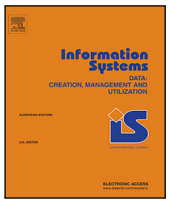
Information Systems 142 (2026) 102770

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is



On the antagonism of explainability and privacy: A comparative study of attacks and explainers

Clemens Müssener [a,*], Gabriela Suntaxi [b], Martin Lange [c], Klemens Böhm [c]

[a] MMK DIGITAL GmbH, Schorndorfer Str. 42, Ludwigsburg, 71638, Germany
[b] Escuela Politécnica Nacional, Quito, 170517, Ecuador
[c] Karlsruhe Institute of Technology, Am Fasanengarten 5, Karlsruhe, 76131, Germany

ARTICLE INFO

Dataset link: https://github.com/Montemaster/XAI

Keywords:
eXplainable Artificial Intelligence (XAI)
Privacy in XAI

ABSTRACT

As explainable artificial intelligence (XAI) becomes more prevalent, concerns arise about the unintended privacy risks associated with model explanations. In this paper, we study the antagonism between explainability and privacy by evaluating the extent to which post-hoc explanations can leak sensitive data. We perform a comparative analysis of three popular XAI methods (SHAP, LIME, and DiCE) applied to Decision Trees, Random Forests, and Neural Networks. We focus on two types of privacy attacks: Membership Inference Attacks and Training Data Extraction. Using datasets of varying complexity, we measure attack success rates and information leakage from explanations. Our results shows that each proposed membership inference attack and training data extraction attack are feasible. These findings highlight the urgent need to design privacy-preserving explainability tools that balance interpretability with user data protection.

1. Introduction

The performance of machine learning (ML) models and their popularity have increased significantly in recent years. ML models tend to be complex and hard to understand for humans. Explainable artificial intelligence (XAI) attempts to address this by releasing information on the models and their presumed decision-making [1]. However, alongside the benefits of transparency, these methods may also introduce new risks. In particular, XAI methods can expose internal behavior of the model and patterns in the training data, raising concerns about privacy leakage.

This paper investigates the antagonism between privacy (whose aim is to protect information) and XAI (whose aim is to provide information) in the context of post-hoc XAI methods. While these methods aim to improve transparency, they can also inadvertently reveal sensitive information about the individuals whose data was used to train the model. Studies [2–5] have shown that post-hoc explanations can inadvertently leak sensitive information about the ML model and the training set. However, these studies typically focus on a single type of explainer or a specific attack scenario. Our article aims to systematically compare privacy attacks against explainers, providing a broader and more realistic assessment of privacy risks across diverse explainability methods. For most explainers, the effect of the particular explainer on privacy is still unclear.

Example 1.1. The popular explainer *Individual Conditional Expectation* (ICE) [6] reveals how a model's prediction changes when a single feature is varied. Fig. 1 shows how the prediction for 'heart disease risk' changes with 'age'. Each line corresponds to one training sample. Although ICE does not directly disclose identities, it still depicts the individual trajectory of each sample, which may allow an attacker to infer sensitive properties of the training data. This illustrates how even widely adopted post-hoc explainers can pose privacy risks.

While our primary focus is on explainers rather than predictive models themselves, we acknowledge that the properties of black-box models influence the post-hoc explanations. Thus, we consider different ML models in our evaluation to capture their indirect but significant influence on privacy risks in XAI. Furthermore, it is unknown if and how this potential effect varies between different explainers. Specifically, we focus on post-hoc XAI approaches, i.e., methods that are applied after the training of machine learning models. These methods interpret already trained models without altering their internal structures or training processes. Post-hoc explainers are among the most widely used in practice due to their model-agnostic nature and ease of integration into existing ML pipelines. Our selected explainers—ranging from global (PDP, ICE, ALE) to local methods (DiCE, KNN, Prototypes & Criticisms)—fall into this category. We, therefore, study these seven popular explainers and conceive different privacy attacks against them.

* Corresponding author.
E-mail address: clemens.muessener@mmkdigital.io (C. Müssener).

https://doi.org/10.1016/j.is.2026.102770
Received 4 November 2023; Received in revised form 7 October 2025; Accepted 9 June 2026
Available online 13 June 2026

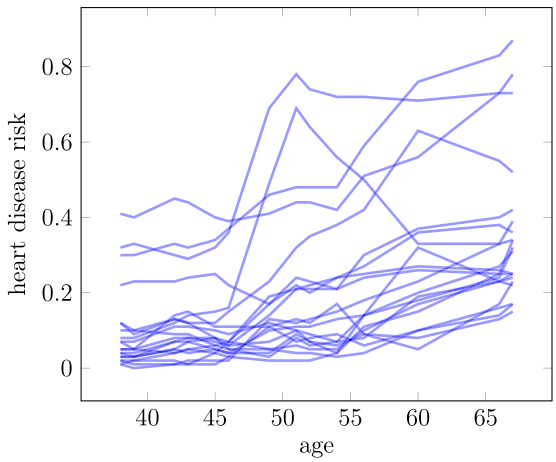


Fig. 1. ICE heart disease example for feature 'age'.

In addition, we carry out these privacy attacks against the different explainers. This is the first comparative study exploring the antagonism between explainability and data privacy in a more general fashion. Understanding the potential conflict between explainability and privacy is essential to developing responsible AI systems. This study aims to contribute to that understanding by analyzing the privacy risks associated with popular post-hoc explainers. Our work confirms the importance of privacy in the context of XAI: It shows that privacy risks in XAI exist.

We have faced the following difficulties comparing privacy attacks against explainers. First, it is necessary to find out which explainers release private data as part of an explanation. Since explainers do not necessarily provide access to training data directly, it is not obvious whether and how one can indeed obtain private data from explanations. Second, it has not been possible to utilize existing implementations of the various attacks, run them in a series of experiments and compile the results. Third, many XAI methods feature unique ways of explaining predictions; hence the type of information that explainers provide as explanation differs significantly. Therefore, one cannot just reuse an attack constructed against one explainer for another one. Instead, one must find different ways of instantiating different attacks for different explainers. For most existing explainers, it is unclear if and how an attacker can successfully access private data. In this paper, we make the following contributions.

We first identify and categorize privacy attacks within the XAI context by adapting two main categories from Papernot et al.'s taxonomy to the explainability setting: membership inference and training data extraction. We treat training data exposed through explanations as private information and formalize these attack types accordingly. Other attack types, such as attribute inference and linkage attacks, do not directly rely on access to the training data itself [7], so they are outside our scope.

Second, we propose a structured adversary model tailored for XAI. We introduce two axes of adversarial capability: (1) access to the prediction function (model access) and (2) access to the explainer itself (explainer access). We analyze both one-time and unlimited access scenarios. To our knowledge, we are the first to introduce this explainer-centric access distinction for privacy attacks in XAI. We consider adversaries with limited capabilities and focus our attack designs on as little access as possible.

Third, we design novel privacy attacks against seven widely used post-hoc explainers selected from Molnar's taxonomy [8]. These explainers, many of which have not been studied before in the context of privacy leakage, expose direct or indirect information about training samples. Thus, we studied and designed privacy attacks against each of these explainers.

Finally, we empirically evaluate the explainers under both membership inference and training data extraction attacks. Our results show that all seven explainers are vulnerable to membership inference, while four also enable training data extraction. Our experiments reveal that these attacks are possible in different scenarios, i.e., different underlying ML models and training data from different settings.

Our work shows that popular explainers are susceptible to privacy breaches. We found that each explainer studied can be attacked to breach privacy, but the success of such attacks depends on several factors, including the type of machine learning model and the characteristics of the training dataset. These findings emphasize the need for a deeper understanding of the interaction between explainability and privacy in real-world applications. The remainder of the article is organized as follows. Section 2 presents an overview of the explainers and privacy attacks considered in this study. Section 3 presents the threat model adopted in this study. Section 4 describes the proposed XAI attacks. Sections 5 and 6 describe the experimental setup and present the results of our evaluation, respectively. Section 7 reviews key literature on explainability and privacy. Finally, Section 8 concludes with insights on the implications for responsible XAI and outlines directions for future research.

2. Explainers – Review

This section focuses on a selected set of widely used post-hoc explainers, both global and local, for which we design and evaluate privacy attacks. While this list is not exhaustive, it covers a representative range of methods that are commonly applied in practice and whose internal mechanisms can be plausibly exploited for privacy breaches. Readers already familiar with the inner workings of the selected explainers may skip this section and proceed directly to the threat model in Section 3.

This section starts with a categorization of explainers, followed by the selection of explainers relevant to our paper. Then, it describes how these explainers work. Table 1 lists the formal notation used in this paper.

2.1. Categorization of explainers

We denote the set of all explanations as $\mathcal{E}$. Explainers return an explanation from this set and are either local or global, as follows [9].

Definition 2.1 (*Global Explainer*)**.** Let e be a function. Given a model $m \in \mathcal{M}$, we call e a global explainer if it provides an explanation $e(m) \in \mathcal{E}$. Thus, a global explainer is of the following form:

$$e_{\text{global}} : \mathcal{M} \to \mathcal{E} \tag{1}$$

Definition 2.2 (*Local Explainer*)**.** Let e be a function. Given a model $m \in \mathcal{M}$ and a concrete sample $\vec{x} \in \mathbb{R}^M$, we call e a local explainer if it provides an explanation $e(m, \vec{x}) \in \mathcal{E}$. Thus, a local explainer is of the following form:

$$e_{\text{local}} : \mathcal{M} \times \mathbb{R}^M \to \mathcal{E} \tag{2}$$

A global explainer describes the decision-making of a model in its entirety. The explanation is not specific to a certain sample. For instance, the explainer may show the average importance of a feature for the decisions by the model. A local explainer explains the prediction of a model for one specific sample, e.g., by showing the importance of each feature for the prediction of the sample. While explainers can be categorized along various dimensions — such as mechanism, model dependency, or output format — we focus specifically on the scope (global vs. local) in this work, as it is central to the design and analysis of our privacy attacks.

Table 1
Notation.

Notation	Description
M	Number of features
$\vec{x} = (x_1, \dots, x_M) \in \mathbb{R}^M$	Sample as a vector of M feature values
x_i	Value of feature i for sample x
X	The set of training samples (training data)
$n = \lvert X \rvert$	Number of training samples
$\hat{X}_i$	Set of all unique feature values of feature i in X
$\mathcal{M}$	The set of all ML models
$f_m : \mathbb{R}^M \to [0, 1]$	Prediction function of model $m \in \mathcal{M}$, that returns a classification score between 0 and 1
$\mathcal{E}$	The set of all explanations
$\vec{x}_{\setminus i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_M)$	Sample $\vec{x}$ without feature value x_i
$\vec{x}_{\setminus i \leftarrow y} = (x_1, \dots, x_{i-1}, y, x_{i+1}, \dots, x_M)$	Sample $\vec{x}$ whose feature value x_i has been replaced by y

Table 2
Selection of explainers.

Explainer	Type	Output
Partial dependence plot [10]	Global	Plot
Indv. Conditional expectation [6]	Global	Plot
Accumulated local effects [11]	Global	Plot
SHAP [12]	Local/ Global	Feature-Importance
K-Nearest-Neighbors [13]	Local	Examples
Counterfactuals [14]	Local	Examples
Prototypes & Criticisms [15]	Local	Examples

2.2. Selection of explainers

Successful privacy attacks are particularly likely when explainers provide access to (private) training data. We, therefore, focus on local and global explainers that provide such access following the definition below.

Definition 2.3 (*Privacy Endangering Explainers*)**.** Let o be the output of a local or global explainer e for a model m. We call e privacy endangering, if its output o contains access to training data. I.e., output o contains at least one unprocessed feature value from the training data.

The book 'Interpretable Machine Learning' [8] features a comprehensive list of popular explainers. We identified seven privacy endangering explainers from [8], whose privacy impact has not yet been studied. See Table 2. Half of the seven privacy endangering explainers (PDP, ICE, ALE, SHAP) are local explainers; the other half are global explainers (SHAP, KNN, Counterfactuals, Prototypes & Criticisms).

2.3. Mechanism of explainers

2.3.1. Partial Dependence Plots

Partial Dependence Plots (PDP) [10] show the average change of the model's prediction when changing a specific feature or a group of features in a sample. PDPs estimate the expected prediction for a feature value $x_i \in \hat{X}_i$ as follows:

$$\overline{F}_i(x_i) = \frac{1}{n} \sum_{j=1}^{n} f_m(\vec{x}^{(j)}_{\setminus i \leftarrow x_i}) \text{ with } \vec{x}^{(j)} \in X \tag{3}$$

A 2D-plot can visualize the PDP for a single feature i:

$$PDP_i = \{(x_i, y) \mid x_i \in \hat{X}_i, y = \overline{F}_i(x_i)\} \tag{4}$$

Each (x_i, y) point is connected to its neighbors on the x-axis by a line. PDPs are a global explanation of the model. A PDP may also include the distribution of the feature in X [8].

2.3.2. Individual Conditional Expectation

Individual Conditional Expectation (ICE) [6] is similar to PDPs; however, no averaging takes place. Thus, it is also a global explanation. An ICE explanation consists of one plot per feature i in X. We denote this with ICE_i:

$$ICE_i = \{ICE_{i,j} \mid j \in \{1, \dots, n\}\} \tag{5}$$

$ICE_{i,j}$ denotes the points of sample $\vec{x}^{(j)} \in X$ in ICE_i:

$$ICE_{i,j} = \{(x_i, y) \mid x_i \in \hat{X}_i, y = f_m(\vec{x}^{(j)}_{\setminus i \leftarrow x_i})\} \tag{6}$$

Each sample $\vec{x}^{(j)} \in X$ has a corresponding line in the 2D plot ICE_i which shows how the prediction of that specific sample changes when changing feature i. The ICE explainer outputs this line by joining together all points $(x, y) \in ICE_{i,j}$ of sample $\vec{x}^{(j)}$ in ICE_i. Like PDPs, the ICE plot ICE_i may show the distribution of feature i in X [8].

2.3.3. Accumulated Local Effects

The explainer Accumulated Local Effects (ALE) [11] also shows the effect of a single feature on the prediction of a model and therefore explains the model globally.

Let K be the number of intervals for which ALE is plotted, selected by the ALE user. Let $z_{k,i}$ be the (k/K)-quantile of the empirical distribution of feature i in the training data X. Let $k_i(x)$ denote the index of the interval that the value x falls into, so that $x \in (z_{k_i(x)-1,i}, z_{k_i(x),i}]$. Similarly, let $N_i(k) \subseteq X$ be the set of training samples that fall into interval k. The so-called uncentered effect is estimated using the following formula:

$$\hat{g}_i(x) = \sum_{k=1}^{k_i(x)} \frac{1}{|N_i(k)|} \sum_{\vec{x} \in N_i(k)} \left(f_m(\vec{x}_{\setminus i \leftarrow z_{k,i}}) - f_m(\vec{x}_{\setminus i \leftarrow z_{k-1,i}}) \right) \tag{7}$$

ALE centers the effects around the mean effect:

$$\hat{f}_i(x) = \hat{g}_i(x) - \frac{1}{n} \sum_{j=1}^{n} \hat{g}_i(\vec{x}_i^{(j)}) \tag{8}$$

This explanation can once again be visualized in a 2D-plot:

$$ALE_i = \{(x, y) \mid x \in \hat{X}_i, y = \hat{f}_i(x)\} \tag{9}$$

Each (x, y) point is connected to its neighbors on the x-axis by a line. Intuitively, ALE explanations divide the values of a single feature into its K quantiles. Each point (x, y) in the ALE plot ALE_i shows how much the prediction changes on average when changing the feature i of a sample to the upper and lower limit of the quantile it falls into. This explainer may also show the distribution of the feature in X [8].

2.3.4. Shapley Additive Explanations

Shapley Additive Explanations (SHAP) [12] attribute either an approximate or exact Shapley value [16] to each feature of a sample. In this paper, we refer to the SHAP implementation that calculates the exact Shapley values. SHAP is a local explanation. For a given sample $\vec{x} = (x_1, \dots, x_M)$, SHAP assigns an importance to each feature i as a scalar value ϕ_i. This importance represents the effect of the respective feature on the model prediction. These ϕ_i correspond to the coefficients of a linear explanation model g. This explanation model g

locally approximates the prediction function with $f_m(\vec{x}) = g(\vec{x}')$ where $\vec{x}'$ is a simplified input of $\vec{x}$.

$$SHAP(\vec{x}) = (\phi_1, \dots, \phi_M) \text{ with } g(\vec{x}') = \phi_0 + \sum_{i=1}^{M} \phi_i x_i' \tag{10}$$

The SHAP Dependence Plot (SDP) shows the SHAP values of each sample in a dataset for a specific feature i. Thus, SHAP may also be used as a global explanation as follows.

$$SDP_i = \{(x_i, SHAP(\vec{x})_i) \mid \vec{x} \in X\} \tag{11}$$

The SHAP Dependence Plot visualizes these points in a scatter plot with x_i on the x-axis and $SHAP(\vec{x})_i$ on the y-axis.

2.3.5. K-Nearest Neighbors

K-Nearest Neighbors (KNN) [13] is a prediction model. It can predict samples by considering the K nearest samples in the training data. In the case of classification, the sample is assigned the class that most of its neighbors share; for regression, KNN predicts the mean of the target values of the neighbors. KNN is deemed interpretable [8], which means that it can serve as its own explanation. The explainer shows the K nearest neighbors of a sample as a local explanation. For a given sample $\vec{x}$, let $\vec{x}^{(1)}, \dots, \vec{x}^{(K)}$ be its K nearest neighbors out of X. Then the KNN explanation outputs these samples:

$$KNN(\vec{x}) = \{\vec{x}^{(1)}, \dots, \vec{x}^{(K)}\} \tag{12}$$

2.3.6. Counterfactuals

Counterfactuals [14] are data objects similar to the given sample but receive the opposite prediction from the model. They are a local explanation that helps users understand what would have to be different to arrive at another outcome. A counterfactual $\vec{c}$ for a given sample $\vec{x}$ minimizes the following objective function for a distance metric d and a new target prediction y (i.e., prediction y is the opposite class to $f_m(\vec{x})$):

$$\underset{\vec{c}}{\operatorname{argmin}} \max_{\lambda} \lambda(f_m(\vec{c}) - y)^2 + d(\vec{x}, \vec{c}) \tag{13}$$

The maximization over the scalar value λ forces $f_m(\vec{c})$ to equal the new target prediction y. If this equality were not given, the maximization of λ would result in an arbitrarily large term value in Eq. (13). This maximization would contradict the minimization over $\vec{c}$. For this paper, we consider the counterfactual explainer DiCE with the constraint that the counterfactuals must come from the training data X. This option is usually chosen to ensure that the counterfactuals are realistic.

$$DiCE(\vec{x}) = \{\vec{c}^{(1)}, \dots, \vec{c}^{(k)}\} \text{ with } \vec{c}^{(j)} \in X \tag{14}$$

2.3.7. Prototypes & Criticisms

Prototypes & Criticisms (PC) [15] are exemplary samples from a dataset. Prototypes represent a specified number of training data samples representing the training data set. Criticisms are samples from the training data where Prototypes either underrepresent or overrepresent the density of the training data. I.e. criticisms are samples that are not well explained by the prototypes. A local explanation may use these samples by showing the nearest prototype and criticism for a sample $\vec{x}'$. Both the selection of prototypes and criticisms can be formulated as the maximization of an objective function F, which calculates a score for given prototypes or criticisms. The number of prototypes/criticisms to generate is k.

$$\underset{S}{\operatorname{argmax}} F(S) \text{ s.t. } S \subseteq X, |S| \leq k \tag{15}$$

Let P be the set of chosen prototypes, and let C be the set of criticisms chosen. The (local) explanation consists of the nearest prototype or criticism:

$$PC(\vec{x}') = \underset{\vec{x} \in P \cup C}{\operatorname{argmin}} \|\vec{x} - \vec{x}'\| \text{ with } P \cup C \subseteq X \tag{16}$$

3. Threat model

This section describes the privacy attacks we adapted from ML to XAI. Since there are several ML attacks against privacy, we first define our focus and then describe each relevant attack. One can model the different types of attacks in any system based on the targets and capabilities of the adversary [17].

3.1. Categories of privacy attacks against ML models

Papernot et al. compile common security and privacy attacks against machine learning models described in the literature [17]. They categorize them based on the adversary's knowledge of the model (black box vs. white box) and the target of the attack (training data, model behavior, model internals, etc.). Here, we focus on black-box model access, which calls for stronger adversary capabilities in contrast to white-box access. Papernot et al. describe three attacks in a black-box setting: (1) Membership inference attacks allow an adversary to infer whether a given sample is included in the training data. (2) Training-data extraction attacks enable adversaries to recover samples partly or entirely. (3) Model extraction is the recovery of internal model parameters by an adversary. – Our work considers access, given by an explainer, to training data and the affiliation of samples to private information from the training data. However, model extraction does not provide such access to training data. Our work focuses on membership inference attacks and training data extraction attacks. We formalize these attacks in Section 3.3.

3.2. Adversary targets and capabilities

Existing attacks on data privacy may have several targets. From the perspective of a threat model, the following information may be private: the training data X, the model m itself, its internal parameters, and its architecture [17]. Training data as the target is critical since the privacy of individuals being objects in the training data is at stake. In white-box attacks, the adversary has access to the model and its internal parameters. In black-box attacks, in turn, the adversary only has access to the prediction function but no knowledge of the internal parameters of the ML model. In terms of privacy, a black-box attack may be a more realistic scenario because of its weaker assumptions.

A common real-world example of black-box access is found in Machine-Learning-as-a-Service (MLaaS) platforms such as Amazon SageMaker, Google Vertex AI, or Azure ML, where users query a model via API and receive only predictions and optional explanation outputs (e.g., SHAP values). In such settings, model parameters and training data remain hidden, making black-box attacks both realistic and practically concerning. This setup reflects weaker assumptions than white-box access, thereby representing a stronger and more generalizable privacy threat model.

Explainer-only attacks. We propose a third category for the XAI context, which we call *explainer-only attacks*. Here, the adversary does not have access to the model internals nor to the prediction function of the model. The adversary only has access to the explanations. In terms of privacy, this is a weaker assumption than the one with black-box models.

For example, an institution might publish SHAP-based explanations alongside loan approval decisions to meet transparency or fairness regulations, while withholding prediction scores or access to the underlying model. In such a setting, an attacker could attempt to extract sensitive training data characteristics using explanations alone. This constitutes a weaker assumption than black-box access, yet still enables effective privacy attacks, as we show in our empirical results.

Our article focuses on black-box and explainer attacks targeting training data in the XAI context. We consider settings with confidentiality risks, i.e., the training data may include personal information.

We define a threat model based on the adversary's access to explanations: black-box access and explainer-only access. These settings involve weaker assumptions than white-box models, making the attacks more generalizable. We also distinguish between one-time and unlimited access to the explainer.

3.3. Privacy attacks

This subsection formally defines the privacy attacks 'membership inference' and 'training data extraction' in the context of XAI.

Definition 3.1 (*Membership Inference*)**.** A membership inference attack MI is a function that takes as input an explainer e, a sample $\vec{x}$, and optionally a prediction function f. MI outputs the inferred membership of $\vec{x}$ in the training data with *true* or $false$. For a dataset S that contains an equal number of training data samples and non-training data samples, MI must fulfill the following condition:

$$\mathbb{P}\left(MI(e,f,\vec{x}) = \begin{cases} true & \text{if } \vec{x} \in X \\ false & \text{if } \vec{x} \notin X \end{cases}\right) > \frac{1}{2} \text{ with } \vec{x} \sim \text{Uniform}(S) \quad (17)$$

Intuitively spoken, a membership inference attack is successful if the chance of the adversary to correctly infer whether $\vec{x}$ is part of the training data X is greater than with random guessing.

Definition 3.2 (*Training Data Extraction*)**.** A training data extraction attack TDE is a function that takes as input an explainer e and outputs a set of extracted samples. TDE must fulfill the following two conditions:

$$|TDE(e)| \geq 1 \quad (18)$$

$$\vec{x} \in TDE(e) \Rightarrow \vec{x} \in X \quad (19)$$

The attack is successful if the adversary extracts at least one sample from the training data.

Both defined attacks are serious privacy violations, but they differ in their severity. With reference to Example 1.1, this means that an attacker can perform membership inference attacks to determine whether an individual is part of a heart disease study, that specifically focused on individuals with particular heart conditions. In addition an attacker can execute training data extraction attacks to obtain the health status of unknown individuals. These attacks can be performed on any dataset.

4. Design of proposed XAI privacy attacks

This section describes membership inference and training data extraction attack designs from Definitions 3.1 and 3.2. Most attack designs are specific to certain explainers, while others apply to multiple explainers. In case a privacy attack applies to several explainers, we describe the attack generally and highlight all affected explainers. Our focus is on a comprehensive comparative study, rather than an in-depth analysis of specific explainers. Table 3 provides an overview of all privacy attack designs in this paper and the access required by an adversary to carry out the attack. Access to the explainer is one-time or unlimited in terms of the number of queries. It is noteworthy, that global explainers tend to provide more information per query than local explainers. For example, a global explainer could provide the average importance of each feature across the entire dataset, whereas a local explainer could only reveal how features contribute to a single prediction.

Access to the model is either only given indirectly through the model (access is explainer-only) or given directly as black-box access.

That is, in our setting the machine learning model itself is treated as a black-box. The privacy leakage does not originate from the explainer in isolation, but rather from the information of the model being revealed through the explainer outputs. Each of the attacks described specifies the particular characteristics of the targeted explainer that are leveraged to execute the attack. This section further describes each design of each privacy attack in detail.

Table 3
Overview of privacy attack designs against explainers.

Attack category	Explainer	Adversary access to the explainer	Adversary access to the model
Membership inference	ICE [6]	One-Time	Black-box
	PDP [10]	One-Time	Explainer-only
	ALE [11]	One-Time	Explainer-only
	SHAP [12]	One-Time	Explainer-only
	DiCE [18]	Unlimited	Explainer-only
	KNN [13]	One-Time	Explainer-only
	Prototypes& Criticisms [15]	One-Time	Explainer-only
Training data extraction	ICE [6]	One-Time	Explainer-only
	DiCE [18]	One-Time	Explainer-only
	KNN [13]	One-Time	Explainer-only
	Prototypes& Criticisms [15]	One-Time	Explainer-only

4.1. Membership inference attacks

We describe five possible membership inference attacks. Two out of these five attacks can be applied to more than one explainer. In other words, we are the first to introduce privacy attack designs against each of the seven explainers listed in Section 2.

4.1.1. ICE membership inference attack

We now propose a design of membership inference attacks against ICE explanations. This attack utilizes the feature values that are part of an ICE explanation to breach privacy as described in Algorithm 1. An ICE explanation is a plot for each feature of the training dataset. The adversary knows a single sample and calculates its ICE line per feature in lines 2 to 6 using the prediction function of the model. The adversary then compares the newly created lines to the original ICE explanation in line 7. If each newly created line matches an existing line in the explanation, the adversary may infer membership. Here, inferred membership of a sample can be both true positives or false positives. We studied the accuracy of the attack experimentally in Section 5.1. The adversary requires one-time access to the global explanation as well as black-box access to the prediction function of the model. This membership inference attack against ICE is the only attack in our paper which requires black-box access to the model.

Algorithm 1 Membership Inference with ICE

Input: Prediction function f_m, ICE_i plot for each feature $i \in \{1, \dots, M\}$, target sample $\vec{t}$

1: **procedure** MEMBERSHIP-INFERENCE-ICE($f_m, ICE, \vec{t}$)
2: **for** each feature $i \in \{1, \dots, M\}$ **do**
3: recreated_line $\leftarrow \emptyset$
4: **for** $x \in$ x values of ICE_i plot for feature i **do**
5: recreated_line $\leftarrow$ recreated_line $\cup$ $(x, f_m((x, \vec{t}_{\setminus i})))$
6: **end for**
7: **if** recreated_line $\notin ICE_i$ **then**
8: **return** false
9: **end if**
10: **end for**
11: **return** true
12: **end procedure**

4.1.2. Feature distribution membership inference attack

We propose a design of the membership inference attack against explanations that include the empirical feature distribution in the training data. One can apply this attack against the explainers PDP, ICE, ALE and SDP, since they contain the empirical feature distributions. Algorithm 2 specifies the attack. Here, an explanation is a line for each feature available in the training data of the model. The line is a connection of (x, y) points. The x values of these points are the x values occurring in the training data for the feature. The adversary knows a sample and has one-time access to the global explanation. This explanation consists of one plot per feature in the training data. The adversary iterates over each feature value of the sample in lines 2 to 6 and checks whether there exists a point in the plot with the exact same feature value in line 3. If this check succeeds for each plot the adversary may infer membership. Inferred membership of a sample can result in true or false positives. We studied this experimentally in Section 5.1.

Algorithm 2 Membership Inference with Feature Distribution

Input: Explainer $e \in \{PDP, ICE, ALE, SDP\}$, target sample $\vec{t}$

1: **procedure** MEMBERSHIP-INFERENCE-FEATURE-DIST($e, \vec{t}$)
2: **for** each feature $i \in \{1, \ldots, M\}$ **do**
3: **if** $\vec{t}_i \notin$ x values of e_i **then**
4: **return** false
5: **end if**
6: **end for**
7: **return** true
8: **end procedure**

4.1.3. SHAP membership inference attack

This SHAP membership inference attack utilizes SHAP together with SHAP Dependence Plot. Algorithm 3 is the pseudocode of the attack. The adversary uses the local SHAP explanation $SHAP(\vec{t})$ of a sample $\vec{t}$ and the global SHAP dependence plot of X as follows. The adversary compares the specific SHAP values $SHAP(\vec{t})_i$ of each feature i of the sample with the SHAP dependence plot SDP_i for feature i in line 3. If there exists a point for each feature i with the exact same feature and SHAP values $SHAP(\vec{t})_i$, the adversary infers membership.

Algorithm 3 Membership Inference with SHAP

Input: $SHAP$ explainer, SHAP dependence plot SDP_i for each feature $i \in \{1, \ldots, M\}$, target sample $\vec{t}$

1: **procedure** MEMBERSHIP-INFERENCE-SHAP($SHAP, SDP, \vec{t}$)
2: **for** each feature $i \in \{1, \ldots, M\}$ **do**
3: **if** $(\vec{t}_i, SHAP(\vec{t})_i) \notin SDP_i$ **then**
4: **return** false
5: **end if**
6: **end for**
7: **return** true
8: **end procedure**

4.1.4. DiCE membership inference

Counterfactuals are either synthetic objects from anywhere in the data space or existing samples from the training data. The following design of a membership inference attack is possible when the counterfactual explainer DiCE is configured to use samples from the training data. See Algorithm 4. The design of the attack relies on the fact that counterfactuals are close to the input sample $\vec{t}$ and have the opposite prediction from the one of $\vec{t}$. Thus, the adversary interacts with DiCE in line 2 to receive counterfactual samples $\vec{c} \in C_1$ for $\vec{t}$ with a opposite prediction compared to $f_m(\vec{t})$. Then, the adversary interacts with DiCE again, to receive the counterfactuals C_2 in line 4. The idea of this attack is that the proximity of counterfactuals to the entered sample $\vec{c} \in C_1$ leads to the adversary receiving $\vec{t}$ as a counterfactual in C_2. This is the only attack in our paper to require repeated access to the explainer, as opposed to one-time access in the other cases.

Algorithm 4 Membership Inference with DiCE

Input: $DiCE$ explainer, target sample $\vec{t}$

1: **procedure** MEMBERSHIP-INFERENCE-DICE($DiCE, \vec{t}$)
2: $C_1 \leftarrow DiCE(\vec{t})$
3: **for** $\vec{c} \in C_1$ **do**
4: $C_2 \leftarrow DiCE(\vec{c})$
5: **if** $\vec{t} \in C_2$ **then**
6: **return** true
7: **end if**
8: **end for**
9: **return** false
10: **end procedure**

4.1.5. Proximity-based membership inference attacks

The respective explainer is trained on the same training data X as the model to be explained. See Algorithm 5. The adversary accesses the explainer e once in line 2 and retrieves the example-based explanations for a given sample $\vec{t}$. The adversary infers membership if sample $\vec{t}$ is part of the explanation. The intuition is that, if $\vec{t}$ is part of the training data it is its own nearest example. Shokri et al. propose an attack on influence functions that identifies training samples with high impact on the model's prediction [2]. In contrast, our proximity-based membership inference attack relies on the proximity of samples in feature space, rather than on influence scores.

Algorithm 5 Membership Inference with Proximity-based Explainers

Input: Explainer $e \in \{KNN, PC\}$, target sample $\vec{t}$

1: **procedure** MEMBERSHIP-INFERENCE-PROXIMITY($e, \vec{t}$)
2: **if** $\vec{t} \in e(\vec{t})$ **then**
3: **return** true
4: **else**
5: **return** false
6: **end if**
7: **end procedure**

4.2. Training data extraction attacks

Here, we propose two training data extraction attacks, called "ICE training data extraction" and "example-based training data extraction attacks", against four explainers. These explainers are ICE, DiCE, KNN, and Prototypes & Criticisms. We were not able to construct training data extraction attacks for the remaining three explainers. Constructing training data extraction attacks (other than against example-based explanations) is particularly difficult, because an adversary has no prior knowledge about the training data. In addition, extracting samples from the training data requires to extract all feature values of a sample.

4.2.1. ICE training data extraction attack

The design of the training data extraction attack against ICE exploits the fact that each ICE line is calculated for an individual sample. See Algorithm 6. The adversary uses the fact that, for all features i, the ICE line $ICE_{i,j}$ of sample $\vec{t}^{(j)} \in X$ must have at least one common value: the prediction $f(\vec{t}^{(j)})$. This common value is determined in line 2. The original feature values of $\vec{t}^{(j)}$ can be extracted if this common value occurs exactly once in each $ICE_{i,j}$ for the different features i. This requirement is checked in lines 8 to 11. The original feature values are the x values of $(x, y) \in ICE_{i,j}$ where y is equal to $f(\vec{t}^{(j)})$. These feature values are extracted one by one in line 12 to construct the extracted sample $\vec{t}$.

Algorithm 6 Training Data Extraction with ICE

Input: ICE_i plot for each feature $i \in \{1, \dots, M\}$, index $j \in \{1, \dots, n\}$ of targeted sample for extraction

1: **procedure** TRAINING-DATA-EXTRACTION-ICE(ICE, j)
2: shared_predictions $\leftarrow \{y \mid \forall i \; \exists x_i : (x_i, y) \in ICE_{i,j}\}$
3: **if** |shared_predictions| $\neq 1$ **then**
4: **fail**
5: **end if**
6: $\vec{t} \leftarrow$ Instantiate array of size M
7: **for** $i \in \{1, \dots, M\}$ **do**
8: x_values $\leftarrow \{x_i \mid (x_i$, shared_predictions[0]$) \in ICE_{i,j}\}$
9: **if** |x_values| $\neq 1$ **then**
10: **fail**
11: **end if**
12: $\vec{t}_i \leftarrow$ x_values[0]
13: **end for**
14: **return** $\vec{t}$
15: **end procedure**

4.2.2. Example-based training data extraction attack

This attack approach is applicable to example-based explainers, such as DiCE, Prototypes & Criticisms, as well as KNN. See Algorithm 7. This attack aims to extract training data returned as explanation. In order to receive the explanation in the first place, the adversary must construct a random sample as input for the explainer in line 2. Since an explanation contains samples from the training data, this attack is relatively easy to construct. However, in order to construct input samples, the adversary must have knowledge about the features of the sample, i.e. the range of numeric features and the possible values of categorical features. This information can be considered common knowledge in real-world scenarios (e.g., for a feature 'age'). Shokri et al. also mention an example-based attack, however specific to influence functions.

Algorithm 7 Training Data Extraction with Example-based Explanations

Input: Explainer $e \in \{DiCE, PC, KNN\}$

1: **procedure** TRAINING-DATA-EXTRACTION-EXAMPLES(e)
2: $\vec{t} \leftarrow$ generate sample
3: **return** $e(\vec{t})$
4: **end procedure**

5. Experimental setup

Objectives. In this section, we describe the experimental design that compare the different privacy attacks proposed in Section 4. The purpose of this evaluation is to learn to what extent an adversary may break privacy for each such privacy attack. In addition, we compare the proposed attacks, to determine if some attacks have scenario specific requirements. For example, an attack may perform successfully for data containing categorical features but not for data containing numeric features. We conduct two kinds of experiments: Both kinds of experiments are instantiated with the various explainers. Furthermore, in each kind of experiment we study different scenarios based on different datasets and different ML models. The source code of the experiments is published on our Git repository [19].

Datasets. We use two datasets, the *census income* and *heart disease* datasets from the UCI Machine Learning Repository [20], and prepared each in three different ways: as-is (with numeric and categorical features), with only numeric features, and with only categorical features. This resulted in a total of six datasets used in our comparative study. In the proess of preprocessing the datasets, we first removed samples with missing data. This affected 93 of 1190 samples for the *heart disease* dataset, and 2399 of 32 561 samples for the *census income* dataset. We then created the numeric and categorical versions of each dataset. For the numeric version, categorical features were omitted. For the categorical version each numeric feature was split into 10 bins of equal width. Each bin corresponds to one new category. Then duplicate samples were removed for each dataset version. For the *heart disease* and *census income* dataset, this resulted in the removal of 271 and 3848 duplicate rows respectively. For the numeric version of the *heart disease* and *census income* dataset, 273 and 19 859 duplicates rows were removed. For the categorical version, we removed 277 and 12 136 duplicate rows. Duplicate rows increase for numeric and categorical versions due to removed features that differentiated samples or samples being placed in the same bins for discretized features.

ML models and scenarios. We consider each combination of ML model, dataset and privacy attack a *scenario*. The ML models used are decision tree, random forest and neural network from the sklearn package [21]. Humans do not understand these popular ML models beyond a certain complexity, which is why they are well suited for explanation using XAI [22]. All ML models are instantiated with default sklearn hyperparameters. We use a neural net with 1 hidden layer of 100 ReLU nodes and cross-entropy loss, and decision tree and random forest classifiers with standard settings. The random forest consists of 100 trees and the splitting criterion is Gini impurity. For membership inference, we analyze 90 scenarios using three ML models, six datasets, and five privacy attack designs. For training data extraction, we conduct 36 scenarios with three ML models, six datasets, and two privacy attack designs. In total, we conducted 126 distinct experimental scenarios.

5.1. Membership inference

Explainers. In this experiment, we evaluate proposed designs of a membership inference attack (as described in Section 4.1). The explainers discussed in Section 4 are likewise susceptible to privacy breaches when subjected to the same attack designs. Specifically, PDP, ICE, ALE, and SDP are susceptible to feature distribution membership inference attack, whereas KNN and PC are vulnerable to the proximity-based membership inference attack. In our evaluation, we chose to implement the feature distribution-based membership inference attack specifically against PDP,We chose to implement the feature distribution-based membership inference attack against PDP, as the success of this attack does not depend on the choice of explainer. The attacker relies solely on the feature distribution, which is determined by the dataset rather than the explanation method. We also implement the proximity-based membership inference attack against KNN to enable a comprehensive comparative analysis of the two attack strategies.

Settings. For each scenario, we train the ML model and explainer on half of their respective datasets to fulfill Definition 3.1. This ensures that each sample $\vec{t}$ has a 50% chance of being in the training data. Our attack is deemed successful if the adversary infers membership more accurately than through random guessing. This setting is crucial, as a highly imbalanced training dataset could make the attack trivial. For example, if a model and explainer are trained on 99% of the data and only 1% non-training data, any attack returning 'true' would have a 99% accuracy. Put differently, such an attack could be become trivial.

Measures. For each scenario, we run an attack for every sample in the corresponding dataset (containing training samples and non-training samples). We then measure the success by calculating the accuracy of membership inference per scenario, i.e., the fraction of memberships the adversary inferred correctly. We present the mean accuracy and its standard deviation for each scenario, calculated from ten experimental trials. Accuracy provides a comprehensive measure of the attack's overall success, since we split the dataset 50/50. In addition we captured the true positive rate - also known as recall, which did not provide further insights. The results are available on our Git repository [19].

5.2. Training data extraction

Explainers. In this experiment, we evaluate both proposed designs of the training data extraction attack (as described in Section 4.2). As noted in Section 4, certain explainers share sufficient structural similarity to allow an attacker to compromise privacy using the same attack design. Specifically, the explainers DiCE, PC, and KNN are capable of supporting the example-based training data extraction attack. For our evaluation, we implement this attack using the widely adopted DiCE explainer. Since this attack involves directly presenting training data as part of the explanation, any of the aforementioned example-based explainers could be substituted without affecting the attack's effectiveness. Therefore, we consider it sufficient to demonstrate this scenario using DiCE alone.

Settings. In each scenario, we train the ML model and the explainer on the entire dataset. Evaluating a training data extraction attack does not require samples which are not in the training data.

Measures. We deem a training data extraction attack successful if one sample $\vec{x}$ from the training data X can be extracted (see Definition 3.2). In a preliminary evaluation, we found it too simple to only extract a single sample in many scenarios. Thus, we extend the scope of our experiment as follows. In the experiment, the adversary attempts to extract the entire training data, not only just one sample. To this end, we run the attack design for each sample in the training data and union their extracted samples. To quantify the success of our training data extraction attacks, we report the recall for each scenario. Recall is defined as the ratio of correctly extracted samples to the total number of samples in the training dataset. Since our attacks do not produce any false positives, precision is consistently 100% and therefore not reported. For each scenario, we present the mean recall along with the corresponding standard deviation, based on ten independent experimental runs.

6. Results

This section features the results of our experiments. We start by evaluating membership inference attacks and then training data extraction attacks.

6.1. Membership inference

In this subsection, we present the results of our experiment on membership inference attacks for different versions of the heart disease and census income datasets. Results are shown in Figs. 2–7, with each figure displaying the attack accuracy per attack design and ML model. The dashed line in each figure is the accuracy of random guessing. We label the design on the x-axis with the explainer under attack.

ICE membership inference attack. On all datasets, membership inference attacks against ICE are very accurate for neural nets and random forests. The attacks are less accurate or not even better than random guessing for decision trees. Compared to the predictions of the neural net and the random forest, the decision tree provides only few different predictions. In particular, the decision tree, with the default parameters, expands all nodes until all leaves of the decision tree are pure (they only contain training data of one class). Therefore, the decision tree only predicts 0 or 1. The neural net and the random forest in turn provide a score between 0 and 1. ICE generally benefits from many different predicted values, because this results in varied ICE lines and thus in multiple unique ICE lines as part of the explanation. The ICE membership inference attack requires unique ICE lines to avoid false positives. However, unique lines are rarely given in our decision tree setting, which explains the low accuracy of the decision tree setting. Our result shows that our proposed membership inference attack performs very accurately. The underlying prediction function of the ML model is crucial for a successful ICE membership inference attack.

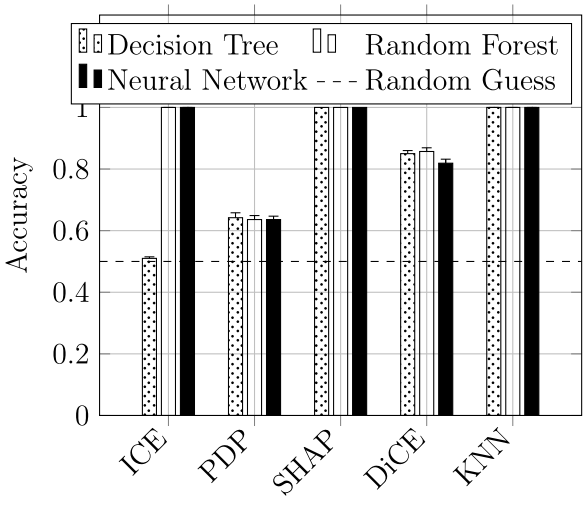


Fig. 2. Accuracy of membership inference attacks using the heart disease dataset.

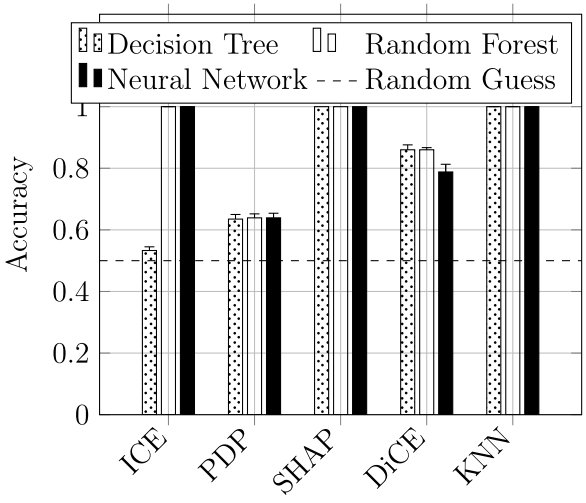


Fig. 3. Accuracy of membership inference attacks using the numeric version of the heart disease dataset.

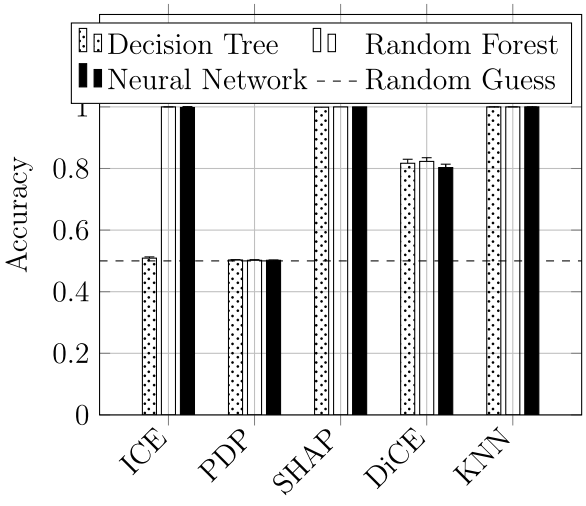


Fig. 4. Accuracy of membership inference attacks using the categorical version of the heart disease dataset.

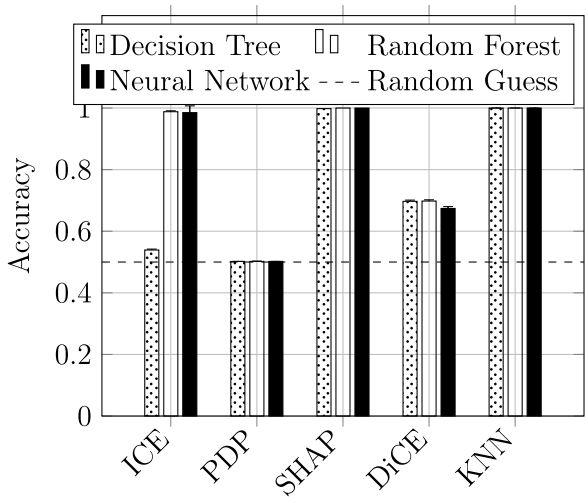


Fig. 5. Accuracy of membership inference attacks using the census dataset.

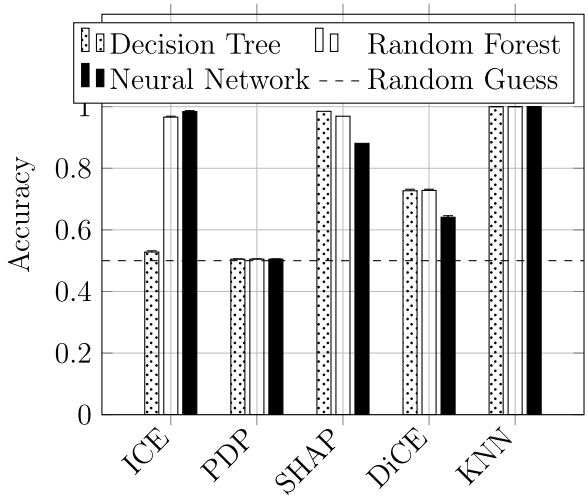


Fig. 6. Accuracy of membership inference attacks using the numeric version of the census dataset.

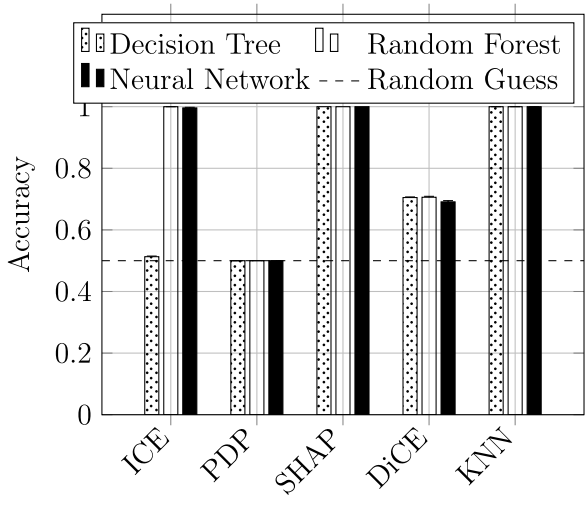


Fig. 7. Accuracy of membership inference attacks using the categorical version of the census dataset.

Feature distribution membership inference attack against PDP. Our experiment shows that membership inference attacks against PDP are better than guessing for the as-is and numeric version of the heart disease datasets but not any other datasets. Our explanation is as follows: The attack uses the distribution of the feature values to infer membership. If a feature has a wide range of possible values – ideally unique values – the attack is likely to succeed (we confirm this in additional experiments, available on our Git repository [19].

The attack is successful for heart disease dataset (as-is and numeric) due to sufficient unique feature values; the accuracy of both attacks is very close. The attack is unsuccessful for categorical versions of heart disease and census datasets due to identifying all samples as members of the training data, including non-members.

However, the attack is not successful for the categorical version of the heart disease dataset nor for the categorical version of the census datasets. This is because the attack identifies all samples as members of the training data, including non-members. For the other two versions of the census dataset, the attack correctly identifies few samples as non-members so that the accuracy is slightly above 50%. In our results, the ML model does not have any influence on the accuracy of the attack, since the attack only depends on the dataset. Overall, our proposed attack design is generally feasible, but strongly dependent on the features of the underlying datasets used by the respective explanations.

SHAP membership inference attack. An adversary can attack SHAP with high accuracy in any of the scenarios studied in our experiment. The results can be seen in Figs. 2–7. These results contain no standard deviation since they were run once per scenario, rather than 10 times, as calculating the SHAP dependence plots is computationally expensive. Nonetheless, we consider them conclusive because one can observe high accuracy across all scenarios. We attribute the high accuracy to the fact that the SHAP values are nearly unique for the sample they are created for, independent from the underlying dataset and ML model. False positives are very unlikely. It is very likely to carry out successful membership inference attacks against SHAP and the SHAP dependence plot.

DiCE membership inference attacks. An adversary can attack DiCE explanations with an accuracy above 60% in any of our scenarios and even higher in some scenarios. Our experiment shows that false negative results become more likely in our attack when the label distribution of the prediction function is unbalanced. We attribute this to the mechanism of DiCE, which provides counterfactuals based on a distance function. One can observe this behavior best in the scenarios that use the neural net. This is because in our experiment the neural net produces the most unbalanced predictions. So the accuracy of the attack is reduced. Our takeaway is that the proposed membership inference attack against DiCE is feasible. Additionally, the balance described above is of high importance for the attack to succeed.

Proximity-based membership inference attacks against KNN. Attacking KNN has an accuracy of 100% in all scenarios of our experiment. The explanation contains the sample as its own neighbor if and only if it is part of the training data. So the adversary infers membership with perfect accuracy.

Runtimes. All experiments were conducted on standard consumer-grade hardware, without the use of specialized high-performance systems for machine learning. While the longest runtimes per scenario remained within a few hours, many completed in just seconds. Therefore, we omit a detailed runtime analysis in the article. For transparency and to support reproducibility on similar setups, complete runtime data for each repetition of the 126 scenarios is available in our Git repository [19].

Table 4
Runtime summary of training data extraction attacks.

Privacy attack	Mean (s)	Max (s)	Min (s)
ICE training data extraction attack	28.56	262.46	0.71
Example-based training data extraction attack against DiCE	569.89	3661.98	9.74

Summary. Our comparative experiments show that all proposed membership inference attacks are feasible. The success of different proposed membership inference attacks hinges on various factors. The attack against ICE highly depends on the underlying ML model. The feature distribution membership inference attack strongly depends on the dataset of the explanation. The attack against DiCE is dependent on the prediction balance of the respective ML model. In our experiment, the membership inference attack against SHAP and the attack against KNN both are very accurate in any scenario. Our experiments show that privacy violations are severe in real-world situations. Dataset affiliation is considered private, but an attacker can confirm if someone belongs to a specific dataset, allowing them to draw conclusions about the individual, such as a study involving people with a specific disease.

6.2. Training data extraction

In this subsection, we describe the results of our experiment for training data extraction attacks with the as-is version (Fig. 8), the numeric version (Fig. 9) and the categorical version (Fig. 10) of the heart disease dataset, as well as the as-is version (Fig. 11), the numeric version (Fig. 12) and the categorical version (Fig. 13) of the census dataset. Each figure shows the recall of the attacks for each attack design per ML model. We label the design on the x-axis with the explainer attacked. We evaluate the example-based training data extraction attack against the explainer DiCE. Training data extraction attacks consistently achieve 100% precision, as they produce no false positives. Therefore, this section only includes figures for recall.

ICE training data extraction attack. Our experiment shows that an adversary can extract training data from ICE in various scenarios. Specifically, using a neural net has a high recall, while using a random forest is low. Next, ICE training data extraction attacks using decision trees are unsuccessful in any scenario. This may be because the ICE training data extraction attack requires diversified predictions to succeed. In particular, our proposed attack is not successful if the original prediction of a sample occurs multiple times in one ICE line. In our setting, the random forest yields few unique prediction values for all samples in the data set compared to the neural net that produces nearly unique predictions per sample. This causes the difference in recall between random forest and neural net. Comparing the different scenarios, our result shows that training data extraction attacks against ICE require diverse predictions of the ML model in order to succeed.

Example-based training data extraction attack against DiCE. In any scenario of our experiment, an adversary can perform a training data extraction attack against DiCE successfully. The recall is under 100%, because some samples are never chosen as counterfactual. This holds for samples far away from samples of the opposite class. Thus, the structure of the datasets is causing the differences.

Runtimes. We conducted our experiments using standard consumer-grade hardware, without access to high-performance systems optimized for machine learning. Despite this limitation, we report the runtimes to provide practical guidance for those intending to replicate the experiments on similar setups. Table 4 summarize the runtimes for each trainings data extraction attack, reporting the mean, maximum, and minimum durations across all scenarios, measured in seconds.

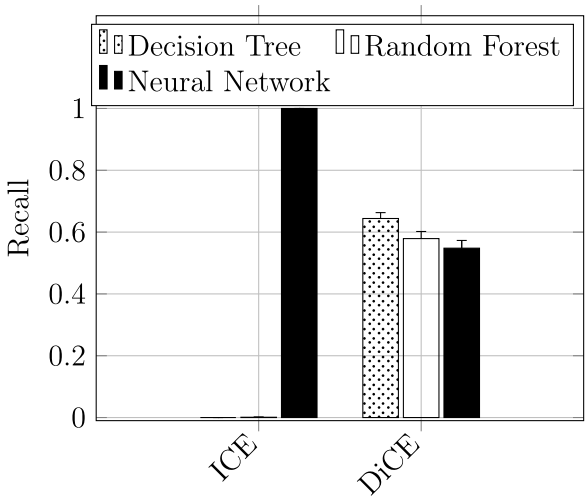


Fig. 8. Recall of training data extraction attacks using the heart disease dataset.

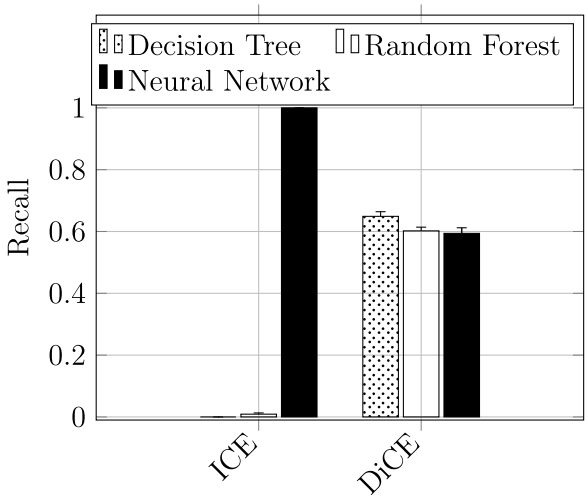


Fig. 9. Recall of training data extraction attacks using the numeric version of the heart disease dataset.

Summary. Our experiment shows that each proposed training data extraction attack is feasible. Attacks against ICE depend on the underlying ML model. The attack against DiCE is dependent on the training dataset and the respective ML model, because both are decisive for the chosen counterfactuals. Our experiments demonstrate that an attacker can extract private information from individuals in the heart disease or census income dataset, thus violating their privacy by revealing their health status and income situation.

7. Related work

The intersection of explainable artificial intelligence (XAI) and data privacy has recently emerged as a critical area of research. While the individual domains of explainability and privacy have been extensively studied, their antagonism remains underexplored systematically and comparatively. Among the most pressing threats are membership inference attacks and training data extraction attacks that exploit the outputs of explanation methods to infer sensitive information about the training data.

Shokri et al. experimentally show successful membership inference against backpropagration-based explainers [2]. In their attack, an adversary uses the variance of the entries of the explanation vector. The

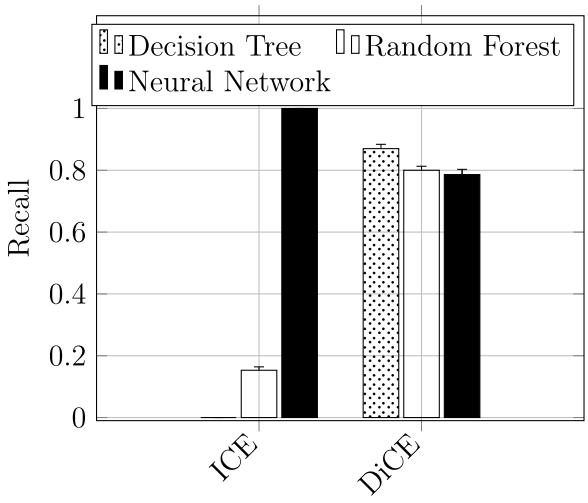


Fig. 10. Recall of training data extraction attacks using the categorical version of the heart disease dataset.

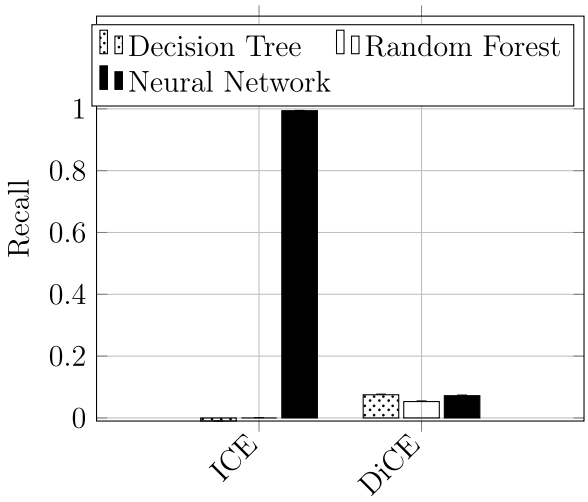


Fig. 11. Recall of training data extraction attacks using the census dataset.

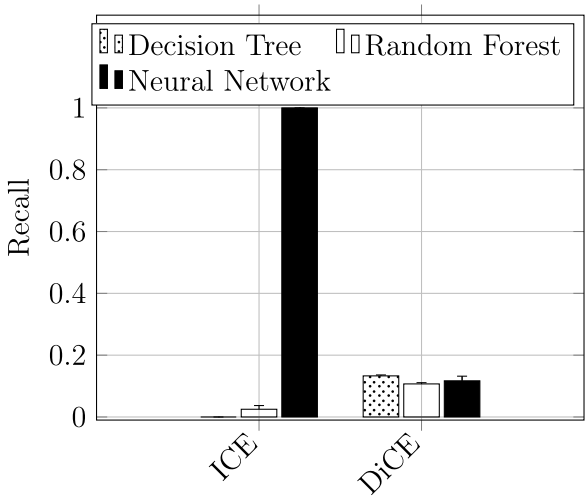


Fig. 12. Recall of training data extraction attacks using the numeric version of the census dataset.

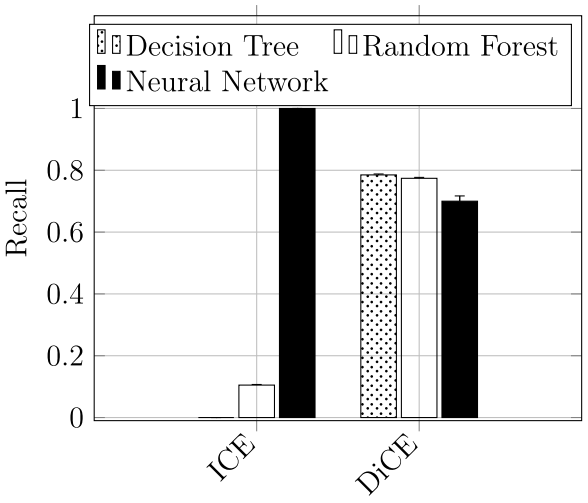


Fig. 13. Recall of training data extraction attacks using the categorical version of the census dataset.

explanation vector for one sample consists of the importance of each feature as determined by the explainer. The authors claim that a variance under a certain threshold indicates that the sample is part of the training data of the model. To find a suitable threshold, the adversary must have some knowledge of the distribution of the training data. The attack reaches an accuracy of up to 60% in the experiments. The authors attempt a similar attack against perturbation-based explainers, but it is unsuccessful. The paper also describes two attacks against the example-based method, Influential Instances, which can be used for membership inference and training data extraction attacks.

Kuppa et al. show that membership inference attacks are possible with access to counterfactual examples [23]. Their attack targets the explainer Latent-CF [24]. Latent-CF uses an autoencoder trained on the training data to create counterfactuals. To infer membership, Kuppa et al. train a 1-nearest-neighbor classifier A_{MemInf} on counterfactuals by the explainer. One compares the output of the original model T to that of A_{MemInf}. If the difference is under a threshold, the adversary infers membership. The authors argue that a similar prediction of T and A_{MemInf} for a sample indicates that the sample is part of the training dataset. In their experiments, this attack reaches an accuracy of 73%.

Nguyen et al. provide a survey on privacy attacks on model explanations and their countermeasures [7]. They analyze research papers in the field of XAI and data privacy. Besides other types of privacy attacks, the authors summarize existing membership inference attacks and training data extraction attacks. They call the latter 'reconstruction attacks'.

Additional work by Aivodji et al. [25] critiques the unanticipated privacy impact of explanation methods like SHAP and LIME. The authors argue that explanations, while intended to promote transparency, may actually leak more about the underlying training data than intended. Their study provides both empirical and theoretical arguments supporting the idea that XAI can be a vector for privacy leakage.

Nguyen et al. present a comprehensive survey on privacy attacks against model explanations, along with corresponding defense mechanisms [7]. Their analysis encompasses a broad range of literature at the intersection of explainable artificial intelligence (XAI) and data privacy. Among various categories of privacy threats, the authors specifically review existing work on membership inference attacks and training data extraction techniques, referring to the latter as "reconstruction attacks".. Their survey also emphasizes the difficulty of designing XAI methods that balance interpretability and privacy protection, calling for new frameworks that systematically evaluate this trade-off. Complementing this, Baniecki et al. [26] examine the landscape of

counterfactual explanations and outline various privacy risks associated with generating purposefully altered instances to justify model decisions. They point out that, although counterfactuals are often seen as privacy-preserving by nature, their generation can inadvertently reveal decision boundaries, feature importance, or even implicit training data properties, particularly when implemented via generative models or autoencoders trained on private data. Both surveys advocate for stronger privacy risk assessments within the development of XAI pipelines and encourage further exploration of the adversarial capabilities specific to explanation methods.

Together, these studies highlight an emerging consensus: explanation methods, particularly post-hoc ones, can be exploited to infer sensitive data. However, most prior work focuses on single explainer types or narrow attack models, i.e., these works assume that an adversary has more knowledge to successfully attack privacy [2,23]. Our work complements these studies by conducting a broader comparative analysis across multiple explainers and designing new attacks tailored to the explainer's mechanisms, filling a significant gap in the literature.

8. Conclusions

Even though the effect on privacy is unclear for most explainers, XAI is a hot topic in research. Also unknown is whether the potential effects vary between different explainers. While the theoretical antagonism between explainability and privacy may seem obvious, its practical analysis is not trivial. Comparing privacy attacks against explainers is challenging because attacking different explainers requires an adversary to design different attacks in most cases. Therefore, our goal is to compare privacy attacks against explainers. Our work is the first comparative study to evaluate the antagonism between explainability and privacy in a broader sense. In this paper, we design attacks against the seven explainers: ICE, PDP, ALE, SHAP, DiCE, KNN, and Prototypes & Criticisms. Our comparative experiments show that an adversary may breach privacy with each attack proposed in this work. Additionally, we observe that the success of privacy attacks can hinge on different factors, such as the ML model or the type of training data.

Future work. We consider it necessary to conduct research on general strategies to ensure privacy in XAI. In addition, we consider developing defense mechanisms to safeguard the privacy of individuals, such as differential privacy, data perturbation, or anonymization, as a topic for future investigation, incl. recommendations to avoid privacy attacks. Another important direction is to extend the study to additional machine learning models, such as Support Vector Machines (SVM) and XGBoost. These models introduce methodological challenges due to the need for model-specific adaptations in post-hoc explainers like SHAP and DiCE. Their inclusion would allow for a broader assessment of how model architecture influences both explanation quality and privacy risks. In addtion, we plan to investigate how model performance, particularly in degraded models, affects explanation behavior and the effectiveness of privacy attacks. Finally, an interesting topic for future work is to cluster explainers by their underlying mechanism — such as surrogate-based, sample-based, or gradient-based approaches — and to evaluate privacy risks with these clusters. This could provide a more nuanced understanding of how structural properties of explanation methods influence their vulnerability to different types of privacy attacks.

CRediT authorship contribution statement

Clemens Müssener: Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Gabriela Suntaxi:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Martin Lange:** Writing – original draft, Software, Data curation. **Klemens Böhm:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.is.2026.102770.

Data availability

The data is public available in the following git repository: https://github.com/Montemaster/XAI.

References

[1] Z.C. Lipton, The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery, Queue 16 (3) (2018) 31–57, http://dx.doi.org/10.1145/3236386.3241340.

[2] R. Shokri, M. Strobel, Y. Zick, On the privacy risks of model explanations, in: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, AIES '21, Association for Computing Machinery, 2021, pp. 231–241.

[3] S. An, Y. Cao, Counterfactual explanation at will, with zero privacy leakage, Proc. ACM Manag. Data 2 (3) (2024) 1–29.

[4] Y. Wang, H. Qian, C. Miao, Dualcf: Efficient model extraction attack from counterfactual explanations, in: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 1318–1329.

[5] M. Pawelczyk, H. Lakkaraju, S. Neel, On the privacy risks of algorithmic recourse, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 9680–9696.

[6] A. Goldstein, A. Kapelner, J. Bleich, E. Pitkin, Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation, J. Comput. Graph. Statist. 24 (1) (2015) 44–65.

[7] T.T. Nguyen, T.T. Huynh, Z. Ren, T.T. Nguyen, P.L. Nguyen, H. Yin, Q.V.H. Nguyen, Privacy-preserving explainable AI: A survey, Sci. China Inf. Sci. 68 (1) (2024) 111101, http://dx.doi.org/10.1007/s11432-024-4123-4.

[8] C. Molnar, Interpretable Machine Learning, second ed., 2022, URL christophm.github.io/interpretable-ml-book/.

[9] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2019) 1–42, http://dx.doi.org/10.1145/3236009.

[10] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Statist. (2001) 1189–1232.

[11] D.W. Apley, J. Zhu, Visualizing the effects of predictor variables in black box supervised learning models, J. R. Stat. Soc. Ser. B Stat. Methodol. 82 (4) (2020) 1059–1086.

[12] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017).

[13] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Amer. Statist. 46 (3) (1992) 175–185.

[14] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, Harv. JL Tech. 31 (2017) 841.

[15] B. Kim, R. Khanna, O.O. Koyejo, Examples are not enough, learn to criticize! criticism for interpretability, Adv. Neural Inf. Process. Syst. 29 (2016).

[16] L.S. Shapley, Notes on the N-Person Game, II: The Value of an N-Person Game, RAND Corporation, 1951.

[17] N. Papernot, P. McDaniel, A. Sinha, M.P. Wellman, SoK: Security and privacy in machine learning, in: 2018 IEEE European Symposium on Security and Privacy, EuroS P, 2018, pp. 399–414, http://dx.doi.org/10.1109/EuroSP.2018.00035.

[18] R.K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, in: FAT* '20, Association for Computing Machinery, 2020, pp. 607–617, http://dx.doi.org/10.1145/3351095.3372850.

[19] C. Müssener, Experiments for the article 'On the antagonism of explainability and privacy: a comparative study of attacks and explainers', 2025, https://github.com/Montemaster/XAI, (Accessed 28 June 2026).

[20] D. Dua, C. Graff, UCI machine learning repository, 2017, URL http://archive.ics.uci.edu/ml.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[22] Y. Lou, R. Caruana, J. Gehrke, Intelligible models for classification and regression, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 150–158, http://dx.doi.org/10.1145/2339530.2339556.

[23] A. Kuppa, N.-A. Le-Khac, Adversarial XAI methods in cybersecurity, IEEE Trans. Inf. Forensics Secur. 16 (2021) 4924–4938, http://dx.doi.org/10.1109/TIFS.2021.3117075.

[24] R. Balasubramanian, S. Sharpe, B. Barr, J.D. Wittenbach, C.B. Bruss, Latent-CF: A simple baseline for reverse counterfactual explanations, 2020, CoRR abs/2012.09301. arXiv:2012.09301. URL https://arxiv.org/abs/2012.09301.

[25] U. Aïvodji, A. Bolot, S. Gambs, Model extraction from counterfactual explanations, 2020, arXiv preprint arXiv:2009.01884.

[26] H. Baniecki, P. Biecek, Adversarial attacks and defenses in explainable artificial intelligence: A survey, Inf. Fusion 107 (2024) 102303, http://dx.doi.org/10.1016/j.inffus.2024.102303, URL https://www.sciencedirect.com/science/article/pii/S1566253524000812.