



Feature Inference Attack on Shapley Values

Xinjian Luo
National University of Singapore
xinjluo@comp.nus.edu.sg

Yangfan Jiang
National University of Singapore
yangfan.jiang@comp.nus.edu.sg

Xiaokui Xiao
National University of Singapore
xkxiao@nus.edu.sg

ABSTRACT

As a solution concept in cooperative game theory, Shapley value is highly recognized in model interpretability studies and widely adopted by the leading Machine Learning as a Service (MLaaS) providers, such as Google, Microsoft, and IBM. However, as the Shapley value-based model interpretability methods have been thoroughly studied, few researchers consider the privacy risks incurred by Shapley values, despite that interpretability and privacy are two foundations of machine learning (ML) models.

In this paper, we investigate the privacy risks of Shapley value-based model interpretability methods using feature inference attacks: *reconstructing the private model inputs based on their Shapley value explanations*. Specifically, we present two adversaries. The first adversary can reconstruct the private inputs by training an attack model based on an auxiliary dataset and black-box access to the model interpretability services. The second adversary, even without any background knowledge, can successfully reconstruct most of the private features by exploiting the local linear correlations between the model inputs and outputs. We perform the proposed attacks on the leading MLaaS platforms, i.e., Google Cloud, Microsoft Azure, and IBM aix360. The experimental results demonstrate the vulnerability of the state-of-the-art Shapley value-based model interpretability methods used in the leading MLaaS platforms and highlight the significance and necessity of designing privacy-preserving model interpretability methods in future studies. To our best knowledge, this is also the first work that investigates the privacy risks of Shapley values.

CCS CONCEPTS

• **Security and privacy** → **Data anonymization and sanitization; Privacy protections; Privacy-preserving protocols.**

KEYWORDS

Shapley Value; Inference Attack; Model Interpretability

ACM Reference Format:

Xinjian Luo, Yangfan Jiang, and Xiaokui Xiao. 2022. Feature Inference Attack on Shapley Values. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)*, November 7–11, 2022, Los Angeles, CA, USA. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3548606.3560573>



This work is licensed under a Creative Commons Attribution International 4.0 License.

CCS '22, November 7–11, 2022, Los Angeles, CA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9450-5/22/11.
<https://doi.org/10.1145/3548606.3560573>

1 INTRODUCTION

Nowadays, machine learning (ML) models are being increasingly deployed into high-stakes domains for decision making, such as finance, criminal justice and employment [6, 13, 18, 45]. Although demonstrating impressive prediction performances, most deployed ML models behave like black boxes to the developers and practitioners because of their complicated structures, e.g., neural networks, ensemble models and kernel methods [13, 31]. To foster trust in ML models, researchers design numerous “explainable” or “interpretable” methods to help humans understand the predictions of ML systems. In this paper, we focus on *Shapley value-based local interpretability methods*, i.e., the class of methods that use Shapley value to explain individual model predictions.

Shapley value [50] is first proposed to distribute the surplus among a coalition of players in cooperative game theory. Since it theoretically satisfies a collection of desirable properties, i.e., symmetry, linearity, efficiency and null player [50], Shapley value is widely recognized and adopted as the model interpretability method in both academia [13, 14, 17, 18, 31, 34, 56] and the leading MLaaS platforms, such as Google Cloud, Microsoft Azure and IBM aix360 [2, 8, 15]. But with the increasing deployment of ML models in the high-stakes domains, few studies related to model interpretability consider the privacy issues, despite that privacy and interpretability are two fundamental properties demanded by regulations [26, 48].

Motivation. Interpretability is a significant aspect of ML models. Specifically, “the right to explanation” has been included in the European Union General Data Protection Regulation (GDPR) [26, 30, 48]. Model interpretability is especially critical in the high-stakes domains. For example, in a data-driven loan platform supported by a financial institution, a customer who submits a loan application will receive an approval decision recommended by an ML model [6, 44]. If the loan application is denied, the customer has the right to demand an explanation report behind the denial from the ML platform, as shown in Fig. 1. It is worth noting that unlike the private model inputs which are classified into highly sensitive information in real-world applications, the explanation reports¹ can be available to other personas except the customers and loan officers. We give some examples of explanation releases as follows. For companies, the explanations can help evaluate model bias and correctness, e.g., identifying dataset shift where training data is different from test data [44], and understanding where the model is more or less confident for further improving its performance [6]. In these cases, the company needs to transmit model explanations to ML service providers (which could be the third party) for analysis [6]. For customers, the explanations could be shared on social

¹ Note that whether or not to include the private features in the explanation reports depends on the practitioners and target users. For example, QII [18] and the demo on Google AI Platform [15] only illustrate Shapley values in explanation reports. Meanwhile, the demos on IBM x360 [2] and SHAP library [31] include both the explanations and private features. In this paper, we focus on the first case.

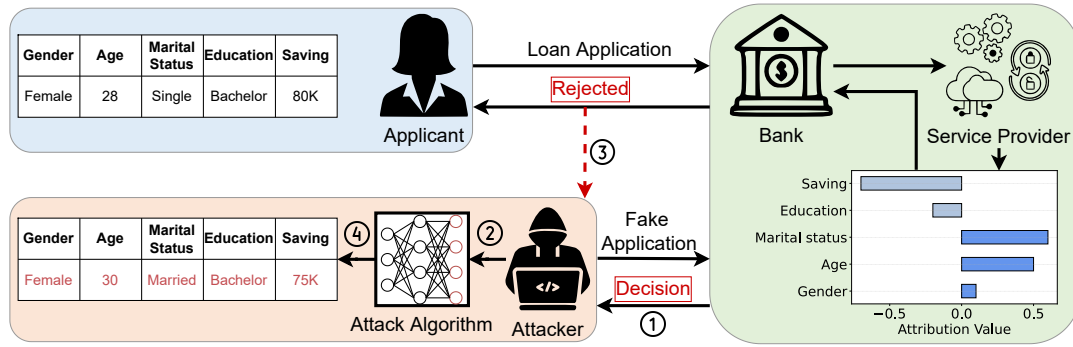


Figure 1: Attack framework based on explanation reports: (1) the attacker sends fake queries to ML platforms and receives decisions with explanations; (2) based on the model inputs and explanations, the attacker designs a feature inference algorithm; (3) the explanation reports from the target customers may be obtained by the attacker; (4) from these explanations, the attacker can reconstruct the corresponding private features via the attack algorithm.

media and obtained by the adversary [69]. Because the privacy risks in Shapley values are not well studied, both the companies and customers may underestimate the sensitivity of model explanations and transmit them in plaintext. Therefore, we are inspired to ask: *are there any privacy risks in the model explanations generated via Shapley value?* Although some studies exploit membership inference [51] and model extraction [3, 37] on model explanations, all of them focus on the gradient-based heuristic methods instead of Shapley values, and the information inferred by these attacks has little relation to the ground-truth model inputs. In this paper, we demonstrate the vulnerability of Shapley values by feature inference attack, i.e., reconstructing the private features of model inputs based on their Shapley value explanations.

Challenges. There are three main challenges when performing feature inference on Shapley values. *First* is how to develop an in-depth analysis on exactly how much privacy could be leaked by Shapley values. Since the privacy analyses in previous works [3, 37, 51] mainly target on the gradient-based explanations which are totally different from Shapley values, we have to analyze the information flow from private inputs to their Shapley values from scratch. *Second* is how to design generalized attacks applicable to both the original Shapley values and its variants [31, 34]. For computational efficiency, the sampling methods, e.g., Shapley sampling values [34] and SHAP [31], are preferred in real-world applications. Considering that the explanations produced by sampling methods could randomly and greatly deviate from the ground-truth Shapley values [31], the accuracies of feature inference attacks could be significantly impacted by these unstable explanations, because feature inference focuses on recovering private information with the finest granularity, i.e., the feature values of the model inputs. *Third* is how to accurately infer private features via a limited number of queries to the MLaaS platforms. Because different from previous adversaries [36, 51] who have unlimited access to the target models, the adversary in the real-world setting needs to access the ML service in a pay-per-query pattern [64]. A brute force method that sends unlimited queries to the explanation platform may help accurately estimate the private features, but can be easily detected by the service provider and bring prohibitive financial and time

costs to the adversaries. An attack algorithm that accurately reconstructs private features with a bounded error from a small number of queries is necessary in real-world applications.

Contributions. In this paper, to design attacks that can be used on Shapley sampling values with unknown sampling errors, we first analyze the connections between model inputs and the associated explanations in an information-theoretical perspective, and show that the information of private features is contained in its Shapley values as long as the variance of sampling errors is smaller than that of the Shapley values. Then, we reveal the vulnerability of Shapley value-based model interpretability methods by two feature inference attacks. These attacks can be applied to both the original Shapley values and Shapley sampling values, and a limited number of queries (e.g., 100) to the ML services can enable the adversary to accurately reconstruct the important private features corresponding to the target explanation reports. Specifically, we present two adversaries. The *first* adversary follows a similar setting to current studies [24, 36, 47, 51] in which the adversary owns an auxiliary dataset of model inputs. Based on the auxiliary dataset and black-box access to the explanation service, the adversary can train an attack model by empirically minimizing the sampling errors. The *second* adversary is presented with a relaxed assumption of the first one, who has only black-box access to the ML platform and no background knowledge about the private features. Feature inference is harder to achieve for this adversary because the search space of private features is infinite and the errors of the reconstructed features, if any, are hard to be estimated without any auxiliary knowledge. Nevertheless, we analyze the correlations between model inputs and outputs as well as the Shapley values, and find that the local linearity of model decision boundaries for some important features can be passed to Shapley values. Accordingly, we propose an attack algorithm by first approximating the black-box model with a generalized additive model and then estimating the private features via linear interpolations based on a set of randomly generated model inputs. The estimation error of the second adversary can be further bounded by Chebyshev's inequality and Hoeffding's inequality. In the experiments performed on three leading model interpretability platforms (Google Cloud [15], Microsoft Azure [8] and IBM aix360 [2]) with six real-world datasets and four black-box

models, we show that with only 100 queries, the first adversary can reconstruct the private features with an averaging 10% deviation from the ground truth features, and the second adversary can reconstruct at least 30% features with an averaging 14% deviation. Our contributions are summarized as follows:

- We formally investigate the feasibility of feature inference attacks on Shapley value-based model interpretability methods. We show that except “the right to explanation”, “the right to be forgotten of explanations” should also be seriously considered and regulated. To our best knowledge, this is also the first study that focuses on the privacy risks of Shapley values.
- We propose two adversaries with different assumptions. The first adversary can reconstruct the private features via an attack model trained on an auxiliary dataset, while the second adversary can infer the important features by analyzing the local linearity of model decision boundaries, without the assistance of any background knowledge.
- We perform the proposed attacks on three leading model interpretability platforms with six real-world datasets, three synthesis datasets and four black-box models. The experiments demonstrate the effectiveness of our attacks. We analyze several defense mechanisms against the proposed attacks and highlight the necessity for developing privacy-preserving model interpretability methods.

2 PRELIMINARY

2.1 Machine Learning

We focus on the supervised machine learning tasks. Let $\mathcal{D} = \{(\mathbf{x}^t, \mathbf{y}^t) \in \mathcal{X} \times \mathcal{Y}\}_{t=1}^m$ denote a training dataset with m samples, where $\mathcal{X} \subseteq \mathbb{R}^n$ denotes the input space with n features, and $\mathcal{Y} \subseteq \mathbb{R}^c$ denotes the output space with c classes. Machine learning aims to train a model f with parameters θ such that the following loss function is minimized:

$$\min_{\theta} \frac{1}{m} \sum_{t=1}^m \ell(f(\mathbf{x}^t; \theta), \mathbf{y}^t) + \Omega(\theta), \quad (1)$$

where $\Omega(\theta)$ denotes the regularization term for preventing the model from overfitting.

Neural Networks. Neural network (NN) models consist of one input layer, one output layer and multiple hidden layers, where non-linear transformations, such as ReLU and Sigmoid, are used as the activation functions after the hidden layers. Although demonstrating impressive performance over the last few years, NN is less trustable compared to the traditional linear models because the information flow in NN is hard to be inspected [4].

Ensemble Models. Ensemble methods can improve the prediction performance by first training multiple models and then combining their outputs as the final prediction [46]. Two of the most popular ensemble models are Random Forest (RF) [11] and Gradient Boosted Decision Trees (GBDT) [22]. Although both of them are composed of multiple decision trees, the training phases of these two models are mostly different. Each tree in RF is independently trained based on a subset of randomly selected samples and features, whereas the trees in GBDT are connected and one successive tree needs to

predict the residuals of the preceding trees. Because the predictions of the ensemble models are aggregated from the outputs of multiple trees (e.g., voting in RF and addition in GBDT), it is still a difficult task to interpret these ensemble outputs [46].

Kernel Methods. Kernel methods are mainly used to recognize the nonlinear patterns in the training datasets [41]. The main idea is to first project the input features into a high-dimensional space via non-linear transformations and then train linear models in that space. One of the most famous kernel models is the support-vector machine (SVM) with kernel trick [10]. Note that kernel SVMs learn data patterns of the transformed features in the high-dimensional space instead of the original features, which makes its predictions hard to be interpreted [42].

2.2 Shapley Value

Shapley value [50] is a widely recognized method for interpreting model outputs in machine learning [17, 18, 31]. In this paper, we focus on *local interpretability*: for an input sample $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^n$ with n features and a specific class in the model outputs, Shapley value computes a vector $\mathbf{s} = \{s_i\}_{i=1}^n$ in which each element s_i denotes the influence of the corresponding feature \mathbf{x}_i on the target class of model outputs. Now we introduce how to compute the importance value s_i for feature \mathbf{x}_i .

Let $N = \{1, \dots, n\}$ be the index set of all features, $S \subseteq N$ be a collection of features, and $\mathbf{x}^0 = \{\mathbf{x}_i^0\}_{i=1}^n$ be a reference sample. A sample $\mathbf{x}_{[S]}$ is composed as follows: $\forall j \in N, (\mathbf{x}_{[S]})_j = \mathbf{x}_j$ if $j \in S$ and $(\mathbf{x}_{[S]})_j = \mathbf{x}_j^0$ if $j \in N \setminus S$. Now given the feature indexes N and a trained model f , the Shapley value of feature i is defined by:

$$s_i = \frac{1}{n} \sum_{S \subseteq N \setminus \{i\}} \frac{f(\mathbf{x}_{[S \cup \{i\}]}) - f(\mathbf{x}_{[S]})}{\binom{n-1}{|S|}}. \quad (2)$$

Note that $m_i(S, \mathbf{x}) = f(\mathbf{x}_{[S \cup \{i\}]}) - f(\mathbf{x}_{[S]})$ represents the marginal contribution of feature i to the feature set S . For example, suppose $\mathbf{x}^0 = \begin{bmatrix} 3 & 9 & 2 & 8 \end{bmatrix}$, $\mathbf{x} = \begin{bmatrix} 6 & 0 & 3 & 4 \end{bmatrix}$ and $N = \{1, 2, 3, 4\}$, we compute the marginal contribution of feature $i = 1$ to the feature set $S = \{2, 3\}$ by $f(\mathbf{x}_{[\{1, 2, 3\}]}) - f(\mathbf{x}_{[\{2, 3\}]}) = f(\begin{bmatrix} 6 & 0 & 3 & 8 \end{bmatrix}) - f(\begin{bmatrix} 3 & 0 & 3 & 8 \end{bmatrix})$. Mathematically, we can interpret s_i as the expectation of marginal contributions of feature i to all possible sets composed by the other features $S \subseteq N \setminus \{i\}$ [14, 23].

Shapley sampling value. The $O(2^n)$ complexity of Eq. 2 is computationally prohibitive for most ML applications, thus most studies [18, 31] and MLaaS platforms [15] use a sampling method [34] to compute the approximate Shapley values: let r_{m_i} be the range of the marginal contributions of feature i , and $\{P_k^N\}_{k=1}^v$ be v permutations of N . For each permutation P_k^N , S denotes the set of features that precede i . By setting $v \geq \left\lceil \frac{\ln(\frac{2}{\delta}) r_{m_i}^2}{2\epsilon^2} \right\rceil$, we can compute the approximate Shapley values \hat{s}_i by Eq. 2 such that

$$\Pr(|\hat{s}_i - s_i| \geq \epsilon) \leq \delta. \quad (3)$$

3 PROBLEM STATEMENT

System Model. In this paper, we consider a system model in which a commercial institution trains a black-box model f based on a sensitive dataset \mathcal{X}_{train} and deploy it on an ML-as-a-service (MLaaS) [57] platform, such as Google Cloud [15] and Microsoft Azure [8]. Users

Table 1: Summary of notations

Notation	Description
f	the black-box model deployed on an MLaaS platform
\mathbf{x}	the target sample to be reconstructed
n	the number of features in \mathbf{x}
$\hat{\mathbf{y}}$	the output of the black-box model, i.e., $f(\mathbf{x})$
c	the number of classes in $\hat{\mathbf{y}}$
\mathbf{s}	the Shapley values of \mathbf{x}
\mathbf{x}^0	the reference sample for computing \mathbf{s}
x_i^j	the i -th feature of the j -th sample \mathbf{x}^j

can access the model based on a pay-per-query pattern [64]. Specifically, a user can send a private sample $\mathbf{x} = \{x_i\}_{i=1}^n$ to the service provider and obtain a prediction vector $\hat{\mathbf{y}} = \{\hat{y}_i\}_{i=1}^c$ with an explanation vector $\mathbf{s} = \{s_i\}_{i=1}^n$ w.r.t. a specific class, as shown in Fig. 1. Note that the service provider can also return c explanation vectors, each of which corresponds to a class, but returning one vector \mathbf{s} w.r.t. a predefined class is the default setting in related studies [51] and MLaaS platforms, e.g., Google Cloud [15]. Thus, for practicability and without loss of generality, we consider one explanation vector w.r.t. a specific class in this paper. We summarize the frequently used notations in Tab. 1 for reference.

Attack Model. The previous studies in ML privacy [36, 40, 47, 51] assume that the adversary can collect an auxiliary dataset \mathcal{X}_{aux} which follows the same underlying distribution of the target sample \mathbf{x} . For example, a bank can synthesize a real-world dataset after obtaining the feature names on the explanation report from its competitors. In this paper, we study two adversaries by relaxing the assumptions of previous studies. The *first* adversary follows the setting in [36, 40, 47, 51], i.e., the attacker aims to infer the private sample \mathbf{x} based on the associated explanation vector \mathbf{s} , an auxiliary dataset \mathcal{X}_{aux} and black-box access to the prediction model f on the MLaaS platforms: $\hat{\mathbf{x}} = \mathcal{A}_1(\mathbf{s}, \mathcal{X}_{aux}, f)$, where $\hat{\mathbf{x}}$ denotes the reconstructed values of \mathbf{x} . For the *second* adversary, we use a more restricted but practical setting that the attacker has only black-box access to the ML services, and no background knowledge about the target sample \mathbf{x} is available: $\hat{\mathbf{x}} = \mathcal{A}_2(\mathbf{s}, f)$.

4 FEATURE INFERENCE ATTACKS BASED ON SHAPLEY VALUES

In this section, we present the proposed feature inference attacks under different settings. Based on an auxiliary dataset, the first adversary can first learn a regression model with empirical risk minimization, then map the explanation vector \mathbf{s} into the private feature space and accordingly obtain the estimation $\hat{\mathbf{x}}$ of \mathbf{x} . With only black-box access to the MLaaS platform, the second adversary can accurately reconstruct the important features of \mathbf{x} by exploiting the linearity of local model decision boundaries.

4.1 Adversary 1: Feature Inference Attack with an Auxiliary Dataset

4.1.1 Motivation. Given an input sample $\mathbf{x} = \{x_i\}_{i=1}^n$, a reference sample \mathbf{x}^0 and the access to a black-box model f , we can simplify the computation of Shapley value in Eq. 2 as follows:

$$s_i = h(x_i; x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, \mathbf{x}^0) + \epsilon, \quad (4)$$

where $h = g \circ f$ is the composition of the black-box model f and a linear transformation (i.e., the mathematical expectation) g on f . ϵ denotes the random noise caused by the sampling phase (Eq. 3).

Now we characterize the dependence between s_i and x_i via a popular information metric called *Mutual Information* [13, 14]:

$$I(x_i; s_i) = \int ds_i \int dx_i P(x_i, s_i) \left[\log \frac{P(x_i, s_i)}{P(x_i)P(s_i)} \right]. \quad (5)$$

To reach a perfect security level of Eq. 4, we need to make $I(x_i; s_i) = 0$ such that the adversary can not obtain any useful information of x_i by observing s_i [68].

For simplicity, we denote the variance and mean of a variable x by σ_x^2 and μ_x , respectively. We assume the noise ϵ caused by random sampling and the Shapley value s_i computed by Eq. 4 follow Gaussian distributions with mean 0 and some variances² because of the central limit theorem [5, 16, 34], thus

$$P(s_i) = \frac{1}{\sqrt{2\pi\sigma_{s_i}^2}} \exp \left[-\frac{s_i^2}{2\sigma_{s_i}^2} \right]. \quad (6)$$

Note that because ϵ is Gaussian noise and $\epsilon = s_i - h(x_i)$, then given x_i , s_i also follows a Gaussian distribution $s_i|x_i \sim N(\mu_{s_i|x_i}, \sigma_{s_i|x_i}^2)$, where $\mu_{s_i|x_i} = h(x_i)$ and $\sigma_{s_i|x_i} = \sigma_\epsilon$. Specifically,

$$P(s_i|x_i) = \frac{1}{\sqrt{2\pi\sigma_\epsilon^2}} \exp \left[-\frac{(s_i - h(x_i))^2}{2\sigma_\epsilon^2} \right]. \quad (7)$$

Then we can rewrite Eq. 5 as follows:

$$I(x_i; s_i) = \int ds_i \int dx_i P(x_i, s_i) \left[\log \frac{P(s_i|x_i)}{P(s_i)} \right] \quad (8)$$

$$= \frac{1}{\ln 2} \mathbb{E} \left[\ln \frac{\sqrt{2\pi\sigma_{s_i}^2}}{\sqrt{2\pi\sigma_\epsilon^2}} - \frac{(s_i - h(x_i))^2}{2\sigma_\epsilon^2} + \frac{s_i^2}{2\sigma_{s_i}^2} \right] \quad (9)$$

$$= \frac{1}{\ln 2} \mathbb{E} \left[\ln \sqrt{\frac{\sigma_{s_i}^2}{\sigma_\epsilon^2}} - \frac{1}{2} + \frac{1}{2} \right] \quad (10)$$

$$= \frac{1}{2} \mathbb{E} \left[\log \frac{\sigma_{s_i}^2}{\sigma_\epsilon^2} \right], \quad (11)$$

where Eq. 10 follows because for the second term of Eq. 9, $\epsilon = s_i - h(x_i)$ and $\mathbb{E}(\epsilon^2) = \sigma_\epsilon^2 + \mu_\epsilon = \sigma_\epsilon^2$, making the numerator cancel with the denominator, and so does for the third term.

Now to achieve a perfect level of security, we need to make $I(x_i; s_i) = 0$, i.e., $\sigma_{s_i}^2 = \sigma_\epsilon^2$ in Eq. 11. But to obtain a reasonable interpretability s_i of x_i , we need to minimize the noise component [13], which means that $\sigma_{s_i}^2 > \sigma_\epsilon^2$ is necessary from the perspective of practicability. In summary, an adversary can always infer useful information of x_i from the explanation report as long as the Shapley values are meaningful.

² We have conducted a χ^2 goodness-of-fit test on the Shapley values generated on the Google platform from two datasets: Bank [39] and Diabetes [55], where H_0 is "Gaussian density (theoretical) distribution can well fit the density distribution of the observed Shapley values". Test results show that for Shapley values generated from Bank and Diabetes, the divergence statistics are 3.84 and 2.48, and p values are 0.998 and 0.999. Therefore, under $p > 0.05$, we can not reject H_0 .

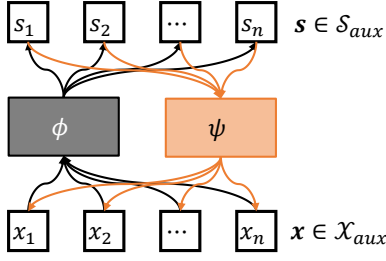


Figure 2: Overview of the attack with an auxiliary dataset. ψ is the attack model.

4.1.2 The Attack with Auxiliary Datasets. Previous studies, such as membership inference [47, 51] and property inference [24, 36], assume that the adversary owns an auxiliary dataset \mathcal{X}_{aux} following the same underlying distribution of the target samples. This assumption is reasonable because the competing companies own datasets that have the same features but different customer bases [29]. The users of the MLaaS platforms can also collect a dataset \mathcal{X}_{aux} by collusion [38]. In this section, we follow this assumption and show that the adversary can reconstruct the target samples by training a regression model on \mathcal{X}_{aux} .

Based on the analysis in Section 4.1.1, we know that the explanation vector s is related to the private sample x . Although the intensity of the noise ϵ is unknown, the adversary can still learn a model to reconstruct x by minimizing the noise component. Specifically, the adversary can send prediction queries for all $x_{aux} \in \mathcal{X}_{aux}$ to the MLaaS platform and obtain the corresponding explanation set \mathcal{S}_{aux} . Now we rewrite Eq. 4 as

$$s = \phi(x; x^0). \quad (12)$$

Before the adversary can learn a hypothesis $\psi : \mathcal{S}_{aux} \rightarrow \mathcal{X}_{aux}$ as the attack model, we need to ensure that the mapping $\phi : \mathcal{X}_{aux} \rightarrow \mathcal{S}_{aux}$ is bijective.

THEOREM 1. *Let $\mathcal{X}_{aux} \subseteq \mathbb{R}^n$ be an input dataset, $\mathcal{S}_{aux} \subseteq \mathbb{R}^n$ be its Shapley values with respect to a specific class computed based on a model f and a reference sample x^0 . The mapping $\phi : \mathcal{X}_{aux} \rightarrow \mathcal{S}_{aux}$ is bijective if $|\mathcal{X}_{aux}|$ is finite.*

PROOF. Because $|\mathcal{X}_{aux}| = |\mathcal{S}_{aux}|$, the proof that ϕ is a bijective function from \mathcal{X}_{aux} to \mathcal{S}_{aux} can be reduced to the proof that for any $s^i \in \mathcal{S}_{aux}$, $s^j \in \mathcal{S}_{aux}$ and $i \neq j$, $s^i \neq s^j$ holds, i.e., there are no repeating samples in \mathcal{S}_{aux} . Now let \mathcal{S} be the space of Shapley values, i.e., $\mathcal{S}_{aux} \subseteq \mathcal{S}$. The probability that there exists no repeating samples in \mathcal{S}_{aux} is:

$$P = \frac{|\mathcal{S}|!}{(|\mathcal{S}| - |\mathcal{S}_{aux}|)! |\mathcal{S}|^{|\mathcal{S}_{aux}|}}. \quad (13)$$

Because the Shapley value space \mathcal{S} is continuous, we have $|\mathcal{S}| \rightarrow \infty$, thus $P \rightarrow 1$ in Eq. 13. The theorem is proved. \square

Note that $\phi : \mathcal{X}_{aux} \rightarrow \mathcal{S}_{aux}$ is a bijective mapping does not mean ϕ is invertible from \mathcal{S} to \mathcal{X} . Nevertheless, the size of the auxiliary dataset $|\mathcal{X}_{aux}|$ is always finite in real-world applications, so the adversary can still learn a hypothesis $\psi : \mathcal{S}_{aux} \rightarrow \mathcal{X}_{aux}$ by

$\rho_{x_i, \hat{y}}$	0.035	0.078	0.210	0.148	0.358	0.602	0.257	0.50 0.25
ρ_{x_i, s_i}	0.048	0.081	0.119	0.233	0.693	0.520	0.188	
	Race	Gender	Age	Admission type	Discharge disposition	Admission source	Time in hospital	

Figure 3: The Pearson correlation coefficient ρ between the first seven features of the Diabetes dataset [55] and the model output \hat{y} as well as the corresponding explanations s . NN and Google Cloud [15] are used as the testing model and platform.

minimizing the empirical risk as shown in Fig. 2:

$$\min_{\psi \in \mathcal{H}} \frac{1}{|\mathcal{X}_{aux}|} \sum L(\psi(\mathcal{S}_{aux}), \mathcal{X}_{aux}) + \Omega(\psi), \quad (14)$$

where L denotes the loss function, \mathcal{H} denotes a set of hypotheses and Ω is the regularization term. After learning a model ψ , the adversary can use it to reconstruct a private sample x based on its Shapley values s : $\hat{x} = \psi(s)$. Note that the regularization term in Eq. 14 is significant to help the model focus on the information of x instead of the noise ϵ in s . The attack with an auxiliary dataset is summarized in Algorithm 1.

4.2 Adversary 2: Feature Inference Attack with Data Independence

4.2.1 Motivation. Suppose the adversary has no auxiliary datasets \mathcal{X}_{aux} , it will be impossible for the adversary to learn an attack model as discussed in Section 4.1 because the distributions of the target private features are unknown. A naive way to restore x from s is that the adversary generates a random dataset \mathcal{X}_{rand} , sends the samples in this dataset to the MLaaS platform separately and obtains the corresponding explanations \mathcal{S}_{rand} . Provided that the size of the random dataset is large enough, there will be some $s_{rand} \in \mathcal{S}_{rand}$ such that $\|s_{rand} - s\|_2 \leq \xi$ where ξ is a small threshold. The adversary can use the x_{rand} associated with s_{rand} as the estimation of x . But there are two problems in this method. First is that to improve the estimating accuracy, the adversary needs to randomly draw a large number of samples, e.g., $|\mathcal{X}_{rand}| > 10000$, from the possible feature space and send them to the MLaaS platform, which can be costly on the pay-per-query service and increase the risk of being detected and blocked by the service provider. Second is that although $\|s_{rand} - s\|_2 \leq \xi$ holds, the corresponding x_{rand} can be a false positive case in which $\|x_{rand} - x\|_2$ is large. Because as discussed in Theorem 1, the bijective relation only holds between \mathcal{X}_{rand} and \mathcal{S}_{rand} instead of between \mathcal{X} and \mathcal{S} .

In the experiments, we observe that the level of correlations between a feature x_i and the model output \hat{y} is closely consistent with the correlations between x_i and its explanation s_i . Fig. 3 shows an example. First, we train a neural network on a real-world Diabetes dataset [55] and deploy it to Google Cloud [15]. Then, we send queries of $x = \{x_i\}_{i=1}^n$ to the deployed model and obtain the model outputs \hat{y} and explanations $s = \{s_i\}_{i=1}^n$. We compute the Pearson correlation coefficients³ between the first seven features and \hat{y} as well as s and show them in Fig. 3, from which we observe that the

³ $\rho(x, y) = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y}$, where $\text{cov}(x, y)$ denotes the covariance between x and y [62].

Algorithm 1: The Attack with an Auxiliary Dataset

Input: the black-box model f , an auxiliary dataset \mathcal{X}_{aux} , a learning rate α , the target Shapley values \mathbf{s}

Output: The reconstructed private input $\hat{\mathbf{x}}$

```

1  $\mathcal{S}_{aux} \leftarrow \phi(\mathcal{X}_{aux}; f)$ ; // Send queries to the ML platform
2  $\theta_\psi \leftarrow \mathcal{N}(0, 1)$ ; // Initialize the attack model
3 for each epoch do
4   for each batch do
5      $loss \leftarrow 0$ ;
6      $B \leftarrow$  randomly select a batch of samples;
7     for  $j \in \{1, \dots, |B|\}$  do
8        $(\hat{\mathbf{x}}_{aux})^j \leftarrow \psi((s_{aux})^j; \theta_\psi)$ ;
9        $loss += L((\hat{\mathbf{x}}_{aux})^j, (\mathbf{x}_{aux})^j)$ ;
10     $\theta_\psi \leftarrow \theta_\psi - \alpha \cdot \nabla_{\theta_\psi} loss$ ; // Update the attack model
11  $\hat{\mathbf{x}} \leftarrow \psi(\mathbf{s}; \theta_\psi)$ ;
12 return  $\hat{\mathbf{x}}$ ;
```

values of $\rho_{x_i, \hat{y}}$ and ρ_{x_i, s_i} are similar for all x_i . Note that the Pearson correlation coefficient measures the linear correlation between variables [62], which inspires that to reduce the number of random queries and improve the estimation accuracies, we can exploit the local linearity of the decision boundaries of the black-box model f w.r.t. some important features x_i . Then by interpolation based on a small random dataset \mathcal{X}_{rand} , we can accurately reconstruct the values of these features from the corresponding Shapley values \mathbf{s} .

4.2.2 The Attack with A Random Dataset. In this attack, because the adversary has no information on the data distribution and feature interactions, different features in the random dataset \mathcal{X}_{rand} have to be drawn independently, i.e., the features are independent of each other in \mathcal{X}_{rand} , which means that the features of \mathbf{x} need to be restored separately from \mathbf{s} . To achieve this task, we introduce a generalized additive model (GAM) [35] to approximate the black-box model f :

$$f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n) + \epsilon_f, \quad (15)$$

where ϵ_f denotes the error caused by GAM approximation, and f_i denotes an unknown smooth univariate function. Note that GAM can theoretically minimize the Kullback-Leibler distance to the original function [35] and has been used for analyzing the data flow in neural networks [1, 66] and random forests [27]. Replace Eq. 15 into Eq. 2, we have

$$s_i = f_i(x_i) - f_i(x_i^0) + \epsilon_s, \quad (16)$$

where $f_i(x_i^0)$ is a constant, and ϵ_s denotes the sampling error. If s_i is linearly correlated with x_i (i.e., f_i is monotonic around x_i), we can first find a set of $\{(x_{rand})_i^j\}_{j=1}^k$ from the random dataset \mathcal{X}_{rand} with $|(s_{rand})_i^j - s_i| \leq \xi$ then estimate x_i by $\hat{x}_i = \frac{1}{k} \sum_{j=1}^k (x_{rand})_i^j$.

Bounding the estimation error. Now we theoretically analyze the estimation error of the interpolation-based attack method. Since the features are reconstructed independently, it suffices to focus on only one feature in the target Shapley values \mathbf{s} , say s_i . Let x_i be the private feature associated with s_i , X be a random variable whose Shapley value s_X is close enough to s_i , i.e., $|s_X - s_i| \leq \xi$, where ξ is a small threshold. Let $\{\tilde{x}^j\}_{j=1}^k$ be k independent sampling points of

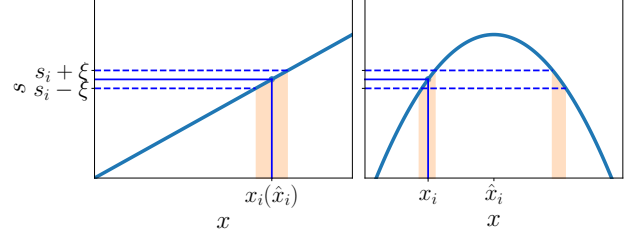


Figure 4: Examples of correlations between private features and the corresponding Shapley values.

X , and $\hat{x}_i = \frac{1}{k} \sum_j \tilde{x}^j$ be the empirical mean of these k points. Now by Chebyshev's inequality [60], we have

$$\Pr(|x_i - \mu| \leq u\sigma) \geq 1 - \frac{1}{u^2}, \quad (17)$$

where u denotes any real number with $u > 1$, μ and σ denotes the expectation and standard deviation of X , respectively. Because we use an empirical mean \hat{x}_i of X instead of its expectation μ to estimate x_i , we need to further bound the deviation between \hat{x}_i and μ via Hoeffding's inequality [61]:

$$\Pr(|\hat{x}_i - \mu| \leq w) \geq 1 - 2 \exp\left(-\frac{2w^2k}{(b-a)^2}\right), \quad (18)$$

where w denotes any real number with $w > 0$, $[a, b]$ denotes the range of the k random points, i.e., $a \leq \tilde{x}^j \leq b$ for all j . By combining Eq. 18 with Eq. 17, we have

$$\Pr(|x_i - \hat{x}_i| \leq u\sigma + w) \geq 1 - \frac{1}{u^2} - 2 \exp\left(\ln(u^2 - 1) - \frac{2w^2k}{(b-a)^2}\right) / u^2. \quad (19)$$

Note that $\sigma < \frac{b-a}{2}$. The result of Eq. 19 shows that when the range $[a, b]$ of the sampling points $\{\tilde{x}^j\}_{j=1}^k$ becomes smaller, the estimation \hat{x}_i becomes more accurate with a larger probability. For example, let $u = 2$, $w = 0.1$ and $k = 30$. If $b - a = 0.1$, we can reconstruct x_i with an error less than 0.2 in a probability of larger than 75%. And if $b - a = 0.5$, the estimation error becomes 0.6 in a probability of 61%.

We explain the intuition behind the error bound via Fig. 4, which depicts the possible correlations between x_i and s_i . The observation is that for important features, a slight change in their values will lead to a great difference in the model predictions and explanations (see the example in Fig. 3), which means that in general, these features are linearly correlated with their Shapley values, e.g., the left figure of Fig. 4. In this case, the range of estimations $[a, b]$ tends to be small, leading to an accurate reconstruction of these features. On the other hand, the less important features have little impact on the model predictions and explanations. This case corresponds to the right figure of Fig. 4, i.e., the feature values scattered in a large range can produce a similar Shapley value s_i . And accordingly, $\sigma_{s_i} \leq \sigma_\epsilon$ (Eq. 11) holds, making the estimation \hat{x}_i far from the ground truth x_i as the information of x_i contained in s_i is overridden by the approximation noise.

Algorithm 2 shows the attack details of adversary 2 without any background knowledge. First, we generate a random dataset \mathcal{X}_{rand} (line 2) and send it to the MLaaS platform to obtain an explanation set \mathcal{S}_{rand} (line 3). Then, we try to reconstruct the private features \mathbf{x} associated with the target Shapley values \mathbf{s} one by one (line 6-23).

Algorithm 2: The Attack with Data Independence

Input: the black-box model f , the threshold for sampling error ξ , the minimum number of interpolation samples m_C , the threshold for the range of estimation values τ , the target Shapley values s

Output: The reconstructed private input \hat{x}

```

1  $\hat{x} \leftarrow \emptyset$ ;
2  $\mathcal{X}_{rand} \leftarrow \text{Random Samples}$ ;
3  $\mathcal{S}_{rand} \leftarrow \phi(\mathcal{X}_{rand}; f)$ ; // Send queries to ML platform
4  $m \leftarrow |\mathcal{X}_{rand}|$ ;
5  $n \leftarrow |s|$ ;
6 for  $i = 1, 2, \dots, n$  do // For each feature
7    $s_t \leftarrow s_i$ ;
8    $D \leftarrow \emptyset$ ; // The distance set from  $s_t$ 
9   for  $j = 1, 2, \dots, m$  do
10     $dist \leftarrow \|s_t - (s_{rand})_i^j\|_2$ ;
11     $\tilde{x} \leftarrow (x_{rand})_i^j$ ;
12     $D \leftarrow D \cup (dist, \tilde{x})$ ;
13   Sort  $D$  on  $dist$  in an ascending order;
14    $C \leftarrow \emptyset$ ; // The candidate set of  $\hat{x}_i$ 
15   for  $j = 1, 2, \dots, m$  do
16     $(dist, \tilde{x}) \leftarrow D_j$ ;
17    if  $|C| < m_C$  or  $dist < \xi$  then // The Shapley error
18       $C \leftarrow C \cup \tilde{x}$ ;
19   if  $\max C - \min C > \tau$  then // The estimation error
20      $\hat{x}_i \leftarrow \perp$ ;
21   else
22      $\hat{x}_i \leftarrow \frac{1}{|C|} \sum C$ ;
23    $\hat{x} \leftarrow \hat{x} \cup \hat{x}_i$ ;
24 return  $\hat{x}$ ;

```

For a target feature x_i , first, we compute the distances between its s_j and the Shapley values of the i -th features in the random samples (line 9-12). After that, we sort the random features based on the Shapley value distances (line 13) and accordingly choose the candidate estimations of x_i whose Shapley value distances are less than a threshold ξ . Note that $\xi (> \epsilon_s)$ indicates the approximation error of Shapley sampling values (see Eq. 3 and 16) and is used to impose the estimation error bound in Eq. 19 on the real value of x_i . We use an empirical threshold m_C as the minimum size of the candidate set to reduce possible estimation bias. In the end, we only reconstruct those private features with value ranges of the candidate estimations less than a threshold τ (line 19). The complexity of Algorithm 2 is $O(nm \log m)$, where n denotes the number of target features and m denotes the size of the random dataset, i.e., $|\mathcal{X}_{rand}|$. It is worth noting that although the accuracy of estimation is limited by the sampling size m , in the experiments, we show that $m = 100$ suffices to reconstruct more than 30% features with high accuracies from the complicated models (e.g., NN).

4.3 Attack Generalization

It is worth noting that our attacks are designed based on the inherent limitations of Shapley values. Specifically, we can simplify the computation of Shapley values into three steps: first use an affine

transformation to generate a feature replacement dataset from x and x^0 , then feed the dataset into the target model, and finally use another affine transformation on the model outputs to obtain Shapley values. Because the mapping from private features to Shapley values is fixed, we can then propose the inverse mapping attack (attack 1) and the GAM transformation-based attack (attack 2) to reconstruct the inputs from their explanations.

On the contrary, other popular explanation methods, e.g., LIME [44] and DeepLIFT [53], focus on approximating the linear decision boundary around the target features via a set of heuristic rules, where the mappings between features and explanations among different samples are unstable and various [4], leading to little convergence of attack 1 and invalidating the GAM transformation and error bounds of attack 2. For justification, we compared the performance of attack 1 on Shapley values, LIME, and DeepLIFT tested on the Google Cloud platform [15], where the testing model and dataset are NN and Adult [19]. Note that attack 2 is not applicable to other explanation methods because the GAM transformation is based on the computational formula of Shapley values. For attack 1, the averaging ℓ_1 error of feature reconstruction tested on Shapley values is 0.0768, which is far better than 0.1878 on LIME, 0.1527 on DeepLIFT, and 0.1933 on the empirical random guess baseline (see Section 5.1). The conclusion is that the proposed attacks are targeted on Shapley values, and specific optimizations are needed to apply them to other heuristic explanation methods. We leave it as our future work.

5 EXPERIMENTS

5.1 Experimental Setting

The proposed algorithms are implemented in Python with *PyTorch*⁴, and all attack experiments are performed on a server equipped with Intel (R) Xeon (R) Gold 6240 CPU @ 2.60GHz×72 and 376GB RAM, running Ubuntu 18.04 LTS.

Explanation Platforms. To validate the effectiveness of the proposed attacks on real-world applications, we conduct experiments on three leading ML platforms: *Google Cloud AI* platform [15], *Microsoft Azure Machine Learning* [8] and *IBM Research Trusted AI* [2]. For the experiments on Google and Microsoft platforms, we first train models and deploy them in the cloud, then send prediction and explanation queries to the deployed models to obtain the corresponding explanation vectors. The experiments on the IBM platform are performed locally because a python package called *AI Explainability 360* [2] is available. In addition, to test the attack performance *w.r.t.* different sampling errors, we implement a *Vanilla* Shapley sampling method [56], because the explanatory API provided by Google only supports up to 50 permutations for sampling computation (e.g., $v \leq 50$ in Section 2.2) while the APIs provided by Microsoft and IBM are based on SHAP [31] which fails to provide a theoretical error bound for Shapley approximation. The experiments are performed on the Google platform by default unless otherwise specified.

Datasets. Six real-world datasets are used in the experiments: *Adult* [19] for predicting whether one's income will exceed 50000 per year, *Bank Marketing* [39] for predicting whether a client will

⁴<https://pytorch.org/>

Table 2: The Experimental Datasets.

Dataset	Instances	Classes	Features
Adult	48842	2	14
Bank marketing	45211	2	20
Credit card	30000	2	23
Diabetes 130-US hospitals	68509	3	24
IDA 2016 Challenge	76000	2	170
Insurance Company Benchmark	90000	2	85
Synthetic dataset×3	100000	5	12

subscribe a deposit, *Credit Card Clients* [67] for predicting whether a client is credible based on payment recordings, *Diabetes 130-US Hospitals* [55] for predicting whether a patient will be readmitted, *IDA 2016 Challenge* [19] for predicting whether a component failure in a truck is related to the air pressure system, and *Insurance Company Benchmark* [58] for predicting whether a customer has a caravan insurance policy. In addition, we use three synthetic datasets generated via the *sklearn*⁵ library to evaluate the impact of correlations between input features and model outputs on the attack performance. The feature values in all datasets are normalized into $[0, 1]$ for ease of comparison [32, 33]. The details of these datasets are summarized in Tab. 2. For each dataset \mathcal{X} , we randomly split it into three portions: a dataset \mathcal{X}_{train} with 60% samples for model training, a dataset \mathcal{X}_{test} with 20% samples for model testing and acting as the auxiliary dataset collected by the attacker, a dataset \mathcal{X}_{val} with the rest samples for attack validation.

Black-Box Models. Four types of models are used in the experiments: Neural Network (NN), Random Forest (RF), Gradient Boosted Decision Trees (GBDT) and Support-Vector machine (SVM) with the Radial Basis Function (RBF) kernel. The NN model consists of an input layer with n dimensions, an output layer with c dimensions, and two hidden layers with $2n$ neurons per layer. The number of trees and maximum depth for each tree are set to $\{100, 5\}$ for RF and $\{100, 3\}$ for GBDT. The models deployed on Google Cloud are trained with TensorFlow⁶, while the models on other platforms are trained with Pytorch (NN) and sklearn (RF, GBDT, SVM). The initial states of models and training hyper-parameters among different platforms are set to be the same for consistency.

Attack Algorithms. For adversary 1, we train a multi-layer perceptron as the attack model ψ , which consists of one input layer with n dimensions, one output layer with n dimensions, and one hidden layer with $4n$ dimensions. Sigmoid is used as the activation function of both the hidden and output layers. The reason is that MLPs with Sigmoid activation can approximate any models with a desired accuracy [28] and are suitable for learning a pseudo inverse mapping of the black-box models with unknown structures. For adversary 2, we set the minimum number of the interpolation samples m_C in Algorithm 2 to be 30, and the threshold τ for the range of estimation values to be 0.4. For the threshold of sampling error ξ , considering that the intensities of Shapley values in different datasets are also different, we determine ξ based on the tight range $r = \max S - \min S$ of the available Shapley values S for different

datasets and set it to be $\frac{\tau}{5}$ by default. Note that S can be generated from \mathcal{X}_{aux} or \mathcal{X}_{rand} .

Metrics. We use the Mean Absolute Error (MAE, i.e., ℓ_1 loss) to measure the distances between the reconstructed private inputs \hat{x} and the ground truth inputs x :

$$\ell_1(\hat{x}, x) = \frac{1}{mn} \sum_{j=1}^m \sum_{i=1}^n |\hat{x}_i^j - x_i^j|, \quad (20)$$

where m is the number of samples in the validation dataset \mathcal{X}_{val} and n is the number of features. Besides, we use the success rate (SR) to evaluate the percentage of features that can be successfully reconstructed by attack 2:

$$SR = \frac{|\hat{\mathcal{X}}_{val} \neq \perp|}{mn}, \quad (21)$$

where $|\hat{\mathcal{X}}_{val} \neq \perp|$ represents the total number of features that are successfully restored in \mathcal{X}_{val} . For each experiment setting, we first randomly select 10 independent samples from \mathcal{X}_{train} as the reference samples x^0 (see Section 2.2), then accordingly conduct 10 independent experiments and report the averaging results.

Baselines. For adversary 1, since the attacker owns an auxiliary dataset that shares the same underlying distribution $P_{\mathcal{X}}$ of the private inputs x , we use a random guess baseline based on the empirical distribution (called RG-E) that estimates the values of x by randomly sampling a vector $\tilde{x} \sim P_{\mathcal{X}}$. For adversary 2, since the attacker does not know the distribution of the private inputs, we use two random guess baselines to estimate the values of x : a Uniform distribution $U(0, 1)$ (called RG-U) and a Gaussian distribution $N(0.5, 0.25^2)$ (called RG-N). Note that RG-N can generate random values falling in $[0, 1]$ with a probability of at least 95% [32].

5.2 Attack Performance w.r.t. Different Numbers of Queries

In both attacks, the adversary sends queries to the MLaaS platforms in a pay-per-query pattern. If the number of queries needed is large, the cost for the adversary could be unacceptable. In this section, we evaluate the performance of attack 1 and 2 under different number of queries. The number of permutations used to compute Shapley sampling values is set to 50. We first vary the number of queries, i.e., the size of the auxiliary dataset $|\mathcal{X}_{aux}|$ or the random dataset $|\mathcal{X}_{rand}|$, in $\{100, 200, 400, 800, 1600\}$, then test the performance of attack 1 and 2, respectively. Fig. 5(a)-5(d) show the ℓ_1 losses of attack 1. Fig. 5(e)-5(h) and Tab. 3 show the ℓ_1 losses and success rates of attack 2. We have two observations from the results.

First, with the increase of queries, the estimation errors of adversary 1 and 2 slightly decrease (Fig. 5(a)-5(h)), and the success rates of adversary 2 generally increase (Tab. 3). The reason is straightforward: in adversary 1, the attack model ψ can obtain a better generalization performance when the training dataset \mathcal{X}_{aux} becomes larger; for adversary 2, a finer sampling of random points can help produce smaller value ranges of the candidate estimations on the private features, thus reducing the estimation error bound (see Eq. 19) and leading to better attack accuracies. Meanwhile, the improvements in the attack accuracies are limited. For example, when varying the number of queries from 100 to 1600, the losses only decrease by 0.03 in adversary 1 and 0.05 in adversary 2 on the

⁵<https://scikit-learn.org/stable/>

⁶<https://www.tensorflow.org/>

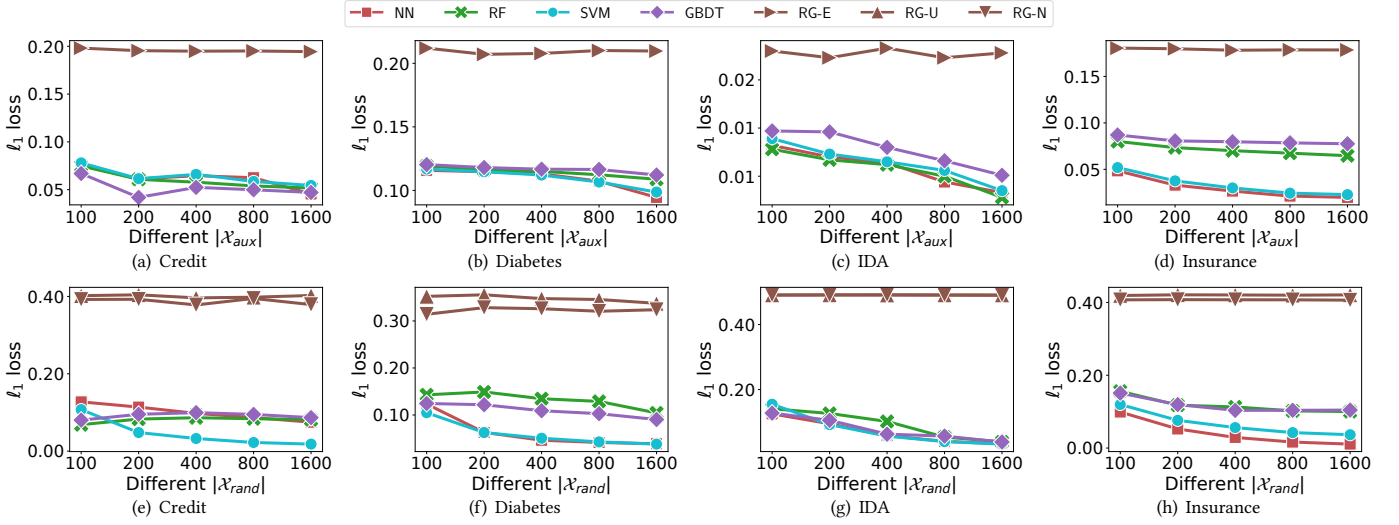


Figure 5: (a)-(d): the performance of attack 1 w.r.t. different sizes of auxiliary datasets; (e)-(h): the performance of attack 2 w.r.t. different sizes of random datasets.

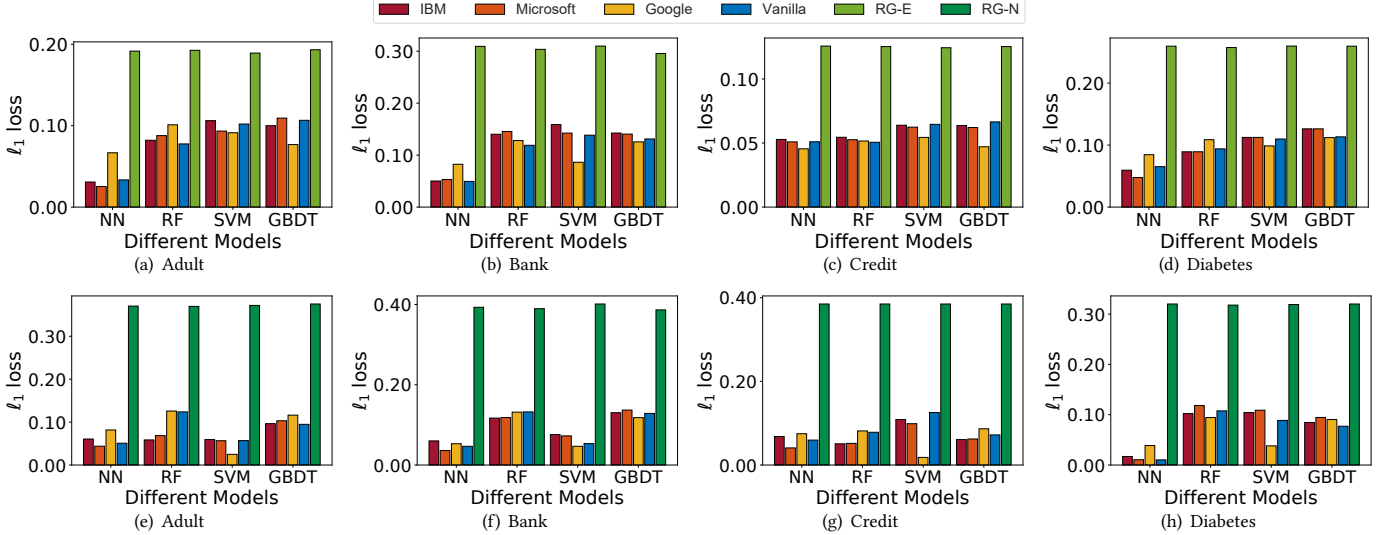


Figure 6: The performance of the proposed attacks tested on different platforms: (a)-(d) attack 1, (d)-(e) attack 2.

NN model of the Credit dataset. This means that a small number of queries (typically 100) suffice to initialize successful attacks.

Second, the proposed attacks perform worse on the ensemble models (RF and GBDT) than on NN and kernel SVM, e.g., Fig. 5(d), 5(f) and 5(h). For adversary 1, the key idea is to learn the inverse mapping of the black-box models with a three-layer MLP. Each of the ensemble models used in our experiments consists of 100 decision trees. Compared to NN and kernel SVM, the decision boundaries of the ensemble models are more complicated and difficult to be accurately imitated by a simple neural network [9, 43]. Therefore, the generalization performances of the attack models trained on ensemble models become worse than on other models. For adversary 2, the decision boundaries of RF and GBDT are more complicated than that of NN and SVM, leading to fewer local linearities between features and Shapley values, and thus larger estimation

errors. As discussed in Section 2.1, each tree in the ensemble models is typically trained on a set of randomly selected samples [46]. This method can prevent overfitting meanwhile introducing more randomness to the decision boundaries. The decision boundaries of NN models, on the other hand, are relatively smoother. In addition, the linearities between the important features and model outputs can be effectively preserved in NN, because the activation functions intercept little information of these features.

5.3 Attack Performance w.r.t. Different MLaaS Platforms

In this section, we evaluate the proposed attacks on different MLaaS explanation platforms. Because the methods for computing Shapley sampling values among different platforms are different, we implement a Vanilla Shapley sampling method based on [56] for

Table 3: The success rates of attack 2 w.r.t. different $|\mathcal{X}_{rand}|$.

Dataset	Model	Different $ \mathcal{X}_{rand} $				
		100	200	400	800	1600
Credit	NN	0.5871	0.7220	0.7520	0.7928	0.7961
	RF	0.4515	0.4976	0.5307	0.5833	0.5858
	SVM	0.5575	0.7607	0.9404	0.9549	0.9552
	GBDT	0.4680	0.6721	0.6833	0.6984	0.7486
Diabetes	NN	0.3911	0.6663	0.7509	0.7947	0.8262
	RF	0.2798	0.3368	0.3830	0.4113	0.4469
	SVM	0.3763	0.5429	0.6826	0.7241	0.6975
	GBDT	0.3151	0.4176	0.4389	0.5043	0.5361
IDA	NN	0.7835	0.9896	0.9972	0.9972	0.9973
	RF	0.3114	0.3464	0.3763	0.4306	0.4857
	SVM	0.8965	0.9614	0.9671	0.9581	0.9638
	GBDT	0.2854	0.3171	0.3463	0.3515	0.3849
Insurance	NN	0.9523	0.9684	0.9708	0.9688	0.9696
	RF	0.3194	0.3339	0.3833	0.4399	0.4518
	SVM	0.6054	0.6764	0.7335	0.7529	0.7539
	GBDT	0.3043	0.3072	0.3176	0.3268	0.3446

comparison. The number of permutations sampled for computing Shapley values is set to 50 in both the Vanilla method and Google Cloud platform. Note that this parameter is not supported in the Microsoft Azure and IBM aix360 platforms because their explanation methods are based on SHAP [31], which computes Shapley values via a heuristic regression method instead of sampling permutations. The numbers of queries in both of the adversaries are set to 1000. The Shapley sampling error ξ in adversary 2 (Algorithm 2) is empirically set to $\frac{r}{5}$ among all platforms for comparison. Fig. 6(a)-6(d) and 6(e)-6(h) show the performance of adversary 1 and 2, respectively. Tab. 4 shows the success rates of adversary 2 corresponding to Fig. 6(e)-6(h).

From Fig. 6(a)-6(d) we observe that the attacks performed on Microsoft and IBM platforms achieve similar performance, which is expectable because these two platforms compute Shapley values via the same regression method. We also observe slight differences between the attacks performed on Vanilla and Google platforms although both of their implementations are based on the same Shapley sampling method [56]. Note that for efficiency, the maximum number of sampling permutations in Google Cloud is fixed to 50, which greatly reduces the computation costs from $O(2^n)$ to $O(50)$, meanwhile producing relatively large sampling error. As discussed in Section 4.1, a large error can override the information of private features in the Shapley values, leading to unstable reconstructions (see Eq. 11). Nevertheless, the proposed attacks can still reconstruct the private features with high accuracy.

From Tab. 4, we observe that the success rates of adversary 2 performed on IBM and Microsoft platforms are lower than the success rates on the Vanilla method and Google Cloud. Considering that SHAP [31] computes Shapley values heuristically without providing a theoretical approximation bound, the real sampling errors computed by IBM and Microsoft could be large. Although the parameter ξ in Algorithm 2 can guarantee that the real features fall into the estimation range $[a, b]$, the reconstructions on IBM and Microsoft could still fall in cases where the real Shapley sampling error is far larger than ξ . But in Tab. 4, we observe that the success rates of adversary 2 performed on IBM and Microsoft are at least 30%, which demonstrates the effectiveness of the proposed attacks.

Table 4: The success rates of attack 2 w.r.t. different platforms.

Dataset	Model	Different Platforms			
		IBM	Microsoft	Google	Vanilla
Adult	NN	0.9119	0.9066	0.9229	0.9788
	RF	0.5207	0.5221	0.5764	0.6181
	SVM	0.5209	0.5192	0.6964	0.6460
	GBDT	0.4086	0.4054	0.4337	0.6426
Bank	NN	0.8808	0.9301	0.9803	0.9807
	RF	0.4133	0.4241	0.4857	0.4954
	SVM	0.4699	0.4561	0.6467	0.6396
	GBDT	0.3404	0.3384	0.5993	0.4774
Credit	NN	0.6405	0.6913	0.7961	0.8784
	RF	0.3913	0.4018	0.5858	0.4858
	SVM	0.4642	0.4759	0.6552	0.6295
	GBDT	0.3787	0.3389	0.4486	0.4595
Diabetes	NN	0.8863	0.9999	0.8262	1.0000
	RF	0.4733	0.4843	0.5469	0.5598
	SVM	0.5079	0.5349	0.6975	0.6027
	GBDT	0.4111	0.4011	0.5361	0.5058

5.4 Attack Performance w.r.t. Different Feature Importance

As discussed in Section 4.2, the important features can be more accurately reconstructed than the less important features. In this section, instead of averaging the reconstruction loss over all features, we give the attack loss per feature and dissect the connections between attack accuracies and feature importance.

Three synthesis datasets with 12 features are used for the experiments (see Tab. 2 for the details). To generate the features of different importance, we first randomly draw five point clusters around five vertices of a three-dimension cube and label each cluster with a unique class. The three features of these random three-dimensional points are called the key features. After that, we generate n_r redundant features by randomly and linearly combining the key features. The rest $12 - 3 - n_r$ features are generated from random noises. The key features and redundant features are called important features. We vary the percentages of important features in $\{25\%, 50\%, 75\%\}$ and accordingly generate three synthesis datasets. In addition, two real-world datasets, Credit and Diabetes, are also used for justification. The importance of feature x_i is defined as the Mean Absolute Correlation Coefficients (MACC) between x_i and model outputs $\hat{\mathbf{y}}$:

$$MACC(x_i, \hat{\mathbf{y}}) = \frac{1}{c} \sum_{j=1}^c abs(\rho(x_i, \hat{y}_j)), \quad (22)$$

where $\hat{\mathbf{y}} = \{\hat{y}_j\}_{j=1}^c$ denotes the model outputs and ρ is the Pearson correlation coefficient [27, 62]. Note that MACC measures the averaging linear correlation between x_i and $\hat{\mathbf{y}}$. A larger MACC indicates that the change of x_i can produce larger variations in $\hat{\mathbf{y}}$. We use the absolute values of correlation coefficients because we focus on the intensities instead of the directions of the correlations.

We first train an NN model on Google Cloud, then compute the MACCs of different features in \mathcal{X}_{val} and show them in Fig. 7(a). We then accordingly perform the proposed attacks on these models and show the ℓ_1 loss per feature of adversary 1 in Fig. 7(b). The losses and success rates per feature of adversary 2 follow a similar pattern and thus are omitted in this paper. From Fig. 7, we can see that the more important features (deeper colors in Fig. 7(a)) will be reconstructed

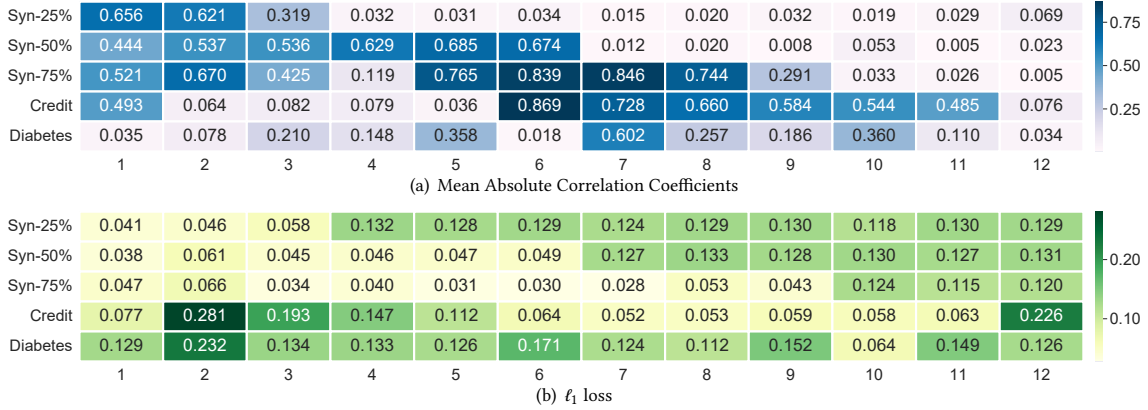


Figure 7: The (a) MACCs and (b) ℓ_1 losses of attack 1 w.r.t. different features in different datasets. The horizontal axes denote all features in the synthesis datasets and the first 12 features in Credit and Diabetes.

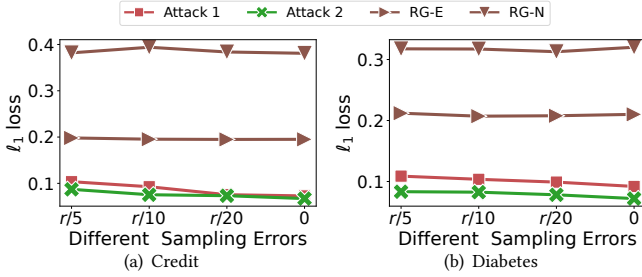


Figure 8: The performance of attack 1 & 2 w.r.t. different sampling errors.

with lower errors (lighter colors in Fig. 7(b)). This observation is consistent with the motivations of model interpretability methods: if a feature is more important, the variations of its value can cause a greater impact on the model outputs, thus producing a greater value on the model explanations. But accordingly, its Shapley value will contain more information of this feature and can be accurately reconstructed in a larger probability.

5.5 Attack Performance w.r.t. Different Shapley Sampling Errors

As discussed in Section 4.1 and 4.2, the proposed attacks can be applied to the explanation methods with unknown sampling errors. To evaluate the robustness of the proposed algorithms, we first vary the Shapley sampling errors ϵ_s in $\{r/5, r/10, r/20, 0\}$ and accordingly generate Shapley values with different errors via the Vanilla method, then test the attack performance on these explanations. The δ in Eq. 3 is set to 0.1. The test model is GBDT. We empirically set ξ in Algorithm 2 to $\frac{r}{5}$ in all experiments, which corresponds to the real-world scenario that the adversary has no information about the sampling error. The numbers of queries in both attacks are set to 1000. The attack losses of adversary 1 and 2 on Credit are shown in Fig. 8(a), with the success rates $\{0.4685, 0.4689, 0.4758, 0.4824\}$ of adversary 2. The losses on Diabetes are shown in Fig. 8(b), with the success rates $\{0.4730, 0.4921, 0.4955, 0.5057\}$ of adversary 2.

We make two observations from the results. First, the reconstruction accuracies of the proposed attacks slightly improve as the sampling errors decrease. This is reasonable because with the

decrease of random noises, the information of the private features contained in Shapley values will increase, which is beneficial to the reconstruction algorithms. But the improvement is slight, demonstrating the robustness and efficiency of the proposed attacks even under large sampling errors. For justification, only 37 random permutations of features are needed to achieve the sampling error of $r/5$, whereas 2^n permutations are needed to achieve zero sampling errors. Second, the losses of adversary 1 are slightly larger than the losses of adversary 2. The reason is that adversary 2 can only reconstruct roughly 50% important features, while adversary 1 can reconstruct all features, among which the less important features contain more noise and thus reduce the overall attack accuracies of adversary 1. Nevertheless, both attacks can achieve far better accuracies than the random guess baselines.

6 COUNTERMEASURES

Reduce the Mutual Information between Inputs and Shapley Values. As discussed in Section 4.1, minimizing the mutual information $I(x_i; s_i)$ can help protect the privacy of \mathbf{x} , which has been used as a defense mechanism in previous works [59]. Now we revisit the mutual information between x_i and s_i :

$$I(x_i; s_i) = H(s_i) - H(s_i|x_i) \stackrel{(a)}{\approx} H(s_i), \quad (23)$$

where $H(s_i)$ denotes the entropy of s_i , and (a) follows because s_i is a function of x_i (see Eq. 2). A way to reduce $I(x_i; s_i)$, i.e., $H(s_i)$, is quantizing the Shapley values. Specifically, we can first force the Shapley values of each feature to be one of 5/10/20 different values, respectively, then evaluate the impact of this quantization strategy on the accuracies of the proposed attacks. The results are shown in Fig. 9(a)-9(d). We observe that the quantization strategy has little impact on the performance of attack 1. Let k denote the number of values defined in quantization, the space size of the target Shapley vector \mathbf{s} is k^n , which could be far larger than $|\mathcal{S}_{aux}|$, indicating that the bijective relation between \mathcal{X}_{aux} and \mathcal{S}_{aux} still holds in a great probability (Theorem 1). Meanwhile, the quantization strategy can reduce the accuracies and success rates of attack 2, which is expected because the range of the candidate estimations for a feature is enlarged and thus leading to larger estimation errors (Eq. 19). Note that quantization can harm the utility of Shapley

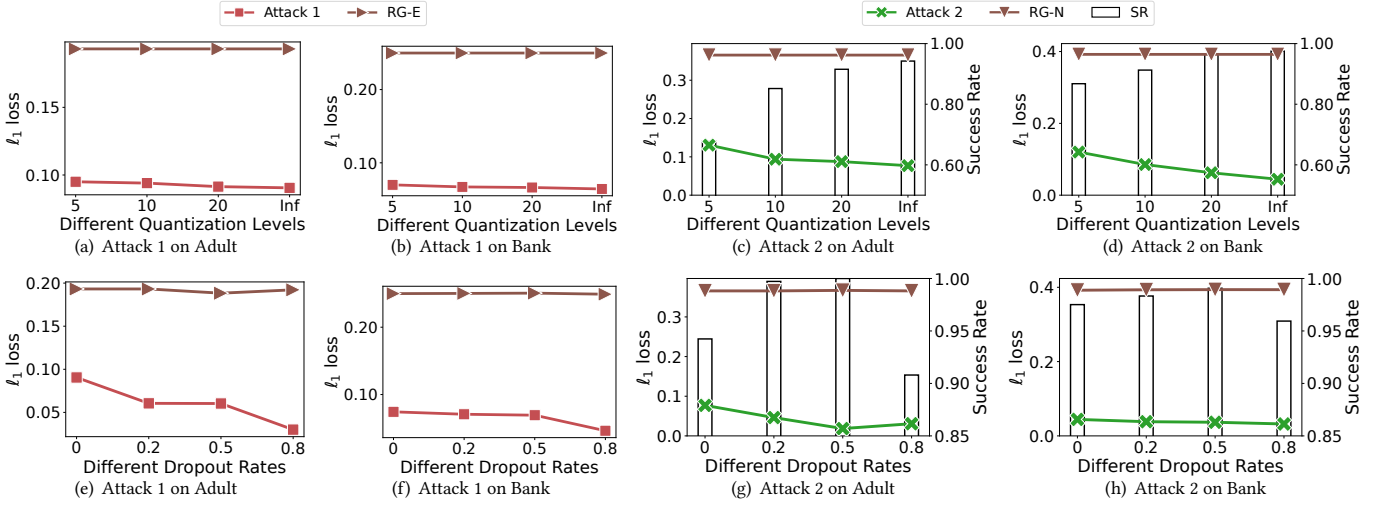


Figure 9: The performance of attack 1 & 2 tested on (a)-(d) different quantization levels and (e)-(h) different dropout rates. The test model and platform are NN and Google Cloud. The number of queries is set to 1600.

values, because two different input samples may produce the same explanation. Developing the privacy-preserving interpretability methods that achieve a good utility-privacy trade-off could be a promising direction in future studies.

Regularization. Previous studies [12, 32, 36, 47, 47, 57] demonstrate that regularization techniques, including dropout and ensemble learning, can prevent the models from memorizing private inputs and thus mitigate the risks of information leakage. In Section 5.2, we discuss the attack performance on different models and conclude that ensemble models can reduce the attack accuracies to a limited degree. In this section, we discuss the impact of the dropout strategy [54] in neural networks on the accuracies of the proposed attacks. In the experiments, we first add a dropout layer after each hidden layer and vary the dropout rate in $\{0.2, 0.5, 0.8\}$, then evaluate the corresponding attack performance. From Fig. 9(e)-9(h), we make an interesting observation that instead of harming the attack accuracies, the Dropout strategy improves the attack accuracies, which is entirely different from previous studies [12, 32, 36, 47]. The reason is that Dropout prevents NN models from overfitting by smoothing the decision boundaries during training, which provides more advantages to the proposed attacks. Note that from Fig. 9(g) and 9(h), we see the success rates of attack 2 drop when the dropout rate is 0.8. The reason is that under this dropout rate, the NN models are underfitting with decision boundaries containing randomness produced in the parameter initialization, thus reducing the linearity between inputs and outputs. However, this randomness causes little impact on attack 1, because the MLP attack model used in attack 1 can learn different decision boundaries with a desired accuracy [28].

Reduce the Dimension of Shapley Values. Because the number of Shapley values *w.r.t.* a class is equal to the number of input features (see Fig. 1), one intuitive defense is to reduce the number of Shapley values released. Note that the variance of Shapley values, instead of the value intensity, indicates the importance of features (see Eq. 11), thus simply releasing the largest k Shapley values may harm the utility of explanations because the important features in some samples may have small Shapley values. Instead,

we can empirically choose the top k important features based on their Shapley variances and only release the Shapley values of these k features as a defense. It is worth noting that this defense method causes little impact on the accuracy of Attack 2 because different features are reconstructed independently based on the corresponding Shapley values (Algorithm 2). Therefore, we mainly test the performance of Attack 1 under this defense. In the experiment, we only release k Shapley values and recover the k corresponding features. The experimental setting is the same as in Fig. 9. When the numbers of released Shapley values vary in $\{20\%n, 50\%n, n\}$, the averaging ℓ_1 losses of Adult and Bank are $\{0.0824, 0.0909, 0.0905\}$ and $\{0.0546, 0.0619, 0.0643\}$, respectively. The results are consistent with our analysis: the Shapley values of more important features leak more private information (Section 4.1.1) since the models tend to build linear connections between these features and their predictions (Section 4.2.2). In summary, releasing only k Shapley values can reduce the success rates of the proposed attacks to at most k/n but impact little on the accuracy of the recovered features.

Differential Privacy (DP). DP [65] is a state-of-the-art privacy-preserving mechanism that can protect the private inputs via a rigorous theoretical guarantee. By requiring that the change of an arbitrary sample in the input dataset should not be reflected in the outputs, DP can prevent the adversaries from inferring the information of private inputs via observing the outputs. However, DP does not apply to local interpretability methods and thus can not be employed as a defense against the proposed attacks. Specifically, for two arbitrary private samples x^i and x^j with $x^i \neq x^j$, local interpretability methods compute two explanations s^i and s^j . To satisfy the requirements of DP, s^i and s^j should be randomized to be indistinguishable, which harms the utility of Shapley values since different samples will produce nearly the same explanations. Therefore, DP is inapplicable to the current problem.

7 RELATED WORK

Model Interpretability. The lack of theoretical techniques for tracking data flows in complicated black-box models fosters plenty

of studies on model interpretability, which aims to explain how these models produce their predictions [45]. The model interpretability methods can be classified into two categories: *global interpretability* for computing the global importance of different features [17, 18, 20], and *local interpretability* for estimating the importance of features in a target sample [31, 44, 53]. In this paper, we focus on the local methods, which can be further classified into Shapley value-based methods and heuristic methods. The *heuristic methods* aim to linearize the model decision boundaries near the target samples and use the linear weights of different features as their importance values, e.g., LIME [44] and gradient-based methods [4, 49, 53]. Although the heuristic methods are computationally efficient, few theoretical analyses are provided to justify their robustness and error bounds. On the other hand, *Shapley value-based methods* [13, 14, 17, 18, 31, 34, 56] can theoretically provide the desirable properties for ML explanations and are thus adopted by the leading MLaaS platforms for explaining tabular data [2, 8, 15].

Attack Algorithms on Machine Learning. Many studies have revealed the privacy risks suffered by machine learning models, such as membership inference [40, 47, 51, 52], property inference [7, 24, 36], model extraction [3, 37, 57] and feature inference [21, 32, 63, 70]. *Membership inference* aims to determine whether a specific sample is in the training dataset or not, and *property inference* aims to restore the statistics of training datasets. Because these two types of attacks are supervised tasks, an auxiliary dataset is necessary for fulfilling the training of attack models. *Model extraction* aims to reconstruct the parameters of black-box models from their predictions. Nevertheless, few studies have considered the privacy risks of model interpretability methods. Although some newest studies focus on membership inference [51] or neural network extraction [3, 37] based on the local explanations produced by heuristic interpretability methods, their settings and tasks are entirely different from ours. In this paper, we focus on *feature inference* attacks on Shapley values, which are more practical and impactful to the real-world applications. To our best knowledge, this is also the first paper that considers the privacy risks in Shapley values.

Defense Mechanisms against Current Attacks. Since the efficiency of current attacks relies on the memorization nature of ML models [12, 32], most defense mechanisms focus on preventing models from memorizing sensitive data via noise injections. For example, using a lower granularity of model outputs can reduce the information of private features contained in them and accordingly decrease the accuracies of membership inference [52] and model extraction [57]. The regularization methods, such as dropout in NN [12, 32, 36, 47] and ensemble learning [47, 57], can reduce the information of training datasets memorized by the model. Differential privacy [65] is one of the most popular defense mechanisms because it provides a rigorous theoretical guarantee of protecting the private model inputs [12, 32, 36, 57]. However, these defenses can not apply to the proposed attacks, because we focus on reconstructing the private inputs in the prediction phase which have never been seen and thus memorized by the model. [59] proposes a defense mechanism by reducing the mutual information between the plaintext outputs and private data, which is similar in spirit to our analysis. However, the method in [59] can not be directly used in our setting. First, [59] aims to change the learning objectives via

mutual information during model training, whereas the model is trained and fixed in the computation of Shapley values. Second, the mutual information for privacy preserving in [59] is loosely related to the primary task and thus can be minimized, whereas in our case, minimizing the mutual information between Shapley values and features can harm the utility of explanations. Nevertheless, [59] still provides some insights on mitigating the proposed attacks.

8 DISCUSSION

The Choice of Datasets. As shown in Fig. 5 and Tab. 3, different choices of experimental datasets impact little on the attack performance. First, the number of samples in the training datasets has no relation to the proposed attacks since we focus on reconstructing private features in the prediction phase. Then, the number of classes also has limited impact on the proposed attacks because we use only one explanation vector *w.r.t.* the class with the highest confidence score (see Section 3). In addition, using larger feature sets may reduce the success rates of Attack 2, which, however, should be attributed to the model behaviors, because Attack 2 can only reconstruct those features that have relatively large influence on the model outputs (see Section 4.2.2).

The Influencing Factors of the Attacks. Our experiments reveal two main factors that impact the attack performance. First, from the results of Section 5.2, we conclude that in general, $NN > SVM > RF > GBDT$ *w.r.t.* the vulnerability to attacks. NN is most vulnerable because the non-linear activations, e.g., ReLU and Sigmoid, can behave in very linear ways for the important features [25], leading to a linear correlation between features and their Shapley values. Second, the results in Section 5.4 show that the important features, which have close relations with model outputs, are more vulnerable to the proposed attacks. Eq. 11 also shows that the variances of Shapley values corresponding to important features are larger than that of less important features. These results provide some insights that can be utilized by future studies to design defenses against privacy attacks on Shapley values.

9 CONCLUSION

In this paper, we focus on the privacy risks suffered in the Shapley value-based model explanation methods. We first analyze the connections between private model inputs and their Shapley values in an information-theoretical perspective, then accordingly propose two feature inference attacks under different settings. Extensive experiments performed on three leading MLaaS platforms demonstrate the effectiveness and robustness of the proposed attacks, highlighting the necessity of developing privacy-preserving model interpretability methods in future studies.

ACKNOWLEDGMENTS

We would like to thank Yuncheng Wu, Xiangqi Zhu, and Xiaochen Zhu for helpful discussions and feedback. This work was supported by the Ministry of Education, Singapore (Number MOE2018-T2-2-091) and A*STAR, Singapore (Number A19E3b0099). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the funding agencies.

REFERENCES

- [1] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Benjamin J. Lengerich, Rich Caruana, and Geoffrey E. Hinton. 2021. Neural Additive Models: Interpretable Machine Learning with Neural Nets. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6–14, 2021, virtual*. 4699–4711.
- [2] IBM Research Trusted AI. 2022. AI Explainability 360 - Resources. <http://aix360.mybluemix.net/resources>. Online; accessed 24-March-2022.
- [3] Ulrich Aivodji, Alexandre Bolot, and Sébastien Gambs. 2020. Model extraction from counterfactual explanations. *CoRR abs/2009.01884* (2020). [arXiv:2009.01884](https://arxiv.org/abs/2009.01884) <https://arxiv.org/abs/2009.01884>
- [4] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- [5] Marco Ancona, Cengiz Öztireli, and Markus H. Gross. 2019. Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA*, Vol. 97. PMLR, 272–281.
- [6] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra M. Josilovic, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John T. Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. 2019. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. *CoRR abs/1909.03012* (2019). [arXiv:1909.03012](https://arxiv.org/abs/1909.03012) <https://arxiv.org/abs/1909.03012>
- [7] Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. 2015. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks* 10, 3 (2015), 137–150.
- [8] Microsoft Azure. 2022. Model interpretability in Azure Machine Learning. <https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability>. Online; accessed 24-March-2022.
- [9] Gérard Biau, Erwan Scornet, and Johannes Welbl. 2016. Neural Random Forests. *CoRR abs/1604.07143* (2016). [arXiv:1604.07143](https://arxiv.org/abs/1604.07143) <https://arxiv.org/abs/1604.07143>
- [10] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27–29, 1992*. ACM, 144–152.
- [11] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.
- [12] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14–16, 2019*. USENIX Association, 267–284.
- [13] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10–15, 2018*, Vol. 80. PMLR, 882–891.
- [14] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2019. L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*. OpenReview.net.
- [15] Google Cloud. 2022. Introduction to AI Explanations for AI Platform. <https://cloud.google.com/ai-platform/prediction/docs/ai-explanations/overview>. Online; accessed 24-March-2022.
- [16] Ian Covert and Su-In Lee. 2021. Improving KernelSHAP: Practical Shapley Value Estimation Using Linear Regression. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13–15, 2021, Virtual Event*, Vol. 130. PMLR, 3457–3465.
- [17] Ian Covert, Scott M. Lundberg, and Su-In Lee. 2020. Understanding Global Feature Contributions With Additive Importance Measures. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual*. 17212–17223.
- [18] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22–26, 2016*. IEEE Computer Society, 598–617.
- [19] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml>
- [20] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. 2019. All Models are Wrong, but Many are Useful: Learning a Variable’s Importance by Studying an Entire Class of Prediction Models Simultaneously. *J. Mach. Learn. Res.* 20 (2019), 177:1–177:81.
- [21] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12–16, 2015*. ACM, 1322–1333.
- [22] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.
- [23] Katsuhige Fujimoto, Ivan Kojadinovic, and Jean-Luc Marichal. 2006. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior* 55, 1 (2006), 72–99.
- [24] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15–19, 2018*. ACM, 619–633.
- [25] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.
- [26] Bryce Goodman and Seth Flaxman. 2017. European Union regulations on algorithmic decision-making and a “right to explanation”. *AI magazine* 38, 3 (2017), 50–57.
- [27] Baptiste Gregorutti, Bertrand Michel, and Philippe Saint-Pierre. 2017. Correlation and variable importance in random forests. *Statistics and Computing* 27, 3 (2017), 659–678.
- [28] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feed-forward networks are universal approximators. *Neural networks* 2, 5 (1989), 359–366.
- [29] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021), 1–210.
- [30] Tae Wan Kim and Bryan R. Routledge. 2018. Informational Privacy, A Right to Explanation, and Interpretable AI. In *2018 IEEE Symposium on Privacy-Aware Computing, PAC 2018, Washington, DC, USA, September 26–28, 2018*. IEEE, 64–74.
- [31] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*. 4765–4774.
- [32] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. 2021. Feature Inference Attack on Model Predictions in Vertical Federated Learning. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19–22, 2021*. IEEE, 181–192.
- [33] Xinjian Luo, Xiaokui Xiao, Yuncheng Wu, Juncheng Liu, and Beng Chin Ooi. 2022. A Fusion-Denoising Attack on InstaHide with Data Augmentation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, February 22 - March 1, 2022*. AAAI Press, 1899–1907.
- [34] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the Estimation Error of Sampling-based Shapley Value Approximation. *CoRR abs/1306.4265* (2013). [arXiv:1306.4265](https://arxiv.org/abs/1306.4265) <https://arxiv.org/abs/1306.4265>
- [35] Daniel F. McCaffrey. 1992. Generalized Additive Models (T. J. Hastie and R. J. Tibshirani). *SIAM Rev.* 34, 4 (1992), 675–678.
- [36] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting Unintended Feature Leakage in Collaborative Learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19–23, 2019*. IEEE, 691–706.
- [37] Smitha Milli, Ludwig Schmidt, Anca D. Dragan, and Moritz Hardt. 2019. Model Reconstruction from Model Explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29–31, 2019*. ACM, 1–9.
- [38] Payman Mohassel and Yupeng Zhang. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22–26, 2017*. IEEE Computer Society, 19–38.
- [39] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.
- [40] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19–23, 2019*. IEEE, 739–753.
- [41] Karl Ezra Pilario, Mahmood Shafiee, Yi Cao, Liyun Lao, and Shuang-Hua Yang. 2019. A review of kernel methods for feature extraction in nonlinear process monitoring. *Processes* 8, 1 (2019), 24.
- [42] Gunnar Rätsch, Sören Sonnenburg, and Christin Schäfer. 2006. Learning interpretable SVMs for biological sequence classification. In *BMC bioinformatics*, Vol. 7. Springer, 1–14.
- [43] Christoph Reindens and Bodo Rosenhahn. 2019. Neural Random Forest Imitation. *CoRR abs/1911.10829* (2019). [arXiv:1911.10829](https://arxiv.org/abs/1911.10829) <https://arxiv.org/abs/1911.10829>
- [44] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why Should I Trust You?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data*

- Mining, San Francisco, CA, USA, August 13–17, 2016. ACM, 1135–1144.
- [45] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215.
 - [46] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1249.
 - [47] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24–27, 2019*. The Internet Society.
 - [48] Andrew Selbst and Julia Powles. 2018. "Meaningful Information" and the Right to Explanation. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23–24 February 2018, New York, NY, USA*, Vol. 81. PMLR, 48.
 - [49] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017*. IEEE Computer Society, 618–626.
 - [50] Lloyd S Shapley. 1953. A value for n-person games. Vol. 2. Princeton University Press, 303–317.
 - [51] Reza Shokri, Martin Strobel, and Yair Zick. 2021. On the Privacy Risks of Model Explanations. In *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19–21, 2021*. ACM, 231–241.
 - [52] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22–26, 2017*. IEEE Computer Society, 3–18.
 - [53] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017*, Vol. 70. PMLR, 3145–3153.
 - [54] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
 - [55] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. 2014. Impact of HbA1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international* 2014 (2014).
 - [56] Erik Štrumbelj and Igor Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems* 41, 3 (2014), 647–665.
 - [57] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10–12, 2016*. USENIX Association, 601–618.
 - [58] Peter Van Der Putten and Maarten van Someren. 2000. *CoIL challenge 2000: The insurance company case*. Technical Report. Technical Report 2000–09, Leiden Institute of Advanced Computer Science.
 - [59] Binghui Wang, Jiayi Guo, Ang Li, Yiran Chen, and Hai Li. 2021. Privacy-Preserving Representation Learning on Graphs: A Mutual Information Perspective. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14–18, 2021*. ACM, 1667–1676.
 - [60] Wikipedia contributors. 2022. Chebyshev's inequality — Wikipedia. https://en.wikipedia.org/wiki/Chebyshev%27s_inequality. Online; accessed 3-April-2022.
 - [61] Wikipedia contributors. 2022. Hoeffding's inequality — Wikipedia. https://en.wikipedia.org/wiki/Hoeffding%27s_inequality. [Online; accessed 3-April-2022].
 - [62] Wikipedia contributors. 2022. Pearson correlation coefficient. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient. Online; accessed 1-April-2022.
 - [63] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F. Naughton. 2016. A Methodology for Formalizing Model-Inversion Attacks. In *IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016*. IEEE Computer Society, 355–370.
 - [64] Yuncheng Wu, Shaofeng Cai, Xiaokui Xiao, Gang Chen, and Beng Chin Ooi. 2020. Privacy Preserving Vertical Federated Learning for Tree-based Models. *Proc. VLDB Endow.* 13, 11 (2020), 2090–2103.
 - [65] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. 2011. Differential Privacy via Wavelet Transforms. *IEEE Trans. Knowl. Data Eng.* 23, 8 (2011), 1200–1214.
 - [66] Zebin Yang, Aijun Zhang, and Agus Sudjianto. 2021. GAMI-Net: An explainable neural network based on generalized additive models with structured interactions. *Pattern Recognition* 120 (2021), 108192.
 - [67] I-Cheng Yeh and Che-hui Lien. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications* 36, 2 (2009), 2473–2480.
 - [68] Raymond W Yeung. 2008. *Information theory and network coding*. Springer Science & Business Media.
 - [69] Xuejun Zhao, Wencan Zhang, Xiaokui Xiao, and Brian Y. Lim. 2021. Exploiting Explanations for Model Inversion Attacks. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10–17, 2021*. IEEE, 662–672.
 - [70] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada*. 14747–14756.