

## A Theoretical Guarantes

In this Appendix, we present comprehensive proofs and assumptions pertaining to the results delineated in Section 4. Specifically, we focus on demonstrating that in the context of the federated averaging estimator, the exclusion of malicious clients prior to the training process facilitates the attainment of an unbiased estimator of the true mean  $\theta^b$ , consequently resulting in a less noisy estimate, as stated in Proposition 1. This outcome is supported through the introduction of two lemmas, namely Lemmas 1 and 2. Without loss of generality, we propose two general assumptions, which reflect the distinct behaviors of benign and malicious clients concerning model distribution. Specifically, we denote the set of benign clients as  $\mathcal{B} \subset \{1, \dots, K\}$  and the set of malicious clients as  $\mathcal{M} \subset \{1, \dots, K\}$ , within a federated system comprising  $K$  clients in total. We assume these sets are mutually exclusive and collectively exhaustive, meaning  $\mathcal{B} \cap \mathcal{M} = \emptyset$  and  $\mathcal{B} \cup \mathcal{M} = \{1, \dots, K\}$ . To adequately address the heterogeneity and the potential adversarial impact on client updates, we employ the following statistical framework.

**Assumption A1.** For each benign client  $k \in \mathcal{B}$ , the local model update  $\theta_k$  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  $\text{Var}[\theta_k] = (\sigma^b)^2$ . Similarly, for malicious clients  $k \in \mathcal{M}$ , the local updates  $\theta_k$  are independent random variables drawn from  $\rho_k(\bar{\theta}^m, \sigma^m)$  with  $\mathbb{E}[\theta_k] = \bar{\theta}^m$  and  $\text{Var}[\theta_k] = (\sigma^m)^2$ .

**Assumption A2.** We posit 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 that there exists  $C > 0$  such that  $(\sigma^m)^2 > C(\sigma^b)^2 > \left(2 + \frac{|\mathcal{M}|}{|\mathcal{B}|}\right) (\sigma^b)^2$ .

Assumption A1 characterizes the statistical distributions corresponding to the two distinct groups of clients. In contrast, Assumption A2 provides that the variance associated with the malicious models is significantly greater than that of the benign client updates. The standard federated averaging estimator is defined as a weighted average of client updates, i.e.

$$\theta_{avg} = \frac{1}{K} \sum_{k=1}^K \theta_k. \quad (7)$$

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 benign and malicious updates have different means (Lemma A1) and one where they share the same mean but differ in variance (Lemma A2).

**Lemma A1.** 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  $\theta_{avg}$  is a **biased estimator** of  $\bar{\theta}^b$ , meaning  $\mathbb{E}[\theta_{avg}] \neq \bar{\theta}^b$ .

*Proof.* Let us first recall that a random variable  $\hat{X}$  is an unbiased estimator of  $\mu$ , if its expectation equals the parameter that we aim to estimate, i.e. if  $\mathbb{E}[\hat{X}] = \mu$ . In case  $\mathbb{E}[\hat{X}] \neq \mu$ , we say that  $\hat{X}$  is a biased estimator of  $\mu$ .

If we compute the expectation of the estimator  $\theta_{avg}$ , defined in Equation 7, using the fact that malicious and benign client  $\{\mathcal{B}, \mathcal{M}\}$  form a partition of  $\{1, \dots, K\}$ , we get

$$\mathbb{E}[\theta_{avg}] = \mathbb{E}\left[\frac{1}{K} \sum_{k=1}^K \theta_k\right] = \mathbb{E}\left[\frac{1}{K} \left(\sum_{k \in \mathcal{B}} \theta_k + \sum_{k \in \mathcal{M}} \theta_k\right)\right]. \quad (8)$$

Let us denote with  $M = |\mathcal{M}|$  and  $B = |\mathcal{B}|$  the number of malicious and benign clients in the federation, by exploiting linearity of the expectation operator, we obtain

$$\begin{aligned} \mathbb{E}[\theta_{avg}] &= \frac{1}{K} \left( \sum_{k \in \mathcal{B}} \mathbb{E}[\theta_k] + \sum_{k \in \mathcal{M}} \mathbb{E}[\theta_k] \right) = \frac{B\bar{\theta}^b + M\bar{\theta}^m}{K} \\ &= \frac{B\bar{\theta}^b + M\bar{\theta}^b - M\bar{\theta}^b + M\bar{\theta}^m}{K} = \bar{\theta}^b + \frac{M}{K} (\bar{\theta}^m - \bar{\theta}^b) \neq \bar{\theta}^b \end{aligned} \quad (9)$$

874 where  $\bar{\theta}^b$  and  $\bar{\theta}^m$  denote the expectation of the model updates for benign and malicious clients,  
875 respectively. Since we obtained that  $\mathbb{E}[\theta_{avg}] \neq \bar{\theta}^b$ , we conclude that the estimator is biased.  $\square$

876 Furthermore, we observe that the drift in the estimate away from the benign model is controlled by  
877 the ratio of malicious clients  $M$  with respect to the number of total clients  $K$ .

878 **Lemma A2.** *Let*

$$\theta_{avg}^B = \frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \theta_k \quad (10)$$

879 *be the federated averaging estimator computed using only benign client updates. Under Assumption*  
880 **2** *the variance of the standard federated averaging estimator is higher than that of our estimator:*  
881  $\mathbb{V}ar[\theta_{avg}] \geq \mathbb{V}ar[\theta_{avg}^B]$ .

882 *Proof.* First, we compute the variance for the two estimators  $\theta_{avg}$  and  $\theta_{avg}^B$  exploiting the independence  
883 between model distributions, posit in Assumption **A1**. In particular

$$\mathbb{V}ar[\theta_{avg}] = \mathbb{V}ar \left[ \frac{1}{K} \sum_{k=1}^K \theta_k \right] = \frac{1}{K^2} \left( \sum_{k \in \mathcal{B}} \mathbb{V}ar[\theta_k] + \sum_{k \in \mathcal{M}} \mathbb{V}ar[\theta_k] \right) = \frac{B(\sigma^b)^2 + M(\sigma^m)^2}{K^2} . \quad (11)$$

884 Similarly, we get that

$$\mathbb{V}ar[\theta_{avg}^B] = \frac{(\sigma^b)^2}{B} . \quad (12)$$

885 If we consider the difference between the variances  $\mathbb{V}ar[\theta_{avg}]$  and  $\mathbb{V}ar[\theta_{avg}^B]$ , and we impose that  
886 this quantity is positive qwe obtain the following inequality

$$\mathbb{V}ar[\theta_{avg}] - \mathbb{V}ar[\theta_{avg}^B] = \frac{B(\sigma^b)^2 + M(\sigma^m)^2}{K^2} - \frac{(\sigma^b)^2}{B} > 0 . \quad (13)$$

887 Recalling that  $K = B + M$ , we get

$$\frac{B^2(\sigma^b)^2 + MB(\sigma^m)^2 - (B + M)^2(\sigma^b)^2}{B(B + M)^2} > 0 \iff MB(\sigma^m)^2 - M(2B + M)(\sigma^b)^2 > 0 \quad (14)$$

888 that is, since  $M > 0$ ,

$$(\sigma^m)^2 > \frac{1}{B}(2B + M)(\sigma^b)^2 \iff (\sigma^m)^2 > \left(2 + \frac{M}{B}\right)(\sigma^b)^2 \quad (15)$$

889 which together with Assumption **A1** concludes the proof.  $\square$

890 Lemma **A2** provides a definitive bound on the variance of the model, thereby resolving the question  
891 *how much larger should the variance of malicious models should be with respect to the benign*  
892 *models' variance.* Nonetheless, given the assumption in **A2** that the variance of malicious models  
893  $\sigma^m$  may exceed that of benign models  $\sigma^b$  to an arbitrary extent, the hypothesis of Lemma **A2** proves  
894 to be non-restrictive and readily achievable.

895 **Proposition A1.** *Under Assumptions **1** and **2**, removing malicious clients (those in  $\mathcal{M}$ ) from the*  
896 *federation yields a superior estimator of the global model. Specifically, the resulting estimator is*  
897 *unbiased (in the sense of Lemma **1**) and exhibits a reduced variance (as shown in Lemma **2**), leading*  
898 *to improved model accuracy and robustness.*

899 *Proof.* The proof is immediately derived from Lemmas **A1** and **A2**. This is due to the fact that upon  
900 the exclusion of malicious clients, the federated averaging estimator reduces to the form presented in  
901  $\theta_{avg}^B$ , which is not only unbiased but also exhibits reduced variance—thereby diminishing noise in  
902 the global model's estimation.  $\square$

## B Implementation Details and Further Experiments

This appendix provides a detailed description of the experimental setup, including the hyperparameters used in `Waffle` and the baseline methods. Section B.1 covers the datasets and implementation details. Section B.2 presents additional plots and experiments. Tables 3 and 4 report the  $2\sigma$  error bars, computed over three runs with different random seeds.

Code is available at <https://anonymous.4open.science/r/Waffle-69C2/>

### B.1 Datasets and Implementation Details

We conducted experiments on common FL benchmark datasets [39], namely FashionMNIST [40], CIFAR-10, and CIFAR-100 [41]. Since our goal was to detect malicious clients based on data characteristics, we sampled benign clients using a Dirichlet distribution with parameter  $\alpha = 1000$  to ensure near-i.i.d. conditions. This setting ensures a fair comparison with the baselines, as `Waffle` neither relies on model updates nor is affected by class imbalance. Moreover, the shuffled training on a distilled dataset already exposes `Waffle` to synthetic heterogeneity. Introducing additional data imbalance would therefore not yield further insights into its performance.

For classification, we used LeNet-5 [44]. The standard version was applied to FashionMNIST, while we adjusted the input channels to three and modified the number of output classes for CIFAR-10 and CIFAR-100. The federation included  $K = 100$  clients, with  $|\mathcal{P}_t| = 10$  clients participating per round. Training was carried out over  $T = 500$  communication rounds, using  $S = 1$  local epoch per round and a batch size of 64. We employed the cross-entropy loss optimized with the ADAM optimizer [45], using an initial learning rate  $\eta = 0.001$ .

For `Waffle`, we used WST parameters  $J = 3$ ,  $L = 6$ , and first-order coefficients [46]. The FT baseline employed a window size of 0.5. The `Waffle` detector was a multilayer perceptron with three hidden layers and hyperbolic tangent activations, trained for 100 epochs using ADAM. The attack parameters  $\beta$  and  $\sigma$  were randomly sampled from  $\text{Unif}\{3, 5, \dots, 19\}$  and  $\text{Unif}(0.5, 2.0)$ , respectively. For baselines, we used `mKrum` with  $k = 5$ , and `TrimmedMean` with a cut-off parameter of 0.2.

All computations were performed using the CPU of a MacBook Pro equipped with an Apple M3 Pro chip. No additional computational resources were employed.

### B.2 Further Experiments

We provide visualizations of client embeddings generated using both the WST and the FT, followed by quantitative results with error bars, obtained by training each method over three different random seeds.

**Waffle : WST vs FT** Figure 2 compares the embeddings  $\varphi_k$  produced by `Waffle` for a federation composed of 40% malicious and 60% benign clients. Both WST and FT generate meaningful representations that enable clear separation between benign and malicious clients, which corresponds to the high detection accuracy reported in Table 3.

In Figure 3, we further inspect the spectral embeddings of three representative clients: a blur attacker, a noise attacker, and a benign client. For each of them, we show the embeddings produced by `Waffle` using FT (left) and WST (right). These embeddings reveal distinctive structural patterns across client types, yet remain non-invertible: as shown in the visualizations, no information about the raw input data can be reconstructed from the communicated statistics. This characteristic aligns with the privacy-preserving goals of federated learning. A deeper discussion on the privacy guarantees of `Waffle` is provided in Appendix D.

Table 3: **Client Detection with  $2\sigma$  error bars.** A comparison between the two variants of `Waffle` : WST and FT. Detection metrics employed are F1 score, precision, recall and accuracy.

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

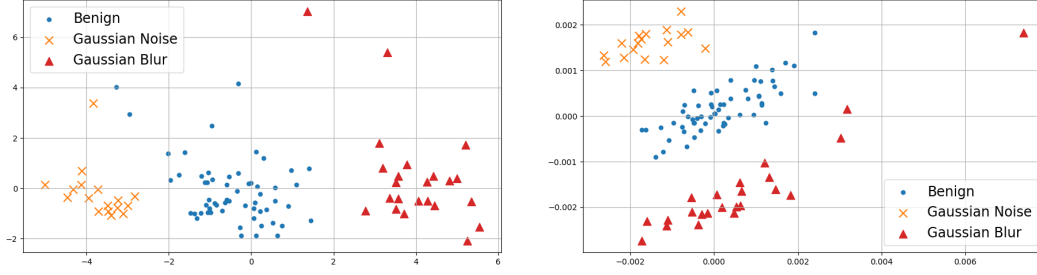


Figure 2: Client distributions of the  $\varphi_k$  for Cifar10 dataset with  $K = 100$  clients on a 2-dimensional space, for **Waffle** + FT (left), and **Waffle** + WST (right). There is a total of 60 benign clients (dots), and 40 attackers: 20 noisy (crosses) and 20 blurred (triangles). Both methods provide a noticeable separation between the clients.

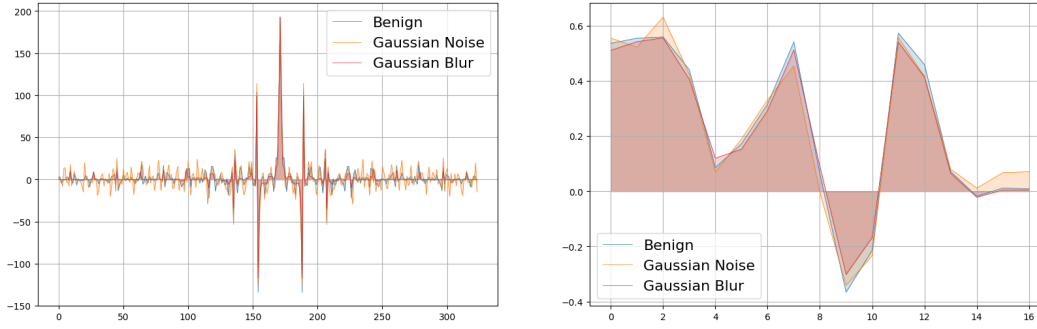


Figure 3: Embeddings  $\varphi_k$  produced by **Waffle** for three clients (blur attacker, noise attacker, and benign) on CIFAR-10. The left panel shows the embeddings obtained using FT, while the right panel shows those obtained using WST.

945 **Comparison with Baselines** We report the result of the comparison with the baseline algorithms  
 946 including the  $2\sigma$  error bar. The table provides the test accuracy averaged across the clients using  
 947 a partition of 40 attackers and 60 benign clients. As an upper bound for our method and all the  
 948 baselines we consider FedAvg trained on the federation without malicious clients, i.e. 60 benign  
 949 clients: on FashionMNIST it achieved a test accuracy of  $75.5 \pm 1.7$ , on CIFAR-10  $50.3 \pm 0.5$ , and  
 on CIFAR-100  $17.0 \pm 1.3$ .

Table 4: Comparison between baselines for detecting malicious clients and **Waffle** (with both WST and FT) with  $2\sigma$  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	$73.7 \pm 1.3$	$73.8 \pm 1.1$	$72.5 \pm 4.0$	$73.4 \pm 1.7$	$74.6 \pm 0.4$
	<b>Waffle</b> - WST	<b><math>74.9 \pm 1.9</math></b>	$70.2 \pm 0.4$	$74.2 \pm 0.9$	$74.6 \pm 1.6$	$74.7 \pm 1.8$
	<b>Waffle</b> - FT	$73.8 \pm 1.1$	$71.4 \pm 2.0$	$74.6 \pm 0.4$	$74.7 \pm 1.0$	<b><math>74.9 \pm 0.5</math></b>
CIFAR-10	w/o detector	$48.7 \pm 1.3$	$44.8 \pm 2.2$	$46.2 \pm 5.9$	$48.3 \pm 0.5$	$48.1 \pm 0.4$
	<b>Waffle</b> - WST	<b><math>49.6 \pm 0.3</math></b>	$46.2 \pm 0.6$	$49.5 \pm 0.6$	$49.0 \pm 1.4$	$49.5 \pm 0.8$
	<b>Waffle</b> - 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	$16.4 \pm 0.1$	$10.1 \pm 0.8$	$14.6 \pm 0.7$	$16.4 \pm 0.7$	$16.5 \pm 1.1$
	<b>Waffle</b> - WST	<b><math>16.5 \pm 1.0</math></b>	$8.8 \pm 2.2$	$14.5 \pm 0.7$	$16.3 \pm 0.3$	$16.2 \pm 0.5$
	<b>Waffle</b> - FT	$11.6 \pm 0.2$	$7.6 \pm 0.6$	$10.6 \pm 0.7$	$12.1 \pm 0.3$	$10.6 \pm 0.5$

950

## 951 C Detection Metrics

952 In evaluating the performance of **Waffle**, we employed several detection metrics [47, 43], each  
 953 offering a different perspective on the detector’s effectiveness in binary classification task. Let TP,

954 TN, FP, and FN represent True Positives (correctly identified malicious clients), True Negatives  
 955 (correctly identified benign clients), False Positives (benign clients incorrectly flagged as malicious),  
 956 and False Negatives (malicious clients incorrectly flagged as benign), respectively.

**Accuracy** Accuracy is one of the most straightforward metrics, representing the overall correctness of the classifier. It is calculated as the ratio of correctly classified instances (both malicious and benign) to the total number of instances.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

957 While intuitive, accuracy can be misleading, especially in scenarios with imbalanced datasets. For  
 958 instance, if 90% of clients are benign, a detector that classifies all clients as benign would achieve  
 959 90% accuracy, despite failing to identify any malicious clients. Therefore, while providing a general  
 960 overview, accuracy alone is often insufficient for evaluating a malicious client detector.

**Precision** Precision measures the proportion of correctly identified malicious clients among all clients classified as malicious by the detector.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

961 High precision indicates a low false positive rate, meaning that when the detector flags a client as  
 962 malicious, it is highly likely to be correct. This is crucial in scenarios where incorrectly blocking  
 963 a benign client (a false positive) has significant negative consequences, such as denying service to  
 964 legitimate users. A low precision score suggests the detector raises many false alarms.

**Recall** Recall measures the proportion of actual malicious clients that are correctly identified by the detector.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

965 High recall indicates a low false negative rate, meaning the detector successfully identifies a large  
 966 fraction of the malicious clients present. This is critical in security applications where failing to detect  
 967 a malicious client (a false negative) can lead to significant damage or compromise. A low recall score  
 968 suggests the detector misses many malicious clients.

**F1-Score** The F1-Score is the harmonic mean of Precision and Recall, providing a single metric that balances both concerns.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}$$

969 The F1-Score is particularly useful when there is an uneven class distribution, as it punishes extreme  
 970 values of precision or recall. A high F1-Score indicates that the detector has both good precision  
 971 and good recall, meaning it is both accurate in its positive predictions and captures a majority of  
 972 the actual positive instances. It is often preferred over accuracy in imbalanced malicious client  
 973 detection scenarios where both minimizing false alarms and maximizing detection of actual threats  
 974 are important.

## 975 D Privacy of Waffle

976 Waffle detector architecture prioritizes client privacy throughout its operation. Throughout the  
 977 learning process, **individual raw data**  $\{x_k^i\}$  **remains strictly on the client's device**. Each client  
 978  $k$  *privately* computes its PCA-derived representation  $\hat{x}_k$  and subsequently its spectral embedding  
 979  $\varphi_k$  locally on its own hardware. Clients only transmit the resulting spectral embedding vector  $\varphi_k$   
 980 to the server, ideally over a secure communication channel to protect these embeddings while in  
 981 transit. This  $\varphi_k$  is explicitly designed to be an aggregate statistic that captures characteristics of the  
 982 data distribution without revealing individual data points, thereby serving as a non-privacy-leaking  
 983 feature. Furthermore, since WST is non-invertible, it is also impossible to reconstruct the PCA  
 984 representant. Our methodology is consistent with approaches in privacy-preserving machine learning  
 985 where transformed or aggregated representations of data are used instead of raw sensitive information

986 to train models or make inferences [48]. Furthermore, the offline training of the Waffle detector and  
987 Algorithm 1 is conducted on a distinct auxiliary dataset  $\mathcal{D}^{\text{aux}}$ , coherently with common practices  
988 [35], ensuring that no actual client data from the federation is used or exposed during the detector’s  
989 training phase. The combination of local feature extraction by clients, the transmission of only  
990 these specialized spectral embeddings, and offline training using auxiliary data ensures that Waffle  
991 functions as a privacy-conscious safeguard within the FL ecosystem.