: Effectiveness of our model under different metrics in Non-IID environment of distribution-based label imbalance.

Method	α	MNIST				F-MNIST				FER2013				AG-NEWS(0.1/ 0.4/ 1)			
		Wass	KL	JS	L1	Wass	KL	JS	L1	Wass	KL	JS	L1	Wass	KL	JS	L1
Dai et al.	0.5 / 0.1	0.1508	0.0282	0.0052	0.0919	0.1008	0.0252	0.0052	0.0849	0.4800	0.1391	0.0363	0.4141	0.1183	0.0507	0.0148	0.1522
Raksha et al.		0.9071	0.2609	0.0694	0.5338	0.4425	0.0847	0.0220	0.3121	0.6742	0.2632	0.0665	0.5502	0.4510	0.2846	0.0864	0.5283
Zhou et al.		1.7926	2.0411	0.2451	1.1215	2.1623	2.4024	0.3052	1.3042	1.2926	1.7846	0.2240	1.0706	0.1434	0.1962	0.0295	0.2163
Ours		**0.1211**	**0.0149**	**0.0041**	**0.0811**	**0.0837**	**0.0085**	**0.0024**	**0.0634**	**0.0972**	**0.0283**	**0.0073**	**0.0972**	**0.0524**	**0.0163**	**0.0049**	**0.0602**
Dai et al.	1 / 0.4	0.1230	0.0352	0.0059	0.0913	0.1098	0.0192	0.0042	0.0853	0.2972	0.1005	0.0213	0.2855	0.2898	0.1199	0.0323	0.3286
Raksha et al.		0.4884	0.0846	0.0226	0.3159	0.2962	0.0586	0.0141	0.2373	0.4807	0.2937	0.0490	0.4400	0.5214	0.2833	0.0780	0.5815
Zhou et al.		1.7856	1.8181	0.2362	1.1302	2.3492	2.2553	0.2976	1.3033	1.6383	2.2295	0.2875	1.2653	0.5403	0.8894	0.1080	0.5978
Ours		**0.0937**	**0.0067**	**0.0018**	**0.0680**	**0.0848**	**0.0080**	**0.0022**	**0.0694**	**0.0689**	**0.0081**	**0.0020**	**0.0799**	**0.0909**	**0.0166**	**0.0043**	**0.1082**
Dai et al.	2 / 1	0.0041	0.0275	0.0050	0.1026	0.1152	0.0255	0.0050	0.0934	0.2476	0.0878	0.0178	0.2503	0.2823	0.1039	0.0286	0.3494
Raksha et al.		0.3527	0.0504	0.0130	0.2204	0.2803	0.0827	0.0140	0.2073	0.6231	0.3442	0.0628	0.5140	0.3667	0.1669	0.0462	0.4348
Zhou et al.		2.3480	2.1417	0.2836	1.2770	2.2807	2.2660	0.3050	1.3314	1.6760	2.2675	0.2828	1.2595	1.2595	1.7653	0.2176	1.0345
Ours		**0.0929**	**0.0075**	**0.0021**	**0.0706**	**0.0896**	**0.0047**	**0.0012**	**0.0613**	**0.0892**	**0.0072**	**0.0018**	**0.0883**	**0.0067**	**0.0124**	**0.0027**	**0.1067**

The Table 3 illustrates the effectiveness of our method compared to SOTA under a non-IID setting with distribution-based label imbalance. Across all four measurement metrics, our method consistently shows lower distances on all datasets, underscoring the close alignment between the predicted and actual label distributions. This indicates that our approach effectively captures both the general and specific features of the la-

bel distribution. Notably, our method reduces the L1 distance by an order of magnitude compared to alternative methods. On the MNIST dataset with an alpha value of 2, the KL divergence between our inferred label distribution and the true distribution is just 0.0027, maintaining a high degree of similarity between the predicted and actual distributions.
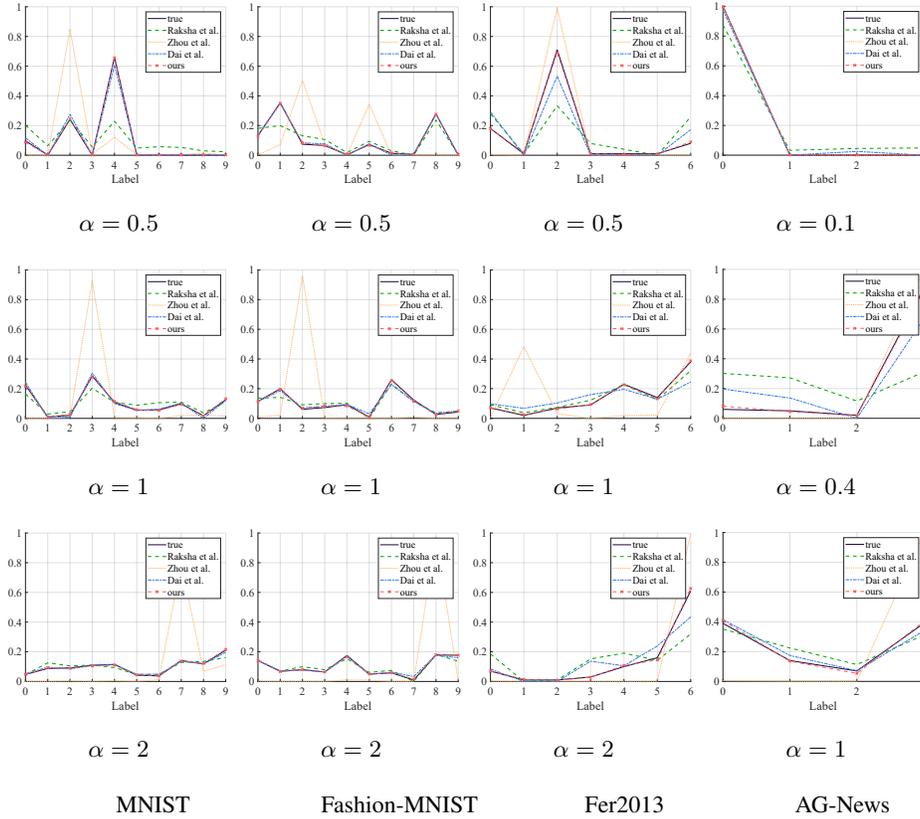

Fig. 5: Predicted distribution in non-IID of distribution-based label imbalance under different α.

The Figure 5 offer a comprehensive view of the performance under varying degrees of data heterogeneity induced by changing the Dirichlet parameter alpha α. Notably, our method consistently demonstrates a superior ability to adhere closely to the true distributions across all datasets and α settings. This indicates that our model maintaining high performance even as data heterogeneity increases. The reveals that all methods struggle to some extent as heterogeneity intensifies. However, our method exhibits a robust response, maintaining closer alignment with the true distribution compared to SOTA. This suggests that our model is particularly well-suited for federated learning applications where client data may vary dramatically, ensuring consistent performance

across a range of complex scenarios. Even in datasets with fewer labels like Fer2013 and AG-News, where modeling might theoretically be simpler, our method still proves effective, handling nuances in data distribution with finesse.

4.4 Inference under Defenses

Local differential privacy (LDP) offers robust mathematical guarantees for data privacy in federated learning by introducing random noise into the results of data publication or queries, thereby protecting individual information from inference. We validate the impact of local differential privacy techniques on the inferential capabilities of the proposed model. We focus on assessing the variations in model performance under different privacy budgets and the specific impact of privacy protection measures on model efficacy. Through these experiments, we aim to demonstrate that even under stringent privacy constraints, the label distribution inference model can still achieve satisfactory performance standards.

Table 4: Performance Metrics at Various Epsilon Values.

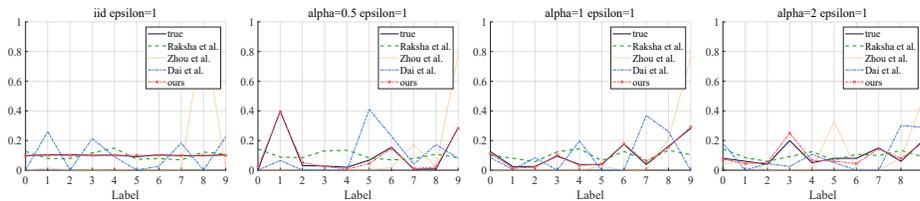| ϵ	1	2	5	10	40
Wass	0.3738	0.2248	0.2041	0.2055	0.1451
KL	0.0689	0.0382	0.0256	0.0257	0.0208
JS	0.0194	0.0112	0.0071	0.0070	0.0058
L1	0.2606	0.1646	0.1483	0.1425	0.1252
ACC(%)	50.46	77.25	85.66	87.78	88.55

Fig. 6: Predicted distribution attack against differential privacy under Non-IID.

Table 4 illustrates the impact of our proposed method under five different ϵ settings for differential privacy defense. As the privacy budget ϵ decreases, the distance between the predicted distribution and the actual label distribution of the target client relatively increases, but this corresponds to a significant reduction in the model's usability. When the privacy budget is 5, the L1 distance is 0.1476, and as the privacy budget reduces to 1, the L1 distance increases to 0.2455. However, the victim client model's prediction accuracy for the main task drastically decreases from 85.66% to 50.46%. Considering

the impact of differential privacy on model usability, we believe that the fluctuations in the precision of label distribution predictions by the attack model are acceptable. Additionally, the server's ability to obtain the epsilon value for local differential privacy and to add corresponding noise in virtual clients can mitigate the effects of differential privacy defense strategies on the inference model.

Figure 6 presents the results of our method compared to SOTA methods under varying levels of data heterogeneity, with a privacy budget of $\epsilon = 1$. The results demonstrate that, both in IID and Non-IID scenarios, our method more precisely predicts the actual label distributions than the SOTAs. The enhanced accuracy is primarily attributed to our approach of simulating multiple virtual clients with different data distributions, whose temporal generalization performance data is used to train the inference model. In contrast, the methods by Dai et al. and Raksha et al. primarily infer by analyzing changes in the model's output layer parameters, that is more susceptible to noise introduced by differential privacy strategies. The presence of differential privacy noise leads to significant deviations in their inference results from the actual label distributions, especially in environments with high data heterogeneity. Our research indicates that analyzing the generalization performance of virtual clients can effectively mitigate the adverse impacts of differential privacy noise on inference accuracy. This capability allows for maintaining precise inference models while upholding high levels of privacy protection. This finding is crucial for designing efficient and privacy-preserving data analysis strategies in practical federated learning applications.

5   Conclusion

This paper introduces a novel label distribution inference attack method for federated learning environments, which leverages temporal variations in the model's generalization performance across different labels. By simulating virtual clients and analyzing the model parameters uploaded by real clients, this method infers the label distribution of their data. Tested across multiple datasets, this study not only confirms the efficacy of the method but also demonstrates its robustness against differential privacy protection measures.

This advancement holds significant value for practical applications, especially in industries where data sensitivity is paramount, such as healthcare and financial services. For instance, in healthcare, accurately understanding the distribution of diseases among populations can optimize resource allocation and predict epidemic trends, but it may also pose risks to patient data privacy. However, several questions remain for future exploration, such as the feasibility of deploying attackers on clients and extracting information from the global model through active attacks. We leave these issues to future research.