






# An Alternative Approach to Federated Learning for Model Security and Data Privacy

William Briguglio<sup>1</sup><sup>a</sup>, Waleed A. Yousef<sup>1,2</sup><sup>b</sup>, Issa Traoré<sup>1</sup><sup>c</sup>, Mohammad Mamun<sup>3</sup><sup>d</sup>  
and Sherif Saad<sup>4</sup><sup>e</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada

<sup>2</sup>Department of CS, HCILab, Helwan University, Cairo, Egypt

<sup>3</sup>National Research Council of Canada, Fredericton, NB, Canada

<sup>4</sup>Department of Computer Science, University of Windsor, Windsor, ON, Canada

Keywords: Data Poisoning, Federated Learning, Model Poisoning, non-IID.

Abstract: Federated learning (FL) enables machine learning on data held across multiple clients without exchanging private data. However, exchanging information for model training can compromise data privacy. Further, participants may be untrustworthy and can attempt to sabotage model performance. Also, data that is not independently and identically distributed (IID) impede the convergence of FL techniques. We present a general framework for federated learning via aggregating multivariate estimated densities (FLAMED). FLAMED aggregates density estimations of clients' data, from which it simulates training datasets to perform centralized learning, bypassing problems arising from non-IID data and contributing to addressing privacy and security concerns. FLAMED does not require a copy of the global model to be distributed to each participant during training, meaning the aggregating server can retain sole proprietorship of the global model without the use of resource-intensive homomorphic encryption. We compared its performance to standard FL approaches using synthetic and real datasets and evaluated its resilience to model poisoning attacks. Our results indicate that FLAMED effectively handles non-IID data in many settings while also being more secure.


## 1 INTRODUCTION


Federated learning (FL) is used to train a machine learning (ML) model from data held by multiple owners, without compromising the privacy of each owner's data. In the standard FL approach, each data owner or client trains a local ML model starting from a shared initial global model. The result of local training is sent in the form of weight or gradient updates to an aggregating server, which then combines the local models to obtain the new global model. This process repeats for several rounds, each starting from the previous round's global model, until convergence is reached. However, (Zhu and Han, 2020) has shown that it is possible to leak training samples from the gradient updates alone. To protect against this, other


approaches in the literature rely on techniques such as homomorphic encryption (HE) and secure multi-party computation (SMPC). But, as discussed in Section 4, these solutions work against securing FL against attacks that target the performance of the global model, posing a trade-off between model security and data privacy.


In FL, during local training, each batch is sampled only from the data available at a given client. However, data in FL settings is non-independent and identically distributed (non-IID), meaning clients have different dataset distributions. This causes local models to be biased away from the global optimum, hampering or preventing convergence. Many approaches have been proposed to overcome this hurdle but they often ignore privacy and security considerations.


We propose an alternative general FL framework using density estimation to simultaneously address non-IID data (see Section 3.1), and privacy and security (see Section 4) concerns. Clients model their local data distributions and share this with the server,

<sup>a</sup> <https://orcid.org/0000-0002-2357-3966>

<sup>b</sup> <https://orcid.org/0000-0001-9669-7241>

<sup>c</sup> <https://orcid.org/0000-0003-2987-8047>

<sup>d</sup> <https://orcid.org/0000-0002-4045-8687>

<sup>e</sup> <https://orcid.org/0000-0002-5506-5261>

allowing the aggregating server to simulate centralized global training. This is a general framework and the methods used for modeling distributions and creating the global model can be decided upon by the practitioner.

The present work makes the following contributions. (1) FLAMED is a general framework for simulated centralized learning that serves as a conceptual basis for alternative FL methods and allows non-IIDness, privacy, and model security to be addressed simultaneously. (2) FLAMED enables the aggregating server to obtain a global model not known to any other participants. Restricting knowledge of the global model to a single participant secures the model’s intellectual property and guards against a malicious participant using shared model weights to attack training data privacy. (3) We evaluated our approach against baseline and state-of-the-art FL approaches on a variety of synthetic datasets and a real-world healthcare dataset from a federated setting with 132 participants. The latter, with 3,069 features, demonstrates FLAMED’s potential with high-dimensionality data. (4) We performed a security analysis of the proposed framework and evaluated its resilience against backdoor attacks as defined in (Bagdasaryan et al., 2020) using the real dataset. (5) We present a technique specific to FLAMED for detecting model backdoor attacks via data poisoning.

The next section provides a brief review of the FL literature. Section 3 presents the general formulation of our approach and a discussion of its strengths relative to standard FL. We also present the proof of concept implementation used for FLAMED in the current paper. In Section 5, we present all the configurations of the FL and model backdoor experiments.<sup>1</sup> Section 6 presents an analysis of the results and highlights the strengths and weaknesses of the proposed approach. Section 7 summarizes our findings and discusses future work.

## 2 RELATED WORK

FedAvg (McMahan et al., 2017) is the basic FL algorithm in which an initial global model is distributed from an aggregation server to clients participating in the FL scheme. Each client trains the model on their local data. The trained local models are sent back to the server, which obtains the updated global model as a weighted average of the local models. This process repeats until convergence.

<sup>1</sup>This research was enabled in part by the Digital Research Alliance of Canada.

The privacy of training data and the integrity of the global model is a principal concern in FL. Although data are never exchanged between clients, (Zhu and Han, 2020) showed it is possible to reproduce training samples using the gradient updates alone, a problem known as gradient leakage. In (Bagdasaryan et al., 2020), the authors demonstrate that one or multiple clients can collaborate to cause misclassifications for specific feature values, without significantly impacting the global model’s overall performance.

Reference (Bonawitz et al., 2017) employed SMPC to provide a private vector summation framework for FL weight aggregation. Their framework is also resilient to clients dropping out of the FL network, ensuring results are still correct even if clients leave part way through the secure summation procedure.

Several approaches improve convergence with non-IID data. FedProx was introduced in (Li et al., 2020), and it improved on FedAvg by penalizing large updates with the addition of a “proximal term” to the clients’ local objective function. The proximal term prevents local updates from pulling the global model away from the global optimum. Researchers in (Karimireddy et al., 2020) introduced SCAFFOLD to account for “client drift,” when a client’s local optimum is not aligned with the average local optimum across all clients, by approximating the ideal unbiased local update, which is the average gradient of the local model across all clients’ data. FedDC, proposed in (Gao et al., 2022), improved on SCAFFOLD by adding a loss term that allows clients to learn their client drift and correct it before submitting updates.

Existing FL approaches have difficulty simultaneously addressing non-IIDness while maintaining clients’ data privacy and global model integrity. In FLAMED, we provide an FL framework that address all these challenges at once.

## 3 METHOD

In this section, we introduce the general FLAMED framework and discuss its benefits before detailing the specific FLAMED implementation used in the experiments presented in this paper. Table 1 provides the notation used throughout.

### 3.1 FLAMED: The General Framework

Non-IID data pose a significant challenge when applying FL in real-world scenarios by causing local models to be biased towards the local solution conditioned only on the locally held data, slowing con-

Table 1: Notation frequently used in this paper.

Symbol	Meaning
$K$	Number of clients
$C_i$	The $i^{\text{th}}$ client
$P$	Global data distribution
$P_i$	Client $i$ 's data distribution
$\hat{P}$	Estimation of $P$
$\mathcal{M}$	Global Model
$X$	Data from all clients
$X_i$	Data belonging to $C_i$
$X'$	Low-dimensional transformation of $X$
$\tilde{X}$	Data simulated from the estimated distribution of $X$
$U_{:r}$	$r$ -dimensional transformation from SVD or FedSVD
$n$	Total number of samples
$n_i$	Number of samples at $C_i$
$m$	Number of features
$c$	Number of classes
$\rho$	Experimental parameter specifying the ratio of $n_i$ to $m$
$\alpha = \beta$	Experimental parameter specifying the level of non-IIDness

vergence towards the global solution. For any observation  $x$  in the IID setting, we have  $x \sim P$ , where  $P$  is the global data distribution, while in the non-IID setting, we have  $x \sim P_i$  for each client  $C_i$ , where  $P_i$  may not equal  $P_j \forall i \neq j$  and  $i, j \in [1, K]$ . To bypass non-IIDness, we estimate the maximum likelihood estimator (MLE)  $\hat{P}^{MLE}$  of  $P$ . In FLAMED, each  $P_i$  is modeled using a density estimation technique to obtain  $\hat{P}_i$ ; then one of two approaches is possible. (1) Each client simulates a dataset  $\tilde{X}_i \sim \hat{P}_i$ , which the server aggregates into a global dataset. (2) The server aggregates the estimated  $\hat{P}_i$ , or its summary statistics, from each client, which the server uses to simulate a global dataset. The first approach is the default in FLAMED and what we adopt in the present article's proof of concept implementation, which can be seen as just one of many possible implementations of the first general approach. In both approaches, a global model is constructed at the server from the global dataset, which then follows a mixture distribution

$$\hat{P} = \sum_i \alpha_i \hat{P}_i, \quad \sum_i \alpha_i = 1, \quad (1)$$

where the weight of the convex combination,  $\alpha_i$ , controls the importance of each client, and the distributions  $\hat{P}_i$ ,  $i = 1, \dots, K$  are the empirical distributions  $\hat{P}_i^{MLE}$  of the data simulated at the client side (approach 1) or the estimated distributions themselves (approach 2). Figure 1 illustrates the general FLAMED framework.

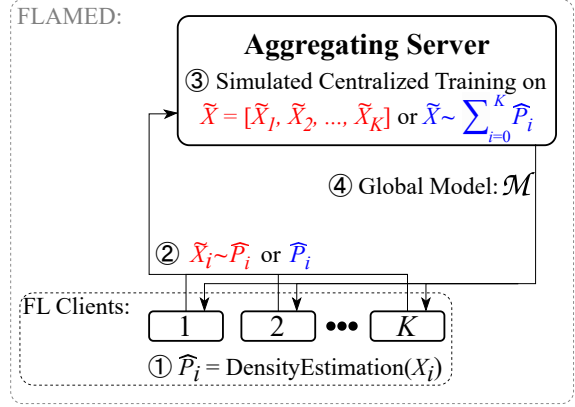


Figure 1: The general FLAMED framework with red indicating approach 1 and blue indicating approach 2.

**Theorem 3.1.** *FLAMED bypasses non-IIDness by approximating the global MLE  $\hat{P}^{MLE}$  of the global data distribution  $P$ .*

*Proof:* It is obvious that  $\hat{P}_i^{MLE} \neq \hat{P}_j^{MLE}$ , where  $\hat{P}_i^{MLE}$  is the MLE, called the empirical nonparametric distribution, that simply puts a mass of  $\frac{1}{n_i}$  on each observation, and where  $n_i$  is the number of observations at  $C_i$ . In both approaches, a global model is constructed at the server from the mixture distribution  $\hat{P}$  defined in Eq. (1). Therefore, the aggregating server obtains  $\hat{P}$ , an estimation of  $\hat{P}^{MLE}$ . ■

In this way, FLAMED simulates a centralized learning task, thus bypassing non-IIDness resulting from varying data distributions  $P_i \neq P_j$ . Further, in contrast to standard FL, there is only a single round of communication. This means each client can contribute once they are available, and the aggregating server can wait to train the global model only when all clients have contributed, without holding up other participants. Therefore, in addition to bypassing non-IIDness resulting from differing  $P_i$ , FLAMED also addresses non-IIDness resulting from client selection bias due to nonuniform client availability.

The general framework of FLAMED proceeds as follows (A specific implementation is given in Section 3.3):

- (Optional for low-dimensionality datasets) The clients use a privacy-preserving distributed dimensionality reduction technique to enable tractable density estimation, distribution modeling, etc., depending on the method used to derive the global model.
- Clients perform the statistical analysis sufficient for the learning task on their (optionally) transformed data, and the resulting information ( $\tilde{X}_i$  or  $\hat{P}_i$  for approach 1 or 2, respectively) is sent to the

aggregating server.

3. The server simulates centralized training to construct the global model  $\mathcal{M}$ .

The default for FLAMED is to follow approach 1 because clients can always simulate the data themselves and send only the simulated data to the server. Here, we are outlining a general approach, so, practitioners themselves must ensure clients do not expose sensitive information in step 2. For example, summary statistics or a simple scaled histogram may be shared with the server but, kernel density estimation (KDE), which uses observations within its estimated probability density function (PDF), would leak observations if its PDF was sent to the server.

### 3.2 FLAMED: Practical Benefits

FLAMED has several practical benefits that distinguish it from traditional FL. Many FL use cases are resource-constrained but standard FL methods (e.g. all methods cited in Section 2) require multiple rounds of communication and computation on participants' devices. FLAMED offers an alternative that requires only a single round of communication and computation, with all model training taking place at the aggregating server. The complexities shown in Table 2 emphasize this point. FLAMED trades multiple rounds of local model training at the clients for a single round of density estimation at the clients and global model training at the server. FLAMED's space requirements are comparable at the client but larger at the server. Communication is also reduced to a single round. Altogether, FLAMED asks for less space, computation, and availability from the clients in exchange for a heavier burden on the server. In general, FLAMED's redistribution of computational burdens may make it more appropriate for settings with low-resource clients, e.g. internet of things applications, provided the central server is able to handle the extra workload. Further, any contribution can easily be individually excluded from global model training by imposing a zero weight on that client (Eq. (1)), making detecting malicious contributions (see Section 6.3) and performing contribution evaluation easier. Also, if a new client joins the FL network, the global model can be updated without repeating the entire FL process with all participants. FLAMED also allows for the streamlined design of the global model because grid search can be performed with little coordination or communication overhead. In contrast, standard FL methods require the entire procedure to be repeated for each choice of hyperparameters or model architecture.

### 3.3 FLAMED: Specific Implementation

**Input:** Data  $X = [X_1, \dots, X_K]$

**Output:** Global model  $\mathcal{M}$

```

begin
  FedSVD( $X_1, \dots, X_K$ ); \\\Clients get  $U_r$ 
  for  $i \in [1, K]$  do
    \\\At Client  $i$ 
     $X'_i \leftarrow U_r X_i$ 
     $\hat{P}_i \leftarrow \text{KDE}(X'_i)$ 
     $\tilde{X}'_i \sim \hat{P}_i$ 
    SendToServer( $\tilde{X}'_i$ )
  end
  \\\At Server
   $\tilde{X}' \leftarrow [\tilde{X}'_1, \dots, \tilde{X}'_K]$ 
   $\mathcal{M} \leftarrow \text{GlobalModel.Train}(\tilde{X}')$ 
  SendToClients( $\mathcal{M}$ )
end

```

Algorithm 1: FLAMED Using Simulation.

In our experiments, to reduce data dimensionality and make simulation tractable, we consider singular value decomposition (SVD) (Halko et al., 2011) for dimensionality reduction. SVD decomposes matrix  $X \in \mathbb{R}^{m \times n}$  as  $X = U\Sigma V^T$ , where  $\Sigma \in \mathbb{R}^{m \times n}$  is diagonal,  $U \in \mathbb{R}^{m \times m}$ , and  $V \in \mathbb{R}^{n \times n}$ . Using the  $r$  columns of  $U$  corresponding to the  $r$  largest singular values in  $\Sigma$ , denoted  $U_{:,r}$ , a low-dimensional transformation  $X' \in \mathbb{R}^{r \times n}$  is obtained with  $X' = U_{:,r} X$ .

However, this approach cannot be directly applied to our problem because each client would obtain a different  $U_i$ , each biased toward their local dataset. Thus, we use FedSVD (Chai et al., 2021), which, with the help of a trusted masking server, can compute  $U_{:,r}$  of the combined data matrix  $X = [X_1, \dots, X_K]$ , which is composed of all  $K$  clients' data matrices  $X_i$ , without compromising the privacy of any of the clients' data. Once each client receives  $U_{:,r}$ , they compute the common  $r$ -dimensional transformation of their data.

The specific algorithm and implementation of FLAMED used in this paper is depicted in Algorithm 1. For simulation, we use KDE with the Gaussian kernel. KDE maps points in the feature space to estimates of the probability density using a weighted sum of kernel distances from said point to each observation in the training set. Logistic regression and a feed forward neural network (NN) were used for the global model. The time, space, and communication complexities for the general FLAMED framework and the specific implementation in our experiments along with the comparison approaches are shown in Table 2.



Table 2: Time and space complexities for all methods.  $D$ ,  $A$ , and  $G$  are placeholders for density estimation, aggregation, and building the global model, respectively.  $n$ ,  $n_i$ , and  $n_l$  are the total number of observations across all clients, the number of observations at client  $i$ , and the max number of observations at any given client, respectively. We assume a feed-forward NN for the global model with  $l$  layers no larger than  $m$  trained for  $e$  epochs and  $R$  rounds of FL.

Method	Time	Space		Communication	
		Client	Server	Size	Rounds
FLAMED(General)	$O(D) + O(A) + O(G)$	$O(D)$	$\max(O(A), O(G))$	$O(K(\text{size}(I_i) + \text{size}(\mathcal{M})))$	1
FLAMED(KDE)	$O(n_l^2) + O(lm^2ne)$	$O(n_i)$	$O(lm^2)$	$O(nm + Klm^2)$	1
FedAvg/Prox/DC	$O(lm^2n_l eR)$	$O(lm^2)$	$O(lm^2)$	$O(RKlm^2)$	$R$

## 4 SECURITY

### 4.1 Threat Model

In our threat model, we assume any subset of the  $K$  participants, including the aggregating server, could be malicious and use any means necessary to attempt to learn something about a particular data sample  $x$  belonging to a benign participant. Although in some settings, it may be inadmissible to allow global properties of data distributions to be leaked, it is not obvious that standard FL approaches can prevent this (Wang et al., 2019; Zhu and Han, 2020). Most data privacy approaches, ours included, focus on the privacy of any particular sample. The European Union’s General Data Protection Regulation (GDPR), a major incentive for the development of FL algorithms in the first place, only applies to “personal data,” which is data relating to an identified or identifiable individual (see (Voigt and Von dem Bussche, 2017, Sec. 2.1.2)). Therefore, we allow knowledge of empirical probability densities of private data to be learned by adversaries.

Given this threat model, because FLAMED is a general method, its security would have to be proved for each individual implementation. Specifically, if approach 1 is followed, it must be shown that sharing  $\tilde{X}$  conforms to a particular privacy requirement, which is application-dependent. If approach 2 is followed, sharing  $\hat{P}_i$  must be shown to conform to a particular privacy requirement (e.g. sharing an approximation of KDE’s PDF is proven secure in (Wagner et al., 2023)). There are many varieties of privacy requirements. Differential privacy requires that any synthetic datasets generated from neighbouring private datasets (i.e. datasets that differ by one element) have a near equal probability of occurring (Ding et al., 2011). This can be accomplished by using methods like PrivBayes for generating synthetic datasets (Zhang et al., 2017). We leave investigation of this approach to future work. Below, we prove the security of the specific FLAMED implementation defined in 3.3 us-

ing a weaker privacy requirement. Specifically, we require that FLAMED releases no certain information about a particular sample.

**Theorem 4.1.** *FLAMED is secure with respect to our threat model. That is, FLAMED leaks no certain information about a particular  $x_i \in X = [X_1, \dots, X_K]$  other than  $\hat{P}$ .*

*Proof:* In approach 1, where each client simulates a dataset  $\tilde{X}_i \sim \hat{P}_i$ , the server only receives  $\tilde{X}_i$  from each client, which cannot be used to accurately reconstruct any particular  $x_i$  with certainty. The server could only use  $\tilde{X} = [\tilde{X}_1, \dots, \tilde{X}_K]$  to construct an empirical PDF that approximates the estimated density  $\hat{P}$  from Eq. (1). Even if the server colludes with all but one participant  $j$ , obtaining the private data belonging to clients  $i \neq j$ , the server will obtain at best a closer approximation to  $\hat{P}$ , which does not leak any certain information about a particular  $x_i$  at the non-colluding participant. Similarly, in approach 2, the server only receives density estimation information  $\hat{P}_i$  from each client. As discussed in Section 3.1, the practitioner should ensure this information does not leak private data (e.g. summary statistics or histograms can be safely shared). Here, we prove the security of our general approach and assume the practitioner will ensure sharing  $\hat{P}_i$  is secure in their specific implementation. By this assumption, in approach 2, the server also cannot accurately reconstruct any particular  $x_i$  after receiving  $\hat{P}_i$ . Therefore, in both approaches, so long as care is taken in specifying  $\hat{P}_i$  when using approach 2, the server does not learn any certain information about a particular  $x_i$ . Conversely, the clients may optionally obtain, at most, the global model trained on  $\tilde{X}$ . Even if they were able to approximately reconstruct much of its training data with model inversion attacks, they would have less certain information than the server, and, at best, would only be able to reconstruct  $\hat{P}$ . Thus, FLAMED leaks no certain information about a particular  $x_i$  other than its estimated probability density. ■

## 4.2 Security Advantages

As discussed in Section 2, training samples can be reconstructed from weight updates exchanged in the standard FL approach (Zhu and Han, 2020). Thus, privacy can be compromised if no measures are taken to hide weight updates from the server. In FLAMED, no gradients are exchanged, so this type of attack is not feasible. Further, (Bagdasaryan et al., 2020) showed that directly manipulating gradient updates (model poisoning) is more effective than manipulating training data (data poisoning). This makes FLAMED inherently more robust against model backdoor attacks, as demonstrated in Section 6.3. FLAMED addresses non-IIDness, gradient leakage, and gradient poisoning simultaneously. Other approaches require observing raw gradients or client state information to correct biases or detect poisoned gradients. This makes securing such methods against gradient leakage attacks more difficult. Conversely, addressing gradient leakage by obscuring raw gradient information makes correcting bias and detecting data poisoning more difficult. FLAMED presents no such trade-off by allowing careful analysis of all client contributions while not exchanging gradient information and maintaining privacy.

FLAMED allows the aggregating server to obtain a global model that is not known to other participants. To share this property with FLAMED, most standard FL methods would require intensive redesign with costly HE or other privacy-preserving methods. It is easy to see the use cases for such an FL method. For example, consider an FL scenario with untrusted participants, such as cell phone users. The participants may want to isolate the trained model at the aggregating server to maintain the aggregating server's sole proprietorship of the global model or to strengthen privacy guarantees because sharing the global model with participants may allow them to perform model inversion attacks, exposing participants' private data to one another.

## 5 EXPERIMENTS

### 5.1 Comparison Approaches

We compare FLAMED with FedAvg, FedProx, and FedDC. All approaches come with a strong theoretical foundation. FedAvg, has over 20,000 citations and is included as a baseline FL approach. A survey of the FL literature showed FedProx reported the largest accuracy increase over FedAvg (Liu et al., 2020, Table 11). FedDC is more recent and outperformed Fed

dcAvg, FedProx, and other approaches from the literature. We thus include FedProx and FedDC to represent the state-of-the-art.

### 5.2 Performance Comparison: Setup and Configurations

**Synthetic Datasets.** To evaluate FLAMED against the comparison FL techniques under a variety of scenarios, we used synthetic datasets generated following the approach used by the authors of FedProx. In their approach, parameters  $\alpha = \beta$  control how non-IID different client datasets are. Where we depart from FedProx is in the configurations of the synthetic datasets used. The mean number of observations held across all clients is determined in proportion to the number of features  $m$  as  $\rho m$ , where  $\rho$  is an experimental hyperparameter. In the following,  $n_i$  is the number of observations held at client  $C_i$ . The distribution of the number of observations across all clients is either uniform (i.e.  $n_i = \rho m \forall i$ ) denoted  $\mathcal{U}$ , or a modified log-normal distribution  $\mathcal{L} = n_i^* + \frac{\rho m}{2}$  where  $n_i^* \sim \text{lognormal}(\mu, 2)$  and  $\mu$  and 2 are the mean and standard deviation, respectively, of the underlying normal distribution;  $\mu$  is chosen such that  $E[n_i^*] = \frac{\rho m}{2}$ , and thus, the mean number of observations at each client  $C_i$  is  $E[\mathcal{L}] = E[n_i^*] + \frac{\rho m}{2} = \rho m$ .

In our initial experiments we applied FLAMED and the comparison methods to 1,536 synthetic distributed dataset configurations defined by the cross product:  $K \in \{2, 4, 8, 16\} \times c \in \{2, 4, 8, 16\} \times m \in \{8, 32, 128, 512\} \times \rho \in \{5, 10, 20\} \times \alpha = \beta \in \{\text{IID}, 0, 0.5, 1\} \times \mathcal{D} \in \{\mathcal{U}, \mathcal{L}\}$  where  $c$  is the number of classes and  $\alpha = \beta = \text{IID}$  denotes IID data across all clients. After our results from these initial experiments (discussed in Section 6.1), we explored how FLAMED handled higher levels of non-IIDness and repeated our initial experiments, but with  $\alpha = \beta \in \{1.5, 2\}$ , adding 768 configurations. We also repeated our initial experiments but with a higher number of clients  $K \in \{32, 64, 128\}$  and  $\rho \in \{5, 10\}$ . Configurations with  $K = 128 \wedge m = 512 \wedge \rho = 10$  were excluded due to time constraints. This added another 736 configurations, for 3,040 configurations in total.

**Real Dataset.** To evaluate FLAMED in a real-world federated setting, we used the eICU Collaborative Research Database (Pollard et al., 2018). This dataset contains real-world medical data from over 200,000 ICU admissions to more than 200 medical centres across the United States. Unlike other commonly used datasets, the eICU dataset represents a real-world federated setting instead of a contrived one obtained by separating a centralized dataset. We use the data contained in the drug infusions table. Each row

in our feature matrix  $X$  corresponds to a patient, while each column corresponds to a drug. If a patient  $i$  receives any dose of a certain drug  $j$  across any of their ICU admissions, then  $X_{ij}$  is set to 1. Otherwise, it is set to 0. A patient is assigned label 0 if their discharge status in the patient table is “alive”, and 1 otherwise. After removing any hospital with less than 10 observations, we are left with 3,069 features and 72,959 patients held across 132 hospitals. In addition to using all 132 clients, we test 22 different configurations defined in the set  $K \in \{2, 4, 8, 16, 32, 64, 128\} \times S \in \{\text{smallest, middle, largest}\}$ . Here,  $K$  is the number of hospitals used and  $S$  denotes the strata of hospitals we select from. That is, if  $S = \text{middle}$ , then we select the  $K$  hospitals with the nearest to the median number of patients, while if  $S = \text{smallest}$  or  $S = \text{largest}$ , then we select the  $K$  hospitals with the least or most number of patients, respectively.

**Model Parameters.** For FedAvg, FedProx, and FedDC, we performed 200 rounds of standard FL to train a feed-forward NN. In all experiments, this was more than enough rounds for the global model to converge. Intermediate layers have  $\lfloor \frac{m}{2} \rfloor$  neurons, unless  $m = 2$ , in which case they have 2 neurons. Other parameters are determined through grid search using balanced accuracy for evaluation to account for class imbalances. For FLAMED, FedSVD is used to transform the dataset into  $r \in \{2, 4, 8\}$  dimensions. Grid search is used to determine the optimal KDE simulation parameters which minimize the log-likelihood score of a held-out local test set. After simulation and the training of a global model, the optimal values for  $r$  and the hyperparameters of the global model are determined using balanced accuracy on a validation set that consists of  $\sim 10\%$  of each local dataset. For the global model, logistic regression (LR) and NNs were compared.

### 5.3 Security Analysis: Setup and Configurations

We recreated the attacks in (Bagdasaryan et al., 2020) which used poisoning to cause the global model to only misclassify observations with certain feature values, called the backdoor, while not affecting the overall accuracy of the global model. We used the full eICU dataset after preprocessing as described in Section 5.2. For all methods, we varied the number of attacking clients, using the clients with the nearest to median number of observations. The backdoor was set when column 377 is 1 with target label 1. Only one observation in the benign data had this column set to 1, and it had the label 0. Thus, the poisoning objective was contradictory to the benign data but

should not have caused overall degradation in model performance. The poisoned training data consisted of the backdoor and some noise in the form of random columns set to 1 to help the model to generalize the learned backdoor.

For attacking FedAvg, FedProx, and FedDC, we followed the model poisoning approach presented in (Bagdasaryan et al., 2020). We also performed the backdoor attack via data poisoning as a baseline comparison with the data poisoning attack against FLAMED. At the server, we tested two different defences that were also presented in (Bagdasaryan et al., 2020), computing the cosine distance and the  $L_2$  distance between each client’s weight update and the global model. It is assumed that updates with higher  $L_2$  or cosine distances are anomalous and represent poisoning attempts. In practice, these defences cannot be deployed in conjunction with secure aggregation; however, we report their effectiveness here as a best case scenario. Further, also following (Bagdasaryan et al., 2020), in order to evade detection, the attacker modifies their loss function to include an “anomalous loss” term, weighted with  $1 - \alpha$ . This term penalizes weight updates with large  $L_2$  or cosine distances, depending on the defence deployed (see (Bagdasaryan et al., 2020, Eq. (4))). The strength of the attackers’ poisoned update and the weight of the anomalous loss term are controlled using the hyperparameters denoted  $\gamma$  and  $\alpha$  in the original paper. In our experiment,  $\gamma$  and  $\alpha$  were varied across  $\{50, 75, 90\}$  and  $\{0.4, 0.5, 0.7, 1\}$ , respectively. Attackers were selected in every round of federated training.

For FLAMED, as discussed in Section 4.2, we are confined to data poisoning because no gradients are exchanged. We injected poisoned training data before FedSVD dimensionality reduction and varied the number of poisoned training observations as a multiple  $p \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.8, 1, 2, 4, 8, 16, 32, 64\}$  of the amount of training data at the attacking client(s). We assume that the backdoor, being inserted by only a few clients and by definition not present in the original data, will be rare. Therefore, any observations containing the backdoor will be an outlier; so, we used anomaly detection to find poisoned observations. We tested six defense methods using two anomaly detection algorithms, local outlier factor (LOF) (Breunig et al., 2000) and isolation forest (IF) (Liu et al., 2008), on the simulated data, the centroid of each client’s simulated data, and the centroid of each class’s simulated observations at each client.

To evaluate the poisoning attacks, we used the backdoor success rate (BSR) which is the percent-

age of observations in the poisoned test data that fool the global model into predicting the target label. The poisoned test data contain the backdoor and some noise, which tests how well the backdoor generalizes. We also record the area under curve (AUC) of the defences by assigning positive labels to poisoned updates or data and using the  $L_2$ /cosine distance or LOF/IF outlier scores as the predictions. The attacker aims to insert an effective model backdoor with a high BSR while also going undetected, meaning the defence method scores a low AUC. Any other result is good for the defender because then the attack is either detected, ineffective, or both.

## 6 RESULTS

### 6.1 Synthetic Data

Table 3: Balanced accuracy for each method averaged over the initial synthetic dataset configurations including and excluding the entirely IID configurations.

Method	Balanced Accuracy	
	Configs. Excluding $\alpha = \beta = \text{IID}$	All Configs.
FedAvg	0.7962	0.8092
FedProx	0.8009	0.8125
FedDC	0.8023	0.7952
FLAMED	0.7741	0.7226

In real-world FL settings, entirely IID datasets are exceedingly rare. We also gain more from non-IID datasets because each client’s contribution holds different information about the global learning task. Therefore, we are more interested in the non-IID configurations and provide the average balanced accuracy scores both including and excluding the configurations with entirely IID data. The averaged balanced accuracy scores for each method across all dataset configurations in our initial experiments are shown in Table 3. FedAvg, FedProx, and FedDC were the overall winners, performing mostly at par, but FLAMED remained competitive in the non-IID experiments. This can already be considered a success because, in addition to FLAMED’s performance (which was within 3% balanced accuracy of the best-performing method across all non-IID settings), it offers the security advantages discussed in Section 4 and the practical benefits mentioned in Section 3.2.

It was not expected that any one approach would perform best across all scenarios, and as we will see, the results in Table 3 represent only a superficial glance at the true utility of each method. We break down our results with respect to configuration

parameters found to heavily affect the relative performance of the compared methods. The average balanced accuracy across all configurations with respect to the level of non-IIDness for our initial experiments and extended experiments with greater levels of non-IIDness are shown in Fig. 2. We can see that the introduction of even slight non-IIDness resulted in a very large improvement in the performance of FLAMED. This trend continued as non-IIDness increased, with diminishing returns, until FLAMED scored just 0.007 average balanced accuracy below the best-performing comparison approach, FedDC. The fact that the difference between the entirely IID configurations and configurations where  $\alpha = \beta = 0$  alone is so great illustrates the importance of considering the non-IID configurations separately. FLAMED performed worse on IID data, and because the comparison methods do not behave in the same way, we cannot consider this an artifact of the synthetic dataset generation. Rather, this may result from FedSVD failing to preserve classification-relevant information or interference from overlapping estimations across similar client distributions.

Increasing the number of clients also increases non-IIDness. Fig. 3 shows, for each method, the balanced accuracy with respect to different numbers of clients averaged across all configurations in our initial experiments and in our extended experiments with a greater number of clients. The plot shows that as the number of clients increased, the relative performance of FLAMED improved. When not considering entirely IID configurations, FLAMED performed better than FedAvg and FedProx, and was competitive with FedDC, if the number of clients  $K \geq 16$ .

A principal characteristic of any ML problem is the number of features being used for prediction. In our case, it is especially important because transformation to a low-dimensionality feature space is required for tractable simulation. Fig. 4 shows the average balanced accuracy for each method with respect to different dataset dimensionalities for the initial experiments, the experiments with a greater number of clients, and the experiments with high non-IIDness. From the figure, we can see that in the initial experiments, when the dimensionality was lowest, FLAMED performed best. Further, the results show that when non-IIDness is increased by increasing the number of clients or by increasing  $\alpha = \beta$ , FLAMED is performant, even at higher dimensionalities, where simulation is not tractable without first using dimensionality reduction. This outcome is corroborated by our results on the real dataset.

Overall, our results show there are several scenarios where FLAMED performs well. Specifically,



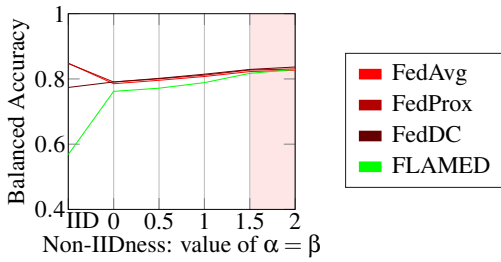


Figure 2: Avg. balanced accuracy for each method across initial (unshaded) and extended configurations with high non-IIDness (shaded) for different levels of non-IIDness.

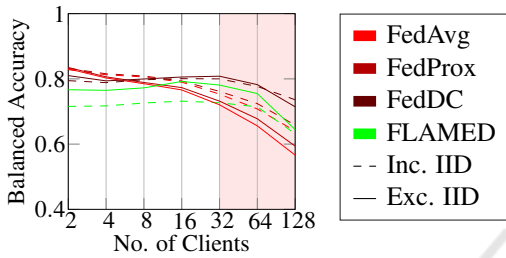


Figure 3: Avg. balanced accuracy versus the number of clients across the initial (unshaded) and extended configurations with a greater number of clients (shaded).

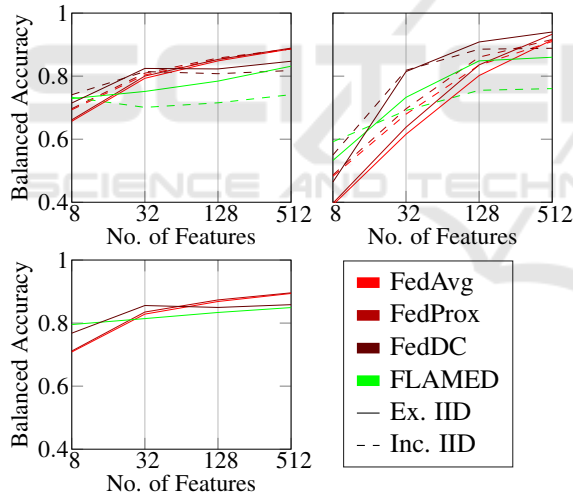


Figure 4: Avg. balanced accuracy versus the number of features for each method across all configurations in the initial experiments (top-left) the extra configurations with more clients (top-right) and higher non-IIDness (bottom).

when the number of features is low, so as to keep simulation tractable (of course this is dependent on the number of informative features, which determines SVD’s ability to successfully preserve all meaningful information) and where non-IIDness is high because of highly heterogenous client distributions or a large number of slightly heterogenous client distributions. It is important to note that FLAMED’s relative performance improves as the number of clients

increases because this amplifies the non-IIDness of the data, which impacts FLAMED’s performance less adversely compared to other methods. However, in scenarios where the data is IID, this advantage may diminish. Regardless of the data distribution, each client must possess sufficient data to achieve reliable local data density estimation.

## 6.2 Real Data

The results of our experiments on the eICU dataset are presented in Fig. 5. The plot shows the best balanced accuracy achieved by each method with varying numbers of clients. The test set is taken from each client that was used in training. Unsurprisingly, when clients with a smaller number of observations were used, the relative performance of FLAMED was lowest because many observations are needed to reliably train a simulator versus a classifier. When larger clients were used, the relative performance of all methods became much closer. When we had a high number of clients, FLAMED had good performance, beating FedAvg and FedProx when using more than 32 of the clients with the largest number of samples, and remaining competitive with FedDC. Notably, FLAMED performed best on the full eICU dataset with all but the three smallest clients. When using the full eICU dataset, these three excluded clients likely contributed poorly simulated data, so FLAMED performed second best.

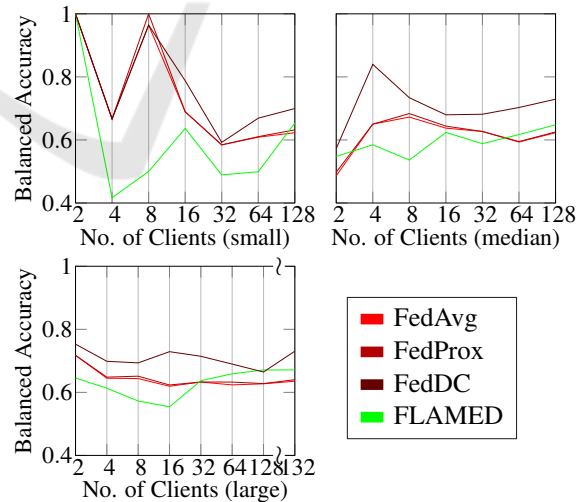


Figure 5: The balanced accuracy for each method evaluated on the eICU dataset with varying number of clients using the clients with the smallest (top-left), nearest to median (top-right), and largest (bottom) number of samples.

### 6.3 Security Analysis: Poisoning Comparison

Here, we present the results of our experiments comparing the practicality of model backdoor attacks against FedAvg, FedProx, FedDC, and FLAMED. In Fig. 6, we show the resulting AUC and BSR of the backdoor attacks via data and model poisoning against FedAvg, FedProx, and FedDC, as described in Section 5.3. Each point corresponds to different attack techniques and parameters. For the data poisoning attacks, in which case no evasion can be performed by the attacker, the maximum AUC from either the  $L_2$  or cosine defence is presented. For the model poisoning attacks, the AUC resulting from the specified evasion technique's corresponding defence method is used. If no evasion technique is used, the maximum AUC from either the  $L_2$  or cosine defence is presented. The results indicate there are many instances where the model poisoning attack is successful, with over 90% BSR, while going undetected by the defence methods used. However, the data poisoning attack was always detected with a high AUC, corroborating the results from (Bagdasaryan et al., 2020). Regardless of the FL method and the defence method used, there were attack configurations where the BSR was greater than 90% and the AUC lower than 60%, meaning the attacks were undetected but potent. The reader should also note that the AUCs presented here represent a best-case scenario because, in practice, client updates are not visible to the aggregating server due to privacy concerns.

It is important to note that in cases where the AUC is below 50, the defender cannot simply flip the prediction to achieve a better-than-random AUC. The AUC is so low primarily because the attacker's evasion techniques minimize the  $L_2$  norm or cosine distances, and therefore the outlier score, of their poisoned updates. Intuitively, any outlier score exceeding a given threshold should be considered anomalous, and the corresponding update ignored. However, this approach would also eliminate many benign

updates because the attackers minimize their outlier score, significantly compromising accuracy.

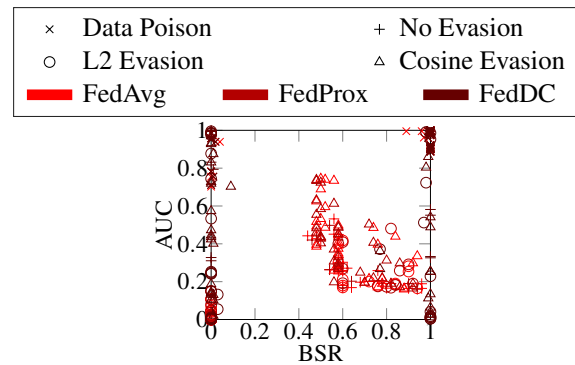


Figure 6: BSR obtained with the different poisoning attack configurations and the AUC of the corresponding defence method.

Fig. 7 shows the results of the backdoor attacks via data poisoning against FLAMED. If used on the simulated data was very effective, scoring a high average AUC regardless of the attack configuration. This demonstrates that FLAMED, unlike standard FL methods, enables effective attack detection without exposing gradients to the aggregating server. Also, many data poisoning attacks achieved poor results, but there is no reliable method to determine good attack parameter settings and risk an ineffective attack or being detected. Therefore, a data poisoning attack on FLAMED is impractical. In contrast, the authors in (Bagdasaryan et al., 2020) present methods for obtaining good model poisoning attack parameters with little prior knowledge of the FL network. However, as stated, model poisoning attacks cannot be performed against FLAMED because it doesn't exchange gradients.

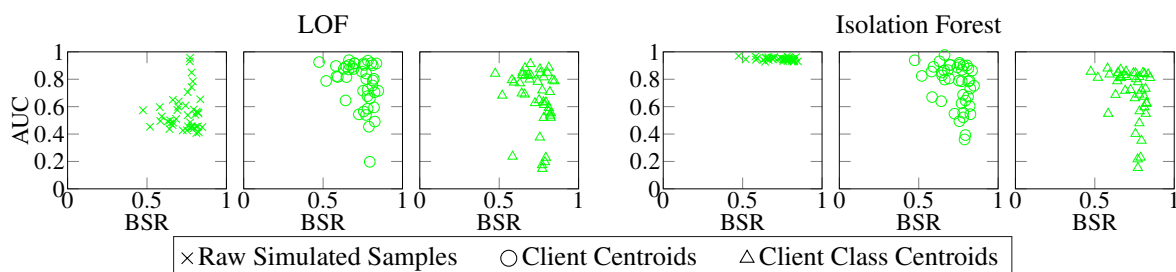


Figure 7: The BSR of data poisoning attacks against FLAMED versus the average AUC across all parameter settings for all defence methods. Repeated marker shapes correspond to different attack configurations.

## 7 CONCLUSION

In this work, we introduced the FLAMED framework and compared it to FedAvg, FedProx, and FedDC. FLAMED demonstrated strong performance in handling non-IID data and detecting attacks against model performance while resisting gradient-based privacy attacks. FedSVD effectively reduced the dimensionality of large datasets (3,069 features) for accurate simulation. While FLAMED’s performance was competitive, it represents an early step in FL with estimated densities, whereas comparison approaches like FedDC represent the culmination of eight years of research interest. Future research directions include developing FedSVD approaches that eliminate the need for a masking server and extending FLAMED to settings such as categorical features, online learning, and vertical FL.

## REFERENCES

- Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2020). How to backdoor federated learning. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR.
- Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., et al. (2017). Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS ’17*, page 1175–1191, New York, NY, USA. Association for Computing Machinery.
- Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.
- Chai, D., Wang, L., Fu, L., Zhang, J., Chen, K., and Yang, Q. (2021). Federated singular vector decomposition. *arXiv preprint arXiv:2105.08925*.
- Ding, B., Winslett, M., Han, J., and Li, Z. (2011). Differentially private data cubes: optimizing noise sources and consistency. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, SIGMOD ’11*, page 217–228, New York, NY, USA. Association for Computing Machinery.
- Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., and Xu, C.-Z. (2022). FedDC: Federated learning with non-iid data via local drift decoupling and correction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10102–10111.
- Halko, N., Martinsson, P. G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.
- Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. (2020). Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR.
- Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450.
- Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM ’08*, page 413–422, USA. IEEE Computer Society.
- Liu, Y., Zhang, L., Ge, N., and Li, G. (2020). A systematic literature review on federated learning: From a model quality perspective. *arXiv preprint arXiv:2012.01973*.
- McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.
- Pollard, T. J., Johnson, A. E. W., Raffa, J. D., Celi, L. A., Mark, R. G., and Badawi, O. (2018). The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1):180178.
- Voigt, P. and Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed.*, Cham: Springer International Publishing, 10(3152676):10–5555.
- Wagner, T., Naamad, Y., and Mishra, N. (2023). Fast private kernel density estimation via locality sensitive quantization. In *International Conference on Machine Learning*, pages 35339–35367. PMLR.
- Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., and Qi, H. (2019). Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, page 2512–2520. IEEE Press.
- Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. (2017). Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4).
- Zhu, L. and Han, S. (2020). *Deep Leakage from Gradients*, pages 17–31. Springer International Publishing, Cham.