

Can Federated Learning Backdoor Defenses Protect Parallel Split-Federated Learning?

Yelyzaveta Klysa, Milica Aleksic, Rudolf Mayer
SBA Research, Floragasse 7, 1040 Vienna, Austria
{yklysa,rmayer}@sba-research.org

Abstract—With the growing use of collaborative learning paradigms in distributed edge applications, hybrid approaches such as Split-Federated Learning (SFL) have emerged to combine the benefits of federated and split learning. Despite the increasing adoption of SFL to improve the trustworthiness of collaborative AI systems, the security of these systems has not been systematically studied, particularly with respect to backdoor attacks known to affect federated learning (FL). While prior work has demonstrated that SFL is vulnerable to such attacks, it remains unexplored whether state-of-the-art FL defenses provide reliable protection in this setting.

In this work, we define multiple realistic threat models based on trust assumptions and the timing of defense deployment. We then analyze representative FL backdoor defenses in the parallel SFL setting under these models. Our findings show that in SFL, defense effectiveness is highly dependent on the entity responsible for defense deployment. Among the considered methods, only one, namely *AlignIns*, is found effective across most threat models. Furthermore, in delayed-defense settings where the backdoor is already (at least partially) embedded into the model, defense effectiveness strongly depends on dataset complexity.

Index Terms—split-federated learning, machine learning security, backdoor defenses

I. INTRODUCTION

Collaborative learning enables multiple participants to jointly train a model, thereby leveraging diverse data sources without sharing their private data. One of the most widely adopted approaches is Federated Learning (FL), in which clients collaboratively train a shared global model by exchanging locally computed model updates without the training data ever leaving the client site. However, this setting assumes that each client has sufficient computational resources to train a full model locally, which may not hold in edge and IoT deployments. Split Learning (SL) has been proposed as an approach to address the limitations of resource-constrained devices. In SL, the model is partitioned at one or more designated (so-called *cut*) layers into two (or more) parts: e.g., a shallow client-side submodel and a deeper server-side submodel that receives the outputs of the client-side submodel as input. Split-Federated Learning (SFL) combines

elements of both FL and SL: clients train only a partial model, as in SL, and the resulting models are aggregated across clients, resembling FL. This hybrid design aims to reduce the computational burden on clients while enabling collaborative training across distributed data.

The residual and novel risks of these collaborative learning paradigms require a careful analysis, and purposefully designed mitigation strategies are crucial for their secure deployment. Among these risks, a malicious participant who controls one or more clients may inject a backdoor [1] into the model, for instance, by data poisoning. A model trained on such data then produces adversary-chosen outputs when presented with specific trigger patterns, while maintaining benign performance on clean inputs. Moreover, the exchanged intermediate representations may still carry sensitive information. By exploiting them, an adversary who controls the server or intercepts communication may attempt to compromise the confidentiality of training data.

While prior work in Federated Learning has extensively studied poisoning attacks [2], existing research in SL and SFL [3], [4] has predominantly focused on confidentiality risks with comparatively limited attention to backdoor attacks [5]. Consequently, SFL still lacks a systematic evaluation of its vulnerability to backdoor attacks and the effectiveness of corresponding defenses. Although numerous defenses against backdoor attacks have been proposed for FL settings, it remains unclear whether these state-of-the-art mechanisms can be effectively transferred to SFL, given its split architecture and more interaction points among entities. Moreover, the flexibility in SFL architectures, particularly in terms of model partitioning and defense location deployment, introduces additional complexity. As a result, the security implications of different design choices in SFL, especially regarding backdoor attacks, remain largely underexplored.

In this work, we investigate the effectiveness of state-of-the-art federated learning defenses against backdoor attacks within a parallel split-federated learning setting. In our preliminary study [6], we examined the impact of defense placement on backdoor mitigation in SFL, finding initial evidence that effectiveness may depend on defense location. Here, we conduct the evaluation across threat models, datasets, and model architectures to assess whether this behavior persists. We define the threat models under which an adversary operates and consider three primary scenarios: (i) the server aims to protect the server-side model from malicious client updates, (ii) clients

This work was partly funded by the European Union’s Horizon Europe Research and Innovation Programme through the CoGNETs project under Grant Agreement No 101135930. This publication reflects only the authors’ view, and the European Commission is not responsible for any use that may be made of the information it contains. The work was also partly funded by SBA Research (SBA-K1 NGC), a COMET Center within the COMET – Competence Centers for Excellent Technologies Programme and funded by BMIMI, BMWET, and the federal state of Vienna, managed by FFG.

seek to protect their part of the model from other malicious clients, and (iii) both parties collaborate to protect against malicious clients by deploying defenses across both sides of the system. In addition, we introduce a delayed-defense scenario for the last setting, in which defenses are activated only after a certain training round r . This setting models situations where security mechanisms are introduced after deployment, e.g., an existing collaborative training pipeline is retrospectively hardened, or suspicious behavior is detected only after training has progressed. It allows assessing whether an already embedded backdoor can still be effectively mitigated. Employing three representative backdoor attacks and five defenses, selected based on the BackFed benchmark [7], we demonstrate that the effectiveness of backdoor defenses in SFL depends heavily on the trust assumptions and access constraints imposed by the threat model. We consider BadNets as a foundational data-poisoning attack, DBA as a distributed variant, and Neurotoxin as a model-poisoning approach, together covering diverse strategies. For defenses, we select representative methods from different categories defined in BackFed, focusing on high-performing defenses suitable for deployment in the SFL setting. SFL introduces distinct entities with different levels of access to model updates and intermediate representations, enabling defense mechanisms to be applied at multiple points in the system. Consequently, the effectiveness of a defense depends not only on the method itself but also on which entity deploys it. Our contributions in this paper are thus as follows:

- To the best of our knowledge, we are the first to provide a structured comparison of backdoor defenses in a parallel split-federated learning setting.
- We show that defense effectiveness strongly depends on the deploying entity.
- We find that state-of-the-art FL backdoor defenses fail to protect client-side models in SFL when applied solely at the client aggregation point.
- We systematically evaluate FL backdoor defenses under diverse threat models, identifying the most effective strategies in the SFL setting.
- Our findings reveal that most of the defenses considered fail under the delayed-defense deployment.
- The code is publicly available on GitHub¹.

The remainder of the paper is structured as follows: Section II provides the necessary background, Section III presents the methodology, Section IV discusses the defense evaluation, and Section V summarizes the main findings.

II. BACKGROUND AND RELATED WORK

This section provides background on collaborative learning paradigms and an overview of prior work on the backdoor attacks and defenses considered.

A. Collaborative Learning Paradigms

In Federated Learning, multiple clients collaboratively train a global model. In horizontal FL, all clients use the same

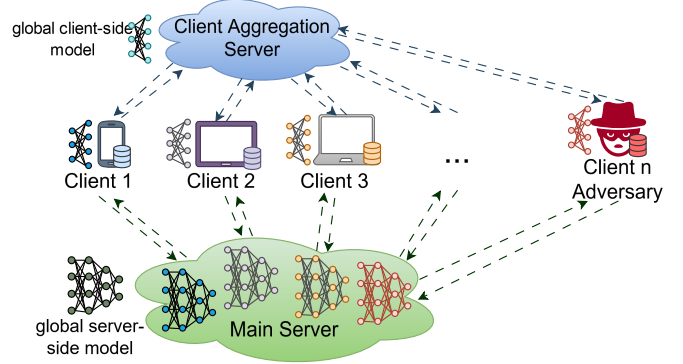


Fig. 1: Overview of actors and communication in the parallel SFL architecture (own work).

model architecture and train it in parallel on their local data partitions, which share the same feature space. The locally computed model updates are then sent to the server, which aggregates them into a global model that is redistributed to clients for subsequent training rounds. In contrast, vertical FL considers settings where clients hold data with different feature spaces, and model training and aggregation are performed differently, often without requiring identical model structures across participants.

Vanilla Split Learning consists of two nodes: a client and a server, with the model split at a *cut layer* [8]. Unlike FL, neither raw data nor model weights are shared; only the intermediate representations, i.e., the outputs of the last layer of the client’s model, also called *smashed data*, are exchanged. A client trains the first, typically less computationally demanding, part of the model on local data and sends the intermediate representations to the server, which holds the second part of the model. The server completes the forward pass, computes the loss, performs model adaptation on the available layers via backpropagation, and finally passes the resulting gradients to the client. In general, SL can be implemented using various splitting strategies, e.g., U-shaped architectures, in which clients hold both the first and last layers of the model. In this work, we focus on the widely used single-cut split setting, with a single cut layer between the client and the server. This configuration serves as a foundation for analyzing the effectiveness of backdoor defenses in split-based learning systems.

Extending split learning to a multi-client setting enables collaborative training across multiple data holders. However, this requires coordinating updates from multiple clients and ensuring a consistent shared model across training rounds. Split-federated learning has emerged to facilitate such a process. In the literature [3], [9]–[11], different deployment settings have been proposed. Thapa et al. [10] introduced two approaches for implementing split-federated learning. One of them, the parallel SFL architecture, which the authors call *SFLVI*, is the architecture we consider in this work. It consists of clients, a Client Aggregation Server (CAS), and a Main Server (MS), as depicted in Figure 1. As in SL, within each

¹<https://github.com/sbaresearch/PSFLBD>

global round, clients train the first part of the model on local data and send smashed data to the Main Server. The Main Server holds a separate server-side model instance for each client during a training round. Each instance is updated independently using the corresponding client's smashed data, and gradients at the split layer are returned to clients for local updates. At the end of each round, as in FL, client-side models are aggregated at the CAS, enabling mutual benefit across clients. Likewise, the server-side models are aggregated into a single global model that is used to initialize the Main Server in the next round. This design mirrors federated learning on both the client and server components. An alternative sequential variant, called *SFLV2*, maintains a single server-side model that is updated *sequentially* with smashed data from different clients in random order. While computationally simpler on the server side, individual client updates are applied one after another, making each client's influence dependent on the update order and the intermediate server model state.

B. Backdoor attacks and defenses

In collaborative learning architectures, one or more malicious clients can inject a backdoor into the model through, e.g., data poisoning. The goal of the attack is for a model during inference to perform normally on benign inputs while producing attacker-specific outputs when a specific trigger pattern is detected in the input [12], thereby affecting the model's integrity. Among the first to demonstrate this vulnerability was the BadNets attack [13], where poisoned samples are used to associate a trigger (e.g., a yellow square) with a target label, leading to high-confidence misclassification at inference time. Since then, various backdoor attacks have been designed in federated [1], [14], [15] and split learning [16], [17]. Dao et al. [7] developed a benchmark to systematically evaluate backdoor attacks and defenses in FL. For SFL, Zahir-Ismail et al. [5] analyzed the effectiveness of data poisoning attacks in a sequential SFL architecture, revealing that it is highly susceptible to such attacks and that appropriate defense mechanisms are required.

In FL, defenses against data poisoning typically modify the aggregation rule or detect and suppress anomalous client updates [1]. In SL, defenses instead analyze smashed data for anomalies [18]. However, (i) how such defenses should be designed and deployed in SFL, (ii) where in the SFL architecture they are most effective, and (iii) whether state-of-the-art FL defenses effectively transfer to SFL remain open questions. Furthermore, the impact of different threat models (cf. Section III-A) on defense effectiveness in SFL is largely unexplored.

III. METHODOLOGY

This section presents the threat models, describes the attacks and defenses considered, defines the evaluation metrics, and explains the evaluation setup.

A. Threat Model

We define the threat model following the methodology of Biggio et al. [19], specifying the actors, their respective goals, knowledge, capabilities, and assumptions.

1) *Adversary*: An adversary is an *insider* (cf. Figure 1) who controls one (or more) clients and deviates from the protocol to embed a backdoor while remaining undetected. Therefore, the adversary has two objectives: (i) achieve a high attack success rate (ASR), causing inputs containing a specific trigger pattern to be classified into a target class; and (ii) preserve model utility, i.e., maintain high accuracy on the main task.

Adversary's knowledge: the adversary has full access to their local training data and client-side model (including its architecture). They do not have access to the raw data of benign clients or to the server's aggregation procedure. Knowledge of the server-side model architecture is not required, as the attacks considered are agnostic to it.

Adversary's capabilities: the adversary can, without restriction, manipulate their local training process, including poisoning and/or relabeling data samples, and modifying the optimization objective to increase the influence of poisoned samples. Furthermore, when they have access to multiple malicious clients, they can coordinate their actions to amplify the attack's effectiveness.

2) *Defender*: We consider multiple defender scenarios, differing by (i) which system component is responsible for protection and (ii) which entities are assumed to be trusted. Specifically, we consider the following cases:

- The client-side aggregation server (CAS) and the server-side aggregation server (Main Server) are operated by different entities. Then, two scenarios arise: (i) the Main Server seeks to protect the server-side model from potentially malicious clients and therefore deploys defenses to filter suspicious updates; (ii) the clients aim to protect their portion of the model from other malicious clients and request the deployment of defenses at the CAS.
- Both servers are controlled by the same entity or cooperate to defend against malicious clients, in which case defenses are deployed on both sides. This setting is further extended to two scenarios: (i) defenses are applied from the beginning of training; (ii) defenses are activated after a certain round r , modeling a delayed-defense scenario, for instance, when the learning process is already ongoing when a defense mechanism becomes available or is activated.

Defender's knowledge: in all scenarios, the defending entity (i.e., the server) does not know which clients are malicious, the exact trigger pattern, or the specific attack strategy employed by an adversary (cf. Section III-B). The defender suspects that a certain number of clients may send malicious updates; however, the majority of clients are assumed to be benign. *Restriction*: the defender has no access to any client's raw local data.

Defender's goals: the defender aims to (i) prevent the embedding of a backdoor in the model under its control, and

(ii) preserve the main task performance.

Defender’s capabilities: the defender can employ custom aggregation rules, apply filtering mechanisms to exclude anomalous updates from aggregation for specific rounds, or completely ban suspicious participants (cf. Section III-C).

B. Backdoor Attack Strategies

We consider the following attack strategies to evaluate their ability to embed a backdoor in the SFL setting and assess the extent to which defenses can mitigate them.

BadNets [13] is a foundational backdoor attack based on static data poisoning, where an adversary injects a fixed trigger into a subset of local training samples and assigns them a target label. The model learns to associate the trigger with the target class while preserving performance on clean data.

Distributed Backdoor Attack (DBA) [14] is proposed to be more difficult to detect, by decomposing a global trigger pattern into m smaller components distributed across malicious clients. Each client embeds its component, keeping individual updates closer to benign ones, while the backdoor is activated when all components appear together at inference time.

Neurotoxin [15] enhances backdoor persistence in FL by mitigating the catastrophic forgetting phenomenon [20], [21], where benign training gradually erases the backdoor. Neurotoxin exploits the sparsity of stochastic gradients by identifying high-magnitude coordinates from prior rounds, assuming they are frequently updated by honest clients. The attack then constrains poisoning updates to low-magnitude coordinates, embedding the backdoor in rarely modified parameters.

Model Replacement [22] is an attack strategy that amplifies malicious client’s influence during aggregation by scaling poisoned updates to counteract averaging effects, effectively replacing the global model in a single round. To evade detection, the scaled update may be constrained to remain statistically similar to benign updates.

C. Backdoor Defense Strategies

In this work, we consider aggregation- and anomaly-detection-based defenses that have been shown to be effective against several attacks [7]. Because both global client-side and server-side models are updated through aggregation, federated learning-inspired defenses are a natural fit for both components of the SFL architecture under study. We use *Federated Averaging (FedAvg)*, defined as a weighted mean of client updates, as an undefended baseline for comparison. The evaluated defenses are described below.

Multi-Krum (MKrum) [23] is a distance-based, robust aggregation that computes Euclidean distances between client updates to assess their similarity. Under the assumption that benign updates are more similar to each other, it selects those most consistent with the majority by minimizing their distances to neighboring updates and aggregates this subset.

MMetric [24] is a robust FL defense that uses multiple distance-based metrics (e.g., Euclidean, Manhattan, and cosine) combined via adaptive dynamic weighting. Each client update is assessed based on its deviation from other updates

across these metrics, thereby mitigating diverse backdoor attacks. The global model is then aggregated only from the updates classified as benign.

Alignment Inspection (AlignIns) [25] filters client updates based on gradient direction consistency. For each parameter, it verifies whether the sign of a client’s update deviates from the majority sign across other clients and removes such updates. This mechanism targets attacks that keep update magnitudes within the benign range but introduce directional inconsistencies in the update space.

FoolsGold [26] is designed to mitigate attacks executed by multiple coordinated malicious clients by identifying unusually similar updates. It assumes the presence of at least one honest client, and sufficient diversity among benign updates (e.g., in non-IID settings). FoolsGold calculates pairwise cosine similarities between client updates and adaptively down-weights those that are consistently similar to others.

FedDLAD [27] is designed in four-stages. First, a Connectivity-Based Outlier Factor (COF) examines the average distance of each update to its neighbors to detect outliers. Second, an IQR module applies inter-quartile range statistics to identify and adjust anomalous updates by reversing their direction. The updates flagged by these stages form an *untrusted group*. Since benign clients may also be incorrectly flagged as anomalous, in the third step, FedDLAD incorporates a so-called *pardoning* mechanism that re-evaluates these updates. Finally, those updates that closely resemble the benign reference (computed from a trusted group) get included back into the aggregation process.

While FL defenses typically rely on robust aggregation or anomaly detection, SL defenses primarily focus on anomaly detection since traditional split learning lacks an aggregation step [16], [18]. For example, Rieger et al. [18] detect anomalous updates by comparing their magnitude and rotational distance to those of recent model states. However, adapting such a defense to the parallel SFL architecture considered in this work would not be trivial, as updates from different clients are aggregated rather than processed sequentially. Therefore, we do not evaluate split learning-specific defenses and leave their adaptation to sequential SFL settings for future work.

D. Metrics

For the evaluation of the selected attacks and defenses, we use three metrics:

- The Attack Success Rate (ASR) measures how effective a backdoor attack is; it is calculated as the ratio of the number of trigger inputs classified as the designated target label to the total number of triggers.
- The Main task Accuracy (Acc) measures how well the model performs on benign data, computed as the proportion of correctly classified samples. Given that the datasets studied are class-balanced, accuracy is an appropriate metric for assessing utility preservation.
- The Robustness Accuracy (RA) [25] measures the proportion of poisoned inputs classified as their original (ground-truth) class instead of the trigger class. RA is

related to, but not the inverse of, the ASR, as some poisoned inputs may be misclassified into other classes.

In the literature (e.g., [5]), attack effectiveness is sometimes assessed by the drop in main task accuracy, reflecting an *availability* attack that degrades model utility [28]. However, when embedding a backdoor with the goal of an *integrity* attack, the objective is to induce specific misbehavior on trigger inputs while preserving normal performance on benign data. Accordingly, we evaluate attacks using the ASR, and a reduction in main task accuracy is treated as a utility drop and is undesirable, since it can reveal the presence of a backdoor.

E. Experiment Setup

Following the setup in [10], we train a ResNet18 model split after the first of four residual blocks for 200 communication rounds on MNIST [29] and CIFAR-10 [30]. Furthermore, we train a GoogLeNet [31] (Inception v1) model on the GTSRB [32] dataset, which contains 43 classes compared to the 10 classes in MNIST and CIFAR-10. The model is split after the Inception 3 block. The results from the MNIST experiments serve as a baseline and are not reported in detail in this paper; however, the trends follow those observed on the other two datasets. We consider a vanilla SFL setting in which the server receives labels to compute the loss. The system consists of 20 clients, four (20%) of which are malicious, each poisoning 50% of its local training data, following [5], [25]. Following prior backdoor studies, we use class seven as the target label for MNIST and CIFAR-10, and class 12 (Priority Road) for GTSRB as a semantically meaningful target. For DBA, malicious clients collaboratively construct a distributed trigger pattern; otherwise, all malicious clients use the same trigger pattern.

For each dataset, we evaluate the three backdoor attacks BadNets, DBA, and Neurotoxin, detailed in Section III-B, both with and without Model Replacement. We then assess defense performance under each attack. According to our threat model, we first evaluate (i) defenses applied only at the Client Aggregation Server (CAS), which aggregates client-side models, and (ii) defenses applied only at the Main Server (MS), which aggregates server-side models. Based on these results, we further evaluate (iii) the joint deployment of defenses on both CAS and MS. Finally, for the most effective defense configurations from scenarios (i)–(iii), we study (iv) a delayed deployment, activating defenses at rounds 50, 100, and 200 to assess their ability to mitigate an already embedded backdoor. When defenses are first applied at round 200, the total training duration is extended to 500 communication rounds. All experiments are run with three seeds; results reflect their mean and standard deviation.

IV. EVALUATION

This section presents the evaluation results, beginning with clean and attacked models without defenses. The effectiveness of defenses deployed by different SFL entities is then evaluated under the threat model in Section III-A, followed by an

TABLE I: Baseline attacks without defenses. All values are reported in %. *MR*: *Model Replacement*, *ASR*: *Attack Success Rate*, *Acc*: *Accuracy*.

Data	MR	BadNets		DBA		Neurotoxin	
		ASR	Acc	ASR	Acc	ASR	Acc
CIFAR-10	no	93.5 $\pm$ 4.3	59.9 $\pm$ 0.6	16.8 $\pm$ 2.3	58.6 $\pm$ 0.6	94.1 $\pm$ 3.8	60.1 $\pm$ 0.8
	yes	93.5 $\pm$ 4.3	59.9 $\pm$ 0.6	16.8 $\pm$ 2.3	58.6 $\pm$ 0.6	94.1 $\pm$ 3.8	60.1 $\pm$ 0.8
GTSRB	no	83.0 $\pm$ 15.0	79.6 $\pm$ 1.1	46.8 $\pm$ 5.1	79.5 $\pm$ 0.5	99.2 $\pm$ 0.3	80.7 $\pm$ 1.4
	yes	83.0 $\pm$ 15.0	79.6 $\pm$ 1.1	46.8 $\pm$ 5.1	79.5 $\pm$ 0.5	99.2 $\pm$ 0.3	80.7 $\pm$ 1.4

evaluation of delayed defense activation and a discussion of the key findings.

A. Baselines

The ResNet18 and GoogLeNet models are trained in the SFL setting for 200 rounds without backdoor attacks. The resulting accuracies of 61.73% (CIFAR-10) and 82.28% (GTSRB) serve as clean baselines for evaluating utility degradation under defenses. While the CIFAR-10 accuracy is lower than in FL (e.g., 77.8% [27]), it is consistent with prior SFL literature [10]. Extending training to 500 rounds increases clean accuracy above 70%, while relative defense performance and ranking remain stable after approximately 150 rounds. The lower clean accuracy at 200 rounds does not affect the observed security behavior of the defenses, and we therefore use 200 rounds for the main experiments.

Table I reports the results of the evaluated attacks, with and without Model Replacement. On CIFAR-10, BadNets and Neurotoxin achieve ASRs of 93.5% and 94.1%, respectively, while DBA performs significantly worse at 16.8%. On GTSRB, BadNets and Neurotoxin reach ASRs of 83.0% and 99.2%, while DBA improves to 46.8%, likely due to the consistent background of centered traffic-sign images that facilitate learning the distributed trigger despite the larger number of classes. On MNIST, DBA achieves 100% ASR, which can be attributed to the dataset’s simplicity and consistent feature representations. Model Replacement has negligible effect on both ASR and clean accuracy.

B. Defense Evaluation

a) Defenses at the Client Aggregation Server: From Table II, we see that applying defenses at the Client Aggregation Server does not mitigate backdoor attacks, as the observed metric values remain largely unchanged across all attack scenarios on CIFAR-10, with the same trend on MNIST. On GTSRB, only minor improvements are observed with MMetric for BadNets, FedDLAD for DBA, and AlignIns and FoolsGold for Neurotoxin, with the best case still leaving BadNets at 67.9% ASR. In all DBA settings, using defenses even increases ASR. Model Replacement has little impact: on CIFAR-10, ASR changes remain below 0.3%, on GTSRB, it often slightly reduces ASR, suggesting that the amplified malicious updates become more distinguishable to the defenses.

b) Defenses at the Main Server: Most defenses applied at the Main Server are effective on CIFAR-10, as shown in Table III. Except for MMetric, all methods reduce ASR to

TABLE II: Evaluation of the defenses applied only on Client Aggregation Server. Best results for each attack are highlighted in bold, while the worst results (based on ASR) are highlighted in red. All values are reported in %. *MR: Model Replacement, ASR: Attack Success Rate, Acc: Accuracy.*

Data	Defense		MR	BadNets		DBA		Neurotoxin	
	CAS	MS		ASR	Acc	ASR	Acc	ASR	Acc
CIFAR-10	MKrum	-	no	93.3 $\pm$ 4.3	59.9 $\pm$ 0.6	16.6 $\pm$ 2.0	58.8 $\pm$ 0.6	94.1 $\pm$ 3.8	60.0 $\pm$ 0.8
		-	yes	93.4 $\pm$ 4.3	59.9 $\pm$ 0.7	16.7 $\pm$ 2.2	58.7 $\pm$ 0.6	94.1 $\pm$ 3.8	60.1 $\pm$ 0.6
	FoolsGold	-	no	93.2 $\pm$ 4.5	59.9 $\pm$ 0.7	16.8 $\pm$ 2.1	58.7 $\pm$ 0.6	93.9 $\pm$ 3.9	60.0 $\pm$ 0.7
		-	yes	93.2 $\pm$ 4.5	59.9 $\pm$ 0.7	16.9 $\pm$ 2.2	58.5 $\pm$ 0.6	94.0 $\pm$ 3.8	59.9 $\pm$ 0.7
	MMetric	-	no	93.4 $\pm$ 4.2	60.0 $\pm$ 0.6	16.7 $\pm$ 2.3	58.7 $\pm$ 0.7	94.1 $\pm$ 3.8	60.1 $\pm$ 0.6
		-	yes	93.5 $\pm$ 4.2	60.0 $\pm$ 0.6	16.7 $\pm$ 2.2	58.6 $\pm$ 0.6	94.1 $\pm$ 3.8	60.1 $\pm$ 0.7
GTSRB	AlignIns	-	no	93.3 $\pm$ 4.4	60.0 $\pm$ 0.7	16.7 $\pm$ 1.9	58.8 $\pm$ 0.7	94.0 $\pm$ 3.9	60.1 $\pm$ 0.7
		-	yes	93.2 $\pm$ 4.4	59.9 $\pm$ 0.6	16.6 $\pm$ 2.1	58.7 $\pm$ 0.6	94.1 $\pm$ 3.7	60.0 $\pm$ 0.7
	FedDLAD	-	no	93.3 $\pm$ 4.4	60.0 $\pm$ 0.7	16.5 $\pm$ 2.0	58.8 $\pm$ 0.7	94.1 $\pm$ 3.8	60.1 $\pm$ 0.7
		-	yes	93.2 $\pm$ 4.3	59.9 $\pm$ 0.6	16.7 $\pm$ 2.3	58.6 $\pm$ 0.6	94.0 $\pm$ 3.9	60.2 $\pm$ 0.6
	MKrum	-	no	76.5 $\pm$ 9.8	78.3 $\pm$ 2.3	67.6 $\pm$ 20.1	78.5 $\pm$ 3.1	94.3 $\pm$ 3.7	73.0 $\pm$ 4.5
		-	yes	79.4 $\pm$ 11.6	73.8 $\pm$ 6.4	69.6 $\pm$ 16.7	77.9 $\pm$ 2.6	93.2 $\pm$ 3.1	71.1 $\pm$ 1.8
	FoolsGold	-	no	92.3 $\pm$ 5.1	74.5 $\pm$ 7.7	76.2 $\pm$ 25.0	77.9 $\pm$ 2.9	87.1 $\pm$ 8.9	79.5 $\pm$ 1.7
		-	yes	92.3 $\pm$ 4.3	74.5 $\pm$ 7.7	77.2 $\pm$ 25.7	74.3 $\pm$ 4.5	80.0 $\pm$ 4.3	79.2 $\pm$ 1.5
	MMetric	-	no	67.9 $\pm$ 13.2	78.8 $\pm$ 1.8	65.7 $\pm$ 2.5	76.9 $\pm$ 7.6	84.2 $\pm$ 12.1	82.3 $\pm$ 1.1
		-	yes	67.9 $\pm$ 13.2	78.8 $\pm$ 1.8	54.3 $\pm$ 17.2	76.3 $\pm$ 7.0	91.1 $\pm$ 8.6	81.5 $\pm$ 0.3
	AlignIns	-	no	92.6 $\pm$ 1.8	77.5 $\pm$ 4.1	56.8 $\pm$ 19.8	77.5 $\pm$ 5.3	81.0 $\pm$ 26.0	78.4 $\pm$ 3.5
		-	yes	92.6 $\pm$ 1.8	77.5 $\pm$ 4.1	52.4 $\pm$ 24.5	77.0 $\pm$ 5.0	86.7 $\pm$ 8.0	78.6 $\pm$ 3.7
	FedDLAD	-	no	93.7 $\pm$ 6.4	77.7 $\pm$ 3.8	51.9 $\pm$ 8.8	75.2 $\pm$ 5.4	96.2 $\pm$ 2.2	79.0 $\pm$ 2.5
		-	yes	93.7 $\pm$ 6.4	77.7 $\pm$ 3.8	51.3 $\pm$ 8.3	78.3 $\pm$ 2.4	94.6 $\pm$ 3.6	80.6 $\pm$ 2.0

TABLE III: Evaluation of defenses applied only on Main Server. Best results for each attack are highlighted in bold, while the worst results (based on ASR) are highlighted in red. All values are reported in %. *MR: Model Replacement, ASR: Attack Success Rate, Acc: Accuracy.*

Data	Defense		MR	BadNets		DBA		Neurotoxin	
	CAS	MS		ASR	Acc	ASR	Acc	ASR	Acc
CIFAR-10	MKrum	-	no	4.7 $\pm$ 0.9	61.0 $\pm$ 1.0	4.5 $\pm$ 1.4	61.2 $\pm$ 0.7	4.8 $\pm$ 1.0	60.9 $\pm$ 0.8
		-	yes	4.9 $\pm$ 1.1	61.2 $\pm$ 0.7	4.6 $\pm$ 1.4	61.2 $\pm$ 0.7	4.7 $\pm$ 1.0	61.0 $\pm$ 0.8
	FoolsGold	-	no	5.6 $\pm$ 1.1	58.7 $\pm$ 0.8	6.9 $\pm$ 2.4	58.5 $\pm$ 0.6	5.4 $\pm$ 1.0	58.6 $\pm$ 0.9
		-	yes	5.6 $\pm$ 1.1	58.8 $\pm$ 0.8	6.9 $\pm$ 2.4	58.5 $\pm$ 0.6	5.4 $\pm$ 1.0	58.6 $\pm$ 0.9
	MMetric	-	no	86.8 $\pm$ 8.6	59.7 $\pm$ 1.6	16.6 $\pm$ 3.4	58.8 $\pm$ 0.2	83.8 $\pm$ 11.3	59.8 $\pm$ 1.4
		-	yes	87.9 $\pm$ 7.8	60.1 $\pm$ 1.5	16.3 $\pm$ 4.0	58.9 $\pm$ 0.2	85.2 $\pm$ 10.2	59.8 $\pm$ 1.4
GTSRB	AlignIns	-	no	4.8 $\pm$ 1.1	61.2 $\pm$ 0.5	4.9 $\pm$ 1.3	61.0 $\pm$ 0.5	4.8 $\pm$ 1.1	61.1 $\pm$ 0.4
		-	yes	4.8 $\pm$ 1.1	61.2 $\pm$ 0.4	4.8 $\pm$ 1.2	61.1 $\pm$ 0.3	4.8 $\pm$ 1.1	61.1 $\pm$ 0.5
	FedDLAD	-	no	5.0 $\pm$ 1.1	61.1 $\pm$ 0.4	4.7 $\pm$ 1.0	61.0 $\pm$ 0.6	5.0 $\pm$ 0.9	61.1 $\pm$ 0.4
		-	yes	5.0 $\pm$ 1.1	61.1 $\pm$ 0.5	4.8 $\pm$ 1.0	61.0 $\pm$ 0.6	5.0 $\pm$ 0.9	61.1 $\pm$ 0.4
	MKrum	-	no	0.2 $\pm$ 0.1	81.4 $\pm$ 1.5	0.3 $\pm$ 0.1	80.6 $\pm$ 1.9	0.3 $\pm$ 0.1	80.1 $\pm$ 2.2
		-	yes	0.2 $\pm$ 0.1	81.4 $\pm$ 1.5	0.3 $\pm$ 0.1	80.6 $\pm$ 1.9	0.3 $\pm$ 0.1	80.1 $\pm$ 2.2
	FoolsGold	-	no	83.3 $\pm$ 15.8	75.4 $\pm$ 4.2	63.8 $\pm$ 5.7	77.3 $\pm$ 4.4	94.0 $\pm$ 4.1	78.5 $\pm$ 3.3
		-	yes	83.3 $\pm$ 15.8	75.4 $\pm$ 4.2	63.8 $\pm$ 5.7	77.3 $\pm$ 4.4	94.0 $\pm$ 4.1	78.5 $\pm$ 3.3
	MMetric	-	no	49.1 $\pm$ 9.8	76.2 $\pm$ 8.6	33.4 $\pm$ 16.7	77.8 $\pm$ 4.0	78.7 $\pm$ 17.0	78.5 $\pm$ 0.9
		-	yes	49.1 $\pm$ 9.8	76.2 $\pm$ 8.6	33.4 $\pm$ 16.7	77.8 $\pm$ 4.0	78.7 $\pm$ 17.0	78.5 $\pm$ 0.9
	AlignIns	-	no	0.3 $\pm$ 0.2	77.0 $\pm$ 4.0	0.4 $\pm$ 0.2	78.9 $\pm$ 3.4	0.2 $\pm$ 0.3	77.3 $\pm$ 5.7
		-	yes	0.3 $\pm$ 0.2	77.0 $\pm$ 4.0	0.4 $\pm$ 0.2	78.9 $\pm$ 3.4	0.2 $\pm$ 0.3	77.3 $\pm$ 5.7
	FedDLAD	-	no	0.3 $\pm$ 0.2	79.7 $\pm$ 1.9	0.3 $\pm$ 0.2	78.5 $\pm$ 4.6	0.2 $\pm$ 0.2	79.8 $\pm$ 3.0
		-	yes	0.3 $\pm$ 0.2	79.7 $\pm$ 1.9	0.3 $\pm$ 0.2	78.5 $\pm$ 4.6	0.2 $\pm$ 0.2	79.8 $\pm$ 3.0

4.5–6.9%, indicating that the effectiveness of defenses in SFL strongly depends on the entity protecting against malicious updates. AlignIns and MKrum perform best, achieving the lowest ASR (4.7% for BadNets and Neurotoxin and 4.5% for DBA) while maintaining utility. In contrast, MMetric provides only marginal reductions (a minimum ASR of 83.8%) and no improvement for DBA, likely due to conflicting signals from combining multiple distance metrics when separating benign and malicious updates. Main task accuracy remains close to the clean baseline, fluctuating between 58.5% and 61.2%.

TABLE IV: Evaluation of defenses applied on both servers. Best results for each attack are highlighted in bold, while the worst results (based on ASR) are highlighted in red. All values are reported in %. *MR: Model Replacement, ASR: Attack Success Rate, Acc: Accuracy.*

Data	Defense		MR	BadNets		DBA		Neurotoxin	
	CAS	MS		ASR	Acc	ASR	Acc	ASR	Acc
CIFAR-10	MKrum	MKrum	-	4.7 $\pm$ 0.7	61.0 $\pm$ 1.0	5.1 $\pm$ 1.5	61.1 $\pm$ 0.8	4.7 $\pm$ 1.4	61.0 $\pm$ 0.8
			-	4.5 $\pm$ 1.4	61.2 $\pm$ 0.8	5.2 $\pm$ 1.5	61.0 $\pm$ 0.8	4.7 $\pm$ 1.4	61.2 $\pm$ 1.0
	AlignIns	AlignIns	-	4.8 $\pm$ 1.1	61.0 $\pm$ 0.3	4.7 $\pm$ 1.3	61.1 $\pm$ 0.3	4.8 $\pm$ 1.2	61.1 $\pm$ 0.4
			-	4.7 $\pm$ 1.0	61.2 $\pm$ 0.4	4.7 $\pm$ 1.4	61.1 $\pm$ 0.2	4.8 $\pm$ 1.1	61.1 $\pm$ 0.3
	FedDLAD	FedDLAD	-	4.8 $\pm$ 1.2	61.1 $\pm$ 0.3	4.7 $\pm$ 1.4	61.1 $\pm$ 0.3	4.7 $\pm$ 1.0	61.1 $\pm$ 0.3
			-	5.0 $\pm$ 1.0	61.1 $\pm$ 0.6	4.7 $\pm$ 1.4	60.9 $\pm$ 0.6	4.9 $\pm$ 1.1	61.1 $\pm$ 0.5
GTSRB	AlignIns	AlignIns	-	5.1 $\pm$ 1.1	61.1 $\pm$ 0.4	4.8 $\pm$ 1.4	60.9 $\pm$ 0.6	5.0 $\pm$ 1.0	61.0 $\pm$ 0.5
			-	5.2 $\pm$ 1.0	61.1 $\pm$ 0.5	4.8 $\pm$ 1.5	60.9 $\pm$ 0.4	4.8 $\pm$ 1.4	61.1 $\pm$ 0.4
	MKrum	MKrum	-	0.2 $\pm$ 0.1	76.9 $\pm$ 3.2	0.2 $\pm$ 0.2	76.4 $\pm$ 6.5	0.3 $\pm$ 0.1	80.5 $\pm$ 2.0
			-	0.4 $\pm$ 0.1	76.9 $\pm$ 7.6	0.2 $\pm$ 0.0	78.6 $\pm$ 5.8	0.3 $\pm$ 0.1	79.2 $\pm$ 3.4
	AlignIns	AlignIns	-	0.3 $\pm$ 0.1	70.4 $\pm$ 18.6	0.4 $\pm$ 0.3	78.2 $\pm$ 5.7	0.3 $\pm$ 0.2	79.6 $\pm$ 5.5
			-	0.4 $\pm$ 0.2	75.9 $\pm$ 4.3	0.2 $\pm$ 0.2	74.2 $\pm$ 7.3	0.3 $\pm$ 0.2	73.7 $\pm$ 7.8
	FedDLAD	FedDLAD	-	0.3 $\pm$ 0.2	78.6 $\pm$ 1.8	0.3 $\pm$ 0.2	78.8 $\pm$ 3.9	0.4 $\pm$ 0.2	77.9 $\pm$ 2.3
			-	0.2 $\pm$ 0.0	74.8 $\pm$ 5.7	0.5 $\pm$ 0.3	76.2 $\pm$ 6.0	0.3 $\pm$ 0.1	75.9 $\pm$ 6.2
	MKrum	MKrum	-	0.3 $\pm$ 0.0	78.5 $\pm$ 4.5	0.2 $\pm$ 0.1	81.5 $\pm$ 1.0	0.3 $\pm$ 0.1	81.6 $\pm$ 1.2
			-	0.1 $\pm$ 0.1	78.6 $\pm$ 4.0	0.4 $\pm$ 0.2	78.9 $\pm$ 5.8	0.2 $\pm$ 0.0	79.4 $\pm$ 5.5
	AlignIns	AlignIns	-	55.1 $\pm$ 49.6	77.5 $\pm$ 3.5	0.2 $\pm$ 0.1	79.3 $\pm$ 3.8	0.2 $\pm$ 0.1	79.4 $\pm$ 4.5
			-	55.1 $\pm$ 49.6	77.5 $\pm$ 3.5	0.2 $\pm$ 0.1	79.3 $\pm$ 3.8	0.2 $\pm$ 0.1	79.4 $\pm$ 4.5

Results on GoogLeNet with GTSRB follow a similar trend, with MKrum, AlignIns, and FedDLAD consistently reducing ASR below 1%. AlignIns and FedDLAD achieve the lowest ASR of 0.2% for Neurotoxin, while MKrum performs best across other settings. In contrast, FoolsGold is least effective, reducing ASR only to 83.3% (BadNets), 63.8% (DBA), and 94.0% (Neurotoxin). Although MMetric performs better, its ASR remains substantially higher than that of the other defenses, with its lowest value being 33.4% for DBA. Moreover, both FoolsGold and MMetric exhibit high standard deviations of up to 17.0%, indicating unstable performance.

Model Replacement has a negligible impact in all settings. The largest ASR increase is 1.4% (ResNet18/CIFAR-10, Neurotoxin with MMetric). This is likely because Model Replacement amplifies malicious client-side updates, while backdoor persistence is primarily driven by the server-side model, as shown in Table III. Likewise, the reduction in main task accuracy does not exceed 1%, indicating no substantial change in the model’s overall behavior. On GoogLeNet with GTSRB, Model Replacement has no measurable effect. Consequently, we do not report the results of the Model Replacement attack technique, FoolsGold, and MMetric defenses in the subsequent experiments due to their consistently limited impact.

c) *Defenses applied on both servers:* Defenses at the CAS alone provide little additional protection, although they introduce minor changes in ASR and accuracy when combined with defenses at the Main Server. Performance is primarily determined by the defense deployed at the MS. On CIFAR-10, the best results (Table IV) are achieved with MKrum and AlignIns at the MS. On GTSRB, FedDLAD exhibits inconsistent performance, allowing the attack to reach up to 55.1% ASR. This instability can be attributed to the COF-based outlier detection of FedDLAD. When a cluster contains a sufficient number of malicious clients, the nearest neighbors

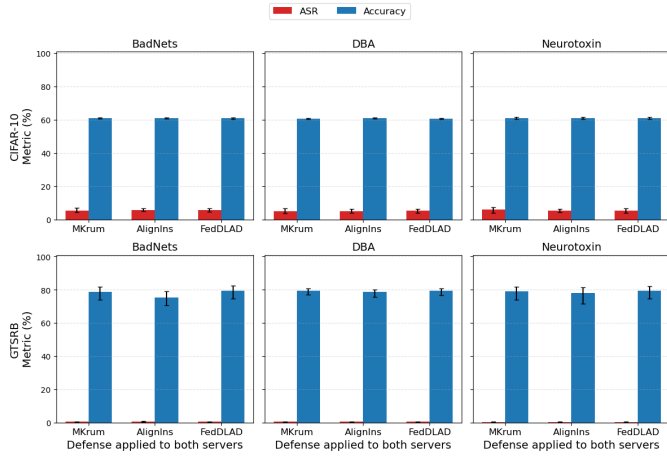


Fig. 2: Evaluation of applied defenses on both servers from the 50th round. The values are taken from the 200th round.

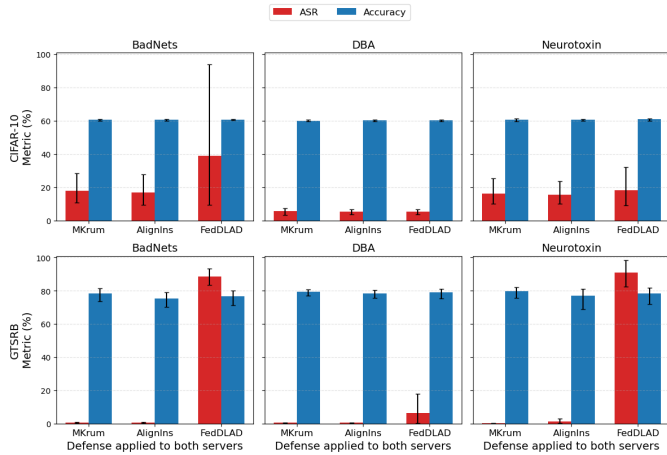


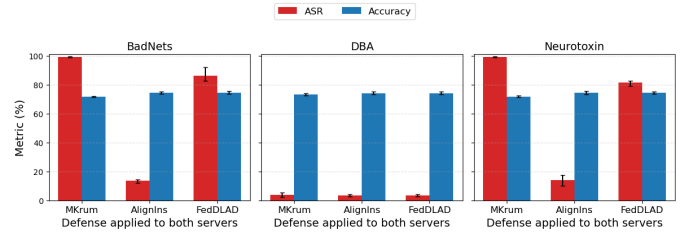
Fig. 3: Evaluation of applied defenses on both servers from the 100th round. The values are taken from the 200th round.

of a malicious update are predominantly other malicious updates, which causes the outlier detection to fail.

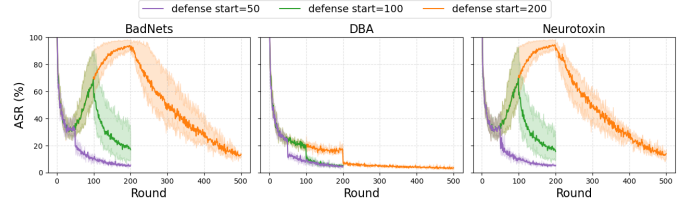
C. Defenses Timing

In the following, we examine the effect of applying defenses not from the start of training. As shown in Figure 2, when defenses are activated at round 50, they successfully mitigate attacks on CIFAR-10 and GTSRB. In contrast, on MNIST, they mostly fail for BadNets and Neurotoxin. After 20 rounds, these attacks reach an ASR of approximately 95%, at which point none of the evaluated defenses can mitigate them. DBA exhibits different behavior: after 50 rounds, it reaches a lower ASR of approximately 93%, allowing defenses such as MKrum, AlignIns, and FedDLAD to partially mitigate the attack, reducing its ASR to near 42% by round 200.

Next, we embed the backdoor for 100 rounds before activating the defenses; the results are shown in Figure 3. The DBA attack embeds a backdoor in ResNet18 trained on MNIST after 80 rounds with 100% ASR. After that, no



(a) Defenses are activated after 200th rounds.



(b) Defenses are activated after 50th, 100th, and 200th rounds. AlignIns is applied on both servers.

Fig. 4: Evaluation of applied defenses for CIFAR-10.

applied defense reduces the ASR. On CIFAR-10, MKrum and AlignIns remain effective, while FedDLAD again shows inconsistent performance against BadNets, just as with the results when the defenses were activated from the first round. On GTSRB, FedDLAD also fails for BadNets and Neurotoxin, reaching more than 80% ASR in both cases.

Given the above observations, we discuss the CIFAR-10 results in detail, as similar trends are observed on MNIST and GTSRB. Figure 4a compares MKrum, AlignIns, and FedDLAD activated in the 200th round. Among them, only AlignIns substantially reduces ASR, from 96% to 11%. Figure 4b further shows a detailed evaluation of AlignIns applied at both servers at the 50th, 100th, and 200th rounds. Earlier activation of the defense consistently results in lower ASR, indicating that the defense is most effective before the backdoor fully converges. However, this behavior depends on the dataset and model. For example, on GTSRB, AlignIns reduces the ASR from 99.71% to 0.20% even when activated after the attack has converged, suggesting that defense effectiveness is also influenced by the underlying architecture and task complexity.

D. Discussion

Our evaluation reveals key differences between federated and split-federated learning in the context of backdoor defenses. In particular, defense effectiveness depends heavily on the entity responsible for deploying it. This highlights a structural security asymmetry in SFL: defenses applied only at the Client Aggregation Server are insufficient, because backdoor-related information is propagated through intermediate representations and becomes encoded in the server-side model. This is in line with prior FL literature [2], which shows that once such associations are established, they become significantly harder to remove. This is further supported by the negligible impact of Model Replacement, which amplifies malicious updates on the client side. Consequently, the Main

Server becomes the critical point for defense deployment; further, FL defenses do not directly transfer to SFL, since the assumption of a single aggregation point no longer holds.

When defenses are applied at the Main Server from the start of training, they effectively prevent backdoor learning. However, once the backdoor has converged (e.g., approaching 100% ASR) and malicious clients remain active, their effectiveness generally decreases, as malicious updates become increasingly difficult to distinguish from benign ones. This is particularly evident for distance- and direction-based defenses, whose underlying assumptions weaken as training converges. Nevertheless, this behavior also depends on the model architecture and dataset. While early activation is beneficial on CIFAR-10, AlignIns remains highly effective on GTSRB even after convergence, whereas on the simpler MNIST task, it struggles to separate poisoned from benign updates.

V. CONCLUSIONS AND FUTURE WORK

In this work, we defined multiple realistic threat models in a split-federated learning setting and evaluated the suitability of state-of-the-art federated learning backdoor defenses for each. Our results show that defense effectiveness in SFL is highly sensitive to both the assumed threat model and the entity responsible for deploying the defense. Across all evaluated methods, MMetric and FoolsGold consistently failed to protect the models against backdoor attacks, with MMetric failing in all and FoolsGold in most scenarios of the considered threat model. In contrast, AlignIns proved the most effective, particularly when deployed on the Main Server.

Furthermore, post-injection detection and mitigation, relevant in settings such as pre-trained models or intermittently participating clients, remain challenging, with most FL defenses failing once the backdoor is fully embedded (near 100% ASR), especially on simpler tasks such as MNIST.

As future work, we plan to investigate alternative SFL architectures, including single server-side model designs and integration with SL-specific defenses.

REFERENCES

- [1] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How To Backdoor Federated Learning," in *Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, PMLR, 2020.
- [2] Z. Li, J. Lan, Z. Yan, and E. Gelenbe, "Backdoor attacks and defense mechanisms in federated learning: A survey," *Information Fusion*, vol. 123, 2025.
- [3] V. Turina, Z. Zhang, F. Esposito, and I. Matta, "Federated or Split? A Performance and Privacy Analysis of Hybrid Split and Federated Learning Architectures," in *IEEE Int. Conf. on Cloud Computing (CLOUD)*, 2021.
- [4] P. Joshi, C. Thapa, S. Camtepe, M. Hasanuzzaman, T. Scully, and H. Afli, "Performance and Information Leakage in Splitfed Learning and Multi-Head Split Learning in Healthcare Data and Beyond," *Methods and Protocols*, vol. 5, no. 4, 2022.
- [5] A.-T. Zahir-Ismael and R. Shukla, "Analyzing the vulnerabilities in Split Federated Learning: assessing the robustness against data poisoning attacks," *Scientific Reports*, vol. 15, no. 1, 2025.
- [6] M. Aleksic, Y. Klysa, and R. Mayer, "Backdoors in split-federated learning," in *Proceedings of the 9th International Students Workshop*, (Świeradów Zdrój, Poland), Wrocław University of Science and Technology, 2026.
- [7] T. Dao, D. T. Nguyen, K. D. Doan, and K.-S. Wong, "BackFed: An Efficient & Standardized Benchmark Suite for Backdoor Attacks in Federated Learning," 2025.
- [8] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," 2018. arXiv:1812.00564 [cs].
- [9] D.-J. Han, H. I. Bhatti, J. Lee, and J. Moon, "Accelerating Federated Learning with Split Learning on Locally Generated Losses," *ICML workshop on federated learning for user privacy and data confidentiality*, 2021.
- [10] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "SplitFed: When Federated Learning Meets Split Learning," *AAAI Conf. on Artificial Intelligence*, vol. 36, no. 8, 2022.
- [11] W. Wu, M. Li, K. Qu, C. Zhou, Xuemin, Shen, W. Zhuang, X. Li, and W. Shi, "Split Learning over Wireless Networks: Parallel Design and Resource Management," 2022.
- [12] T. D. Nguyen, T. Nguyen, P. L. Nguyen, H. H. Pham, K. Doan, and K.-S. Wong, "Backdoor Attacks and Defenses in Federated Learning: Survey, Challenges and Future Research Directions," 2023.
- [13] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating Backdooring Attacks on Deep Neural Networks," *IEEE Access*, 2019.
- [14] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "DBA: Distributed Backdoor Attacks against Federated Learning," 2019.
- [15] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable Backdoors in Federated Learning," in *Int. Conf. on Machine Learning (ICML)*, PMLR, 2022.
- [16] F. Yu, L. Wang, B. Zeng, K. Zhao, Z. Pang, and T. Wu, "How to backdoor split learning," *Neural Networks*, vol. 168, 2023.
- [17] F. Yu, B. Zeng, K. Zhao, Z. Pang, and L. Wang, "Chronic Poisoning: Backdoor Attack against Split Learning," *AAAI Conf. on Artificial Intelligence*, vol. 38, no. 15, 2024.
- [18] P. Rieger, A. Pegoraro, K. Kumari, T. Abera, J. Knauer, and A.-R. Sadeghi, "SafeSplit: A Novel Defense Against Client-Side Backdoor Attacks in Split Learning," in *Network and Distributed System Security Symposium (NDSS)*, (San Diego, CA, USA), Internet Society, 2025.
- [19] B. Biggio and F. Roli, "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning," in *ACM SIGSAC Conf. on Computer and Communications Security (CCS)*, (Toronto Canada), ACM, 2018.
- [20] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the Tails: Yes, You Really Can Backdoor Federated Learning," in *Neural Information Processing Systems (NeurIPS)*, 2020.
- [21] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in Cognitive Sciences*, vol. 3, no. 4, 1999.
- [22] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can You Really Backdoor Federated Learning?," 2019.
- [23] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," in *Neural Information Processing Systems (NeurIPS)*, 2017.
- [24] S. Huang, Y. Li, C. Chen, L. Shi, and Y. Gao, "Multi-metrics adaptively identifies backdoors in Federated learning," in *IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, (Paris, France), IEEE, 2023.
- [25] J. Xu, Z. Zhang, and R. Hu, "Detecting Backdoor Attacks in Federated Learning via Direction Alignment Inspection," 2025.
- [26] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "The Limitations of Federated Learning in Sybil Settings," 2020.
- [27] B. Ding, P. Yang, and S.-J. Huang, "FedDLAD: A Federated Learning Dual-Layer Anomaly Detection Framework for Enhancing Resilience Against Backdoor Attacks," vol. 6, 2025.
- [28] M. Nieves, K. Dempsey, and V. Y. Pillitteri, "An introduction to information security," Tech. Rep. NIST SP 800-12r1, National Institute of Standards and Technology, Gaithersburg, MD, 2017.
- [29] Y. LeCun, C. Cortes, and C. J. Burges, "MNIST handwritten digit database," 2010.
- [30] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," 2009.
- [31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [32] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, 2012.