Wavelet Scattering Transform and Fourier Representation for Offline Detection of Malicious Clients in Federated Learning

Wavelet Scattering Transform and Fourier Representation for Offline Detection of Malicious Clients in Federated Learning

Alessandro Licciardi, Davide Leo, and Davide Carbone

***Abstract*—Federated learning (FL) enables the training of machine learning models across decentralized clients while preserving data privacy. However, the presence of anomalous or corrupted clients—such as those with faulty sensors or nonrepresentative data distributions—can significantly degrade model performance. Detecting such clients without accessing raw data remains a key challenge. We propose wavelet and fourier representations for FL), a detection algorithm that labels malicious clients *before training*, using locally computed compressed representations derived from either the wavelet scattering transform (WST) or the Fourier transform (FT). Both approaches provide low-dimensional, task-agnostic embeddings suitable for unsupervised client separation. A lightweight detector, trained on a distilled public dataset, performs the labeling with minimal communication and computational overhead. While both transforms enable effective detection, WST offers theoretical advantages, such as noninvertibility and stability to local deformations, that make it particularly well-suited to federated scenarios. Experiments on benchmark datasets demonstrate that our method improves both detection accuracy and downstream classification performance compared to existing FL anomaly detection algorithms, validating its effectiveness as an offline alternative to online detection strategies. Source code for this article is publicly available at https://github.com/davedleo/Waffle**

***Index Terms*—Anomaly detection, federated learning (FL), offline detection, signal processing, wavelet scattering transform (WST).**

I. Introduction

FEDERATED Learning (FL) is a key paradigm for enabling decentralized intelligence in large-scale Internet of Things (IoT) systems, allowing numerous devices to train a shared model without exposing raw data [1], [2]. This approach is vital for privacy-sensitive applications in domains like smart cities, industrial automation, connected healthcare systems, and EV charging station networks [3], [4], [5]. However, the success of FL in IoT is threatened by two intertwined challenges: vast data heterogeneity from diverse devices and the system's vulnerability to malicious or faulty clients [6], [7], [8].

In these physically exposed networks, ensuring data integrity is critical. Consider an Industrial IoT deployment monitoring equipment health [9]; sensors may become miscalibrated, suffer damage, or be deliberately compromised to inject anomalous data. Such poisoning attacks can severely degrade the global model, leading to costly operational failures. Current defenses largely fail to address this efficiently. Robust aggregation methods [5], [7], [10] operate *during* the aggregation phase; they merely mitigate the impact of malicious updates rather than eliminating the source, and often fail when attackers form a majority. Similarly, online detection methods [11] monitor clients throughout training, introducing significant communication overhead that is untenable for resource-constrained IoT devices. These approaches are *reactive*—they identify threats only after the training process (and potential damage) has already begun.

To bridge this gap, we propose `Waffle` (wavelet and Fourier representations for FL), a lightweight, *proactive* detector designed to identify and exclude clients with malicious data strictly before FL training begins. By shifting detection prior to the FL training phase, we avoid the heavy communication costs of online monitoring. `Waffle` trains a classifier on stable spectral features—extracted via the Fourier transform (FT) and wavelet scattering transform (WST) [12]—which provide robust representations of client data distributions. The detection is performed using a model trained offline on a public dataset, ensuring efficiency and privacy. Clients only need to compute low-dimensional spectral statistics and send a secure, noninvertible summary to the server. Unlike existing methods, `Waffle` remains effective even when malicious clients form a large majority. Our experiments demonstrate its high efficacy in diverse settings, including under challenging non-Gaussian data attacks, and we showcase its versatility with a proof-of-concept on a natural language processing (NLP) task.

Received 4 January 2026; revised 13 February 2026; accepted 4 March 2026. Date of publication 6 March 2026; date of current version 25 May 2026. The work of Alessandro Licciardi was supported by the Project Piano Nazionale di Ripresa e Resilienza-Next Generation European Union (PNRR-NGEU) from Italian Ministry of University and Research (MUR) under Grant DM 117/2023. The work of Davide Carbone was supported by the Government Funding managed by the National Research Agency through the France 2030 Program under Grant ANR-23-IACL-0008. *(Alessandro Licciardi and Davide Leo contributed equally to this work.) (Corresponding author: Alessandro Licciardi.)*

Alessandro Licciardi is with the Department of Mathematical Sciences, Politecnico di Torino, 10129 Turin, Italy, and also with the Istituto Nazionale di Fisica Nucleare, Sezione di Torino, 10125 Turin, Italy (e-mail: alessandro.licciardi@polito.it).

Davide Leo is with Tynk S.R.L., 10141 Turin, Italy.

Davide Carbone is with the Laboratoire de Physique de l'École normale supérieure ENS, Université PSL, CNRS, Sorbonne Université, Université de Paris, 75005 Paris, France.

This article has supplementary downloadable material available at https://doi.org/10.1109/JIOT.2026.3671698, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2026.3671698

The article is organized as follows: Section II defines the FL setting, the data attacks considered, and the spectral representations (FT and WST). Section III details `Waffle` 's training and detection. Theoretical guarantees are provided in Section IV, showing the benefits of removing malicious clients. Section V reports experimental results validating `Waffle` on benchmark datasets.

A. Related Work and Contributions

1) Wavelet Scattering Transform: WST was introduced by Mallat [12] to construct translation-invariant and deformation-stable representations via cascaded wavelet convolutions with modulus nonlinearities. Bruna and Mallat [13], Bruna [14] formalized scattering networks for image classification, while Andén and Mallat [15] extended the framework to audio signals, establishing stability to time-warping. These properties make WST robust to input perturbations—critical for handling heterogeneous data distributions. Beyond classification, WST has been used for robust signal characterization in bioacoustics [16], [17], astrophysics [18], and fault detection [19], [20]. Hybrid architectures combine scattering with learned filters [21], [22], with efficient implementations available through Kymatio [23]. Recent work applies WST to speech deepfake detection [24] and time-series forecasting [25], while geometric extensions enable applications on graphs and manifolds [26]. To the best of our knowledge, this is the first work applying WST to federated learning (FL) for malicious client detection. Unlike existing methods that monitor model updates during training or require multiround trajectory analysis, we extract WST features directly from raw client data distributions before federated optimization begins, enabling lightweight, one-shot detection that is both model-agnostic and communication-efficient.

2) Malicious Client Detection in FL: Detection-based approaches classify clients as benign or malicious based on anomalies in their updates or data distribution [27]. `FLDetector` [11] identifies malicious clients by analyzing the consistency of their updates over time—benign updates follow predictable patterns, while malicious ones are erratic. `MuDHog` [28] leverages historical update trajectories with model-agnostic meta-learning, and `VAE` [29] uses variational autoencoders to flag deviations from benign distributions. However, these methods suffer from a common structural limitation: they are inherently reactive. They rely on observing multiround update trajectories, which requires significant communication overhead before a verdict is reached. In contrast, our approach enables "one-shot" detection before the expensive training loops begin.

3) Robust Aggregation in FL: Robust aggregation methods aim to mitigate the influence of malicious clients without explicitly identifying them [7], [30]. `KRUM` [10] selects the most central update in ℓ_2 distance, while `TrimmedMean` [7] discards extreme values per coordinate. `FLTrust` [8] enhances robustness by normalizing updates against a trusted server-side dataset. Secure aggregation protocols [31], [32] focus on privacy but not adversarial robustness. Although these mechanisms dampen the impact of attacks, they do not remove compromised nodes, and Consequently, malicious clients continue to drain bandwidth in subsequent rounds; Furthermore, these defenses theoretically degrade when the proportion of attackers exceeds 50%, a constraint our offline filtering approach avoids.

4) Spectral Analysis and Frequency-Based Defenses: Spectral methods analyze updates in the frequency domain to identify anomalies [33], [34], [35]. `FreqFeD` [36] filters high-frequency components in updates assumed to be adversarial noise. `FedSSP` [37] targets backdoor attacks by pruning suspicious spectral patterns in model weights. The primary shortcoming of these works is that they analyze *model updates*—a downstream proxy that can obscure the original data characteristics and requires access to model parameters. By contrast, `Waffle` extracts embeddings directly from client-side *data* distributions, enabling a more precise, model-agnostic detection that is independent of the specific learning architecture.

Our *main contributions* are summarized as follows.

1) We propose `Waffle`, a novel offline detector for identifying clients with data attacks, introducing the use of WST for anomaly detection in FL.
2) We provide a theoretical framework motivating WST and FT as robust data representations. Furthermore, we present some of the first statistical results demonstrating the explicit benefit of excluding malicious clients *before* training in Federated Averaging, proving that removing malicious clients before training yields tighter global error bounds compared to relying solely on robust aggregation.
3) We present experiments on benchmark datasets showing that `Waffle` significantly improves model performance and robustness compared to training with contaminated data or using only robust aggregation.

II. Theoretical Framework

In this section, we introduce the mathematical framework that provides the foundation for our algorithm. Section II-A presents the FL setting and defines the class of attacks considered on clients' data. Section II-B introduces the WST and the FT, recalling their basic properties that are relevant for anomaly detection.

A. Problem Formulation

Consider a standard FL setting [1] with $K \in \mathbb{N}$ clients and a central server. Each client k possesses n_k data samples $\{(x_k^i, y_k^i)\}_{i=1}^{n_k} \sim \mathcal{D}_k$ supported in $\mathcal{X} \times \mathcal{Y}$. The objective of FL is to learn a shared global model θ that generalizes across all clients, by solving the following optimization problem:

$$\theta^* \in \arg\min_{\theta\in\Theta} \frac{1}{N} \sum_{k=1}^{K} n_k \mathcal{L}_k(\theta) \tag{1}$$

where Θ denotes the model's parameter space, $N = \sum_{k=1}^{K} n_k$ is the total number of data samples, and $\mathcal{L}_k$ represents the empirical loss function for client k with respect to its local data distribution $\mathcal{D}_k$. In each communication round $t \in \{1, \dots, T\}$,

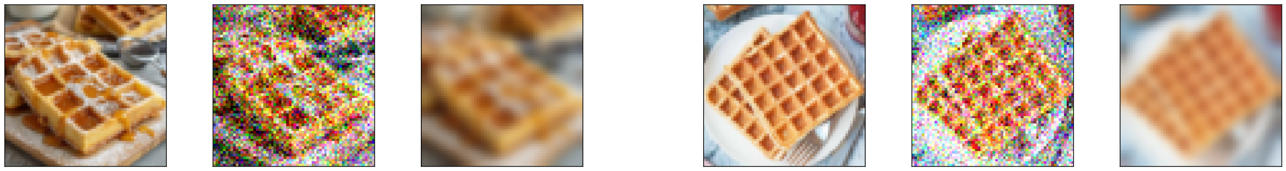

Fig. 1. Examples of attacked data. Two images downloaded from link1 and link2. For each image: *left*: clean client, *center*: noisy attack with magnitude $\sigma = 0.2$, *right*: blur attack with spread $\beta = 11$.

a subset of clients $\mathcal{P}_t$ is randomly selected to participate in training. Each participating client $k \in \mathcal{P}_t$ performs $S \in \mathbb{N}$ local iterations of a stochastic optimizer. Subsequently, clients send their updated parameters to the server, which aggregates these updates to derive a new global model.

A critical challenge in realistic FL deployments is the *non-i.i.d.* nature of client data, which can hinder the convergence and performance of the global model. In this work, we specifically address non-i.i.d. settings where the data distribution discrepancies are caused by malicious clients perturbing their original data samples. This differs from typical attack detection scenarios focusing on model poisoning during training.

1) Type of Attacks: We define two types of feature-level attacks that our algorithms aim to address: noisy and blur attackers. Examples of the effect of these attacks are displayed in Fig. 1. This focus is motivated by the fact that noise and blur are common consequences of real-world faults [38], [39]—such as sensor degradation, miscalibration, or environmental interference–that can subtly compromise data quality and model performance without exhibiting overtly malicious behavior.

Definition 1: Let $k \in [K]$ and $\sigma_k > 0$. Client k is a **noisy attacker** if its data samples are perturbed as $\tilde{x}_k^i = x_k^i + \sigma_k \epsilon_k^i$, where x_k^i is the clean sample, and, $(\epsilon_k^i)_{i=1}^{n_k}$ is a family of independent Wiener processes supported in $\mathcal{X}$.

Let us observe that the severity of the attack is determined by the magnitude of σ_k. Smaller values of σ_k might represent natural noise inherent in data collection or random transformations, requiring careful consideration of what constitutes a "malicious" level of perturbation.

Another feature-wise attack we formally define is the **blur attacker**. This attack is particularly relevant for image or signal data where x_k^i can be treated as a function over $\mathcal{X}$.

Definition 2: Let $k \in [K]$ and $\beta_k > 0$. Client k is a **blurred attacker** if it provides samples perturbed according to a convolution operation

$$\tilde{x}_k^i = x_k^i \star \zeta_k = \int_{\mathcal{X}} x_k^i \left(u'\right) \zeta_k \left(u - u'\right) du' \quad i = 1, \ldots, n_k \tag{2}$$

where $\star$ denotes the convolution operation. Typically, ζ_k is a smooth kernel, and the parameter β_k controls its spread or blur radius.

A common choice for the kernel ζ_k is a Gaussian kernel, and the scalar β_k has a role of controlling the spread of the kernel. similar to noisy attacks, in blur attacks the magnitude of the perturbation is controlled by the parameter β_k, the larger it is, the stronger the perturbation.

B. Representation Operators: WST and FT

In this section, we recall the notion of a representation operator Φ, which maps a signal x (e.g., an image or a time-series) onto a transformed space. This transformation induces a metric $d(x, x') = \|\Phi[x] - \Phi[x']\|$ in the new space [13]. The core idea is that an effective representation operator Φ should possess properties instrumental for accurately detecting and differentiating between data samples. Specifically, for the purpose of identifying perturbed data, Φ should be able to separate distinct data characteristics while exhibiting robustness to common variations like slight translations or small, nonmalicious perturbations. We propose two variants for the representation layer of our detection algorithm: one based on the FT and the other on the WST [12], [13]. The FT is by far the most widely used tool for spectral analysis in signal processing and data science due to its simplicity and interpretability. However, it has been surprisingly underutilized in the context of FL. We therefore include it as an internal baseline in our study, allowing us to contrast its performance against the more structured and hierarchical WST.

1) Fourier Representation: We first formally define the FT.

Definition 3: Let $x \in L^1(\mathcal{X}, du)$, the **FT** of x, denoted by $\mathcal{F}[x]$, is a complex valued function defined as

$$\mathcal{F}[x](\omega) = \int_{\mathcal{X}} x(u)\, e^{-2\pi i (u \cdot \omega)} du. \tag{3}$$

FT can be efficiently computed using the FFT algorithm [40]. Beyond its computational efficiency, the FT offers several critical advantages for feature extraction, particularly in the context of analyzing data perturbations. As a linear operator ($\mathcal{F}[ax + bx'] = a\mathcal{F}[x] + b\mathcal{F}[x']$ for scalars a, b and integrable signals x, x'), the FT maps additive perturbations directly to additive components in the frequency domain. For instance, in the case of a *noisy attacker* where $\tilde{x} = x + \epsilon$, we have $\mathcal{F}[\tilde{x}] = \mathcal{F}[x] + \mathcal{F}[\epsilon]$. This linearity simplifies the analysis of such perturbations. Moreover, the convolution theorem [41] states that convolution in the spatial domain corresponds to point-wise multiplication in the frequency domain ($\mathcal{F}[x \star \delta] = \mathcal{F}[x] \cdot \mathcal{F}[\delta]$). This property is highly advantageous for detecting perturbations induced by *blur attackers*, which are defined as convolutions. By examining the frequency spectrum, different types of data manipulations, like blurring (attenuating high frequencies) or specific noise patterns, reveal distinct signatures. However, FT is an invertible operator: on the one hand, it preserves all information present in the original signal; on the other hand, it allows reconstruction of the original data.

2) Wavelet Scattering Transform: WST is a nonlinear operator that, unlike Fourier-based representations, has been designed to be stable to additive perturbations, locally translation invariant and stable to small continuous deformation. Moreover, the fact that WST is not invertible makes it particularly attractive for privacy-enhancing applications in FL, as reconstructing the original input data from the scattering coefficients is a challenging task. Following the construction in [12], [13] we define the WST and discuss its most relevant properties.

Let $\psi(u) \in L^2(\mathcal{X}, du)$ be a function referred to as the **mother wavelet**, and let $\{a^j\}_{j\in\mathbb{Z}}$ be a family of scale factors defined with respect to a fixed scalar $a > 1$. Let $r \in G$ denote a discrete rotation, where G is the group of discrete rotations acting on the domain $\mathcal{X}$. The jth **wavelet function** is then defined as $\psi_j(u) = a^{-dj}\psi(a^{-j}r^{-1}u)$. For a fixed maximal depth $J \in \mathbb{Z}$, we define the set of admissible scale-rotation operators as $\Lambda_J = \{\lambda = a^j r : |\lambda| = a^j < 2^J\}$. In most implementations, Morlet wavelets are employed as the mother wavelet, and the scale factor is typically chosen as $a = 2^{1/Q}$ for some $Q \in \mathbb{N}$ [23].

To streamline notation, following [12], we introduce the **propagator operator**, which acts on a signal $x \in L^1(\mathcal{X})$ by cascading modulus and convolution operations. Given a path of scale-rotation operators $p = (\lambda_1, \lambda_2)$, the propagator applied to x is defined as

$$U\left[p\right]x = \left|\,\left|x \star \psi_{\lambda_1}\right| \star \psi_{\lambda_2}\right|.$$

The definition of the WST naturally follows.

Definition 4: Let $p = (\lambda_1, \ldots, \lambda_m) \subset \Lambda_J$ be a path of length m. For any signal $x \in L^1(\mathcal{X})$, the WST along p is defined as

$$S_J\left[p\right]x = U\left[p\right]x \star \phi_J \tag{4}$$

where ϕ_J is a low-pass filter rescaled to recover low-frequency content.

The WST representation shares structural similarities with convolutional neural networks (CNNs), with the key distinction that the wavelet filters are fixed rather than learned. The WST defines a norm with properties desirable for detection and classification. Notably, the operator is **nonexpansive**: for any $x, x' \in L^2(\mathcal{X}, du)$, the following inequality holds:

$$\left\|S_J\left[p\right]x - S_J\left[p\right]x'\right\| \leq \left\|x - x'\right\|. \tag{5}$$

This implies that small, nonadversarial perturbations do not substantially affect the representation.

Additionally, WST is **translation invariant** in the limit: for a translated signal $x_c(u) = x(u - c)$ with $c \in \mathcal{X}$, we have

$$\lim_{J\to\infty} \left\|S_J\left[p\right]x - S_J\left[p\right]x_c\right\| = 0.$$

Finally, the WST is **Lipschitz continuous** with respect to small C^2-diffeomorphisms. That is, if a signal x undergoes a smooth deformation with small norm, the resulting change in the WST representation remains bounded.

III. Malicious Client Detector: Waffle

This section details the architecture and training of our server-side detector, wavelet, and Fourier representations for FL), designed to identify clients contributing potentially harmful updates based on their data characteristics. `Waffle` is a parametric classification model, trained offline on a generated auxiliary dataset $\mathcal{D}^{\text{aux}}$ to distinguish between benign and malicious clients. It operates by analyzing aggregated, privacy-preserving spectral embeddings of client data distributions.

Algorithm 1 `Waffle` Offline Training

Require: Auxiliary dataset $\mathcal{D}^{\text{aux}}$, Number of epochs E, Number of fictitious clients $\tilde{K}$, Number of top PCs r, Spectral operator Φ, Learning rate η
Ensure: Trained detector weights w
Initialize detector weights w
for $e = 1 \ldots E$ **do**
  // **Phase 1: Simulation**
  $\mathcal{D}_e^{\text{simulated}} \leftarrow \text{SimulateAttackedData}(\mathcal{D}^{\text{aux}})$ ▹Applies random attacks to $\mathcal{D}^{\text{aux}}$
  $\{(\mathcal{D}_k, \mu_k)\}_{k=1}^{\tilde{K}} \leftarrow \text{PartitionData}(\mathcal{D}_e^{\text{simulated}}, \tilde{K})$ ▹Creates $\tilde{K}$ clients with labels
  // **Phase 2: Feature Extraction**
  Initialize epoch dataset $\mathcal{S}_e = \emptyset$ ▹Stores (φ_k, μ_k) pairs
  for $k = 1 \ldots \tilde{K}$ **do**
    $\{x_k^i\}_{i=1}^{n_k} \leftarrow \mathcal{D}_k$
    Compute PCA-derived representation $\hat{x}_k$ from $\{x_k^i\}$▹ Eq. (6)
    Compute spectral embedding $\varphi_k \leftarrow |\Phi[\hat{x}_k]|$ ▹Apply FT or WST to $\hat{x}_k$
    Add (φ_k, μ_k) to $\mathcal{S}_e$
  end for
  // **Phase 3: Optimization**
  $w \leftarrow \text{Opt}(\mathcal{L}_{\text{BCE}}(w; \mathcal{S}_e))$ ▹Optimization step
end for
return w

A. *Offline Detector Training*

The training of the `Waffle` detector is conducted entirely offline, prior to the FL process. This approach offers several advantages: it avoids interfering with live FL rounds, allows for controlled generation of diverse malicious scenarios, and ensures the detector is fully trained and ready when FL begins. Consistent with common practices in FL frameworks utilizing auxiliary data [42], the server has access to a representative auxiliary dataset $\mathcal{D}^{\text{aux}}$. Algorithm 1 summarizes the procedure.

To improve the robustness of the detector, the training process is structured into epochs. In each epoch $e \in \{1, \ldots, E\}$, we simulate a complete FL round by generating a fresh set of $\tilde{K}$ fictitious clients. This dynamic generation strategy [43] ensures the model encounters diverse data distributions and attack variations, mitigating overfitting. The procedure within each epoch is organized into three logical phases: data simulation, feature extraction, and model optimization.

1) Phase 1: Attack Simulation and Client Generation: The first phase focuses on generating a labeled dataset of fictitious clients. For each sample $x \in \mathcal{D}^{\text{aux}}$, the server decides whether to simulate an attack based on a Bernoulli trial ($p = 1/2$). If selected for attack, a perturbation type is chosen uniformly at random as follows.

1) *Blurring:* A severity parameter $\beta \sim \mathrm{Unif}(\beta_0, \beta_1)$ is sampled to apply a blurring operation (Definition 2), simulating low-quality or obscured sensor data.
2) *Noise Injection:* A noise variance $\sigma \sim \mathrm{Unif}(\sigma_0, \sigma_1)$ is sampled to apply additive noise (Definition 1), simulating sensor corruption or adversarial perturbations.

Once the data is processed, the dataset is partitioned among $\tilde{K}$ fictitious clients, equally split into benign (clean data) and malicious (attacked data) groups. Let $\{x_k^i\}_{i=1}^{n_k}$ denote the resulting local dataset for the kth fictitious client.

2) Phase 2: Privacy-Preserving Feature Extraction: In the second phase, we compute the spectral embedding φ_k for each client, mirroring the privacy-preserving protocol of the live system. First, we apply PCA [44] to the client's local dataset $\{x_k^i\}_{i=1}^{n_k}$ to analyze the covariance structure and extract the top r principal components v_k^i with eigenvalues λ_k^i. We then compute a compact representation vector

$$\hat{x}_k = \sum_{i=1}^{r} \alpha_k^i v_k^i, \quad \text{with} \quad \alpha_k^i = \frac{\lambda_k^i}{\sum_{j=1}^{r} \lambda_k^j}. \tag{6}$$

Next, a spectral operator Φ (WST or FT) is applied to $\hat{x}_k$ to capture frequency and texture anomalies introduced by the attacks. The final embedding is given by $\varphi_k = |\Phi[\hat{x}_k]|$. This two-step process—PCA for structural summarization followed by spectral analysis—produces a fixed-size feature vector that characterizes the data distribution without exposing raw samples.

3) Phase 3: Detector Optimization: Finally, the collected embeddings and their labels $\{(\varphi_k, \mu_k)\}_{k=1}^{\tilde{K}}$ form the training batch for the current epoch, where $\mu_k \in \{\text{Benign}, \text{Attacker}\}$. The detector weights w are updated using a stochastic optimizer (e.g., SGD, Adam) to minimize a binary classification loss, such as binary cross-entropy (BCE) [45], between the detector's prediction based on φ_k and the ground-truth label μ_k. Consistently with established practices in FL frameworks that leverage server-side data to mitigate statistical heterogeneity [46], [47], we assume the server has access to a representative auxiliary dataset $\mathcal{D}^{\text{aux}}$. The selection of $\mathcal{D}^{\text{aux}}$ is guided by **domain alignment** criteria [46], [47]. While the server cannot access private client samples, the learning task is known. Therefore, $\mathcal{D}^{\text{aux}}$ is composed of publicly available data that shares the same modality (e.g., image versus text), resolution, and channel depth as the target client data. Importantly, because `Waffle` detects spectral anomalies (such as high-frequency noise or loss of detail due to blur) rather than semantic class shifts, the auxiliary data need not perfectly match the clients' class distribution. It suffices that the auxiliary images possess similar low-level signal statistics (texture, edges) to allow the detector to learn the spectral signature of the attack patterns.

B. Offline Detection and Filtering

Once the `Waffle` detector model w has been trained offline on the simulated auxiliary dataset $\mathcal{D}^{\text{aux}}$ and prior to the first FL communication round, each client $k \in \{1, \ldots, K\}$ in the federation processes its local training data $\{x_k^i\}_{i=1}^{n_k}$ *privately* on the client device. This processing involves a sequence of steps performed locally. First, each client computes the PCA of their local training samples to derive the representation vector $\hat{x}_k$, as defined in (6). Then, each client computes its spectral embedding $\varphi_k = \Phi[\hat{x}_k]$, by applying the spectral operator Φ (WST or FT).

After completing these local computations and obtaining φ_k, each client k securely transmits only this resulting spectral embedding vector to the server. The server, upon receiving φ_k from each participating client, inputs it into the pretrained `Waffle` detector w. Clients that are classified as malicious by the detector are then excluded from participating in the federated training process for the global model θ. This preemptive filtering step enhances the stability and reliability of the global model training process, leading to potentially faster and more robust convergence by ensuring that aggregation occurs over updates from predominantly benign sources.

Moreover, due to its modular nature, `Waffle` operates as an initial defense layer. The set of clients validated as benign by `Waffle` can proceed with any FL aggregation methods, allowing `Waffle` to be easily combined with other online robust aggregation techniques to further strengthen the overall defense strategy.

IV. Theoretical Guarantees

In this section, we establish a theoretical foundation for our proposed algorithm, `Waffle`. Our primary focus is to demonstrate the benefits of removing adversarial clients in FL scenarios. We show that by filtering out malicious updates, `Waffle` provides a more accurate estimate of the true global model compared to standard `FedAvg` [1], which is susceptible to adversarial poisoning. We provide general error bounds with detailed proofs presented in Appendix A in the Supplementary Material.

Let $\mathcal{B} \subset \{1, \ldots, K\}$ denote the set of benign clients and $\mathcal{M} \subset \{1, \ldots, K\}$ the set of malicious clients in a federated system with K total clients. We assume these sets are disjoint and their union covers all clients, i.e., $\mathcal{B} \cap \mathcal{M} = \emptyset$ and $\mathcal{B} \cup \mathcal{M} = \{1, \ldots, K\}$. To model the heterogeneity and potential adversarial influence in client updates, we adopt the following statistical framework.

Assumption 1: For each benign client $k \in \mathcal{B}$, the local model update θ_k is an independent random variable drawn from a distribution $\rho_k(\bar{\theta}^b, \sigma^b)$. This distribution is centered around a common benign mean $\bar{\theta}^b$ with variance $(\sigma^b)^2$, i.e., $\mathbb{E}[\theta_k] = \bar{\theta}^b$ and $\mathbb{V}ar[\theta_k] = (\sigma^b)^2$. Similarly, for malicious clients $k \in \mathcal{M}$, the local updates θ_k are independent random variables drawn from $\rho_k(\bar{\theta}^m, \sigma^m)$ with $\mathbb{E}[\theta_k] = \bar{\theta}^m$ and $\mathbb{V}ar[\theta_k] = (\sigma^m)^2$.

Assumption 2: We assume that malicious clients exhibit significantly higher update variance compared to benign clients, reflecting a diverse range of attack strategies and the potential for large, destabilizing updates. Formally, we assume $\sigma^m \gg \sigma^b$.

The standard federated averaging estimator is defined as a weighted average of client updates: $\theta_{avg} = 1/K \sum_{k=1}^{K} \theta_k$. Our objective is to obtain an estimator that is unbiased with respect to the benign client distribution, meaning $\mathbb{E}[\theta_{avg}] = \bar{\theta}^b$. We demonstrate that removing malicious clients is crucial for achieving this goal. We analyze two scenarios: one where the

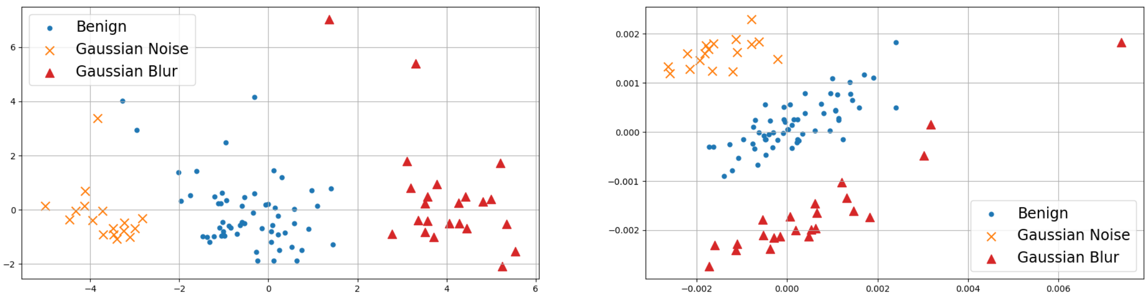

Fig. 2. Client distributions of the φ_k for the Cifar10 dataset with $K = 100$ clients projected onto a 2-D space, for `Waffle` + FT (left), and `Waffle` + WST (right). There are a total of 60 benign clients (dots), and 40 attackers: 20 noisy (crosses) and 20 blurred (triangles). Both methods provide a noticeable separation between the clients.

benign and malicious updates have different means (Lemma 1) and one where they share the same mean but differ in variance (Lemma 2).

Lemma 1: If the benign and malicious client updates have different mean parameter values, i.e., $\bar{\theta}^m \neq \bar{\theta}^b$, then the standard federated averaging estimator θ_{avg} is a **biased estimator** of $\bar{\theta}^b$, meaning $\mathbb{E}[\theta_{avg}] \neq \bar{\theta}^b$.

Lemma 2: Let $\theta_{avg}^{\mathcal{B}} = 1/|\mathcal{B}| \sum_{k\in\mathcal{B}} \theta_k$ be the federated averaging estimator computed using only benign client updates. Under Assumption 2, if $(\sigma^m)^2 > (2 + |\mathcal{M}|/|\mathcal{B}|)(\sigma^b)^2$, then the variance of the standard federated averaging estimator is higher than that of our estimator: $\mathbb{V}ar[\theta_{avg}] \geq \mathbb{V}ar[\theta_{avg}^{\mathcal{B}}]$.

Lemmas 1 and 2 provide the foundation for the following proposition, which formally establishes the advantage of removing malicious clients from the federated aggregation process.

Proposition 1: Under Assumptions 1 and 2, removing malicious clients (those in $\mathcal{M}$) from the federation yields a superior estimator of the global model. Specifically, the resulting estimator is unbiased (in the sense of Lemma 1) and exhibits reduced variance (as shown in Lemma 2), leading to improved model accuracy and robustness.

We observe that Assumption 2 assumes that $\sigma^m \gg \sigma^b$, characterizing active destabilization attacks where malicious updates introduce significant noise. We briefly discuss the implications if this condition does not hold below.

1) *Biased Updates ($\bar{\theta}^m \neq \bar{\theta}^b$):* If the mean of the malicious updates differs from the benign mean, Lemma 1 holds regardless of the variance. In this case, removing malicious clients is mandatory to eliminate the systematic bias in the global estimator θ_{avg}, irrespective of whether σ^m is large or small.
2) *Unbiased, Low-Variance Updates ($\bar{\theta}^m = \bar{\theta}^b, \sigma^m \leq \sigma^b$):* In this theoretical edge case, malicious clients provide updates that are centered on the true objective and have variance comparable to or lower than benign clients. Mathematically, these updates are indistinguishable from high-quality benign contributions. Including them would actually *reduce* the variance of the global estimator without introducing bias. Therefore, detection in this regime is unnecessary, as such clients do not degrade the learning process.

Thus, our theoretical analysis and the proposed `Waffle` detector focus on the critical regimes where malicious contributions are actively harmful—either by shifting the model parameters (bias) or by destabilizing convergence (high variance).

V. Experiments

In this section, we present experimental results on widely used FL benchmark datasets [48], [49], [50], comparing the performance of `Waffle` in its two variants—one using the WST representation and the other using FT—with established baselines from the Byzantine-resilient FL literature. Details on implementation settings, datasets, and models are provided in Appendix D in the Supplementary Material.

Section V-A evaluates the detection performance of the two variants of `Waffle`, highlighting the differences between the WST and FT representations. In Section V-B, we compare `Waffle` against standard Byzantine-resilient FL baselines, including `FedAvg` [1], `Krum` and `mKrum` [10], `GeoMed` [51], and `TrimmedMean` [7]. Additionally, we demonstrate that `Waffle` can be integrated with any aggregation algorithm, improving their performance. Further experiments, comparisons, and code release details are reported in Appendix D in the Supplementary Material, and the metrics used for evaluation—both for detection and classification—are detailed in Appendix E in the Supplementary Material.

A. `Waffle`: WST Versus Fourier

We compare the detection performance of `Waffle` to assess the differences between the WST and FT representations. As illustrated in Fig. 2, both representations yield a clear separation between benign and malicious clients. The visualizations—obtained via 2-D PCA embeddings—show that the method effectively distinguishes between the different attacker groups and benign clients, regardless of the chosen representation. Fig. 3 reports the average embedding for the spectral features, comparing FT and WST representation. However, as shown in Table I, the quantitative results at the client level differ between the two variants. We report standard detection metrics: precision, $F1$ score, recall, and accuracy [52], with 40% and 90% malicious clients. The WST variant consistently achieves higher precision and $F1$ scores, while the FT variant tends to yield higher recall. In the context

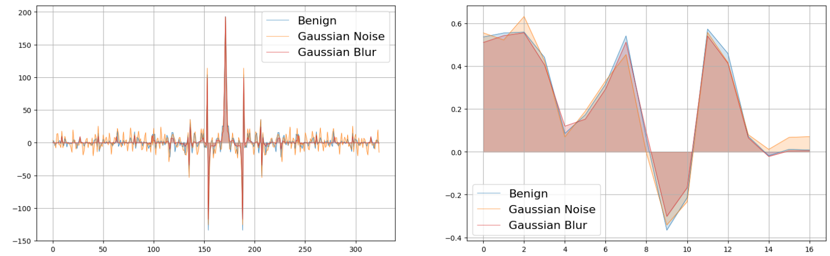

Fig. 3. Embeddings φ_k produced by `Waffle` for three clients (blur attacker, noise attacker, and benign) on CIFAR-10. The left panel shows the embeddings obtained using FT, while the right panel shows those obtained using WST.

TABLE I

CLIENT DETECTION. COMPARISON BETWEEN VARIANTS OF WAFFLE USING WST AND FT REPRESENTATIONS, UNDER TWO ATTACK SCENARIOS (40% TOP, 90% BOTTOM). METRICS ($F1$ SCORE, PRECISION, RECALL, ACCURACY [52]) REFER TO THE DETECTION OF MALICIOUS CLIENTS

	Method	FashionMNIST				CIFAR-10				CIFAR-100			
		F1	Prec.	Rec.	Acc.	F1	Prec.	Rec.	Acc.	F1	Prec.	Rec.	Acc.
40%	`Waffle - FT`	65.1 ±3.1	59.9±3.1	**69.1** ±3.1	**69.2**±3.1	80.2±2.6	69.1±2.6	**96.1**±2.6	67.0±2.6	55.1 ±3.2	40.5±2.6	**89.7**±2.6	44.1±2.6
	`Waffle - WST`	**72.7** ±1.1	**96.3**±1.1	58.2±1.1	**82.4**±2.6	**95.2**±1.0	**97.6**±1.0	92.9±1.0	**96.1** ±1.0	**83.0** ±1.2	**93.1**±1.2	75.1±1.2	**87.0**±1.2
90%	`Waffle - FT`	**80.9**±2.6	94.2 ±2.6	**70.7**±2.6	**71.2**±2.6	**93.3**±1.6	89.2±1.6	**95.7**±1.6	**86.2**±1.6	89.0 ±1.6	88.2±1.6	**88.4**±1.6	**81.1**±1.6
	`Waffle - WST`	65.6 ±0.2	**100.0**±0.0	49.1±0.2	54.0±0.2	91.1±0.5	**100.0**±0.0	83.8 ±0.5	87.0±0.5	**88.1**±0.3	**100.0**±0.0	68.3±0.3	72.2±0.3

TABLE II

COMPARISON BETWEEN BASELINES FOR DETECTING MALICIOUS CLIENTS AND WAFFLE (WITH BOTH WST AND FT) WITH 2σ ERROR BARS. WE CONSIDER AS UPPER-BOUND FOR ALL METHODS FEDAVG TRAINED ON THE WHOLE BENIGN FEDERATION WITHOUT MALICIOUS CLIENTS—FASHIONMNIST 75.5 $\pm$ 1.7, CIFAR-10 50.3 $\pm$ 0.5, AND CIFAR-100 17.0 $\pm$ 1.3

Dataset	Setting	`FedAvg`	`Krum`	`mKrum`	`GeoMed`	`TrimmedMean`
FashionMNIST	w/o detector	73.7 $\pm$ 1.3	73.8 $\pm$ 1.1	72.5 $\pm$ 4.0	73.4 $\pm$ 1.7	74.6 $\pm$ 0.4
	`Waffle - WST`	**74.9 $\pm$ 1.9**	70.2 $\pm$ 0.4	74.2 $\pm$ 0.9	74.6 $\pm$ 1.6	74.7 $\pm$ 1.8
	`Waffle - FT`	73.8 $\pm$ 1.1	71.4 $\pm$ 2.0	74.6 $\pm$ 0.4	74.7 $\pm$ 1.0	**74.9 $\pm$ 0.5**
CIFAR-10	w/o detector	48.7 $\pm$ 1.3	44.8 $\pm$ 2.2	46.2 $\pm$ 5.9	48.3 $\pm$ 0.5	48.1 $\pm$ 0.4
	`Waffle - WST`	**49.6 $\pm$ 0.3**	46.2 $\pm$ 0.6	49.5 $\pm$ 0.6	49.0 $\pm$ 1.4	49.5 $\pm$ 0.8
	`Waffle - FT`	47.1 $\pm$ 0.4	43.8 $\pm$ 1.8	46.7 $\pm$ 1.3	47.2 $\pm$ 0.3	46.8 $\pm$ 1.1
CIFAR-100	w/o detector	16.4 $\pm$ 0.1	10.1 $\pm$ 0.8	14.6 $\pm$ 0.7	16.4 $\pm$ 0.7	16.5 $\pm$ 1.1
	`Waffle - WST`	**16.5 $\pm$ 1.0**	8.8 $\pm$ 2.2	14.5 $\pm$ 0.7	16.3 $\pm$ 0.3	16.2 $\pm$ 0.5
	`Waffle - FT`	11.6 $\pm$ 0.2	7.6 $\pm$ 0.6	10.6 $\pm$ 0.7	12.1 $\pm$ 0.3	10.6 $\pm$ 0.5

of malicious client detection, higher recall is often desirable, as it reduces the likelihood of overlooking faulty clients. Table I highlights the robustness of `Waffle`: unlike most Byzantine-resilient FL methods, it maintains strong predictive performance even when the vast majority of clients are malicious. Notably, in the extreme case with 90% adversarial clients, `Waffle` with WST achieves 100% precision across all datasets.

Experimental results reveal that while the Fourier-based detector is computationally efficient, its performance varies significantly across different datasets and attack intensities. This inconsistency can be attributed to the theoretical limitations of the Fourier modulus. While invariant to global translation, the FT is unstable to local deformations: small spatial warps or high-frequency noise can cause large fluctuations in the spectral coefficients [12]. In contrast, the WST separates scales and linearizes small deformations, providing a representation that is Lipschitz continuous to such distortions. This structural stability explains why the WST variant consistently outperforms the FT baseline.

To mitigate the limitations of individual transforms, future work could explore **hybrid spectral architectures** that fuse global Fourier features with local Wavelet descriptors. Such a multiview approach could potentially enhance detection sensitivity by capturing both the absolute frequency content (FT) and the deformation-robust texture statistics (WST), as presented in [16] and [18].

TABLE III

PERFORMANCE COMPARISON WITH OTHER DETECTION METHODS UNDER A RANDOM BLOCK ATTACK WITH 40% MALICIOUS CLIENTS

Dataset	`FedAvg`	`Waffle` -WST	`Waffle` -FT	`FLDetector`	`VAEDetector`
FashionMNIST	73.7 ± 1.3	$\mathbf{74.9 \pm 1.9}$	73.8 ± 1.1	71.4 ± 0.9	73.1 ± 0.8
CIFAR-10	48.7 ± 1.3	$\mathbf{49.6 \pm 0.3}$	47.1 ± 0.4	45.4 ± 0.6	47.0 ± 0.5
CIFAR-100	16.4 ± 0.1	$\mathbf{16.5 \pm 1.0}$	11.6 ± 0.2	16.2 ± 0.1	15.5 ± 0.7

B. *Comparison With Baselines and Orthogonality of `Waffle`*

In this section, we compare `Waffle` with established Byzantine-resilient FL methods, highlighting its advantages in two complementary settings: 1) we evaluate the impact of applying the two `Waffle` variants to `FedAvg`, compared to using different aggregation rules without detection and 2) we assess the effect of applying `Waffle` on top of robust aggregation algorithms. As shown in Table II, the WST variant of `Waffle` combined with `FedAvg` consistently outperforms all baselines across all datasets. Furthermore, `Waffle` improves the performance of each aggregation method it is applied to,

TABLE IV

PERFORMANCE UNDER RANDOM BLOCK ATTACK. WE REPORT MEAN TEST ACCURACY AND 2-SIGMA ERROR BARS OVER MULTIPLE RUNS. WE CONSIDER AS UPPER-BOUND FOR ALL METHODS FEDAVG TRAINED ON THE WHOLE BENIGN FEDERATION WITHOUT MALICIOUS CLIENTS—FASHIONMNIST 75.5 $\pm$ 1.7, CIFAR-10 50.3 $\pm$ 0.5, AND CIFAR-100 17.0 $\pm$ 1.3

Dataset	`FedAvg`	`Krum`	`MultiKrum`	`TrimmedMean`	`GeoMed`	`Waffle-WST`	`Waffle-FT`
CIFAR-10	48.7 ± 1.3	44.5 ± 0.2	47.8 ± 0.2	47.9 ± 0.3	48.2 ± 0.2	$\mathbf{49.8 \pm 0.2}$	48.5 ± 0.2
CIFAR-100	16.4 ± 0.1	9.4 ± 0.1	15.3 ± 0.1	16.2 ± 0.1	15.1 ± 0.1	$\mathbf{16.9 \pm 0.1}$	16.3 ± 0.1
Fashion-MNIST	73.7 ± 1.3	74.3 ± 0.3	71.7 ± 0.3	75.0 ± 0.2	71.4 ± 0.3	$\mathbf{75.4 \pm 0.5}$	75.3 ± 0.6

demonstrating its orthogonality to the choice of aggregator. These results indicate that `Waffle` is effective in identifying and removing malicious clients without compromising benign contributions. In contrast, the FT variant exhibits more variable performance, further confirming the suitability of WST representations for this detection task. For reference, we also report the test accuracy of `FedAvg` trained on a clean federation (i.e., without malicious clients, corresponding to $\theta_{avg}^{\mathcal{B}}$ in the notation of Lemma 2): FashionMNIST 75.08%, CIFAR-10 50.24%, CIFAR-100 17.72%. These values demonstrate that `Waffle` enables recovery of near-optimal performance, effectively neutralizing the impact of adversarial clients.

C. Comparison With Anomaly Detection Baselines

To provide a comprehensive comparison, we also benchmark our approach against other recent detection methods from the literature, namely `FLDetector` [11] and `VAEDetector` [29]. These methods operate as online techniques, analyzing model updates across multiple training rounds to identify malicious behavior. We evaluate their performance in a scenario with 60% benign clients under the same random block attack. The accuracy achieved by `FedAvg` when integrated with these detectors is reported in Table III. The results show that under this challenging, non-Gaussian attack scenario, these benchmarks were unable to reliably detect the attacks, leading to a significant drop in performance compared to our method.

D. `Waffle` With Non-Gaussian Attacks

In Section II, we formalized two scenarios of Gaussian attacks (noisy and blurred clients). In this section, we extend our evaluation to non-Gaussian attacks. The primary framework we analyze consists of an attack in which a random subset of pixels of each client's data is perturbed; in this case, 50% are substituted with red pixels.

However, the `Waffle` framework is not limited to detecting these types of attacks. To validate its robustness against more complex, non-Gaussian structural attacks, we conducted further experiments. In this new scenario, malicious clients apply a random dropout attack on part of the image, where 50% of the image pixels, grouped into small random blocks, are set to zero. This introduces sharp, non-Gaussian artifacts that are structurally different from simple noise. `Waffle-WST` obtained an almost perfect detection performance, as summarized in Table IV.

We report in Table IV the results of the random block attack across CIFAR-10, CIFAR-100, and Fashion-MNIST, including standard robust aggregation baselines.

TABLE V

MODEL ACCURACY ON THE NLP TASK UNDER A COMPOSITE SHIFT-AND-NOISE ATTACK. THE TASK IS CLASSIFICATION OF SENTIMENTS, THEREFORE THE EVALUATION METRIC IS STILL ACCURACY

Scenario and method	**Test Accuracy (%)**
`FedAvg` w/o malicious clients (No Attack)	44.81 ± 2.1
`FedAvg` w/ `Waffle` -WST (Under Attack)	42.71 ± 1.9
`FedAvg` w/ `Waffle` -FT (Under Attack)	41.91 ± 1.6
`FedAvg` w/o Detector (Under Attack)	38.53 ± 2.0

As expected from our theoretical results, if the detector is perfect, we reach a performance of the federated training that is comparable to the situation without malicious clients. The `WST` variant is particularly effective, as it is designed to capture local structural information and textures. The random dropout attack fundamentally disrupts these local patterns, creating a strong and detectable signal for our framework. The benefit of `Waffle`, especially the WST variant, is particularly prominent on color images (CIFAR-10/100), where the attack disrupts chromatic and texture patterns that `WST` is well-suited to detect. In contrast, Fashion-MNIST consists of grayscale images, where the attack is more subtle and less disruptive to local statistics. Nonetheless, `Waffle-WST` still achieves performance very close to the clean-case baseline. This demonstrates that `Waffle` is a robust solution capable of identifying a broader class of feature-level data integrity attacks beyond simple Gaussian perturbations.

E. `Waffle` in NLP Tasks

As a proof of concept for tasks beyond computer vision, we extend our evaluation to a NLP task. We do not compare against other baselines here, as this is intended to demonstrate the versatility of our framework. For this experiment, we implemented a composite Shift-and-Noise Attack on the 50-D GloVe embeddings [53] in its most recent version [54] for 40% of the 100 clients in the federation. The attack consists of two components: 1) applying random permutations to the embedding vectors and 2) adding Gaussian noise. Our `Waffle-WST` method demonstrated strong detection capabilities against this attack, achieving an $F1$-score of 0.73 and, notably, a perfect precision of 1.0, ensuring no honest clients were penalized. This successful detection directly translated to a significant performance recovery in the global model. As shown in Table V, the `Waffle` detector is able to raise the final test accuracy from a compromised 38.53% (without our detector) to 42.71%. This result brings the model's performance remarkably close to the ideal, attack-free scenario of 44.81%.

VI. Conclusion

We propose `Waffle`, a novel offline algorithm to detect malicious client data in FL before training begins. Exploiting stable spectral features extracted via the WST and FT, our method enables robust anomaly detection from private, low-dimensional client-side summaries calibrated on publicly available data. By filtering out compromised clients prior to the aggregation process, `Waffle` significantly improves convergence speed, final model accuracy, and robustness to data contamination. Our benchmarks show it achieves near-perfect precision, even in extreme scenarios with up to 90% malicious clients, outperforming strategies that rely solely on robust aggregation.

A key advantage of `Waffle` is its role as a proactive and complementary security layer. By specializing in the detection of data-level attacks, it acts as an essential first line of defense, sanitizing the client pool before resource-intensive training begins. This model-agnostic approach is not intended to replace in-training defenses but rather to fortify them. It can be seamlessly integrated with existing FL defenses, such as robust aggregation mechanisms that target model-level threats, to create a more comprehensive, multilayered security architecture against a wider spectrum of attacks.

This early-detection mechanism also yields substantial practical benefits by reducing training time, communication overhead, and energy consumption—factors that are crucial in large-scale and resource-constrained deployments, like IoT. By enhancing the robustness, trustworthiness, and efficiency of the training pipeline, our method helps pave the way for secure FL deployments in sensitive domains like connected healthcare, autonomous systems, and smart infrastructure.

Future work will focus on extending `Waffle` to defend against more sophisticated threats, including backdoor attacks, model poisoning, and Sybil attacks. In parallel, we plan to adapt the approach to support diverse neural architectures capable of handling more complex and high-dimensional datasets, such as CIFAR-100 or ImageNet-scale benchmarks. These directions aim to broaden the applicability and impact of `Waffle` in advancing secure and efficient decentralized machine learning.

Acknowledgment

Alessandro Licciardi and Davide Carbone were worked under the auspices of Italian National Group of Mathematical Physics (GNFM) of INdAM. Alessandro Licciardi expresses his gratitude to Prof. Lamberto Rondoni for the valuable discussions and support. Davide Carbone expresses their gratitude to Marylou Gabrié for the support.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[2] K. Bonawitz et al., "Towards federated learning at scale: System design," in *Proc. Mach. Learn. Syst.*, vol. 1, 2019, pp. 374–388.

[3] R. S. Antunes, C. André da Costa, A. Küderle, I. A. Yari, and B. Eskofier, "Federated learning for healthcare: Systematic review and architecture proposal," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, pp. 1–23, Aug. 2022.

[4] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated Learning: Privacy and Incentive*. Cham, Switzerland: Springer, 2020, pp. 240–254.

[5] Y. Liu et al., "Eliminate conflicts and attacks: Fair and robust federated learning for anomaly detection of charging stations," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–13, 2025, doi: 10.1109/TITS.2025.3579885.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, 2020, pp. 429–450.

[7] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.

[8] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2021, pp. 1–18.

[9] M. Dzaferagic, N. Marchetti, and I. Macaluso, "Fault detection and classification in industrial IoT in case of missing sensor data," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8892–8900, Jun. 2022.

[10] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[11] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2022, pp. 2545–2555.

[12] S. Mallat, "Group invariant scattering," *Commun. Pure Appl. Math.*, vol. 65, no. 10, pp. 1331–1398, Oct. 2012.

[13] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.

[14] J. Bruna, "Scattering representations for recognition," Ph.D. dissertation, Ecole Polytechnique X, Palaiseau, France, Feb.~2013. [Online]. Available: https://pastel.hal.science/pastel-00905109

[15] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4114–4128, Aug. 2014.

[16] A. Licciardi and D. Carbone, "WhaleNet: A novel deep learning architecture for marine mammals vocalizations on watkins marine mammal sound database," *IEEE Access*, vol. 12, pp. 154182–154194, 2024.

[17] A. Licciardi, D. Carbone, and L. Rondoni, "Wavelet scattering operators for multiscale processes: The case study of marine mammal vocalizations," in *Proc. Int. Conf. Nonlinear Dyn. Appl.* Cham, Switzerland: Springer, 2024, pp. 173–191.

[18] A. Licciardi, D. Carbone, L. Rondoni, and A. Nagar, "Wavelet scattering transform for gravitational wave analysis: An application to glitch characterization," *Phys. Rev. D, Part. Fields*, vol. 111, no. 8, Apr. 2025, Art. no. 084044.

[19] T. Bourgana, R. Brijder, T. Ooijevaar, and A. P. Ompusunggu, "Wavelet scattering network based bearing fault detection," in *Proc. PHM Soc. Eur. Conf.*, 2021, vol. 6, no. 1, p. 8.

[20] H. Khan, A. Sharma, N. Upadhyay, and V. Shivhare, "Bearing defects classification using wavelet time scattering features and ensemble techniques," in *Proc. Unified Int. Conf. Emerg. Technol. Cyber-Phys. Syst. Ind. AI*. Cham, Switzerland: Springer, 2024, pp. 667–675.

[21] E. Oyallon and S. Mallat, "Deep roto-translation scattering for object classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2865–2873.

[22] E. Oyallon, E. Belilovsky, S. Zagoruyko, and M. Valko, "Compressing the input for CNNs with the first-order scattering transform," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 301–316.

[23] M. Andreux et al., "Kymatio: Scattering transforms in Python," *J. Mach. Learn. Res.*, vol. 21, no. 60, pp. 1–6, 2020.

[24] X. Xuan, D. Carbone, R. Pandey, W. Zhang, and T. H. Kinnunen, "WST-X series: Wavelet scattering transform for interpretable speech deepfake detection," 2026, *arXiv:2602.02980*.

[25] W. Li, "ScatterFusion: A hierarchical scattering transform framework for enhanced time series forecasting," 2026, *arXiv:2601.20401*.

[26] J. Chew et al., "Geometric scattering on measure spaces," *Appl. Comput. Harmon. Anal.*, vol. 70, May 2024, Art. no. 101635.

[27] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," 2018, *arXiv:1808.04866*.

[28] A. Gupta, T. Luo, M. V. Ngo, and S. K. Das, "Long-short history of gradients is all you need: Detecting malicious and unreliable clients in federated learning," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, 2022, pp. 445–465.

[29] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," 2020, *arXiv:2002.00211*.

[30] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 3521–3530.
[31] P. Mai, Y. Pang, and R. Yan, "RFLPA: A robust federated learning framework against poisoning attacks with secure aggregation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024, pp. 104329–104356.
[32] H. Lycklama, L. Burkhalter, A. Viand, N. Küchler, and A. Hithnawi, "RoFL: Robustness of secure federated learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2023, pp. 453–476.
[33] Z. Wang et al., "Revisiting VAE for unsupervised time series anomaly detection: A frequency perspective," in *Proc. ACM Web Conf.*, May 2024, pp. 3096–3105.
[34] C.-H. Chan and G. K. H. Pang, "Fabric defect detection by Fourier analysis," *IEEE Trans. Ind. Appl.*, vol. 36, no. 5, pp. 1267–1276, Sep./Oct. 2000.
[35] R. Tao, X. Zhao, W. Li, H.-C. Li, and Q. Du, "Hyperspectral anomaly detection by fractional Fourier entropy," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4920–4929, Dec. 2019.
[36] H. Fereidooni, A. Pegoraro, P. Rieger, A. Dmitrienko, and A.-R. Sadeghi, "FreqFed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning," 2023, *arXiv:2312.04432*.
[37] Y. Chen and Z. Tan, "FedSSP: Federated graph learning with spectral knowledge and personalized preference," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, Dec. 2024, pp. 1–21. [Online]. Available: https://proceedings.neurips.cc/paper/2024/hash/3d226fb8fbd6ee6ec70d0427f1319707-Abstract-Conference.html
[38] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Trans. Sensor Netw.*, vol. 6, no. 3, pp. 1–39, Jun. 2010.
[39] Y. Peng, Z. Tang, G. Zhao, G. Cao, and C. Wu, "Motion blur removal for uav-based wind turbine blade images using synthetic datasets," *Remote Sens.*, vol. 14, no. 1, p. 87, Dec. 2021.
[40] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.
[41] R. N. Bracewell, "The Fourier transform," *Sci. Amer.*, vol. 260, no. 6, pp. 86–95, 1989.
[42] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," 2018, *arXiv:1811.10959*.
[43] W. Bao, H. Wang, J. Wu, and J. He, "Optimizing the collaboration structure in cross-silo federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 1718–1736.
[44] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.
[45] . U. R. Dr. A, "Binary cross entropy with deep learning technique for image classification," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020.
[46] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. NIPS*, vol. 33, 2020, pp. 2351–2363.
[47] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "FedAUX: Leveraging unlabeled auxiliary data in federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 5531–5543, Sep. 2023.
[48] S. Caldas et al., "LEAF: A benchmark for federated settings," 2018, *arXiv:1812.01097*.
[49] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[50] A. Krizhevsky, V. Nair, and G. Hinton. *CIFAR-10 (Canadian Institute for Advanced Research)*. Accessed: 2010. [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html
[51] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 1–25, 2017.
[52] J. Lever, "Classification evaluation: It is important to understand both what a classification metric expresses and what it hides," *Nature Methods*, vol. 13, no. 8, pp. 603–605, 2016.
[53] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[54] R. Carlson, J. Bauer, and C. D. Manning, "A new pair of GloVes," 2025, *arXiv:2507.18103*.
[55] M. Bari, A. Ambaw, and M. Doroslovacki, "Comparison of machine learning algorithms for raw handwritten digits recognition," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Perth, WA, Australia, Oct. 2018, pp. 1512–1516.
[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[57] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015.
[58] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.
[59] D. Goswami, S. Magistri, K. Wang, B. Twardowski, A. D. Bagdanov, and J. van de Weijer, "Covariances for free: Exploiting mean distributions for training-free federated learning," in *Proc. 39th Annu. Conf. Neural Inf. Process. Syst.*, 2025, pp. 1–28.

Alessandro Licciardi received the B.S. degree (cum laude) in mathematics and the M.Sc. degree (cum laude) in mathematical engineering from Politecnico di Torino, Turin, Italy, in 2021 and 2023, respectively, where he is currently pursuing the Ph.D. degree in mathematical sciences.

He is a Visiting Researcher with the Department of Theoretical Chemistry, Stanford University, Stanford, CA, USA. At Stanford, his research focuses on statistical mechanics, generative models, and molecular dynamics. His work also focuses on theoretical aspects of federated learning in heterogeneous scenarios, a research line began with the ELLIS Turin Unit. Broader research interests include industrial applications of physics-informed machine learning, and time series analysis, from whale vocalizations to gravitational waves detection.

Davide Leo received the M.Sc. degree in mathematical engineering and in data science and engineering from Politecnico di Torino, Turin, Italy, in 2023 and 2024, respectively.

He is currently a Machine Learning Quantitative Researcher in a start-up environment, working on the development of mathematically grounded models for trading systems. During his M.Sc. thesis he worked with the ELLIS Unit of Turin, supervised by Prof. Barbara Caputo, working on federated learning. His background includes statistical learning, optimization, and deep learning, with a focus on the design and validation of predictive models. His research interests include federated learning and quantitative finance, with particular attention to the connection between theoretical methods and real-world applications.

Davide Carbone received the bachelor's degree in physics and the master's degree in physics of complex systems from the University of Turin, Turin, Italy, in 2018 and 2020, respectively, and the Ph.D. degree in pure and applied mathematics from Politecnico di Torino, Turin, in 2024.

His bachelor's thesis focused on Palatini-f(R) gravitational models, while his master's thesis addressed generalized time-reversal symmetry and Onsager relations. During his Ph.D., he was a Visiting Scholar at École Normale Supérieure, Paris, France, and the Courant Institute of Mathematical Sciences, New York University, New York, NY, USA.He is currently a Fellow in artificial intelligence at PRAIRIE-PSAI, PSL University, Paris, and a Post-Doctoral Researcher at École Normale Supérieure, Paris. His research focuses on theoretical and applied aspects of machine learning and generative models from a statistical mechanics perspective, with particular attention to MCMC-based sampling, Energy-Based Models and nonequilibrium systems of interacting particles. He has also worked on applications of wavelet scattering transforms, time series analysis, and stochastic sampling techniques for both physical and artificial data.

Open Access funding provided by 'Politecnico di Torino' within the CRUI CARE Agreement