Sibai: A Few-Shot Meta-Classifier for Poisoning Detection in Federated Learning

Melanie Götz, Torsten Krauß, Alexandra Dmitrenko
University of Würzburg
Sanderring 2, 97070 Würzburg, Germany
melanie.goetz@uni-wuerzburg.de, torsten.krauss@uni-wuerzburg.de,
alexandra.dmitrienko@uni-wuerzburg.de

Abstract

Federated Learning (FL) enables collaborative machine learning across decentralized clients without sharing raw data, which offers enhanced privacy and improved performance. However, FL is vulnerable to poisoning attacks, compromising model integrity through both untargeted performance degradation and targeted backdoor attacks. Detecting backdoors in FL is challenging due to their stealthy nature and variability in local datasets. Existing defenses struggle against adaptive adversaries and distinguishing between poisoning and genuine dataset anomalies. This paper introduces the Siamese Backdoor Inspector (Sibai), a novel meta-classifier-based poisoning defense for FL. Leveraging the staple few-shot learning technique of Siamese networks, Sibai effectively detects malicious contributions in various scenarios, including settings with strong variations between clients' datasets and encounters with adaptive adversaries. Sibai achieves high detection rates, prevents backdoors, minimizes performance impact, and outperforms eight recent defenses regarding F1 score, poisoning prevention, and consistency across complex scenarios.

1. Introduction

Federated Learning (FL) [59] enables collaborative deep neural network (DNN) training across decentralized clients by sharing model updates instead of raw data, preserving privacy. While FL offers significant benefits, it is susceptible to *poisoning attacks* by untrustworthy or compromised clients who manipulate training data, labels, hyperparameters, or model parameters. These attacks can be *untargeted* [15, 31, 55, 57], degrading global model performance, or *targeted* [2–4, 11, 18, 19, 33, 40, 42, 45, 52, 54, 56], introducing backdoors that misclassify inputs with specific triggers while maintaining normal performance otherwise. Backdoors are particularly dangerous due to their stealth and severe risks in security-critical applications like autonomous vehicles [32, 41, 62]. FL poisoning defenses include *influence reduction (IR)* [1, 3, 39, 50], *robust aggregation (RA)* [36, 61], and the more common *detection and filtering (DF)* [5, 8, 17, 27, 43, 47, 48, 63], which excludes poisoned models before aggregation. Although IR and RA can leverage information from all local models, their methods, such as noising and clipping, often degrade benign performance when applied broadly.

Challenges. Most DF-based defenses rely on predefined metrics to detect poisoning, but this approach faces two key challenges. First, adaptive adversaries can tailor poisoned models to evade detection by appearing normal within these metrics [27, 29]. Expanding the metric sets [27] partially addresses this, but still relies on rigid criteria which are vulnerable to adaptation. Second, the natural diversity of *non-identically and independently distributed* (non-i.i.d.) datasets in real-world scenarios makes distinguishing poisoned models from those reflecting unusual data distributions difficult.

Meta-Classifiers. This paper introduces a dynamic meta-classifier approach to FL poisoning defense, which avoids rigid metrics and instead learns to differentiate malicious directly from observing local models. Recent methods [9, 21, 22, 25, 58] show that meta-classifiers can effectively detect poisoned models in an offline scenario. However, applying this to FL poses challenges: training requires a diverse set of so-called *shadow models*, including both benign and poisoned models, and existing methods depend on time-consuming preprocessing.

Contributions. To tackle the aforementioned challenges, this paper introduces the Siamese Backdoor Inspector (Sibai), a novel meta-classifier-based poisoning detection method specifically tailored to function as a DF-based defense for FL. Sibai is based on the staple few-shot learning technique of Siamese networks. Our contributions can be summarized as follows:

- Sibai is the first application of a meta-classifier in DF-based poisoning defenses for FL. Unlike conventional approaches that are reliant on predefined fixed metrics for detection, Sibai dynamically learns the criteria to distin-

guish between benign and malicious local models. Using a dynamic approach based on ML, Sibai poses a challenge for sophisticated attackers that try to adapt to predefined metrics, as it utilizes a diverse set of learned, complex model characteristics for its decision-making.
- We utilize Siamese Networks in the context of a poisoning detection meta-classifier. These networks are designed to compare and find dissimilarities between two inputs, in the case of Sibai DNN models.
- Sibai operates directly on flattened DNN representations using a novel layer-wise dimensionality reduction pipeline, using Principal Component Analysis (PCA) enhanced with engineered features. This feature-rich approach improves Sibai's ability to learn complex criteria, enhancing robustness against adaptation. Additionally, it enables Sibai to efficiently inspect all local models in a federation without requiring extensive time-consuming preprocessing steps.
- We extensively evaluate Sibai in various FL scenarios against five different poisoning attacks, assessing its effectiveness against stealthy and adaptive attackers and in scenarios with non-i.i.d. data. To conduct our evaluation, we created a substantial dataset of shadow models, including 5720 benign and 4145 malicious models. Sibai consistently achieves high detection rates across all tested scenarios without notable degradation of the global model's performance. Notably, Sibai successfully prevents the establishment of effective backdoors in all experiments, and outperforms the eight existing FL poisoning defenses, demonstrating higher F1 scores and consistently maintaining the lowest backdoor accuracy across all experiments.

In summary, Sibai introduces a robust ML-based FL poisoning defense to effectively address adaptive adversaries and non-i.i.d. scenarios.

2. Background

This section introduces concepts relevant to this paper.

Federated Learning FL [26, 36, 59] enables collaborative ML by training on local data while preserving privacy. In each round, the server sends the global model to selected clients, who train it locally and return updates. The server aggregates updates, often using FedAVG [36], a weighted average with a global learning rate δ. The algorithm is explained in more detail in Appendix A.

IID vs. non-IID Data Distributions ML models capture the characteristics of their datasets, emphasizing patterns in the training data. In an ideal *identically and independently distributed* (IID) scenario, all clients have similar data distributions, making outlier detection an effective strategy to detect poisoning. However, the real world rarely resembles those conditions; instead, clients usually exhibit non-i.i.d. data distributions which complicate the distinction between anomalies and poisoning. For classification tasks, *'1-class non-i.i.d.'* describes datasets dominated by a single label, with the disproportionality degree regulated by a parameter $q \in [0, 1]$. *Inter-client non-i.i.d.* [27] however, is even more complex and allows arbitrary and disjoint distributions, which resemble real-world conditions.

Poisoning Attacks Poisoning attacks [51, 54] on DNNs involve attackers altering the training process to manipulate the model. *Untargeted attacks* lower main task accuracy (MA) by reducing local model accuracy, while *targeted (backdoor) attacks* embed triggers causing specific misclassifications with minimal impact on MA [3, 18, 34]. Poisoning involves *data* (e.g., malicious labels or triggers [49]) or *model* modifications (e.g., altered gradients [28]). This paper mainly considers five specific poisoning types: pixel trigger backdoors [19], blend backdoors [10], edge-case attacks [54], semantic backdoors [3], and random label flipping [27], which are discussed in more detail in Appendix D

Adaptive Attackers Adaptive attackers can bypass FL defenses that focus on fixed metrics by manipulating their local models to evade detection. Bagdasaryan et al. [3] introduced a method that uses an additional loss component that balances backdoor accuracy (BA) and stealthiness to the metrics the defense considers, as shown in Equation (1) [3].

$$Loss = \alpha * Loss^{MA\&BA} + (1 - \alpha) * Loss^{Adaptation} \quad (1)$$

Siamese Neural Networks Siamese networks [7, 24] are few-shot classifiers that compare embedding vectors from twin networks with shared parameters. Using a distance metric like Euclidean distance [13], they assess similarity between inputs. The Triplet loss [20], shown in Equation (2), is used to minimize distances between the feature vectors of same-class pairs and maximize them for different-class pairs.

$$L(a, p, n) = \max(||f(a) - f(p)||^2 - ||f(a) - f(n)||^2 + \gamma, 0) \quad (2)$$

3. Sibai

In this section, we present Siamese Backdoor Inspector (Sibai), a novel defense mechanism against poisoning attacks in FL. Sibai addresses non-i.i.d. scenarios and adaptive attackers by dynamically learning discriminative features of malicious local models using a meta-classifier. This approach avoids the reliance on predefined metrics. By directly operating on flattened model representations, Sibai does not require time-consuming preprocessing and enables detailed analysis that hinders adversarial adaptation. Sibai quickly adapts to federation behavior and new attack strategies with minimal training samples. The neural network architecture adds a layer of opacity, complicating the task for attackers to even discern the specific criteria they must adapt to.

3.1. Threat Model

We consider an FL scenario with a centralized server using FedAVG [36] and a fixed global learning rate. Consistent with prior defenses [1, 5, 6, 27, 35, 37] we assume adversaries control fewer than half of the clients, however, they can fully manipulate their data, training process, and model parameters. Adversaries are aware of the algorithms and defenses deployed on the server. Importantly, Sibai's's design, training, and evaluation make no assumptions about local dataset distributions. We evaluate Sibai across i.i.d., 1-class non-i.i.d., and inter-client non-i.i.d. scenarios, the latter closely reflecting real-world FL conditions as described in Sec. 2.

3.2. Design

Sibai operates on the central server to detect and filter suspicious local models before aggregation. It employs a Siamese network [16, 24] meta-classifier trained to identify dissimilarities between malicious and benign client updates. Sibai consists of two phases: (I) *Offline training*, where the Siamese network learns from shadow models, including both benign and malicious examples; (II) *Online inference*, where Sibai evaluates local models against the global model during federation to identify and exclude poisoned models based on learned criteria.

Architecture. As shown in Fig. 1, Sibai takes the global model and a local model as input, processing them through four components: Dimensionality Reduction, Twin Networks, Distance Function, and Decision Network. The output is a binary decision indicating the likelihood of a local model being malicious. Detailed descriptions of each component follow in subsequent sections.

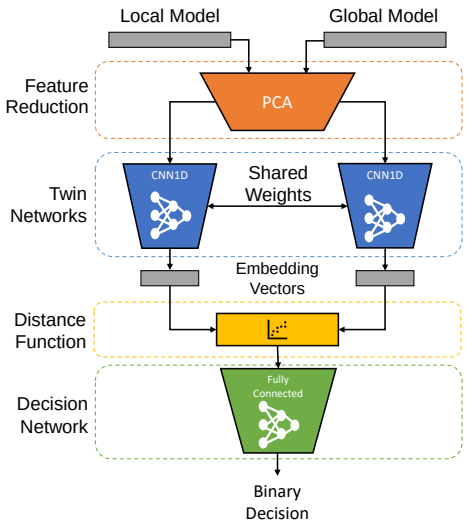

Figure 1. Sibai Architecture

Dimensionality Reduction. Sibai processes models of varying sizes by condensing input data while retaining key features. Local models, often containing hundreds of thousands of parameters, are broken into slices, and Principal Component Analysis (PCA) extracts the most significant components for each slice. PCA identifies and ranks features by variance, ensuring compact yet informative representations. Additional metrics, specifically Cosine and Euclidean distances to the global model, that were shown to be effective by prior research [5, 17, 27, 43], are also incorporated into the input vector. Experiments showed that these metrics consistently improve detection accuracy by up to 10%. More details about the implementation of the dimensionality reduction pipeline are provided in Appendix C.2.

Twin Networks. The Twin Networks consist of two identical 1D Convolutional Neural Networks (CNNs), serving as the core of the Siamese network. Each CNN processes one of the two pre-processed data samples, acting as a feature extractor. They map the samples to fixed-length embedding vectors using the same filters. These two vectors can then be compared to assess similarity. A high dissimilarity between embedding vectors indicates potential maliciousness, as the twins are explicitly trained to focus on the distinguishing features of malicious local models. Appendix C.4 provides further details about the architecture of the twin models.

Distance Function. The embedding vectors generated by the Twin Networks are compared using the L_2-distance, which demonstrated superior performance compared to other metrics like L_1-distance or cosine distance. This distance is used in the triplet loss function, which is utilized to train the twin networks.

Decision Network. To avoid using a fixed distance threshold, a fully connected neural network, called a decision network, is used to classify the output of the distance function. The last layer features a sigmoid activation function [38] that produces a value between 0 and 1, with values closer to 1 indicating a higher probability of maliciousness. We classify local models that when compared to the global model produce an output > 0.5 as malicious.

Shadow Model Dataset. Sibai's performance depends on a shadow model dataset used during offline training. This dataset should cover a range of attacks and scenarios to ensure effective generalization by the Siamese Network. To create the shadow model set we simulate FL across various federations with different non-i.i.d. degrees attacks and PDRs. Incorporating stealthy models from adaptive adversaries with low PDR allows Sibai to focus on nuanced criteria while including poisoned models with high PDR enhances generalization. It's crucial to maintain consistent local model architectures in the shadow set, and federations with varied input or output layer sizes cannot be mixed. Further details are provided in Appendix C.3.

4. Evaluation

This section evaluates Sibai during online inference against five poisoning attacks in challenging scenarios, including complex non-i.i.d. settings, low PDRs, and adaptive adversaries.

Hardware. Training and evaluation were performed on a Linux server with an Intel Xeon Gold 6342 CPU, 251GB RAM, CUDA [44], and an NVIDIA A30 GPU with 24GB memory.

Reference Defenses. We compare Sibai with eight defenses: HDBSCAN [35] which represents naïve clustering via cosine distance, Krum and M-Krum [5] which select the new global model based on Euclidean distance, MESAS [27] which tries to counter adaptive adversary by using six metrics, FLTrust [8] which uses trusted reference-based weighting, FoolsGold [17] which analyzes the final layer cosine distance, and the robust aggregation method Trimmed Mean/-Median [60]. Hyperparameters are detailed in Appendix E. However, it is important to note that a direct comparison of methods with fundamentally different defense paradigms, such as a DF-based method like ours and IR-based defenses like FLTrust, cannot capture the whole picture since these paradigms approach poisoning defense with fundamentally different philosophies.

Attacks. Sibai is tested against five attacks: *Random* label flipping [27], *Pixel* trigger backdoor [19], *Blend* backdoor [10], *Edge* case backdoor [54], and *Semantic* backdoor [3]. These attacks are introduced in more detail in Appendix D

Datasets and Models. We evaluate federations collaboratively training Squeezenet [23] on CIFAR-10 [30]. Additionally, we conducted experiments with a custom DNN and MNIST, which are provided in Sec. 4.5.

Shadow Model Dataset. Sibai's offline training uses 520 simulated federations, each with 20 clients (11 benign, 9 malicious), totaling 5720 benign and 4145 malicious models. Scenarios include i.i.d. and non-i.i.d. distributions with varying degrees (e.g., 0.3, 0.6, 0.9 1-class non-i.i.d. and inter-client non-i.i.d. [27]). Realistic and exaggerated settings enhance the distinction between benign and malicious models. Each federation experiences one of the five attacks. Federations with local backdoor accuracies below 45% were excluded to ensure poisoned models exhibit detectable traits. Distribution details are in Tab. 1 and a detailed breakdown is listed in Appendix H.

Default Federation Parameters. We evaluate Sibai in simulated federations collaboratively training Squeezenet [23] with SGD (learning rate: 0.01, momentum: 0.9, decay: 0.005) and a batch size of 64. Each client trains locally for 10 epochs on i.i.d. datasets with 2560 samples. The global model is initialized with pre-trained weights from PyTorch [46] and undergoes 49 benign rounds to mitigate early backdoor persistence [3], followed by poisoning in the 50th round.

Category		Number of Federations
IID/non-i.i.d.	i.i.d.	225
	0.3 1-class non-i.i.d.	66
	0.6 1-class non-i.i.d.	71
	0.9 1-class non-i.i.d.	96
	inter-client non-i.i.d. [27]	62
Attacks	Random [27]	90
	Pixel [19]	91
	Blend [10]	119
	Edge [54]	121
	Semantic [3]	99
Total Number of Federations		520

Table 1. Composition of the shadow model dataset

Sibai Parameters. During offline training, the twin and decision networks are trained separately for 25 epochs using SGD. The twin networks use a learning rate of 0.001, while the decision network uses 0.005. Model slices contain up to 20,000 parameters, with larger layers partitioned. PCA retains 200 components per slice. Siamese network training does not use batch processing due to memory constraints, while the decision network processes batches of 16 distances derived from 8 Siamese network triples.

Metrics. We evaluate Sibai with precision, recall, and F1 score. True positives are correctly flagged malicious models, and true negatives are benign models. Precision measures the ratio of correctly identified malicious models to all flagged models, while recall assesses the proportion of flagged malicious models among all actual malicious models. The F1 score balances these metrics. Additionally, we track the global model's BA to evaluate attack mitigation and MA changes to assess benign scenario impact. The MA diff. represents the difference between the MA after Sibai application and perfect benign aggregation.

4.1. IID Scenarios

This section evaluates Sibai's online performance after training on the shadow model set. We test Sibai on 12 unseen federation scenarios[1] of increasing difficulty and adversary stealthiness. In the baseline scenario (0) (Tab. 2), federations training Squeezenet [23] on CIFAR-10 [30] in an i.i.d. setting were attacked with the five poisoning attacks. Adversaries employed a PDR of 0.5, which is a relatively overt attack. Sibai flawlessly detected all malicious clients without false positives. Scenario (1) increased stealthiness by minimizing PDRs while maintaining local BAs above 45%. The blend [10] and semantic [3] backdoors used a PDR of 0.04 due to their stronger effects, while the edge case backdoor [54] required a PDR of 0.16 for consistent backdoor introduction. These adjustments mirror real-world attacker strategies. Sibai achieved perfect recall and F1 scores for

[1] Each scenario is designated by a circled number

Scenario		Random	Pixel	Blend	Edge	Semantic
⓪	PDR	0.5	0.5	0.5	0.5	0.5
	Precision	1.0000	1.0000	1.0000	1.0000	1.0000
	Recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.09%	0.20%	18.87%	0.0%
①	PDR	0.4	0.08	0.04	0.16	0.04
	Precision	1.0000	1.0000	1.0000	1.0000	1.0000
	Recall	0.8888	1.0000	1.0000	1.0000	0.7777
	F1	0.9411	1.0000	1.0000	1.0000	0.8750
	MA diff.	-0.88%	0.0%	0.0%	0.0%	+4.12%
	BA	-	10.43%	0.0%	18.87%	0.0%

⓪ = Default + high PDRs
① = Default + low PDRs

Table 2. Detection performance of Sibai in i.i.d. scenarios

q	Scenario		Random	Pixel	Blend	Edge	Semantic
0.6	②	PDR	0.5	0.5	0.5	0.5	0.5
		Precision	1.0000	1.0000	1.0000	1.0000	1.0000
		Recall	1.0000	1.0000	1.0000	1.0000	1.0000
		F1	1.0000	1.0000	1.0000	1.0000	1.0000
		MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
		BA	-	2.65%	0.20%	6.12%	0.0%
	③	PDR	0.4	0.08	0.04	0.16	0.04
		Precision	1.0000	1.0000	1.0000	1.0000	1.0000
		Recall	1.0000	1.0000	1.0000	0.8888	1.0000
		F1	1.0000	1.0000	1.0000	0.9411	1.0000
		MA diff.	0.0%	0.0%	0.0%	-6.71%	0.0%
		BA	-	2.65%	0.21%	8.67%	0.0%
0.9	④	PDR	0.5	0.5	0.5	0.5	0.5
		Precision	0.8750	1.0000	1.0000	0.9000	1.0000
		Recall	0.7777	0.8888	1.0000	1.0000	1.0000
		F1	0.8235	0.9411	1.0000	0.9473	1.0000
		MA diff.	+4.80%	+1.08%	0.0%	+2.44%	0.0%
		BA	-	0.11%	0.0%	0.0%	0.0%
	⑤	PDR	0.4	0.08	0.04	0.16	0.04
		Precision	0.8181	0.7500	1.0000	0.8181	1.0000
		Recall	1.0000	1.0000	1.0000	1.0000	0.8888
		F1	0.9000	0.8571	1.0000	0.9000	0.9411
		MA diff.	+0.48%	-4.26%	0.0%	-1.65%	+0.48%
		BA	-	0.0%	0.21%	0.0%	0.0%

② = Default + 0.6 1-class non-i.i.d. + high PDR
③ = Default + 0.6 1-class non-i.i.d. + low PDR
④ = Default + 0.9 1-class non-i.i.d. + high PDR
⑤ = Default + 0.9 1-class non-i.i.d. + low PDR

Table 3. Detection performance of Sibai in 1-class non-i.i.d. scenarios, with different level of disproportionality q

pixel trigger and edge case backdoors, identifying all malicious clients. In the random label flip and semantic backdoor attacks, one and two malicious clients, respectively, evaded detection. However, Sibai identified over 77% of malicious clients in all cases, preventing effective poisoning. Across all i.i.d. scenarios, Sibai maintained BA below 19%, ensuring attacks were unsuccessful. Interestingly, in the semantic backdoor scenario, Sibai mistakenly aggregated two malicious models, resulting in a slight MA improvement over perfect aggregation. This occurred because the aggregated models provided additional knowledge, but the low PDR and limited bypassed defenses still prevented backdoor establishment.

4.2. Non-IID Scenarios

In this section, we evaluate Sibai in challenging non-i.i.d. scenarios, including varying degrees of 1-class non-i.i.d. and inter-client non-i.i.d. setups as described in Sect. 2 and [27].

Scenario		Random	Pixel	Blend	Edge	Semantic
⑥	PDR	0.5	0.5	0.5	0.5	0.5
	Precision	1.0000	1.0000	1.0000	1.0000	0.9000
	Recall	0.7777	0.8888	1.0000	1.0000	1.0000
	F1	0.8750	0.9411	1.0000	1.0000	0.9473
	MA diff.	+2.61%	+0.86%	0.0%	0.0%	0.0%
	BA	-	4.03%	0.0%	0.0%	0.0%
⑦	PDR	0.4	0.08	0.04	0.16	0.04
	Precision	1.0000	1.0000	1.0000	1.0000	1.0000
	Recall	1.0000	0.8888	0.6666	1.0000	0.8888
	F1	1.0000	0.9411	0.8000	1.0000	0.9411
	MA diff.	0.0%	+1.38%	-1.43%	0.0%	-1.48%
	BA	-	13.57%	0.0%	0.0%	0.0%

⑥ = Default + inter-client non-i.i.d. + high PDR
⑦ = Default + inter-client non-i.i.d. + low PDR

Table 4. Detection performance of Sibai in inter-client non-i.i.d. scenarios [27]

0.6 1-class non-i.i.d. For the scenarios ② and ③ $q = 0.6$ was used. Sibai achieved perfect detection in the 0.5 PDR case ②. For the low PDRs scenario ③, F1 scores were 1.0 for all attacks except the edge case backdoor, where one malicious client evaded detection. Despite this, the MA dropped by less than 7%, and BA stayed below 9%, ensuring no effective backdoor was introduced.

0.9 1-class non-i.i.d. In the more extreme 0.9 1-class non-i.i.d. scenarios ④ and ⑤ in Tab. 10, Sibai occasionally flagged benign models, with three false positives in the worst-case pixel trigger backdoor scenario (⑤). However, these errors had minimal impact on MA. Sibai consistently eliminated enough malicious models to prevent effective backdoors, as no attack achieved a BA higher than 1%.

Inter-client non-i.i.d. Inter-client non-i.i.d. features erratic and highly unbalanced data distributions, reflecting real-world scenarios. In the 0.5 PDR scenario ⑥, Sibai achieved perfect detection for blend and edge case backdoors, perfect precision for untargeted and pixel attacks, and perfect recall for the semantic attack. Only one false positive occurred during the semantic attack. The lowest F1 score of 0.8750 occurred during the untargeted attack) In the attacker-adapted low PDR scenario ⑦, Sibai maintained perfect precision for all attacks despite high non-i.i.d.-ness. Recall was lower, yielding F1 scores of 0.9411 for pixel and semantic backdoors and 0.8000 for the blend attack. Nonetheless, MA was minimally affected, and BA remained low across all attacks. Overall, Sibai demonstrated robust performance in 1-class and inter-client non-i.i.d. scenarios, with minimal false positives and no effective backdoors.

4.3. Adaptive Adversaries

Sibai incorporates slice-wise Euclidean and cosine distances to the global model into its input vectors to enhance detection. While effective, these metrics may be vulnerable to adaptive attackers. To test Sibai's robustness, we evaluated four scenarios involving attackers using Bagdasaryan *et al.*'s [3] adaptation method with an α parameter balancing adaptation

Form of Adaptation	Scenario		Random	Pixel	Blend	Edge	Semantic
Cosine dist.	⑧	PDR	0.5	0.5	0.5	0.5	0.5
		Precision	1.0000	1.0000	1.0000	1.0000	1.0000
		Recall	1.0000	1.0000	1.0000	1.0000	1.0000
		F1	1.0000	1.0000	1.0000	1.0000	1.0000
		MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
		BA	-	1.08%	0.20%	18.87%	0.0%
	⑨	PDR	0.4	0.08	0.04	0.16	0.04
		Precision	1.0000	1.0000	1.0000	1.0000	1.0000
		Recall	1.0000	1.0000	1.0000	1.0000	1.0000
		F1	1.0000	1.0000	1.0000	1.0000	1.0000
		MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
		BA	-	1.08%	0.14%	18.87%	0.0%
Euclidian dist.	⑩	PDR	0.5	0.5	0.5	0.5	0.5
		Precision	1.0000	1.0000	1.0000	1.0000	1.0000
		Recall	1.0000	1.0000	1.0000	1.0000	1.0000
		F1	1.0000	1.0000	1.0000	1.0000	1.0000
		MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
		BA	-	1.08%	0.20%	18.87%	0.0%
	⑪	PDR	0.4	0.08	0.04	0.16	0.04
		Precision	1.0000	1.0000	1.0000	1.0000	1.0000
		Recall	1.0000	0.8888	0.8888	1.0000	1.0000
		F1	1.0000	0.9411	0.9411	1.0000	1.0000
		MA diff.	0.0%	+3.1%	-6.3%	0.0%	0.0%
		BA	-	5.98%	0.0%	18.87%	0.0%

⑧ = Default + cosine distance adaption with 0.3 α by mal. clients + high PDR
⑨ = Default + cosine distance adaption with 0.3 α by mal. clients + low PDR
⑩ = Default + Euclidean distance adaption with 0.3 α by mal. clients + high PDR
⑪ = Default + Euclidean distance adaption with 0.3 α by mal. clients + low PDR

Table 5. Detection performance of Sibai against adaptive attackers in i.i.d. scenarios

and task losses. Results for $\alpha = 0.3$, shown in Tab. 5, highlight Sibai 's resilience. In cosine-adapted scenarios, Sibai achieved perfect F1 scores (1.0), showing immunity to such attacks. Similarly, Sibai maintained perfect performance in Euclidean-adapted Scenario ⑫. In low-PDR Euclidean-adapted scenarios, Sibai showed flawless precision, with one false negative for pixel and blend backdoors. However, no attack established an effective backdoor, with BA remaining below 19%. Overall, Sibai demonstrates strong resilience against adaptive attacks, proving it does not overly rely on engineered features. Its ability to internalize specific patterns makes it impractical for attackers to adapt using Bagdasaryan *et al*. 's method.

4.4. Comparison to Existing Defenses

To assess the effectiveness of Sibai compared to metric-based poisoning detection methods in FL, we also evaluate existing defenses in some of our previous scenarios. Due to space limitations, we focus on a selection of the most representative scenarios, including scenario ① as the i.i.d. baseline, scenario ④ as the highest degree of 1-class non-i.i.d., and scenario ⑥ to showcase inter-client non-i.i.d. We also include scenario ⓪ as the overall simplest scenario and scenario ⑪ to cover adaptive adversaries. Since some reference defenses rely on modified aggregation methods instead of strictly flagging and filtering out malicious models, calculating F1 scores for all reference defenses faced challenges. Further details are provided in Appendix F.

Scenario	Defense	Random	Pixel	Blend	Edge	Semantic
⓪	PDR	0.5	0.5	0.5	0.5	0.5
	MESAS	0.9000	0.9000	0.9000	**1.0000**	0.9473
	Krum	0.5714	0.5714	0.5789	0.5714	0.5714
	M-Krum	0.2608	0.2608	0.6428	0.2608	0.2608
	HDBSCAN	**1.0000**	0.0000	0.1111	**1.0000**	**1.0000**
	FLTrust (0.09)	0.0000	0.2608	0.3333	0.0000	0.2608
	Sibai (ours)	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
①	PDR	0.4	0.08	0.04	0.16	0.04
	MESAS	**0.9473**	0.0000	0.1818	0.1666	0.3076
	Krum	0.5714	0.6428	0.6428	0.5714	0.5714
	M-Krum	0.2608	0.6086	0.6086	0.2608	0.4347
	HDBSCAN	0.0000	0.5555	0.4444	0.0000	0.1176
	FLTrust (0.09)	0.0952	0.6666	0.5714	0.5185	0.5714
	Sibai (ours)	0.9411	**1.0000**	**1.0000**	**1.0000**	**0.8750**
④	PDR	0.5	0.5	0.5	0.5	0.5
	MESAS	0.0000	0.1666	0.8000	**0.9473**	0.9000
	Krum	0.5714	0.5714	0.5714	0.6428	0.6428
	M-Krum	0.2608	0.6956	0.6956	0.7826	0.7826
	HDBSCAN	0.0000	0.4000	0.7777	0.8235	0.8235
	FLTrust (0.09)	0.5714	0.8888	0.8888	0.8888	0.8571
	Sibai (ours)	**0.8235**	**0.9411**	**1.0000**	**0.9473**	**1.0000**
⑥	PDR	0.5	0.5	0.5	0.5	0.5
	MESAS	0.6428	0.4761	0.2105	0.6666	0.3157
	Krum	0.5714	0.5714	0.5714	0.5714	0.5714
	M-Krum	0.2608	0.2608	0.2608	0.5217	0.3478
	HDBSCAN	0.0000	0.0000	0.1428	0.3076	0.1428
	FLTrust (0.09)	0.0000	0.5714	0.5714	0.5714	0.6206
	Sibai (ours)	**0.8750**	**0.9411**	**1.0000**	**1.0000**	**0.9473**
⑪	PDR	0.4	0.08	0.04	0.16	0.04
	MESAS	**1.0000**	0.6666	0.9000	0.6153	0.9473
	Krum	0.5714	0.5714	0.6428	0.6428	0.6428
	M-Krum	0.2608	0.2608	0.7826	0.6086	0.6956
	HDBSCAN	0.0000	**1.0000**	**1.0000**	**1.0000**	0.2000
	FLTrust (0.09)	0.5714	0.6206	0.6666	0.6206	0.6206
	Sibai (ours)	**1.0000**	0.9411	0.9411	**1.0000**	**1.0000**

Table 6. F1 score comparison between Sibai and current FL poisoning defenses

As shown in Tab. 6, Sibai achieved superior F1 scores across most scenarios. In high PDR i.i.d. scenario ⓪, it matched or outperformed all defenses. In low PDR i.i.d. scenario ①, Sibai achieved perfect F1 scores for pixel, blend, and edge case backdoors, outperforming MESAS for these attacks. Even when MESAS maintained low BA, it highlighted the fragility of backdoor attacks in low PDR settings. For 1-class non-i.i.d. scenario ④, Sibai significantly outperformed other methods, and in inter-client non-i.i.d. scenario ⑦, its worst-case F1 score of 0.8750 surpassed the best scores of other defenses. In adaptive scenario ⑪, Sibai slightly trailed HDBSCAN for pixel and blend backdoors however, it excelled in the detection of the semantic and untargeted attacks, achieving perfect F1 scores. Overall, Sibai consistently outperformed existing methods in identifying malicious models across diverse scenarios and attacks.

Table 7, also considered methods for which the F1 was unavailable, specifically FoolsGold [17] and trimmed mean/-median [60]. Sibai showed performance close to perfect benign aggregation, effectively preventing poisoning without significant MA reduction. Notably, Sibai and MESAS achieved low BA in challenging scenarios ④ and ⑥, where other defenses failed to stop backdoors. Although FLTrust [8], produced higher MA in non-i.i.d. scenarios by using a weighted aggregation that particularly emphasizes the old global model, it also often allowed higher BA and

Scenario	Defense	Random	Pixel		Blend		Edge		Semantic	
		MA diff.	MA diff.	BA	MA diff.	BA	MA diff.	BA	MA diff.	BA
①	PDR	0.4	0.08		0.04		0.16		0.04	
	benign agg.	0.0	0.0	10.4	0.0	0.0	0.0	18.8	0.0	0.0
	MESAS [27]	**-0.1**	**+5.1**	9.5	**0.0**	20.4	+0.2	7.1	0.0	**0.0**
	Krum [5]	-35.0	-43.2	**0.0**	-14.1	**0.0**	-35.5	**0.0**	-21.1	100.0
	M-Krum [5]	-24.5	-8.5	12.5	-9.5	2.9	-16.3	**0.0**	-11.4	**0.0**
	FoolsGold [17]	-6.3	+3.0	11.0	-1.7	18.8	-8.3	6.1	**+0.1**	**0.0**
	HDBSCAN [35]	-16.0	-1.9	10.9	-6.5	11.4	-13.8	**0.0**	-5.8	**0.0**
	Trimmed mean [60]	-8.3	+3.5	10.2	-2.5	12.8	-9.0	6.6	-0.6	**0.0**
	Trimmed median [60]	-22.3	-2.6	**5.2**	-11.6	0.1	-14.7	2.5	-6.4	**0.0**
	FLTrust [8]	-22.3	-0.1	17.2	-5.2	13.0	**+1.5**	16.8	-4.8	**0.0**
	Sibai (ours)	-0.8	0.0	10.4	**0.0**	**0.0**	0.0	18.8	0.0	**0.0**
④	PDR	0.4	0.5		0.5		0.5		0.5	
	benign agg.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	MESAS [27]	-5.7	+0.3	34.3	+5.0	0.6	+2.4	**0.0**	+10.6	**0.0**
	Krum [5]	-7.7	-6.1	100.0	-0.2	100.0	-4.8	**0.0**	-5.3	**0.0**
	M-Krum [5]	-2.1	+5.5	1.7	+1.0	100.0	-5.1	0.0	+4.0	**0.0**
	FoolsGold [17]	-2.0	-5-0	30.2	-0.2	100.0	-10.8	**0.0**	-7.3	**0.0**
	HDBSCAN [35]	-7.5	+1.0	13.2	+1.4	0.0	-8.7	**0.0**	-0.9	20.0
	Trimmed mean [60]	-5.3	-0.4	28.9	-0.2	100.0	-11.1	**0.0**	-7.8	40.0
	Trimmed median [60]	-6.6	-5.1	0.0	-0.4	100.0	-10.4	**0.0**	-6.5	**0.0**
	FLTrust [8]	**+19.5**	**+14.7**	53.1	**+10.8**	98.2	**+4.8**	**0.0**	**+11.1**	20.0
	Sibai (ours)	+4.8	+1.0	**0.0**	0.0	**0.0**	+2.4	**0.0**	0.0	**0.0**
⑥	PDR	0.5	0.5		0.5		0.5		0.5	
	benign agg.	0.0	13.9	2.9	0.0	2.3	0.0	2.4	0.0	0.0
	MESAS [27]	+11.9	-0.3	1.6	-0.1	32.7	+9.7	**0.0**	0.0	**0.0**
	Krum [5]	+4.8	-3.9	100.0	-5.1	100.0	**+16.8**	85.7	**+18.7**	80.0
	M-Krum [5]	-1.6	-3.0	85.3	-0.3	100.0	+2.9	**0.0**	0.0	**0.0**
	FoolsGold [17]	-1.6	+1.5	7.0	-5.0	100.0	-2.0	**0.0**	0.0	**0.0**
	HDBSCAN [35]	-5.8	-0.7	19.0	+3.8	99.9	-0.3	**0.0**	-0.4	**0.0**
	Trimmed mean [60]	-0.5	+1.1	7.3	-3.8.1	100.0	-3.0	**0.0**	0.0	**0.0**
	Trimmed median [60]	-2.5	-3.5	**0.0**	-4.9	100.0	-2.3	**0.0**	-0.3	**0.0**
	FLTrust [8]	**+14.7**	**+14.3**	47.4	**+13.0**	99.8	+14.0	**0.0**	+8.3	20.0
	Sibai (ours)	+2.6	+0.8	4.0	0.0	**0.0**	0.0	**0.0**	0.0	**0.0**

Table 7. Comparison of MA diff. and BA (both in percentage) between Sibai and existing FL poisoning defenses. The best value for each attack in each scenario is highlighted in bold.

therefore failed to prevent effective backdoors. In contrast, Sibai excelled at malicious model detection and consistently maintained low BA while achieving MAs aligned with benign aggregation, making Sibai an overall reliable defense.

4.5. Alternative Dataset and DNN Architecture

To demonstrate that Sibai is not limited to Squeezenet or the CIFAR-10 dataset, we conducted additional experiments using a custom CNN and the MNIST dataset [12]. The CNN model consists of 2 convolutional layers, 2 max-pooling layers, and 3 linear layers. We excluded the dataset-specific semantic and edge case backdoors and instead included a label swap attack [27], which targets samples of one label and swaps them to another to reduce the global model's accuracy. The shadow-model dataset for these experiments comprised 1401 federations. The F1 scores in Tab. 8 indicate that Sibai effectively detected malicious models, and therefore that the approach is not specific to Squeezenet or CIFAR-10.

5. Multiple Poisonings in one Model

To evaluate Sibai's effectiveness in detecting models poisoned with multiple backdoors simultaneously, we conducted experiments using federations where each malicious local model contained both a blend backdoor [10] and a pixel trigger backdoor [19] simultaneously. For those local models, the PDR quantifies the ratio of the total poisoned samples, i.e. the sum used for both backdoors, to the complete local model dataset. In our implementation of this multi-backdoor half of the poisoned samples were used to introduce the pixel trigger backdoor, while the other half was used to apply the blend backdoor. Importantly, our shadow model dataset, used to train Sibai, did not include

PDR	Distribution	Random [27]	Pixel [19]	Blend [10]	Label Swap [27]
0.4	i.i.d.	1.0000	1.0000	1.0000	1.0000
0.4	0.3 1-class non-i.i.d.	1.0000	1.0000	1.0000	1.0000
0.4	0.6 1-class non-i.i.d.	0.6923	0.7200	0.7200	0.7200
0.4	0.9 1-class non-i.i.d.	0.8750	0.9411	1.0000	1.0000
0.4	inter-client non-i.i.d.	0.9000	0.9411	0.9473	0.9000
0.5	i.i.d.	1.0000	1.0000	1.0000	1.0000
0.5	0.3 1-class non-i.i.d.	1.0000	1.0000	1.0000	1.0000
0.5	0.6 1-class non-i.i.d.	0.7200	0.6666	0.7200	0.7200
0.5	0.9 1-class non-i.i.d.	1.0000	0.9411	1.0000	1.0000
0.5	inter-client non-i.i.d.	0.9473	0.9411	0.9473	0.9000

Table 8. F1 score of Sibai in federations training a small CNN with the MNIST dataset

PDR	Distribution	Precision	Recall	F1
0.2	i.i.d.	1.0000	1.000	1.0000
0.2	0.3 1-class non-i.i.d.	1.0000	1.0000	1.0000
0.2	0.6 1-class non-i.i.d.	1.0000	1.0000	1.0000
0.2	0.9 1-class non-i.i.d.	1.0000	0.8888	0.9411
0.2	inter-client non-i.i.d.	1.0000	0.8888	0.9411
0.5	i.i.d.	1.0000	1.000	1.0000
0.5	0.3 1-class non-i.i.d.	1.0000	1.0000	1.0000
0.5	0.6 1-class non-i.i.d.	1.0000	1.0000	1.0000
0.5	0.9 1-class non-i.i.d.	1.0000	1.0000	1.0000
0.5	inter-client non-i.i.d.	1.0000	1.0000	1.0000

Table 9. Detection performance of Sibai against local models poisoned with two backdoors (pixel trigger backdoor [19] and blend backdoor [10]) simultaneously in different scenarios

any models with such multi-backdoors. As illustrated in Tab. 9, Sibai consistently identified local models containing multiple backdoors across all tested scenarios.

6. Related Work

This section provides a brief overview of relevant literature. First, Sect. 6.1 outlines poisoning defenses tailored for FL. Then, Sect. 6.2 discusses machine learning methods for general offline poisoning analysis.

6.1. Defenses against Poisoning Attacks in Federated Learning

There exists a diverse range of techniques that employ fixed metrics to detect poisoning attacks in FL. FoolsGold [17] uses a client-specific update history which is compared to the update histories of other clients using the cosine similarity. Krum [5] selects a new global model based on the Euclidean distance to other local models. M-Krum [5] extends this method to select multiple models simultaneously. Trimmed-Mean/Median [60] are robust aggregation methods that replace FedAVG [36] and calculate the updated global model by using coordinate-wise means or medians from local models. FLAME [43] clusters models using HDBSCAN [35] and filters out smaller clusters. It also applies differential privacy techniques, weight clipping, and noising. DeepSight [47] combines filtering with differential privacy. It relies on two metrics: The cosine distance between models and values extracted from the output layer of the model. Auror [48] uses K-Means clustering on model parameters, however, unfortunately, it exhibited an extreme performance overhead in our experiments. FLTrust [8] uses a weighted aggregation system for which it assigns weights to local models based on their cosine and Euclidean distances to a benign reference model, which is trained server-side on a small trusted dataset. FLDetector [63] is a historical update-based defense that mathematically predicts client updates and compares the predictions to the actual updates. MESAS [27] was specifically designed to operate in non-i.i.d. scenario and counter adaptive adversaries and employs six metrics. These approaches differ from Sibai as they rely on fixed metrics or criteria, making them susceptible to adaptive attacks [27]. Defenses focusing on specific layers, like FoolsGold [17], are prone to adversaries exploiting advanced model poisoning techniques specifically targeting those layers. Similarly, methods that heavily focus on cosine or Euclidean distance are often vulnerable to adaptation methods using additional losses [3, 27], as detailed in Sect. 2. These rigid metrics also face challenges in non-i.i.d. scenarios, especially in inter-client non-i.i.d. settings, where it has been demonstrated that most defenses, except MESAS [27] which was specifically designed to handle such scenarios, face difficulties. In contrast, Sibai dynamically learns criteria based on its shadow model dataset, making it challenging for attackers to adapt without access to the Siamese Network. Sibai can also differentiate between variations in local models due to data distribution and those caused by poisoning.

6.2. Non-FL Specific Offline Poisoning Detection

This section discusses several machine-learning-based detection methods against poisoning attacks, not specifically designed for FL. Xu et al. [58] propose Meta Neural Trojan Detection (MNTD), a black-box backdoor detection approach utilizing a meta-classifier that assesses if a model's output behavior on a given input aligns more closely with a set of known benign models or a set of known malicious models. Huang et al. [21] introduce the one-pixel signature concept for DNNs, consisting of a set of single-pixel adversarial perturbations that have the highest probability of triggering misclassifications on a specific benign image dataset. Subsequently, a meta-classifier was trained on these signatures to discern between poisoned and benign models. Kolouri et al. [25] use universal litmus patterns and a meta-classifier to detect backdoors across different network architectures. Chen et al. [9] present DeepInspect, a black-box method utilizing model inversion and anomaly detection for backdoor detection. Huang et al. [22] also introduced Neuroninspect, employing explanation heatmaps and outlier detection for identifying abnormal neurons. Unlike these methods, Sibai is tailored to FL, avoiding intensive preprocessing steps. DeepInspect [9] involves extensive reverse engineering of potential triggers for every label of each local model, and Huang *et al*. [21] requires testing every local model with a large set of single-pixel adversarial perturbations. Conducting such intricate analyses for every model in a federation during each round is impractical, as these methods were originally designed for detailed offline analysis of individual models. Sibai instead only uses PCA as the sole dimensionality reduction mechanism, facilitating straightforward inference.

7. Conclusion

To counter non-i.i.d. data distribution and adaptive adversaries in FL, we introduce Siamese Backdoor Inspector (Sibai), a novel FL poisoning defense mechanism utilizing a Siamese network meta-classifier. Inspired by successful offline backdoor detection [21, 25, 58], Sibai dynamically learns discriminative features directly from PCA decompositions of local models. In diverse evaluations, Sibai consistently achieves high detection rates without significantly impacting global model performance. It effectively prevents backdoor establishment and outperforms current defenses in terms of f1 score while maintaining low false positive rates. Future work may explore selectively applying influence reduction to models identified by Sibai for targeted defense.

8. Acknowledgment

This research has been funded by the Federal Ministry of Education and Research of Germany (BMBF) within the program Digital. Sicher. Souverän.“ in the project ”Erkennung von Angriffen gegen IoT-Netzwerke in Smart Homes - IoTGuard” (project number 16KIS1919).

References

[1] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. Baffle: Backdoor detection via feedback-based federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 852–863. IEEE, 2021. 1, 3

[2] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1505–1521, 2021. 1

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020. 1, 2, 4, 5, 8, 13, 14, 15, 16, 17, 18, 19

[4] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019. 1

[5] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017. 1, 3, 4, 7, 8, 15, 16

[6] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017. 3

[7] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a” siamese” time delay neural network. *Advances in neural information processing systems*, 6, 1993. 2

[8] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *Network and Distributed Systems Security (NDSS) Symposium 2021*, 2021. 1, 4, 6, 7, 8, 15, 16

[9] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, page 8, 2019. 1, 8

[10] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 2, 4, 7, 13, 14, 17, 18, 19

[11] Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual computer security applications conference*, pages 554–569, 2021. 1

[12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 7

[13] Elena Deza, Michel Marie Deza, Michel Marie Deza, and Elena Deza. *Encyclopedia of distances*. Springer, 2009. 2

[14] Cynthia Dwork et al. The mathematics of information coding, extraction, and distribution. *The IMA Volumes in Mathematics and its applications*, 107:31–47, 1999. 12

[15] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622, 2020. 1

[16] Hossein Fereidooni, Jan König, Phillip Rieger, Marco Chilese, Bora Gökbakan, Moritz Finke, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. Authentisense: A scalable behavioral biometrics authentication scheme using few-shot learning for mobile platforms. *arXiv preprint arXiv:2302.02740*, 2023. 3

[17] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 301–316, 2020. 1, 3, 4, 6, 7, 8, 15, 16

[18] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760*, 2020. 1, 2

[19] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 1, 2, 4, 7, 13, 17, 18, 19

[20] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, pages 84–92. Springer, 2015. 2

[21] Shanjiaoyang Huang, Weiqi Peng, Zhiwei Jia, and Zhuowen Tu. One-pixel signature: Characterizing cnn models for backdoor detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 326–341. Springer, 2020. 1, 8

[22] Xijie Huang, Moustafa Alzantot, and Mani Srivastava. Neuroninspect: Detecting backdoors in neural networks via output explanations. *arXiv preprint arXiv:1911.07399*, 2019. 1, 8

[23] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 4

[24] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*. Lille, 2015. 2, 3

[25] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 301–310, 2020. 1, 8

[26] Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016. 2

[27] Torsten Krauß and Alexandra Dmitrienko. Mesas: Poisoning defense for federated learning resilient against adaptive attackers. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, page 1526–1540, New York, NY, USA, 2023. Association for Computing Machinery. 1, 2, 3, 4, 5, 7, 8, 13, 15, 16, 17, 18, 19

[28] Torsten Krauß, Raphael Götz, and Alexandra Dmitrienko. Security of nvme offloaded data in large-scale machine learning. In *Computer Security – ESORICS 2023*, pages 143–163, Cham, 2024. Springer Nature Switzerland. 2

[29] Torsten Krauß, Jan König, Alexandra Dmitrienko, and Christian Kanzow. Automatic Adversarial Adaption for Stealthy Poisoning Attacks in Federated Learning. *Network and Distributed Systems Security (NDSS) Symposium 2024*, 2024. 1

[30] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4, 13, 14, 15

[31] Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1544–1551, 2019. 1

[32] Yijing Li, Xiaofeng Tao, Xuefei Zhang, Junjie Liu, and Jin Xu. Privacy-preserved federated learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8423–8434, 2021. 1

[33] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 1

[34] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018. 2

[35] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. 3, 4, 7, 8, 15, 16

[36] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 3, 8, 15

[37] Luis Muñoz-González, Kenneth T Co, and Emil C Lupu. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125*, 2019. 3

[38] Sridhar Narayan. The generalized sigmoid activation function: Competitive supervised learning. *Information sciences*, 99 (1-2):69–82, 1997. 3, 13

[39] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. *arXiv preprint arXiv:2009.03561*, 2020. 1

[40] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D Joseph, Benjamin IP Rubinstein, Udam Saini, Charles Sutton, J Doug Tygar, and Kai Xia. Exploiting machine learning to subvert your spam filter. *LEET*, 8(1-9):16–17, 2008. 1

[41] Anh Nguyen, Tuong Do, Minh Tran, Binh X Nguyen, Chien Duong, Tu Phan, Erman Tjiputra, and Quang D Tran. Deep federated learning for autonomous driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1824–1830. IEEE, 2022. 1

[42] Thien Duc Nguyen, Phillip Rieger, Markus Miettinen, and Ahmad-Reza Sadeghi. Poisoning attacks on federated learning-based iot intrusion detection system. In *Proc. Workshop Decentralized IoT Syst. Secur.(DISS)*, pages 1–7, 2020. 1

[43] Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1415–1432, 2022. 1, 3, 8

[44] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020. 4

[45] Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. Hidden trigger backdoor attack on {NLP} models via linguistic style manipulation. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3611–3628, 2022. 1

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4, 12

[47] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. *NNetwork and Distributed Systems Security (NDSS) Symposium 2022*, 2022. 1, 8

[48] Shiqi Shen, Shruti Tople, and Prateek Saxena. Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519, 2016. 1, 8, 16

[49] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, Lingjuan Lyu, and Ji Liu. Data poisoning attacks on federated machine learning. *IEEE Internet of Things Journal*, 9(13):11365–11375, 2021. 2

[50] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019. 1

[51] Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8):1–35, 2022. 2

[52] Alexander Turner, Dimitris Tsipras, and Aleksander

Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. 1

[53] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. 12

[54] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020. 1, 2, 4, 13, 14, 17, 18, 19

[55] Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B Giannakis. Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing*, 68:4583–4596, 2020. 1

[56] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019. 1

[57] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, pages 261–270. PMLR, 2020. 1

[58] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 103–120. IEEE, 2021. 1, 8

[59] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019. 1, 2

[60] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. 4, 6, 7, 8, 15, 16

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

[62] Hongyi Zhang, Jan Bosch, and Helena Holmström Olsson. End-to-end federated learning for autonomous driving vehicles. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 1

[63] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2545–2555, 2022. 1, 8

A. Federated Averaging

In a federation comprising multiple clients, denoted as $C_k \in \{C_1, ...C_N\}$, the server selects a subset $C_i \in \{C_1, ...C_n\}$ of size n from the N available clients. Each chosen client initializes its local model with the distributed global model ($L_i^r = G^r$) before training a new local model L_i^{r+1} using its local dataset D_i. The disparity between this local model and the global model is termed the *update*, denoted as $U_i^r = L_i^{r+1} - G^r$. In the application of the FedAVG algorithm, the server aggregates these updates by computing the weighted average of all updates, utilizing the global learning rate δ as expressed in Equation (3).

$$G^{r+1} = G^r + \delta \left(\frac{1}{n} \sum_{i=0}^{n-1} U_i^r \right) \tag{3}$$

B. Model Accuracies

For a benign test set D_{test} consisting of samples with correctly labeled predictions y, we evaluate the prediction performance of a model, referred to as the model accuracy (MA). As expressed in Equation (4), the MA is determined as the fraction of samples d in D_{test} that the new global model G^{r+1} correctly classifies, divided by the total size of D_{test},

$$MA = \frac{|(d, y) \in D_{test} : f(d, G^{r+1}) = y|}{|D_{test}|} \tag{4}$$

In the assessment of a model's prediction performance for a backdoor task, an adversary A, controlling one or more clients (C_i) within the federation ($A_C \subseteq C_1, ...C_N$), aims to submit manipulated local models to the server. The goal is to influence the aggregated model G^{r+1} to produce a predefined target prediction p when presented with an input sample d^T containing the attacker-designated trigger T. Both the target label p and the trigger T are freely chosen by the attacker. The success of a backdoor attack is measured by the global model's prediction performance on triggered inputs, quantified as the backdoor accuracy (BA). As indicated in Equation (5), the BA is the fraction of triggered input samples d^T in a malicious test set D_{test}^T, which exclusively contains such samples, relative to the total size of the malicious test set D_{test}^T,

$$BA = \frac{|(d^T, P) \in D_{test}^T : f(d^T, G^{r+1}) = P|}{|D_{test}^T|} \tag{5}$$

C. Implementation

This section provides an in-depth look into our implementation of Sibai, focusing on key aspects and challenges encountered during development. As aforementioned, a primary concern was efficiently managing the substantial input size, since keeping multiple shadow models concurrently in GPU memory can be difficult because of their substantial size. The detailed architectures of the twin networks and the decision network are listed in Appendix C.4. Our implementations used PyTorch [46], a popular Python [53] machine learning library.

C.1. Training Strategy

To ensure stability both networks undergo training individually. The Siamese network is trained first using the triplet loss method. Each epoch employs all global models as anchors, with positives and negatives randomly selected from available local models. This approach effectively increases the distance between benign and malicious samples, aligning with Sibai's goals. Once the Siamese network stabilizes, training for the decision network begins. It utilizes the binary cross entropy (BCE) loss [14], the standard loss function for binary classifiers with a Sigmoid output layer.

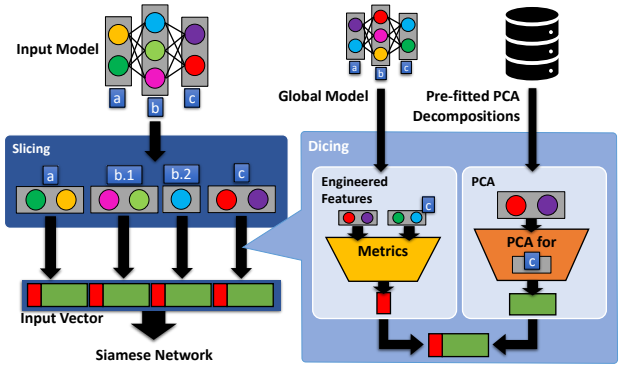

Figure 2. Application of the PCA pipeline to a local model. The dicing process is applied to each slice individually, using a slice-specific PCA decomposition (in this example for slice c) that was fitted before the training started.

C.2. Dimensionality Reduction and Feature Engineering

Our PCA-based feature reduction operates in two phases. Initially, before training begins, a PCA decomposition is pre-computed for each slice individually in a process called *fitting*, and the results are stored for subsequent use. As illustrated in Fig. 2, when processing a model with the Siamese network, the pre-computed PCA decomposition is applied to each slice, and Euclidean and cosine distances to the respective slice of the global model are calculated. The resulting slice vectors are concatenated to generate a unified flat representation. The PCA decomposition retains a fixed number of principal components, which means it preserves all information for slices with fewer parameters than that number of components.

C.3. Shadow Model Dataset

We employed our custom FL simulation framework to generate shadow models for our dataset. Each federation consisted of 20 models: 11 benign and 9 malicious, poisoned with one of five backdoors detailed in Sect. D. To ensure uniqueness, each federation had a distinct seed and dataset split. Federations were divided into a training set, used for Sibai's training, and a test set for performance metrics. However, that test set was not used in our experiments; instead, Sibai was integrated into the FL framework as a DF-based defense and applied before aggregation in entirely new federations, distinct from both training and test sets.

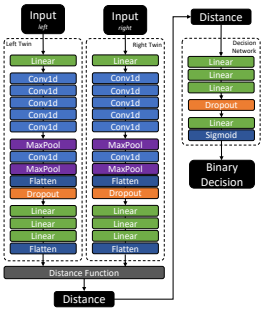

Figure 3. Model Architecture of Sibai: Siamese twin networks followed by a small fully connected decision network

C.4. Model Architecture

This section elaborates on the detailed architectures of the Siamese networks of Sibai used throughout this paper. At the core of Sibai lies the twin conventional neural networks designed to learn feature representations that effectively maximize the distances between benign and poisoned models. As shown in Fig. 3 the structure of each of the twins unfolds as follows: Beginning with a linear layer, the model compresses the dimensions of the input vector. Subsequently, a sequence of five 1-dimensional convolutional layers follows, gradually increasing the number of filters from 64 up to 1024. As the layers progress, the kernel size decreases from 5 to 3, facilitating more nuanced computations. A critical down-sampling step follows through one-dimensional max pooling with a spatial window of 4, ensuring the retention of crucial features. The process continues with another convolutional layer that reduces the number of filters back to 64. This is complemented by a second max pooling layer, filtering out more irrelevant features. The output is then flattened and subjected to a dropout layer to prevent overfitting. To address the issue of large flattened vectors, a sequence of three linear layers progressively reduces the length to a fixed size of 512 elements. The Siamese network takes two local models as input and guides each through the twin convolutional networks. Afterward, the L_2-distance of the two resulting 512-element long feature vectors is computed and then handed to the compact decision network. This network is tasked to learn a distance threshold at which one of the models should be flagged as potentially malicious. As shown in Fig. 3 the architecture of this network consists of four linear layers: Initially, the first two layers work in tandem, expanding the input distance vector to encompass 64 and 128 features, respectively. The third layer reduces the feature vector to a compact 32-dimensional representation. To ensure effective generalization, a dropout layer is applied. The final linear layer then brings the representation down to a single feature. A Sigmoid [38] activation function is employed at this stage to produce an output from 0 to 1. This output signifies the probability that one of the models is potentially malicious.

D. Poisoning Methods

In this section, we explain the backdoor trigger methods, that malicious clients can leverage to poison the local models. In our evaluation we use pixel triggers [19], semantic backdoor [3], edge case backdoor [54], and the pervasive backdoor, also known as blend backdoor [10]. Additionally, we employ one untargeted attack in the form of random label flipping [27].

D.0.1. Pixel Trigger Backdoor

Figure 4. Visualization of the pixel trigger backdoor [19] on an example from the CIFAR-10 [30] dataset. The color of the pixel patch is the maximal RGB color of the image. (a) shows a benign sample and (b) shows a malicious one with the embedded trigger

The pixel trigger backdoor [19] is a well-established image classification-specific attack. During model training, a pixel pattern is added to benign input images which are then labeled with an attacker-chosen label. The model associates the pixel pattern with the attacker label, learning to disregard the underlying image in the presence of the trigger. During inference, when the trigger is added to any arbitrary input image, the model focuses on it and classifies the image as the attacker-chosen target label, regardless of the actual underlying image. Fig. 4 depicts an example of the pixel triggers used throughout this paper. The color, value, and location of the pixel trigger significantly impact the BA. For color selection, we opt for the maximum color of the first image seen by any adversary and broadcast this color to other adversarial clients. This choice makes the color less conspicuous and challenging for defenders to detect. The trigger has a quadratic shape, with a size of $\frac{1}{16}$ of the sample width, positioned in the upper left corner of the image.

D.0.2. Blend Backdoor

Figure 5. Visualization of the blend backdoor [10] on an example from the CIFAR-10 [30] dataset. The trigger is a pattern of seemingly random noise overlayed on the image. (a) shows a benign sample and (b) shows a malicious one with the noise pattern overlayed at 10% opacity

The pixel trigger backdoor has a significant drawback: malicious input images are easily recognizable as poisoned by any human observer. To address this issue, the blend backdoor [10] conceals the trigger throughout the entire image, by overlaying a layer of seemingly random noise onto it. At low intensity, this noise may be invisible to humans, yet it still alters the DNN's perception which has learned to associate these noise patterns with the attacker's target label, leading to misclassifications. Examples of a poisoned sample can be seen in Fig. 5

D.0.3. Edge Case Backdoor

An edge-case backdoor [54] specifically targets inputs located at the tail of the input distribution. These inputs are commonly prone to confusion by the model. Unlike other attacks this backdoor does not incorporate an additional trigger, instead, it simply involves identifying and deliberately mislabeling the specific edge case specific samples in the adversary's dataset. Consequently, the likelihood of accidentally triggering the backdoor is higher, however, due to the inherent difficulty of correctly classifying these samples, the model's misclassifications may appear as honest mistakes. It's noteworthy that even on a completely benign model, the BA for this attack might seem somewhat high, especially when compared with those of other attacks, given that the misclassifications are somewhat natural and simply get artificially amplified

Figure 6. Samples from the CIFAR-10 [30] dataset used for the edge-case backdoor [54]. The blue airplanes of Southwest airline are on the tail end of the distribution for the airplane class, and therefore an adversary deliberately mislabeled them as trucks

by the adversary. We implemented CIFAR-10 [30] version of this backdoor attack, deliberately mislabeling images of airplanes from the Southwest airline as trucks. An example of such images can be seen in Fig. 6

D.0.4. Semantic Backdoor

In the semantic backdoor attack [3], attackers intentionally mislabel benign images containing specific characteristics. Therefore, similar to the edge case backdoor [54], the semantic backdoor causes the model to produce an attacker-chosen output on completely unmodified inputs. For instance, in an image classification model, the attacker assigns a chosen label to all images with certain features, such as misclassifying all cars in front of a striped background. Similar to the blend backdoor, is exceptionally difficult for humans to detect that an image contains a semantic backdoor simply be observing it, especially considering that the trigger may encompass a varied set of arbitrarily complex image characteristics. Fig. 7

Figure 7. Visualisation of a semantic backdoor [3] with samples from the CIFAR-10 [30] dataset with cars in front of a striped background as a trigger. (a) is an example of a benign car sample without the trigger, while (b) and (c) contain the combination of car and striped background that triggers the misclassification.

illustrates a semantic backdoor on CIFAR-10 [30], which we utilize in our experiments. To ensure the uniqueness of the trigger, samples containing it are excluded from the training datasets of benign FL clients.

D.0.5. Random Label Flipping

Unlike targeted attacks, which aim to induce misclassifications in very specific cases that are unlikely to be triggered by accident, untargeted attacks are less covert. They focus on degrading the model's overall performance, introducing arbitrary misclassification, and generally hindering its correct functioning. Random Label Flipping [27] is a direct method for executing untargeted attacks, achieved by assigning random incorrect labels to benign samples. This causes the model to generate high loss values for correct classifications, leading to a process of unlearning. While this type of attack might be noticeable to federation operators due to a reduced MA, it still holds the potential to cause significant damage. In the best cases, it may necessitate a model rollback to a previous checkpoint, resulting in time and potential financial losses. In the worst-case scenario, a successful untargeted poisoning attack might require a complete reset of the training process.

E. Evaluation Hyperparameters

This section outlines the hyperparameters for the defenses Sibai was compared in our evaluations, aiming to offer a comprehensive overview of our experimental setup. In the case of trimmed mean [60], we trim the upper and lower 5% to mitigate the outliers. The threshold for both Krum and M-Krum [5] is set to 0.7 and the rate of clients considered for M-Krum is 0.3. For FLTrust [8], a sizable root dataset containing 200 elements was utilized.

F. F1 Score Calculations for Reference Defenses

Calculating F1 scores for all reference poisoning defenses posed challenges, especially for IR-based methods that rely on modified aggregation rather than strictly flagging and filtering out malicious models. For instance, FoolsGold [17] utilizes a weighted aggregation system, making it unsuitable for explicit classification of models as malicious or benign. The same applies to Trimmed mean [60] and trimmed median [60] which replace FedAVG [36] with their own robust aggregation algorithms designed to diminish the influence of poisoning. DF-based methods like Krum [5], M-Krum [5], HDBSCAN [35], and MESAS [27] directly identify and remove malicious models, allowing direct calculations of precision, recall and F1 score. However, FLTrust [8] also adopts a weighted aggregation system, assigning weights based on cosine and Euclidean distances to a benign reference model, which is trained server-side on a small trusted dataset. However, in our experiments, we observed that the malicious models usually get assigned exceedingly low weights by FLTrust, which make them effectively have no tangible impact on the aggregated global model. We, therefore, calculate FLTrust's F1 score using a 0.09 weight threshold, designating models below it as effectively filtered. This threshold stems from observations where overtly malicious models, recognized by most defenses, received weights of 0.08 or below. Note that using a different threshold would alter FLTrust's F1 scores, but it wound not affect its MA and BA performance.

G. Additional Evaluation

This section provides evaluation results that were omitted from the main paper for space reasons.

G.1. 0.3 1-Class non-i.i.d.

In scenarios (A) and (B), shown in Tab. 10, a 1-class non-i.i.d. disproportionality level q of 0.3 was employed. Sibai achieved perfect detection with F1 scores of 1.0 in the 0.5 PDR scenario (A) and exhibited perfect precision in the low PDR scenario (B), with no false positives. There was one false negative for the semantic backdoor, however, the global model's BA remained at 0.

q	Scenario		Random	Pixel	Blend	Edge	Semantic
0.3	(A)	PDR	0.5	0.5	0.5	0.5	0.5
		Precision	1.0000	1.0000	1.0000	1.0000	1.0000
		Recall	1.0000	1.0000	1.0000	1.0000	1.0000
		F1	1.0000	1.0000	1.0000	1.0000	1.0000
		MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
		BA	-	3.52%	0.0%	13.77%	0.0%
	(B)	PDR	0.4	0.08	0.04	0.16	0.04
		Precision	1.0000	1.0000	1.0000	1.0000	1.0000
		Recall	1.0000	1.0000	1.0000	1.0000	0.8888
		F1	1.0000	1.0000	1.0000	1.0000	0.9411
		MA diff.	0.0%	0.0%	0.0%	0.0%	+1.34%
		BA	-	2.65%	0.20%	8.67%	0.0%

(A) = Default + 0.3 1-class non-i.i.d. + high PDR

(B) = Default + 0.3 1-class non-i.i.d. + low PDR

Table 10. Detection performance of Sibai in 1-class non-i.i.d. scenarios, with different level of disproportionality q

G.2. Runtime Evaluation

Defense	Runtime
MESAS [27]	37.25s
Krum [5]	5.61s
M-Krum [5]	5.70s
FoolsGold [17]	0.17s
HDBSCAN [35]	0.18s
trimmed mean [60]	0.61s
trimmed median [60]	0.03s
FLTrust [8]	25.8s
Auror [48]	12 hours
Sibai (ours)	5.27s

Table 11. Runtime of defenses in seconds during the aggregation process

This section assesses the runtime performance of Sibai. The offline training phase took 23 hours and 40 minutes, with each epoch lasting about half an hour. Despite this initial investment, given the large-scale nature of FL and the advantages of Sibai, such as robust defense against adaptive attacks, this one-time effort before launching the federation is generally acceptable. During the online inference phase, Sibai demonstrates a reasonably average performance impact. As shown in Tab. 11, in scenario (0), Sibai's runtime, including PCA decomposition and Siamese network inference for all 20 models, is 5.27 seconds. This aligns with other recent defenses like Krum and M-Krumm [5]. Naïve cosine-distance based HDBSCAN [35] and robust aggregation methods such as trimmed mean, median [60] show minimal impact (less than one second) but also exhibit inferior detection performance compared to Sibai. The Auror [48] method introduces extreme performance overhead and was excluded from other experiments in our evaluation.

G.3. Alpha Parameter for Adaptive Adversaries

In this section, we present an overview of experiments testing Sibai against adaptive adversaries, providing results for experiments conducted against attackers employing different levels of adaptation. These attackers utilize the method introduced by Bagdasaryan *et al*. [3], adjusting the α parameter to balance the trade-off between optimizing BA and stealthiness. Our experiments cover a range of alpha values from 0.1 to 0.9 and maintain a consistent seed across all experiments to mitigate the impact of ML-related randomness on the results.

The results in Tab. 12 show the results of experiments involving an attacker utilizing both high PDR and cosine distance adaptation. Sibai demonstrates robust detection for all α values, reliably identifying all malicious models.

In Tab. 13 an attacker employs cosine distance adaptation alongside low PDR values. False negatives occur only at very high levels of adaptation (α values of 0.8 and 0.9), however, at this point, the malicious local models are so focused on adaptation that they fail to effectively introduce a backdoor, resulting in minimal impact on MA and very low BA values.

The outcomes for the adversary employing Euclidean adaptation and high PDR in Tab. 14 closely mirror the high PDR cosine adaptation experiment results in Tab. 12. Sibai consistently identifies all malicious models, regardless of the α value employed by the attacker.

The scenario involving Euclidean adaptation and low PDRs appears to be the most challenging for Sibai among all adaptive attacker scenarios. For high α values (0.7 to 0.9), Sibai easily identifies all malicious models, possibly because their strong focus on adaptation amplifies the indicators of poisoning that Sibai has learned to detect. However, for balanced α values (0.2 to 0.5), Sibai exhibits some false negatives, particularly for the pixel and blend backdoors. Despite these instances, the BA remains extremely low, indicating that no effective backdoor was introduced. The impact on model accuracy MA was also minimal; for instance, in the worst case, there were 3 false negatives for the pixel trigger backdoor with an α of 0.5, reducing the MA by 9.27%. These experiments suggest that Sibai is resilient against adaptive adversaries targeting cosine and Euclidean distance. Generally, an α between 0.2 and 0.5 has the most impact, although this impact is still very limited. We, therefore, chose to showcase the results for an α of 0.3 in the main paper.

H. Shadow Model Dataset Composition

Table 16 provides a detailed breakdown of the composition of the shadow model dataset used to train the Siamese network that was evaluated in the experiments showcased in the main paper. Initially, each federation trained 11 benign and 9 malicious models. However, malicious models with a local BA below 45% were subsequently removed from the dataset. This exclusion was necessary because such models exhibit backdoors that are too inefficient to effectively manifest the typical characteristics of poisoned models. Moreover, detecting such models is somewhat futile because of their weak local poisoning, they are very unlikely to introduce any significant backdoor to the global model when aggregating. Therefore, removing them improves the quality of the classifier.

α	Metrics	Random [27]	Pixel [19]	Blend [10]	Edge [54]	Semantic [3]
0.1	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.2	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.3	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.4	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.5	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.6	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.7	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.8	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.9	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%

Table 12. Detection performance of Sibai against attackers using cosine distance adaptation with different α levels and high PDR

α	Metrics	Random [27]	Pixel [19]	Blend [10]	Edge [54]	Semantic [3]
0.1	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%
0.2	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%
0.3	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%
0.4	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%
0.5	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%
0.6	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%
0.7	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%
0.8	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	0.8888	0.8888	0.8888
	F1	1.0000	1.0000	0.9411	0.9411	0.9411
	MA diff.	0.0%	0.0%	-0.95%	-1.46%	+2.59%
	BA	-	1.08%	0.18%	19.38%	0.0%
0.9	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	0.7777	1.0000	0.6666	1.0000	1.0000
	F1	0.8750	1.0000	0.8000	1.0000	1.0000
	MA diff.	-1.27%	0.0%	+1.83%	0.0%	0.0%
	BA	-	1.08%	0.72%	18.87%	0.0%

Table 13. Detection performance of Sibai against attackers using cosine distance adaptation with different α levels and low PDR

α	Metrics	Random [27]	Pixel [19]	Blend [10]	Edge [54]	Semantic [3]
0.1	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.2	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.2%	18.87%	0.0%
0.3	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.4	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.5	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.6	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.7	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.8	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%
0.9	PDR	0.5	0.5	0.5	0.5	0.5
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.20%	18.87%	0.0%

Table 14. Detection performance of Sibai against attackers using Euclidean distance adaptation with different α levels and high PDR

α	Metrics	Random [27]	Pixel [19]	Blend [10]	Edge [54]	Semantic [3]
0.1	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	0.7777	1.0000	1.0000	1.0000
	F1	1.0000	0.8750	1.0000	1.0000	1.0000
	MA diff.	0.0%	-1.1%	0.0%	0.0%	0.0%
	BA	-	1.34%	0.14%	18.87%	0.0%
0.2	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	0.8888	0.7777	1.0000	0.8888
	F1	1.0000	0.9411	0.8750	1.0000	0.9411
	MA diff.	0.0%	+2.09%	-6.02%	0.0%	-6.42%
	BA	-	2.05%	0.27%	18.87%	0.0%
0.3	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	0.8888	0.8888	1.0000	1.0000
	F1	1.0000	0.9411	0.9411	1.0000	1.0000
	MA diff.	0.0%	+3.1%	-6.3%	0.0%	0.0%
	BA	-	5.98%	0.0%	18.87%	0.0%
0.4	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	0.8888	1.0000	1.0000	1.0000
	F1	1.0000	0.9411	1.0000	1.0000	1.0000
	MA diff.	0.0%	+1.05%	0.0%	0.0%	0.0%
	BA	-	0.0%	0.07%	18.87%	0.0%
0.5	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	0.7777	0.8888	1.0000	1.0000
	F1	1.0000	0.8750	0.9411	1.0000	1.0000
	MA diff.	0.0%	-9.27%	-7.86%	0.0%	0.0%
	BA	-	0.11%	0.0%	18.87%	0.0%
0.6	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	0.8888	1.0000	1.0000
	F1	1.0000	1.0000	0.9411	1.0000	1.0000
	MA diff.	0.0%	0.0%	-8.36%	0.0%	0.0%
	BA	-	1.08%	0.03%	18.87%	0.0%
0.7	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	0.8888	1.0000	1.0000
	F1	1.0000	1.0000	0.9411	1.0000	1.0000
	MA diff.	0.0%	0.0%	-8.01%	0.0%	0.0%
	BA	-	1.08%	0.05%	18.87%	0.0%
0.8	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%
0.9	PDR	0.4	0.08	0.04	0.16	0.04
	precision	1.0000	1.0000	1.0000	1.0000	1.0000
	recall	1.0000	1.0000	1.0000	1.0000	1.0000
	F1	1.0000	1.0000	1.0000	1.0000	1.0000
	MA diff.	0.0%	0.0%	0.0%	0.0%	0.0%
	BA	-	1.08%	0.14%	18.87%	0.0%

Table 15. Detection performance of Sibai against attackers using Euclidean distance adaptation with different α levels and low PDR

Attack	PDR	i.i.d./non-i.i.d.-ness	Adv. Adaptation	# Federations	# Benign Models	# Malicious Models
Blend [10]	0.02	i.i.d.	none	8	88	72
	0.04	i.i.d.	none	8	88	63
	0.08	i.i.d.	none	8	88	72
	0.5	i.i.d.	none	1	11	9
	0.8	i.i.d.	none	2	22	18
	0.02	0.3 1-class	none	8	88	68
	0.02	0.6 1-class	none	8	88	59
	0.02	0.9 1-class	none	8	88	54
	0.04	0.3 1-class	none	7	77	62
	0.04	0.6 1-class	none	7	77	57
	0.04	0.9 1-class	none	7	77	60
	0.5	0.9 1-class	none	2	22	18
	0.8	0.9 1-class	none	1	11	9
	0.9	0.9 1-class	none	1	11	9
	0.04	inter-client	none	7	77	52
	0.5	inter-client	none	7	77	61
	0.4	i.i.d.	cosine $\alpha = 0.3$	8	88	72
	0.4	i.i.d.	Euclid $\alpha = 0.3$	8	88	66
	0.5	i.i.d.	cosine $\alpha = 0.3$	7	77	63
	0.5	i.i.d.	Euclid $\alpha = 0.3$	7	77	58
Edge [54]	0.08	i.i.d.	none	5	55	13
	0.16	i.i.d.	none	7	77	30
	0.2	i.i.d.	none	4	44	27
	0.4	i.i.d.	none	1	11	9
	0.6	i.i.d.	none	1	11	9
	0.8	i.i.d.	none	1	11	9
	0.9	i.i.d.	none	1	11	9
	0.08	0.9 1-class	none	8	88	34
	0.16	0.3 1-class	none	7	77	37
	0.16	0.6 1-class	none	7	77	41
	0.16	0.9 1-class	none	7	77	35
	0.18	0.3 1-class	none	8	88	49
	0.18	0.6 1-class	none	8	88	53
	0.18	0.9 1-class	none	8	88	46
	0.5	0.9 1-class	none	2	22	18
	0.8	0.9 1-class	none	1	11	9
	0.9	0.9 1-class	none	1	11	9
	0.16	inter-client	none	7	77	48
	0.5	inter-client	none	7	77	60
	0.4	i.i.d.	cosine $\alpha = 0.3$	9	99	81
	0.4	i.i.d.	Euclid $\alpha = 0.3$	8	88	39
	0.5	i.i.d.	cosine $\alpha = 0.3$	8	88	72
	0.5	i.i.d.	Euclid $\alpha = 0.3$	7	77	50
Pixel [19]	0.04	i.i.d.	none	6	66	36
	0.05	i.i.d.	none	1	11	3
	0.1	i.i.d.	none	1	11	9
	0.04	0.3 1-class	none	3	33	20
	0.04	0.6 1-class	none	8	88	46
	0.04	0.9 1-class	none	8	88	33
	0.08	0.3 1-class	none	7	77	48
	0.08	0.6 1-class	none	3	33	23
	0.08	0.9 1-class	none	7	77	42
	0.5	0.9 1-class	none	2	22	18
	0.8	0.9 1-class	none	1	11	9
	0.9	0.9 1-class	none	1	11	9
	0.08	inter-client	none	7	77	52
	0.5	inter-client	none	7	77	63
	0.4	i.i.d.	cosine $\alpha = 0.3$	8	88	72
	0.4	i.i.d.	Euclid $\alpha = 0.3$	8	88	72
	0.5	i.i.d.	cosine $\alpha = 0.3$	7	77	63
	0.5	i.i.d.	Euclid $\alpha = 0.3$	7	77	63
Semantic [3]	0.02	i.i.d.	none	6	66	53
	0.05	i.i.d.	none	1	11	9
	0.08	i.i.d.	none	1	11	9
	0.02	0.3 1-class	none	8	88	72
	0.02	0.6 1-class	none	8	88	68
	0.02	0.9 1-class	none	8	88	61
	0.5	0.9 1-class	none	2	22	9
	0.8	0.9 1-class	none	1	11	9
	0.9	0.9 1-class	none	1	11	9
	0.04	0.3 1-class	none	7	77	63
	0.04	0.6 1-class	none	7	77	61
	0.04	0.9 1-class	none	7	77	59
	0.04	inter-client	none	7	77	59
	0.5	inter-client	none	7	77	54
	0.4	i.i.d.	cosine $\alpha = 0.3$	8	88	72
	0.4	i.i.d.	Euclid $\alpha = 0.3$	8	88	71
	0.5	i.i.d.	cosine $\alpha = 0.3$	7	77	63
	0.5	i.i.d.	Euclid $\alpha = 0.3$	7	77	63
Random [27]	0.4	i.i.d.	none	1	11	9
	0.5	i.i.d.	none	2	22	18
	0.8	i.i.d.	none	1	11	9
	0.2	i.i.d.	none	8	88	72
	0.2	0.3 1-class	none	8	88	72
	0.2	0.6 1-class	none	8	88	72
	0.2	0.9 1-class	none	8	88	72
	0.4	0.3 1-class	none	3	33	27
	0.4	0.6 1-class	none	7	77	63
	0.4	0.9 1-class	none	7	77	63
	0.8	inter-client	none	7	77	63
	0.4	i.i.d.	cosine $\alpha = 0.3$	8	88	72
	0.4	i.i.d.	Euclid $\alpha = 0.3$	8	88	72
	0.5	i.i.d.	cosine $\alpha = 0.3$	7	77	63
	0.5	i.i.d.	Euclid $\alpha = 0.3$	7	77	63
Sum				520	5720	4145

Table 16. Detailed breakdown of the shadow model dataset composition used to train Sibai