# Investigating Inference Costs and Security in Ensemble-Based Private Collaborative Machine Learning

Alex Derhacobian
*Stanford University*

Maggie Gray
*Stanford University*

Sasha Ronaghi
*Stanford University*

## Abstract

Secure multiparty computation (SMPC) is a cryptographic method that allows multiple parties to jointly compute a function over private inputs. Recently, SMPC has been utilized to allow multiple data owners to jointly train machine learning (ML) models over secret-shared data. Many SMPC techniques for privacy-preserving ML have been shown to be vulnerable to data poisoning attacks, a common adversarial ML attack. One framework proposed by Chaudhari et. al. to defend SMPC ML systems against data poisoning attacks is SAFENET (6). While SAFENET is able to significantly reduce the likelihood of data poisoning attacks, its use of model ensembles makes inference tasks computationally expensive. Chaudhari et. al. have proposed a cost-efficient variation of SAFENET to reduce its inference costs by using transfer learning, which we refer to as SAFENETLITE. In this paper, we explore whether SAFENETLITE provides the same security guarantees as the original SAFENET framework. We show that reducing the inference costs of SAFENETLITE often results in weaker security against data poisoning attacks. Additionally, we show that, in our experiments, the inference costs of SAFENETLITE ensembles are generally a better predictor of the success rate of data poisoning attacks than the number of corrupted parties in the SMPC.

## 1 Introduction

Machine learning (ML) has shown tremendous success in a wide array of applications. ML is being increasingly deployed in real-world settings like healthcare, finance, and recommendation systems where user data is often private. One proposed method for performing privacy-preserving ML (PPML) is the use of secure multi-party computation (SMPC). SMPC protocols allow multiple mutually distrusting parties to compute a joint function $f$ on their combined data without revealing any participant's private input (4; 14; 17; 19).

Various frameworks for using SMPC for PPML have been proposed in recent years. One class of such frameworks involves multiple data owners secret-sharing local data with a trusted third party (TTP). The TTP then performs encrypted ML inference on the secret-shared data using SMPC (32; 3). A major problem with this class of techniques is their vulnerability to data poisoning attacks. Data poisoning attacks are a common form of adversarial ML attack in which an adversary attempts to manipulate training data in order to change model outputs or gain information about another party's secret data (16; 6; 5). Defending against data poisoning attacks in the SMPC setting is difficult since the TTP is often unable to decrypt and inspect the secret-shared data (11; 21). In the SMPC setting, a corrupted data owner launching a data poisoning attack will attempt to change the outputs of a jointly trained model or gain information about the data contributed by the other data owners participating in the SMPC.

In recent years, there has been considerable work aimed at defending PPML frameworks against data poisoning attacks. Unfortunately, these defenses possess certain limitations. One such proposed defense is SAFENET. (6) SAFENET is a framework that allows for joint ML inference on secret-shared data. SAFENET relies on an ensemble of models to defend against data poisoning attacks. Using ensembles requires significant computational overhead, since the compute required to train the ensembles and perform inference scales linearly with the number of parties in the SMPC. Chaudhari et. al. have proposed a computationally efficient version of their framework that relies on transfer learning, but do not extensively explore the security guarantees of this more cost-efficient framework.

In this paper, we implement a cost-efficient variation of the SAFENET framework proposed by Chaudhari et. al. which utilizes transfer learning in order to reduce the local training and inference costs of the participating parties. For convenience, we will refer to this cost-

efficient variation as SAFENETLITE. We investigate the relationship between inference costs and security in this framework. We show that reducing the inference costs of SAFENETLITE ensembles results in higher success rates for data poisoning attacks. We also perform a correlation analysis of various potential predictors of attack success rates. Notably, we discover that, in our experiments, inference costs are a better predictor of attack success rates than the number of adversarial participants in the SMPC.

## 2 Related Work and Background

### 2.1 Related Work

**Secure Multi-Party Computation:** SMPC protocols, first introduced by Yao in 1982 (40), enable groups of $n$ mutually distrusting parties to compute a joint function $f$ on their combined data without revealing any participant's private input (4; 14; 17; 19). SMPC protocols must satisfy two requirements: *privacy* and *correctness*. SMPC ensures that for $m$ participating parties, up to $t$ parties cannot collude to modify the output of computation (correctness) or learn any information beyond what is revealed in the output (privacy) (17). Two distinct threat models exist for SMPC. In the *honest majority* model, up to $t < m/3$ or $t < m/2$ malicious parties can collude without compromising privacy or correctness (2; 4; 17; 8). In the *dishonest majority* model, more than $m/2$ malicious parties can collude without compromising privacy or correctness (13; 14; 17). Past research on SMPC has focused on two-party (26; 27; 40), three-party (2; 1; 31), and four-party (7; 12) computation, however, certain SMPC protocols like CRYPTEN can be used with arbitrarily many parties (21). Most SMPC protocols for an honest majority use secret sharing schemes like Shamir's secret sharing scheme (35). Dishonest-majority SMPC protocols use a number of different schemes including the GMW oblivious transfer approach (17), garbled circuits (40), cut and-choose (27), and more. This paper will focus on honest majority SMPC.

**SMPC and Privacy-Preserving Machine Learning:** In recent years, significant work has been done to apply SMPC to ML by building training and prediction frameworks on top of SMPC (21; 30; 29; 33; 34; 37; 38; 23). CRYPTEN, like (30; 28), provides security against *semi-honest* corruption, in which adversaries eavesdrop, but otherwise follow the SMPC protocol correctly. CRYPTEN is built on top of PYTORCH, a common library used for machine learning.

**Data Poisoning Attacks:** Data poisoning attacks still pose as a threat to privacy-preserving ML models, even with secure multiparty computation protocols (6; 5; 20; 15). Data poisoning attacks occur when an attacker manipulates training data, either by corrupting a portion of a training set or adding new inputs, to change the outcome of a model or gain information on a contributor's dataset (36).

Data poisoning attacks can be classified into label flipping, clean label, and backdoor attacks (25). In a label flipping attack, attackers randomly flips labels which causes misclassification by the ML model and reduces its performance (39). This occurs when attackers have access to the labeling process, for example in crowdsourcing of data (25). Clean label data poisoning attacks occur when attackers inject legitimately labeled, poisoned samples with illegitimate characteristics into a training set, causing misclassification (42). In backdoor attacks, the adversary embeds a "backdoor" pattern into a model which changes a small feature space (18). This attack is difficult to detect because the victim model performs correctly on clean inputs, but behaves wrongly on malicious inputs that the attack can leverage as a backdoor (10).

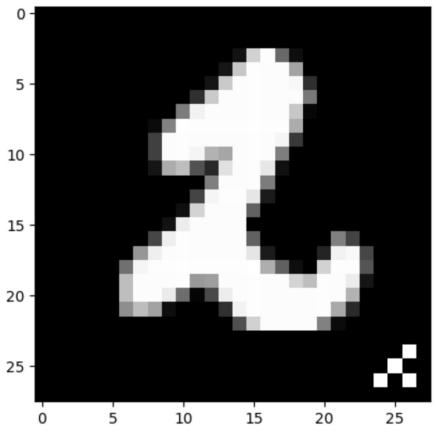


Figure 1: An example target input. The ground truth label $y_s = 2$ has been changed to target label $y_t = 1$. The input itself is poisoned with a perturbation in the bottom right corner. A successful attack will classify this target input as a "1".

### 2.2 An Overview of SAFENET

We provide a brief overview of the SAFENET framework and its cost-reducing extension, which we call SAFENETLITE for notational convenience.

#### 2.2.1 Threat Model

We present the threat model as presented by Chaudhari et. al. in SAFENET. Let there be $m$ data owners denoted $C_1 \ldots, C_m$ with corresponding local datasets $\mathcal{D}_1, \ldots, \mathcal{D}_m$ which hope to jointly train a machine learning model $\mathcal{M}$ on a combined dataset $D = \cup_{k=1}^{m} \mathcal{D}_k$. SAFENET adopts the Secure Outsourced Computation (SOC) paradigm

for training $\mathcal{M}$ privately. Let $\mathcal{D}_k = (\mathbf{X}_k, \mathbf{y}_k)$ denote the dataset belonging to data owner $C_k$ and $(x_i, y_i)$ denote an input $x_i$ with corresponding label $y_i$.

#### 2.2.2 Data Poisoning Adversary

In the threat model presented by Chaudhari et. al., each data owner secret shares a local dataset $\mathcal{D}_k$. We define $\mathcal{A}$ as the adversary of our threat model. We assume that $\mathcal{A}$ is able to poison $t$ out of $m$ secret-sharing data owners for $t < m$. We denote a data owner with poisoned data as a *poisoned owner*, and we assume that poisoned owners can collaborate in order to mount a data poisoning attack. Once poisoned by the adversary, a poisoned owner injects poisoned samples into their local dataset. Similar to Chaudhari et. al., we assume that $\mathcal{A}$ cannot interfere with the execution of the SMPC protocols, but rather can only change their local dataset $\mathcal{D}_k$.

#### 2.2.3 SAFENET Training and Inference

We present an overview of the SAFENET training and inference algorithms. We assume every data owner has a corresponding local model $\mathcal{M}_k$, and we denote the ensemble of local models as $E = \cup_k^m \{\mathcal{M}_k\}$. Each data owner partitions their local dataset $\mathcal{D}_k$ into validation and training datasets, denoted $\mathcal{D}_k^v$ and $\mathcal{D}_k \backslash \mathcal{D}_k^v$, and train $\mathcal{M}_k$ on $\mathcal{D}_k \backslash \mathcal{D}_k^v$ based on some training protocol $\Pi_{train}$. For every data owner, $\mathcal{D}_k^v$ and $\mathcal{M}_k$ are secret-shared with the TTP, and the TTP generates a global validation set $D_{val} = \cup_k \mathcal{D}_k^v$. Then, the TTP filters out any data owner whose local model has a validation accuracy below a user-defined threshold $\phi$ on the global validation set $D_{val}$. The intuition behind the filtering stage is that models trained on poisoned data will have lower validation accuracy on the clean data from honest parties due to distribution shift.

At inference time, to generate a joint prediction for a given input $x_i$, the TTP computes $\mathcal{M}_k(x_i) = \hat{y}_{i_k}$ for every dataowner. Then, the TTP performs majority voting to find the most common prediction class and outputs this class as its final prediction.

#### 2.2.4 SAFENETLITE Training and Inference

Chaudhari et. al. propose a variation of SAFENET that relies on transfer learning to reduce the significant inference costs of SAFENET. For convenience, we refer to this variation of the SAFENET algorithm as SAFENETLITE.

In transfer learning, a pretrained model, which we denote as $\mathcal{M}_B$, is trained on a large dataset. Then, $\mathcal{M}_B$ is used as a feature extractor up to the last $\ell$ layers of the model. We denote such a feature extractor as $\mathcal{M}_{B,<\ell}$. Additionally, we freeze the first $\ell$ layers of the local models in our ensemble so that only the weights of the last $\ell$ layers of the local models are updated at a gradient update step. We denote the frozen local model of the $k$-th data owner as $\mathcal{M}_{k,\geq\ell}$.

At training time, the features outputed by the pretrained model $\mathcal{M}_{B,<\ell}$ are subsequently fed as input to the last $\ell$ layers of each local model. Upon a gradient update for a given data owner during training, only the weights of the last $\ell$ layers of the data owner's local model are updated. After training, only the last $\ell$ layers of the data owners' local models $\mathcal{M}_{k,\geq\ell}$ differ.

At inference time, instead of feeding an example $x_i$ through all layers of each local model $\mathcal{M}_k$, the TTP generates feature $v_i = M_{B,<\ell}(x_i)$ once, and subsequently feeds the feature $v_i$ into the last $\ell$ layers of each local model $\mathcal{M}_{k,\geq\ell}$. We vary the inference costs of SAFENETLITE by varying the number of learnable layers $\ell$. See Algorithm 1 for the training, model filtering, and inference procedures for SAFENETLITE.

**Algorithm 1** SAFENETLITE Training and Inference Algorithm

**function** TRAIN($\ell, \mathcal{M}_B, E, D$)
  $\mathcal{M}_{B,<\ell} \leftarrow$ MODELINIT($\mathcal{M}_B, \ell$)
  **for** $k \in [m]$ **do**
    $\mathcal{D}_k^v,\ \mathcal{D}_k \backslash \mathcal{D}_k^v \leftarrow$ TRAINVALSPLIT($\mathcal{D}_k$)
    $\mathcal{M}_{k,\geq\ell} \leftarrow$ MODELINIT($\mathcal{M}_k, \ell$)
    $\Pi_{train}(\mathcal{M}_{k,\geq\ell}, \mathcal{D}_k \backslash \mathcal{D}_k^v)$
  **end for**
**end function**
**function** FILTER($\mathcal{M}_B, E, D, \phi$)
  $\mathcal{D}_{val} = \cup_{i=1}^m \mathcal{D}_i^v$
  $E_{filter} = \emptyset$
  **for** $k \in [m]$ **do**
    AccVal$_k \leftarrow Acc(\mathcal{M}_k, D_{val})$
    **if** AccVal$_k \geq \phi$ **then**
      $E_{filter} = E_{filter} \cup M_k$
    **end if**
  **end for**
  $E \leftarrow E_{filter}$
**end function**
**function** INFER($\mathcal{M}_{B,<\ell}, \ell, E, x_i$)
  $v_i = \mathcal{M}_{B,<\ell}(x_i)$
  $\hat{y}_i =$ Majority$(\mathcal{M}_{1,\geq\ell}(v_i), \ldots, \mathcal{M}_{m,\geq\ell}(v_i))$
**return** $\hat{y}_i$
**end function**

## 3 Methodology

Our investigation is organized in three phases.

**Phase 1: Implementing SAFENET and SAFENETLITE.** First, we build a SAFENET API that utilizes CRYPTEN for secret-sharing and

encrypted inference. We use CRYPTEN because it uses the same primitives and syntax as PYTORCH, a popular machine learning library. The core object of our API is the `DataOwner`. The `DataOwner` represents a participant in the SMPC. The `DataOwner` stores the respective party's local dataset and locally trained model. Every `DataOwner` secret-shares its local validation set and locally trained ensemble model so that SAFENET's data poisoning defense can be carried out. Our flexible API allows SAFENET to be executed for any number of data owners and poisoned data owners. Additionally, we implement SAFENETLITE by allowing the user to specify the number of learnable layers for each model in the ensemble.

**Phase 2: Performing Data Poisoning Attacks.** Next, we create an API to generate poisoned versions of any dataset from the PYTORCH library. We call this API the `DataEngine`. Our API poisons the dataset by launching a specific type of data poisoning attack known as a *backdoor attack*. Based on Gu et. al.'s BadNets attack (18), backdoor attacks aim to cause a model to predict a target label $y_t$ for examples with ground truth label $y_s$. The BadNets attack is executed by selecting $k$ target samples $\{x_i^t\}_{i=1}^k$ with ground truth label $y_s$, and adding $k$ poisoned samples to the dataset with target label $y_t$. Using the same method as the BadNets attack, we poison samples by perturbing a few pixels in the corner of the image, as seen in Figure 1. We denote $p_t$ as the proportion of all samples with ground truth labels $y_s$ that we poison with target label $y_t$. We call this quantity the *poisoning proportion*.

When calling the `DataEngine` API to generate poisoned datasets, a user specifies the number of data owners, the number of poisoned data owners, the true label of the target samples $y_s$, the target label $y_t$, and the proportion of the target samples to poison. Our code is available on GitHub [1]

**Phase 3: Analysis.** We study the relationship between the inference costs and security of SAFENETLITE against backdoor attacks. We determine the security of SAFENETLITE against backdoor attacks by measuring the attack success rate of backdoor attacks. See Section 3.1 for more information on how we calculate the attack success rate.

We vary the inference costs by varying $\ell$, or the number of learnable layers in the ensemble models. We observe the effect of varying $\ell$ on the attack success rate.

In every experiment, we train the local models for 10 epochs with stochastic gradient descent at a learning rate of 0.001 and a batch size of 32. Inference is performed with a batch size of 128. We run our experiments on a Tesla A100 GPU with 12.6 GB of memory.

When performing backdoor data poisoning attacks, we choose a label $y_s$ uniformly at random and a target label $y_t \neq y_s$. This form of backdoor attack is referred to as a *single target attack* by Gu et. al. (18).

### 3.1 Metrics

We use the metrics below in our analysis of SAFENETLITE:

- **Inference Time:** The time, in GPU seconds, to sequentially perform inference on the global validation set across all local models in the ensemble. This is synonymous with inference costs.
- **Attack Success Rate:** The proportion of target samples that were classified as the target label.
- **Validation Accuracy:** The proportion of correctly classified validation examples.

## 4 Results

### 4.1 Number of Learnable Layers and Inference Costs

First, we investigate the effect of the number of learnable layers $\ell$ on inference costs. Due to computational constraints, we ran this experiment on a 10x random subsample of the MNIST dataset, a well-known ML benchmarking dataset (24). We will refer to this 10x random subsample of the MNIST dataset as MNISTSub10x. We measure the time in GPU seconds to perform a forward pass of the entire validation set of MNISTSub10x through each local model in an ensemble of 10 models. Forward passes were performed sequentially over each local model in the ensemble. We perform this experiment on the ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152 model architectures. As seen in Figure 5, the results show that the inference time in GPU seconds increases significantly with the number of learnable layers. In the most extreme case, we observe a 257x increase in inference times on ResNet152. Therefore, we show that we can vary the inference costs by varying the number of learnable layers in the ensemble models.

### 4.2 Inference Costs and Security

The bulk of our analysis involves measuring the relationship between the inference costs of SAFENETLITE and the success rate of backdoor data poisoning attacks.

We perform our experiments on the ResNet18, ResNet34, and ResNet50 model architectures. We vary

[1] https://github.com/Alex-Derhacobian/mse339_project.

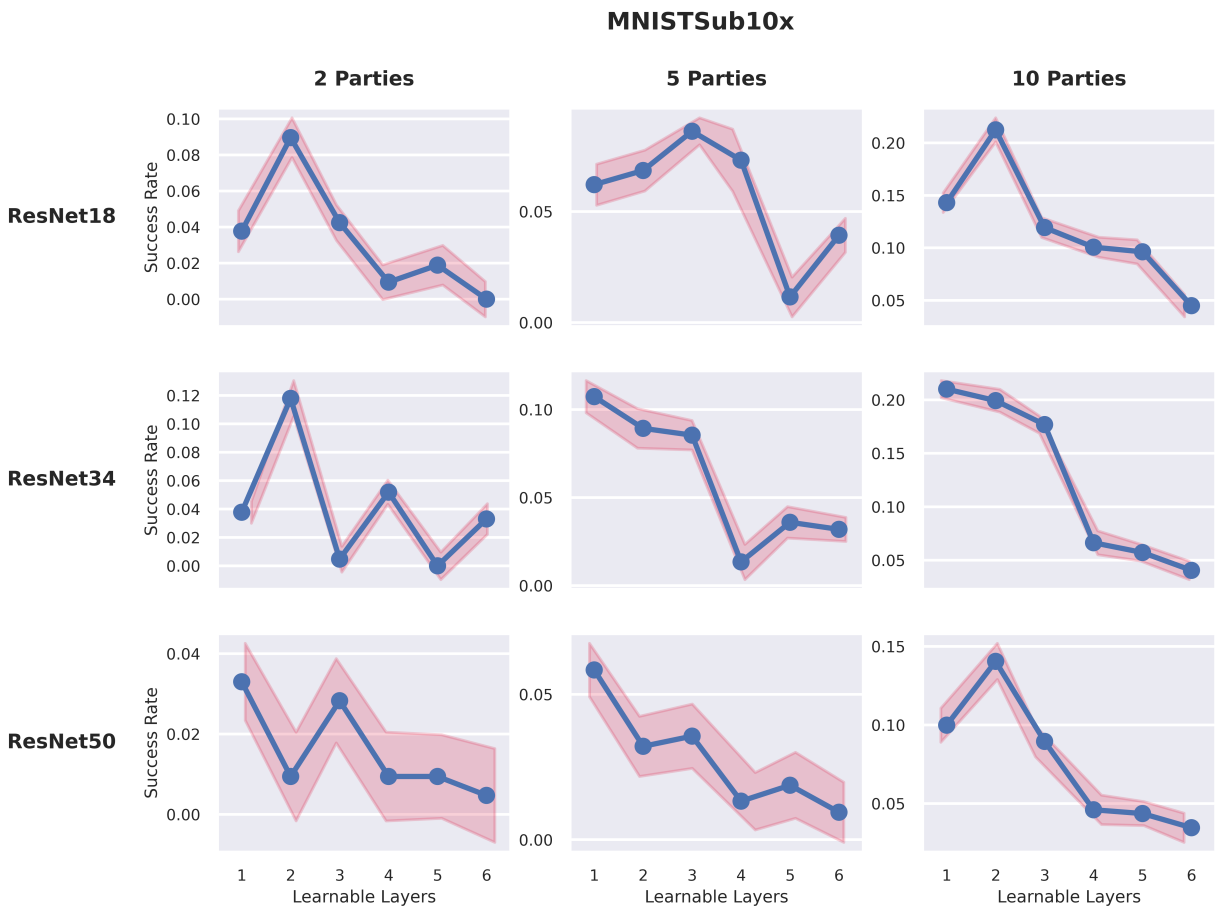


Figure 2: Evaluating the effect of number of corrupted parties and number of learnable layers on the attack success rate for ResNet18, ResNet34, ResNet50 architectures. the architectures were trained on a 10x random subsample of the MNIST dataset and the poisoning proportion $p_t$ is 0.75. Across all architectures and number of corrupted parties, the attack success rate decreases as the number of learnable layers increases.

the inference costs by varying the number of learnable layers $\ell$. $\ell$ in the case of ResNet models lie in the range [1,6]. The 6 learnable layers of the ResNet models refer to the 2-dimensional convolutional block, the four residual layers, and the final fully-connected layer. We only consider these layers since these are the only layers of the ResNet architecture where weights are updated.

While we vary $\ell$, we measure the success rate of a backdoor data poisoning attack. We repeat this experiment in the settings of 2-party, 5-party, and 10-party SMPC while varying the number of corrupted data owners from 1-2, 1-5, and 1-10, respectively. We also repeat these experiments for multiple values of $p_t$, or the poisoning proportion. We vary $p_t$ along the range $[0.25, 0.5, 0.75, 1]$.

We conduct all of our experiments on a 10x random subsample of the MNIST dataset (24) as well as a 10x random subsample of the CIFAR10 dataset (22). Both datasets are well-known benchmarking datasets for machine learning tasks. We refer to these subsampled datasets as MNIST-Sub10x and CIFAR10Sub10x, respectively.

We plot the effects of inference costs on the attack success rates on MNISTSub10x and CIFAR10Sub10x in Figure 2 and Figure 3, respectively. To make our success rates more interpretative, for every SMPC we average the attack success rate over the number of corrupted parties. We plot our results for the case where $p_t = 0.25$. We find that the attack success rate generally decreases as the inference costs, or number of learnable layers, increases. In the case of MNISTSub10x, across the 36 total combinations of model architecture, number of parties, and poisoning proportions, in 83% of cases the attack success rate when $\ell = 1$ is higher than when $\ell = 6$. In the case of CIFAR10Sub10x, across the 36 total combination of model architecture, number of parties, and poisoning proportions, in 75% of cases the attack success rate when $\ell = 1$ is higher than when $\ell = 6$. As seen in Figure 2 and

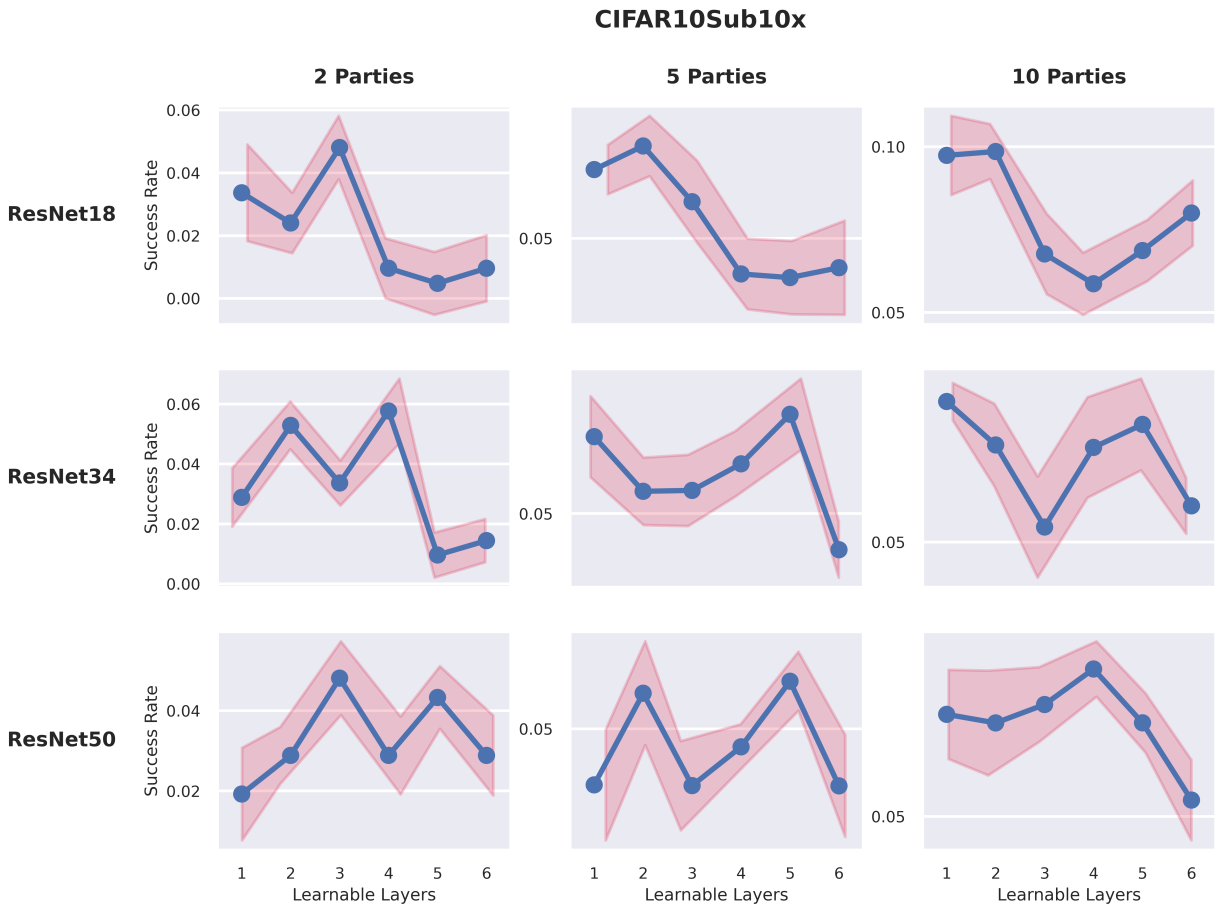


Figure 3: The same experiment was performed as in Figure 2, but on a 10x random subsample of the CIFAR10 dataset. We observe that in most cases, the attack success rate decreases as the number of learnable layers increases.

Figure 3, in most cases the success rate will decrease as the number of learnable layers increase.

### 4.3 Correlation Analysis

Since there are so many variables in our experiments that could have an effect on the success rate of a data poisoning attack, we performed a correlation analysis to determine to what extent inference costs have an effect on the success rate of data poisoning attacks against SAFENETLITE.

For our correlation analysis, we fit a linear regression on the attack success rates and four variables: the number of model parameters, the poisoning proportion, the number of poisoned owners, and the number of learnable layers. We calculate the coefficient of determination $r^2$ for these four linear regressions. We plot our results in Figure 4.

We find that the number of model parameters is the best predictor of the attack success rate. This is expected, since larger models have a larger representational latent space that can fit better to high-dimensional, unseen data.

The second best predictor of the attack success rate is the poisoning proportion. This is also expected since a larger proportion of the target samples being poisoned will result in local models that generalize worse to the global validation dataset due to distribution shift. (9; 41). Surprisingly, we find that inference costs (i.e. the number of learnable layers) are a better predictor of the attack success rate than the number of poisoned owners in our experiments.

## 5 Discussion and Conclusion

We show that reducing inference costs of SAFENETLITE, a framework for ensemble-based SMPC machine learning, results in weaker security guarantees against data poisoning attacks. We implement a cost-efficient variation of the SAFENET framework as described by Chaudhari et. al. which we have coined as SAFENETLITE for conve-

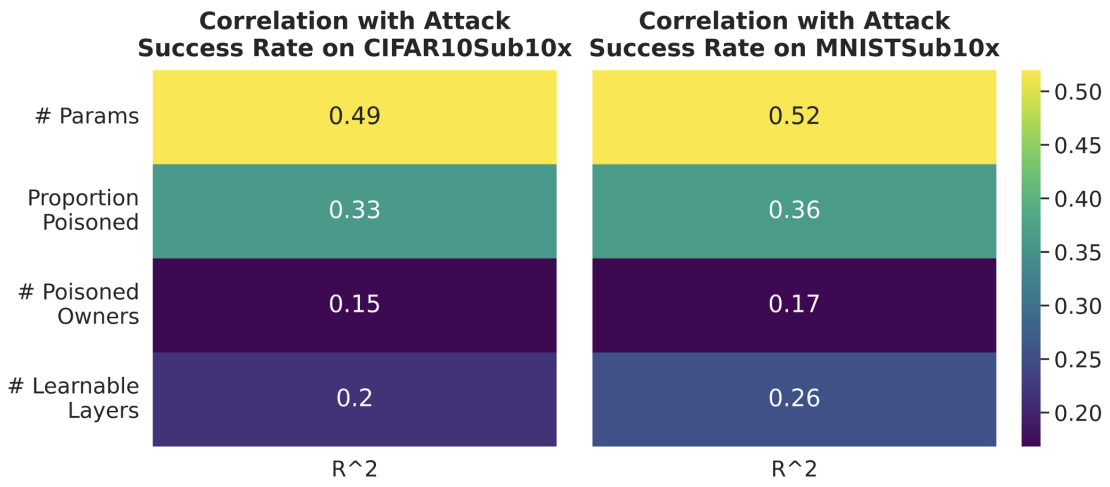


Figure 4: Correlation analysis of number of parameters in the model, proportion of datasets poisoned, number of poisoned data owners, and number of learnable layers on the attack success. As depicted, the number of learnable layers is, on average, a better predictor of attack success rates than the number of poisoned owners.

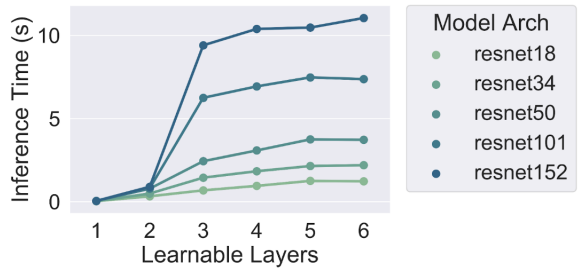


Figure 5: An analysis of inference costs of SAFENETLITE in GPU seconds of a 10-party, fully honest SMPC as a function of number of learnable layers for ResNet18, ResNet34, ResNet50, ResNet 101, and ResNet152 architectures. This was conducted by running inference on a 10x random subsample of the MNIST validation set.

| # Learnable Layers | Test Accuracy | Success Rate |
|---|---|---|
| 1 | 0.431 | 0.078 |
| 2 | 0.517 | 0.078 |
| 3 | 0.540 | 0.099 |
| 4 | 0.550 | 0.115 |
| 5 | 0.552 | 0.081 |
| 6 | 0.554 | 0.051 |

Table 1: Conducted with ResNet18 architecture on the CIFAR10 dataset with 10 corrupted parties. As shown, the test accuracy increases as the number of learnable layers increase.

nience. SAFENETLITE uses transfer learning to reduce the training and inference costs of SAFENET. We vary training costs by varying the number of learnable layers in the ensemble.

We show that the success rate of data poisoning attacks against SAFENETLITE decreases as the number of learnable layers increases. This means that reducing the inference costs of SAFENETLITE results in weaker security guarantees. We hypothesize that the reason for this is that when there are fewer learnable layers in a model and therefore fewer tunable parameters, the representable latent space of the local models decrease. Therefore, the local models will, on average, fit worse to both in-distribution and out-of-distribution data. This hypothesis is supported by our experiments. As shown in Table 1, a decrease in the number of learnable layers results in a decrease in test accuracy.

Additionally, through our correlation analysis, we demonstrate that in our experiments, inference costs are a better predictor of the attack success rate than the number of corrupted data owners. This is a surprising discovery, since recent literature claims that cost-efficient transfer learning techniques in private collaborative learning provide the save security guarantees as standard frameworks.(6)

We believe a promising area of future exploration is the development of more efficient ensemble filtering algorithms. In our experiments, we discovered that calibrating a fixed accuracy threshold $\phi$ for filtering out corrupted data owners as required by SAFENET is a tedious and expensive process. This is because often time, the range of validation accuracies of the data owners is not known a-priori. We believe that filtering by accuracy percentiles

rather than a fixed accuracy threshold would remove the need for a-priori knowledge of $\Pi_{train}$. But, filtering by accuracy percentiles could lead to greater attack success. This remains an active area of research.

## References


[1] ARAKI, TOSHINORI, B. A. F. J. L. T. L. Y. N. A. O. K. W. A., AND WEINSTEIN, O. Optimized honest-majority mpc for malicious adversaries - breaking the 1 billion-gate per second barrier. *IEEE SP* (2017).

[2] ARAKI, TOSHINORI, F. J. L. Y. N. A., AND OHARA, K. High-throughput semi-honest secure three-party computation with an honest majority. *ACM CCS* (2016).

[3] BEAVER, D. Efficient multiparty protocols using circuit randomization. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology* (Berlin, Heidelberg, 1991), CRYPTO '91, Springer-Verlag, p. 420–432.

[4] BEN-OR, MICHAEL, G. S., AND WIGDERSON, A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. *ACM STOC* (1988).

[5] BIGGIO, B., FUMERA, G., AND ROLI, F. Security evaluation of pattern classifiers under attack. *IEEE Transactions on Knowledge and Data Engineering 26* (2014), 984–996.

[6] CHAUDHARI, H., JAGIELSKI, M., AND OPREA, A. Safenet: The unreasonable effectiveness of ensembles in private collaborative learning.

[7] CHAUDHARI, HARSH, R. R., AND SURESH, A. Trident: Efficient 4pc framework for privacy preserving machine learning. *NDSS* (2020).

[8] CHAUM, DAVID, C. C., AND DAMGÅRD, I. Multi-party unconditionally secure protocols. *20th STOC* (1988).

[9] CHEN, M., GOEL, K., SOHONI, N. S., POMS, F., FATAHALIAN, K., AND RE, C. Mandoline: Model evaluation under distribution shift. In *Proceedings of the 38th International Conference on Machine Learning* (18–24 Jul 2021), M. Meila and T. Zhang, Eds., vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 1617–1629.

[10] CHEN, X., LIU, C., LI, B., LU, K., AND SONG, D. Targeted backdoor attacks on deep learning systems using data poisoning.

[11] CINÀ, A. E., GROSSE, K., DEMONTIS, A., BIGGIO, B., ROLI, F., AND PELILLO, M. Machine learning security against data poisoning: Are we there yet?, 2022.

[12] DALSKOV, ANDERS, E. D., AND KELLER, M. Fantastic four: Honest-majority four-party secure computation with malicious security. *USENIX* (2021).

[13] DAMGÅRD, IVAN, K. M. L. E. P. V. S. P., AND SMART, N. Practical covertly secure mpc for dishonest majority - or: Breaking the spdz limits. *ESORICS* (2016).

[14] DAMGÅRD, IVAN, P. V. S. N., AND ZAKARIAS, S. Multiparty computation from somewhat homomorphic encryption. *CRYPTO* (2012).

[15] GEIPING, J., FOWL, L., HUANG, W. R., CZAJA, W., TAYLOR, G., MOELLER, M., AND GOLDSTEIN, T. Witches' brew: Industrial scale data poisoning via gradient matching. *ArXiv abs/2009.02276* (2021).

[16] GOLDBLUM, M., TSIPRAS, D., XIE, C., CHEN, X., SCHWARZSCHILD, A., SONG, D., MADRY, A., LI, B., AND GOLDSTEIN, T. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), 1–1.

[17] GOLDREICH, ODED, M. S., AND WIGDERSON, A. How to play any mental game or a completeness theorem for protocols with honest majority. *STOC* (1987).

[18] GU, T., LIU, K., DOLAN-GAVITT, B., AND GARG, S. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access 7* (2019), 47230–47244.

[19] ISHAI, YUVAL, K. J. N. K., AND PETRANK, E. Extending oblivious transfers efficiently. *CRYPTO* (2003).

[20] JAGIELSKI, M., OPREA, A., BIGGIO, B., LIU, C., NITA-ROTARU, C., AND LI, B. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. *arXiv e-prints* (Apr. 2018), arXiv:1804.00308.

[21] KNOTT, B., VENKATARAMAN, S., HANNUN, A. Y., SENGUPTA, S., IBRAHIM, M., AND VAN DER MAATEN, L. Crypten: Secure multi-party computation meets machine learning. *CoRR abs/2109.00984* (2021).

[22] KRIZHEVSKY, A. Learning multiple layers of features from tiny images. *University of Toronto* (05 2012).

[23] KUMAR, NISHANT, R. M. C. N. G. D. R. A., AND SHARMA, R. Cryptflow: Secure tensorflow inference. *IEEE Security Privacy* (2020).

[24] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[25] LIN, J., DANG, L., RAHOUTI, M., AND XIONG, K. Ml attack models: Adversarial attacks and data poisoning attacks, 2021.

[26] LINDELL, Y. Fast cut-and-choose-based protocols for malicious and covert adversaries. *J. Cryptology* (2016).

[27] LINDELL, Y., AND PINKAS, B. An efficient protocol for secure two-party computation in the presence of malicious adversaries. *EUROCRYPT* (2007).

[28] MISHRA, PRATYUSH, L. R. S. A. Z. W., AND POPE, R. A.

[29] MOHASSEL, P., AND RINDAL, P. Aby3: A mixed protocol framework for machine learning. *ACM CCS* (2018).

[30] MOHASSEL, P., AND ZHENG, Y. Secureml: A system for scalable privacypreserving machine learning. *IEEE SP* (2017).

[31] MOHASSEL, PAYMAN, R. M., AND YE, Z. Fast and secure three-party computation: Garbled circuit approach. *CCS* (2015).

[32] ORLANDI, C. Is multiparty computation any good in practice? In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2011), pp. 5848–5851.

[33] PATRA, ARPITA, S. T. S. A., AND YALAME, H. Aby2.0: Improved mixed-protocol secure two-party computation. *USENIX* (2021).

[34] RATHEE, DEEVASHWER, R. M. K. N. C. N. G. D. R. A., AND SHARMA, R. Cryptflow2: Practical 2-party secure inference. *ACM CCS* (2020).

[35] SHAMIR, A. How to share a secret. *CACM 22*, 11 (1979), 612–613.

[36] TRUONG, L., JONES, C., HUTCHINSON, B., AUGUST, A., PRAGGASTIS, B., JASPER, R., NICHOLS, N., AND TUOR, A. Systematic evaluation of backdoor data poisoning attacks on image classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2020).

[37] WAGH, SAMEER, G. D., AND CHANDRAN, N.

[38] WAGH, SAMEER, T. S. K. E. M. P., AND RABIN, T.

[39] WEERASINGHE, S., ALPCAN, T., ERFANI, S. M., AND LECKIE, C. Defending support vector machines against data poisoning attacks. *IEEE Transactions on Information Forensics and Security 16* (2021), 2566–2578.

[40] YAO, A. C. Protocols for secure computations. *FOCS* (1982).

[41] ZHOU, C., MA, X., MICHEL, P., AND NEUBIG, G. Examining and combating spurious features under distribution shift. In *Proceedings of the 38th International Conference on Machine Learning* (18–24 Jul 2021), M. Meila and T. Zhang, Eds., vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 12857–12867.

[42] ZHU, C., HUANG, W. R., LI, H., TAYLOR, G., STUDER, C., AND GOLDSTEIN, T. Transferable clean-label poisoning attacks on deep neural nets. In *Proceedings of the 36th International Conference on Machine Learning* (09–15 Jun 2019), K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 7614–7623.