Building Privacy-Protected and Interpretable Model Surrogates on Synthetic Data

Francesca Naretto
University of Pisa
Computer Science Department
Pisa, Italy
francesca.naretto@unipi.it

Rami Haffar
Intellias Spain SL
Málaga, Spain
rami.haffar@intellias.com

Josep Domingo-Ferrer
Universitat Rovira i Virgili
Dept. of Computer Engineering and Mathematics
Tarragona, Catalonia
josep.domingo@urv.cat

***Abstract*—High-stakes machine learning applications require models that are both privacy-preserving and explainable. However, existing approaches focus on one of these objectives, either protecting the training data at the cost of transparency or providing explanations while relying on the original data. Achieving both properties while maintaining predictive performance remains challenging. In this work, we present ECHOFOREST, a methodology for building interpretable surrogate classifiers without accessing the original data or any derived statistics. ECHOFOREST combines an agnostic synthetic data generation, a decision-tree ensemble trained on the generated data, and local rule-based explanations extracted directly from the surrogate model. The resulting predictor is designed for trustworthy release: it remains detached from the original training data, it reproduces faithfully the behavior of the target model, and can be inspected through decision rules, exploiting a post-processing step that summarizes local explanations by selecting a compact subset of most important rules. We validate ECHOFOREST on twenty tabular datasets, showing strong fidelity and competitive performance compared with state-of-the-art synthetic data generation methods that rely on access to the original training data.**

***Index Terms*—Trustworthy machine learning, Synthetic data, Model fidelity, Interpretability**

I. INTRODUCTION

The EU Artificial Intelligence (AI) Act mandates Artificial Intelligence systems to meet requirements beyond accuracy, such as transparency, accountability, and data protection [25]. Other jurisdictions also promote responsible AI[1], but there is still a gap between regulations and practical techniques. A central challenge in predictive machine learning (ML) is to distribute a surrogate predictor that remains faithful to the original predictor but is detached from the original predictor's training data and understandable enough to be inspected by external users. In this work, we address this problem considering both data protection and interpretability.

Regarding data protection, existing techniques such as k-anonymity [24] and Differential Privacy (DP) [12], [6] aim to safeguard user information. However, these approaches often lead to a significant reduction in data utility, making them difficult to adopt in real-world scenarios [4], [3]. An alternative is the generation of synthetic data, but most data synthesizers require direct access to the original training data or to statistics derived from them, which makes them less suitable for obtaining a surrogate detached from the original training data. Recent approaches such as DP-CTGAN [7] mitigate this issue only partially, since they still rely on private data access during generation and may additionally inherit the utility loss typical of DP-based protection.

Regarding interpretability, most of the abundant literature focuses on *post hoc* explanations generated after model training [5]. These methods aim to understand the behavior of the model from an external perspective. However, this can limit the fidelity of the explanations, reducing user trust in AI systems. For this reason, there is growing interest in developing AI models that are interpretable *by design*, where the internal decision process is transparent and understandable. These models perform classification while providing explanations that are faithful by construction. Unfortunately, they typically rely on the training data and the knowledge extracted during training to construct explanations, which can expose sensitive information. Several studies have shown that explanations can compromise the privacy of users in the training set [20], [16].

In contrast, our goal is to develop an AI system that supports trustworthy release. Our target scenario is a company that owns private data and trains a predictive ML model, *e.g.*, a neural network (NN), that offers high prediction accuracy, but is opaque and tightly coupled to private training data. For external use, the company would benefit from a surrogate that is not trained on the original data or on statistics derived from them and that remains faithful to the original predictor. We propose ECHOFOREST, a methodology that builds such an interpretable surrogate model.

Our approach consists of three steps. First, we generate *synthetic records* without using any statistical information from the original data, exploiting only the prediction probability vectors of the original model. The intuition is that the probability vectors output by the model when records similar to the original ones are fed as input can be leveraged to mimic the model behavior. The resulting synthetic data are detached from the original records and can be used to train a surrogate. Additional privacy mechanisms, such as k-anonymity or DP, can optionally be incorporated during synthetic data generation. Second, the synthetic data are used

[1]China, Japan

to train an ensemble classifier based on decision trees. Third, we provide interpretability by extracting local explanations directly from the surrogate model. The explanations correspond to the decision rules supporting the final prediction and are faithful by construction. Furthermore, users can choose between the complete set of supporting rules and a compact explanation obtained through a selection step that extracts the most reliable and diverse rules. From a data protection perspective, our goal is to come up with a surrogate that is empirically much less vulnerable than the original model, for example, to membership inference attacks.

We evaluated ECHOFOREST using various synthetic data generation methods and compared it with competitors, performing an extensive experimental analysis on 20 tabular data sets. The results empirically demonstrate that ECHOFOREST has good generalization capabilities, with high fidelity to the original model being mimicked. The performance of our method is better than or on par with that of competitors, despite the fact that the latter leverage information about the original data during training. In addition, privacy attacks on the released surrogates remain close to random guessing, even when the original models exhibit non-negligible vulnerability.

In the remainder of the paper, Section II presents background on privacy protection, synthetic data generation, and explanation methods . In Section III, we provide a detailed description of our ECHOFOREST methodology. Section IV reports extensive experiments to validate our approach. Finally, Section V concludes the paper and discusses future directions.

II. BACKGROUND

In this section we discuss privacy protection techniques, competitors for synthetic data generation considered in our evaluation, and interpretable explanations.

A. Privacy protection

In the usual case that training data contain personal data, they should comply with data protection regulations, such as the EU General Data Protection Regulation [26]. Several privacy guarantees and techniques have been proposed to anonymize personal data and turn them into non-personal data for safe release and use in AI training [2]. A well-known privacy guarantee is k-anonymity, along with its l-diversity extension. k-Anonymity protects against re-identification by ensuring that each record is indistinguishable from at least k others in terms of the quasi-identifier attributes that might be used to link with external identified data sources [24]. l-Diversity adds protection against the disclosure of sensitive attributes [12]. Enforcing k-anonymity involves data modification (*e.g.*, using techniques such as generalization, suppression, or microaggregation) and therefore utility loss for AI training.

Another privacy model is Differential Privacy (DP) [6], which adds calibrated noise to the results of a query. This ensures that the inclusion or exclusion of any individual data has a limited impact on the query output, quantified by the privacy budget ϵ. A smaller ϵ provides stronger privacy but requires more noise, reducing the utility of the data. In fact, DP faces challenges when applied to anonymizing training data for AI. When applying DP to tabular data, strong privacy (low ϵ) can introduce substantial distortion, especially for high-dimensional data [3]. Moreover, model training often involves iterative updates on the same data set, increasing cumulative privacy loss and requiring additional noise, which further degrades model accuracy.

In summary, directly enforcing the above privacy guarantees on AI training data can be difficult. For this reason, our approach relies on an *ad hoc* synthetic data generation procedure.

B. Synthetic data generation

In recent years, synthetic data generation has become a key approach to address privacy concerns and the scarcity of real data. Classical approaches generate synthetic data by estimating the statistical structure of the original data set and then sampling the resulting model, for example, using mixture models or copulas [17]. More recent methods rely on deep generative architectures, such as generative adversarial networks (GANs), which learn to produce realistic samples through adversarial training [27].

To better balance utility and privacy, several *differentially private* tabular synthesizers have been proposed. DP-GAN variants, such as DP-CTGAN inject noise during training to provide formal privacy guarantees, while conditioning-based methods, such as ADS-GAN, aim to reduce identity disclosure by restricting generation to a subset of attributes [7], [29]. Beyond GANs, other approaches for DP tabular synthesis include methods that privately estimate low-dimensional marginals or contingency tables and then reconstruct a synthetic data set, such as MST, AIM, and PRIVBAYES [13], [14], [30].

Even if these synthesizers address privacy protection, they are based on original training data and may still be vulnerable to emerging privacy attacks of different kinds [21]. For this reason, we propose a data-agnostic synthetic data generation technique for tabular data.

C. Explainable artificial intelligence

The increasing complexity and deployment of AI systems in various domains have increased the need for interpretability, and explainable AI (XAI) has received growing attention in the research community [5]. A first distinction in XAI is between methods interpretable by design, designed to be transparent by construction, and post-hoc explainers, which provide explanations after the model has been trained. A second distinction is between *local* methods, which are instance-specific interpretations describing the decision-making for individual inputs [1], and *global* methods that tackle the overall behavior of the model across the data set.

A general technique for post-hoc explanation is to exploit surrogate interpretable models, such as linear models (*e.g.*, LIME [18]) or rule-based models [9]. These approaches only approximate the behavior of complex opaque models. Several criteria have been proposed to assess explanations, including fidelity and comprehensibility; however, surrogate explainers still face nontrivial trade-offs across such desiderata.

In [10], the authors proposed a surrogate interpretable model based on random forests (RF). This explainer delves into the forest and examines each individual tree, computing the importance of features from all the trees that align with the opaque model's decision. Then, it calculates the average feature importance across these trees. This line of work highlights the potential of forest-based surrogates to provide faithful, model-grounded explanations, while also raising the practical question of how to present such explanations at an appropriate level of detail. In our setting, we address this point by exposing a *modular* explanation interface, allowing users to choose the desired granularity (from a compact headline explanation to richer supporting evidence) depending on their preferences and application needs.

III. Methodology

We present EchoForest, a methodology for creating trustworthy surrogate predictors. The main structure of our approach is depicted in Figure 1: (1) generate synthetic records, without relying on the original data or any derived statistics (Section III-A); (2) train an interpretable random forest surrogate on the synthetic data, with fidelity to the original model as the main objective (Section III-B); and (3) extract local rule-based explanations from the surrogate and summarize them through a selection step (Section III-C).

A. Synthetic data generation

First, a synthetic data set is generated on which the interpretable surrogate can be trained. The goal is to explore the feature space while avoiding any use of the original training data or derived statistics, which could increase the risk of privacy breaches. No assumptions are made about data distributions or feature ranges. The only information used is the number of input features of the opaque model and the prediction probability vectors returned when it is queried, as in a platform-as-a-service AI setting. Our data generation procedure is detailed in Algorithm 1. Since no information about the original data distribution is available in our setting, our generation cannot rely on data set specific assumptions. To address this, EchoForest adopts an *exploration-exploitation* strategy, alternating between trying new candidate distributions and reusing those that performed well in previous iterations. This allows the generator to focus on the samplers that better approximate the decision regions of the target model, without relying on data statistics. The generator uses a modular pool of candidate distributions covering univariate, multivariate, and mixture-based samplers. In addition, this pool can be extended or restricted, depending on the application setting.

In Algorithm 1, we initialize variable *exploitation* to *False*, as no historical data are available during the first iteration. Then, in the while loop (line 4), each iteration follows one of two paths: if $exploitation$ is $False$, we randomly select a distribution for synthetic data generation (line 6); otherwise, we leverage the generation method that has obtained the best results so far (line 8). After selecting the generation process through the exploration-exploitation approach, we randomly choose a number of records to generate within the range $(1, N/2)$. This choice promotes iterative generation and supports the exploration-exploitation mechanism.

Once $data$ is generated, we apply a *filtering process* (Algorithm 2) to discard *lower-quality* records and rank each generation for potential reuse during the exploitation phase. To filter the generated records, we analyze the prediction probability vectors returned by the opaque model. For an input x, we denote by:

$$bb(x) = V(x) = \langle v_1, \ldots, v_C \rangle \in [0,1]^C \tag{1}$$

the probability vector returned by the opaque model, where C is the number of classes. Following [23], vectors with a sharp peak for one class indicate high confidence and suggest proximity to the data space of the original data. In contrast, a more evenly distributed vector reflects uncertainty and likely corresponds to points outside the training distribution.

From this intuition, we define four selection scores based on queries to the opaque model, computed on $V(x)$: ENTROPY, MARGIN, LOGIT, and KAPPA. The ENTROPY score is defined as $filter_{entropy}(x) = -\sum_{i=1}^{C} v_i \log(v_i)$, after standard numerical clipping. It measures the overall uncertainty of the full prediction probability vector. In this way, it captures not only the top prediction, but also how the remaining probability mass is distributed across classes. The MARGIN score measures the gap between the two most likely classes in the prediction probability vector. It is high when one class clearly dominates the second-best alternative, and low when the classifier is less certain. The LOGIT score follows the same intuition as MARGIN, but is based on the logit space rather than directly on probabilities. In this way, the score is more sensitive to relative differences with respect to MARGIN. The KAPPA score also relies on the top-two class separation, but rescales it by the residual probability mass, accounting for how much uncertainty remains outside the two leading classes.

These filtering strategies capture different aspects of the prediction profile. ENTROPY exploits the full probability vector and is well suited to identifying broad uncertainty patterns, although it is less specifically tied to the decision boundary. MARGIN focuses on the separation between the two most likely classes but ignores the remaining probability mass. LOGIT preserves the same focus while measuring the separation in logit space, making it more sensitive when probabilities are strongly compressed. KAPPA rescales the top-two separation by considering how much probability mass remains outside these classes. As a result, it distinguishes between vectors with a similar top two gap but a different overall uncertainty structure. Together, these filters provide different criteria for deciding which queried synthetic records to retain before training the surrogate.

After obtaining the $filtered$ data, the primary algorithm (Algorithm 1) extracts relevant statistical information and stores it in $history$, along with the type of generation method used (*e.g.*, Wald, Zipf, Normal). This process allows us to record the performance, enabling the exploitation phase to leverage the most effective configuration identified so far. The

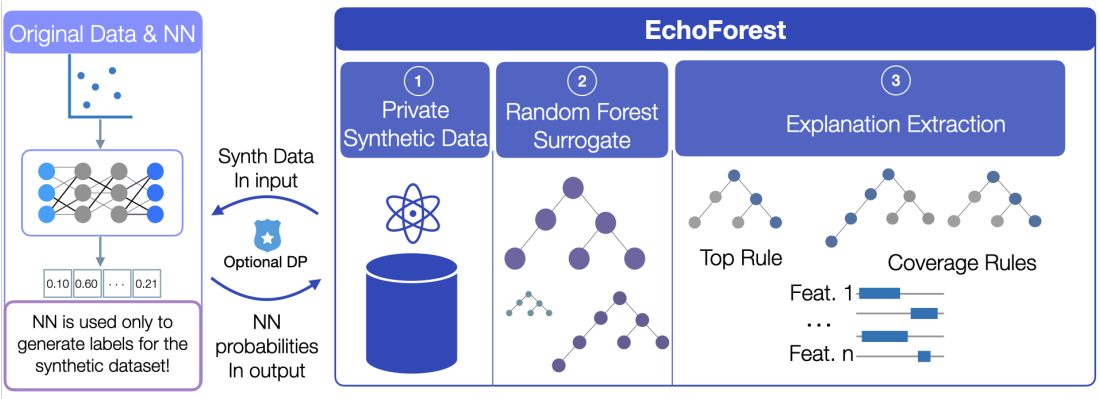


Fig. 1. The ECHOFOREST methodology. BB stands for the opaque ML model.

main algorithm ends when N records are generated. Up to this point, our methodology provides a practical form of protection by avoiding any direct or statistical use of the original data and by releasing only a surrogate trained on synthetic records. We do not claim that this core procedure provides a formal privacy guarantee. As an additional privacy-enhancing mechanism, ECHOFOREST can perturb the prediction probability vectors returned by the opaque model before they are used by the generator. In the following, we describe this Laplace perturbation as an optional additional layer.

Algorithm 1 PrivateSynthDataGen(N, metric, threshold, B)

Input: N (number of data points), $metric$ (evaluation metric), $thresh.$ (filter threshold), B (opaque model), $\mathcal{D}$ (user-defined set of statistical distributions)
Initialize: $exploitation \leftarrow$ False, $generated \leftarrow 0$, $synth_data \leftarrow [\,]$
Output: $synth_data$ (synthesized data with desired properties)
while $generated < N$ **do**
  if $exploitation =$ False **then**
    $gen \leftarrow$ random_choice($\mathcal{D}$)
  else
    $gen \leftarrow$ select_best($history$)
  end if
  $data \leftarrow$ generate(random$(1, N/2), gen$)
  $filtered \leftarrow$ Filter$(data, metric, thresh., B)$
  $stats \leftarrow$ extract_stats$(data, filtered)$
  $history$.add$((gen, stats))$
  $exploitation \leftarrow$ random(True, False)
  $generated \leftarrow generated +$ len$(filtered)$
  $synth_data \leftarrow synth_data + filtered$
end while

Laplace perturbation during synthetic data generation. Although ECHOFOREST never directly accesses the original training records, the probability vectors returned by the opaque model may still leak information when the model is queried [21]. Even if these attacks are often limited in practice [11], they may still reveal whether specific individuals were part of the training data, thus introducing a potential privacy risk. To investigate whether perturbing these outputs can further reduce the information exposed through model queries, we optionally add Laplace noise to each probability vector before it is used by the generator. We stress that this output-perturbation mechanism should not be interpreted as providing an end-to-end differential privacy guarantee, as model sensitivity and query composition are not accounted for.

Algorithm 2 Filter(data, metric, threshold, B)

Input: $data$ (input data), $metric$ (evaluation metric), $thresh.$ (threshold to filter data), B (opaque model)
Output: Filtered data based on the metric threshold
$prob_vectors \leftarrow B$.predict_proba$(data)$
$evaluation \leftarrow$ rank_metric$(prob_vectors, metric)$
$filtered \leftarrow$ apply_thresh$(data, evaluation, thresh.)$
return $filtered$

Let $V(x) = bb(x)$ be the probability vector as defined in Eq. (1). We produce a protected vector $\tilde{V}(x)$ using the following query mechanism, based on [6]:

1) **Stabilization.** We clip each probability to avoid numerical issues:
$$\bar{v}_i = \text{clip}(v_i, \tau, 1-\tau),$$
where we use a small constant τ (e.g., $\tau = 10^{-6}$). We then renormalize the clipped vector as
$$\bar{v}_i \leftarrow \frac{\bar{v}_i}{\sum_{j=1}^{C} \bar{v}_j}.$$
2) **Noise injection.** We add i.i.d. noise to each component:
$$z_i = \bar{v}_i + \eta_i, \qquad \eta_i \sim \text{Laplace}\left(0, \frac{\Delta}{\epsilon}\right). \tag{2}$$
We use the conventional Laplace parameterization Δ/ϵ to control the amount of perturbation. In our experiments, ϵ controls the noise level.
3) **Projection to a valid probability vector.** Since noise may break non-negativity and the unit-sum constraint, we project back to the probability simplex: $\tilde{V} = \Pi_{\Delta_C}(z)$, where $\Delta_C = \{u \in \mathbb{R}^C : u_i \geq 0,\ \sum_i u_i = 1\}$.

In the case of the optional application of DP, all the other components of ECHOFOREST data generation, *i.e.* record filtering, the exploration-exploitation procedure, are computed from $\tilde{V}(x)$ and not anymore from $V(x)$. This introduces a tunable perturbation–utility trade-off controlled by ϵ.

B. Model generation

After generating the synthetic data set, we train a Random Forest (RF), chosen for its strong performance on tabular data and its ability to produce rule-based explanations (Section III-C). Given opaque model bb, we label $synth_data$ by querying $y_synth = bb(synth_data)$, and perform a grid search to train a surrogate RF classifier, $EchoForest(\cdot)$, that closely mimics bb while maintaining generalization. In our setting, the goal of ECHOFOREST is not to solve the original task with a new predictor, but to release a surrogate that is as close as possible to the original opaque predictor.

Algorithm 3 Explain(x, rf, MODE)

Input: x (explained instance), rf (surrogate)
Params: θ (max Jaccard overlap), k (top-k), $k_{\max}$, $c^\star$ (coverage target)
Output: E (selected, simplified supporting rules)
$\hat{y} \leftarrow rf(x)$ {majority vote}
$S \leftarrow [\,]$ {supporting rules with scores}
for each tree t in rf **do**
  if $t(x) = \hat{y}$ **then**
    $r \leftarrow \mathrm{ExtractRule}(t, x)$
    $score(r) \leftarrow \dfrac{purity(r) \cdot support(r)}{1 + length(r)}$
    $S \leftarrow S \cup \{r\}$
  end if
end for
sort S by decreasing $score(\cdot)$
$E \leftarrow [\,]$, $C \leftarrow 0$
for each r in S **do**
  if $\max_{r' \in E} \mathrm{Jaccard}(F(r), F(r')) \leq \theta$ **then**
    $E \leftarrow E \cup \{r\}$
    $C \leftarrow 1 - \prod_{u \in E}(1 - support(u))$
  end if
  if MODE = TOPK **and** $|E| = k$ **then**
    break
  end if
  if MODE = COVERAGE **and** ($C \geq c^\star$ **or** $|E| = k_{\max}$) **then**
    break
  end if
end for
for each rule r in E **do**
  $r \leftarrow \mathrm{PrunePremises}(r)$ {merge repeated per-feature bounds}
end for
return E

C. Explanation extraction

Since ECHOFOREST is based on a RF, it naturally admits a rule-based interpretation: each tree prediction can be expressed as a conjunction of feature-threshold tests along the path from the root to a leaf. We leverage this property to extract local explanations directly from the surrogate, rather than relying on post-hoc approximations. For the explanations, we consider only rules coming from trees that support the final ensemble prediction, following the ensemble-based approach adopted in [8]. Algorithm 3 summarizes the procedure of explanation extraction. For an input x, let $\hat{y} = EchoForest(x)$ be the majority-vote prediction of the surrogate. For each tree t_i in ECHOFOREST, we compute the individual prediction $\hat{y}_{t_i} = t_i(x)$. If $\hat{y}_{t_i} = \hat{y}$, the corresponding decision rule $R_{t_i}(x)$ is extracted and then added to the supporting set:

$$exp_rules(x) = \bigcup_{t_i(x) = EchoForest(x)} R_{t_i}(x). \tag{3}$$

Thus, this set contains all the rules in the ensemble that support the final prediction. $exp_rules(x)$ can be very large, with dozens to hundreds of rules, and is therefore not directly usable as an explanation. We address this limit by (1) ranking and selecting a compact, diverse subset of supporting rules, (2) simplifying each selected rule by pruning redundant premises, and (3) exposing a *modular* explanation interface that lets users choose the desired level of detail, from a high-level explanation to richer supporting evidence.

*a) **Selection of rules**:* Let $r \in exp_rules(x)$ be a supporting rule, and let $purity(r)$ denote the class concentration at the corresponding leaf, $support(r)$ the fraction of synthetic training instances covered by that leaf, and $length(r)$ the number of premises in the rule. We rank rules according to:

$$score(r) = \frac{purity(r) \cdot support(r)}{1 + length(r)}. \tag{4}$$

Starting from this ranking, we iteratively construct a set of explanations by balancing rule relevance and diversity. At each step, the highest-scoring candidate rule is added only if the overlap between its feature set and those of the already selected rules remains sufficiently low. Formally, if $F(r)$ denotes the set of features appearing in rule r, a candidate rule is retained only if its *Jaccard overlap* with every previously selected rule remains below a user-defined threshold. At the end of this step, we obtain $Rules_{rank}(x)$.

*b) **Pruning of premises**:* Given $Rules_{rank}(x)$, we simplify each rule by removing redundant premises. Technically, a decision path may test the same feature multiple times, resulting in repeated constraints in the rule. We therefore merge repeated bounds for each feature: all constraints of the form $f \leq \tau$ are replaced by a single upper bound $f \leq \min \tau$, while all constraints of the form $f > \tau$ are replaced by a single lower bound $f > \max \tau$. When both a lower and an upper bound are present, they define an interval constraint. This transformation preserves the semantics of the rule while reducing its length.

*c) **Modular interface**:* Different users and application settings may require different levels of explanatory detail. To accommodate these needs, ECHOFOREST provides a modular explanation interface that allows users to choose among multiple explanation modes.

In *top-k mode*, we return exactly k simplified rules from $Rules_{rank}(x)$. This allows users to directly control the amount of supporting evidence they aim at inspecting. Setting $k = 1$ is the most concise explanation, while increasing k progressively provides additional supporting evidence.

In the *coverage-driven mode*, rules are selected from $Rules_{rank}(x)$ until the estimated cumulative coverage of the selected set exceeds a user-defined threshold, or until a maximum budget of $k_{\max}$ rules is reached. Cumulative coverage is approximated using per-leaf training supports and serves as a stopping criterion. This mode aims to provide a compact set of rules that captures most of the available supporting evidence while remaining smaller than $exp_rules(x)$.

Finally, for rapid inspection, we also provide a *feature-view* summary that aggregates $Rules_{rank}(x)$ into per-feature interval constraints. This representation offers the most compact explanation, highlighting the features and value ranges that occur most frequently across the selected supporting rules.

IV. EXPERIMENTS

In this section, we describe the experiments conducted to empirically evaluate ECHOFOREST. The code and reproducibility information is available at Code. For details on computational costs and reproducibility: Reproducibility. [2]

A. Data sets and opaque models

To validate ECHOFOREST, we consider 20 publicly available benchmark tabular classification data sets from different domains, summarized in Table I. We remove missing values and apply one-hot encoding where needed. All features are standardized using StandardScaler.

Each data set D and its labels C are divided into train and test sets (70%–30%), with $(D^{\text{train}}, C^{\text{train}})$ and $(D^{\text{test}}, C^{\text{test}})$. For each train set, we train a neural network (NN) as opaque model, with a grid search and 3-fold cross-validation for hyper-parameter tuning. Results are reported in Table I.

B. Synthetic data generation

After creating the opaque models, we apply our ECHOFOREST methodology. The first step is the generation of the synthetic data sets, described in Section III-A. ECHOFOREST exploits an exploration-exploitation strategy that alternates between different sampling distributions and filters the generated candidates to keep those that best capture the behavior of the opaque model. This design requires: (1) selecting the statistical distributions used for candidate generation (Algorithm 1) and (2) defining the filtering criteria used to discard less informative candidates (Algorithm 2). To keep the approach generic, we use a pool of 19 candidate generators spanning different statistical families, including symmetric, asymmetric, bounded, heavy-tailed, discrete, mixture-based, and multivariate correlated distributions. This diversity allows the generator to explore the feature space under different distributional and geometric biases without relying on statistics from the original data. The pool is user-defined and can be extended or restricted depending on the application setting. The complete set of generators and their parameterizations used in our experiments are provided in the code. This allows the generator to explore the feature space under different geometric biases without relying on statistics from the original data. To filter queried candidates, we evaluate five selection configurations, based on the opaque model's prediction vectors: E25, E50, MARGIN, LOGIT, and KAPPA, described in Section III. We also compare our synthetic data generation with existing privacy-aware approaches: DP-CTGAN (with $\epsilon = 0.5$), MST, AIM, and PRIVBAYES. They were trained on all the original training data set. For all methods, we generate 80000 synthetic records and use the predictions of the opaque model as labels, since the objective is to mimic its behavior. The impact of the size of the synthetic data set is discussed later in Section IV-D.

*a) **Quality of the synthetic data**:* To evaluate the quality of the synthetic data sets generated, we used Synthcity metrics [3]. Figure 2 summarizes the results in all data sets for two of the most relevant metrics: *XGB synth performance* and *nearest-neighbor separation*. The first metric measures the predictive utility of the synthetic data by training an XGB classifier on synthetic samples and evaluating it on real data. Among the evaluated configurations, LOGIT achieved the highest average score (0.69), closely followed by E50 (0.68) and E25 (0.65). All ECHOFOREST variants outperformed competitors.

TABLE I
OVERVIEW OF THE DATA SETS AND MACRO-F1 SCORE OF THE NN USED AS OPAQUE MODEL. 'NUM' DENOTES NUMERICAL FEATURES, 'CAT' CATEGORICAL FEATURES, AND 'MIX' A COMBINATION OF BOTH TYPES.

Data set	Classes	#Feat.	Input	NN Macro F1
INCOME	2	14	mix	0.79 ± 0.02
ACTIVITY	2	42	num	1.00 ± 0.01
POL	2	49	num	0.98 ± 0.03
SPOTIFY	2	17	mix	0.98 ± 0.00
SPOTIFY R	2	17	mix	0.97 ± 0.01
WAVE B	2	40	num	0.86 ± 0.01
WAVE B 2	2	40	num	0.88 ± 0.02
CREDIT	2	20	mix	0.62 ± 0.05
DIAMOND B	2	9	mix	0.83 ± 0.02
HTRU 2	2	8	num	0.94 ± 0.03
LANDSAT B	2	5	num	0.97 ± 0.01
LANDSAT B 2	2	5	num	0.96 ± 0.02
MAGIC	2	10	num	0.84 ± 0.01
HELOC	2	23	num	0.72 ± 0.02
ELEC	2	8	num	0.78 ± 0.01
CALI	2	8	num	0.85 ± 0.00
SPLICE	3	60	cat	0.93 ± 0.03
WAVE M	3	40	num	0.87 ± 0.01
LANDSAT M	6	5	num	0.74 ± 0.02
PENDIGITS	10	16	num	0.77 ± 0.05

[2]The experiments were conducted on a server with Ubuntu 20.04.6 LTS as operating system, equipped with an Intel(R) Xeon(R) Gold 6342 processor (96 cores) and 1.0 TiB of RAM.

[3]https://github.com/vanderschaarlab/synthcity/tree/main

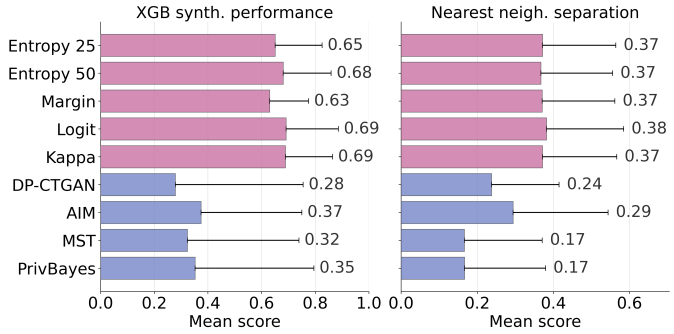


Fig. 2. Quality of the generated synthetic data sets in terms of XGB performance (higher values are better) and nearest-neighbor separation (higher values indicate greater separation from the original data).

The *nearest-neighbor separation* metric measures the distance between synthetic and original data. Values closer to 0 indicate that synthetic records lie very close to the original, while larger values correspond to greater separation. From a privacy perspective, this metric provides an indication of how closely the generated samples resemble records from the original data set. The ECHOFOREST variants achieved the highest separation scores, with LOGIT reaching an average value of 0.38, while MST and PRIVBAYES obtained the lowest values (around 0.17). These results suggest that the synthetic data sets of ECHOFOREST are less coupled to the original samples. However, a larger separation is consistent with our design, which never accesses the original training data and does not explicitly attempt to reproduce its distribution.

We also examined the *common row proportion*, which measures the presence of exact duplicates of training records and is 0 in all ECHOFOREST configurations. In contrast, MST and PRIVBAYES occasionally produced non-zero duplication rates. Furthermore, although ECHOFOREST does not explicitly optimize datatype constraints, the generated data sets remained structurally coherent, with only minor mismatches.

*b) **Ablation study on the statistical distributions.**:* Next, we investigate the impact of the different sampling distributions used during synthetic data generation. Figure 3 shows that the 19 candidate distributions do not contribute equally to the final E25 dataset. In particular, exponential, centered Poisson, and Bernoulli-based samplers account for the highest fraction of accepted samples, whereas other distributions, such as Gaussian mixtures and Student-t, contribute less. This behavior indicates that different data sets benefit from different distributional biases, which supports the use of a diverse pool of generators rather than a single sampling strategy. The standard deviations indicate that the preferred distributional mix is not identical across data sets, which further supports the need for a flexible multi-distribution generator.

C. Model generation

After generating the synthetic data sets, we train a RF surrogate for each generation strategy. Since the goal of ECHOFOREST is to replace the original opaque model, the evaluation focuses on the ability of the surrogate to reproduce the opaque

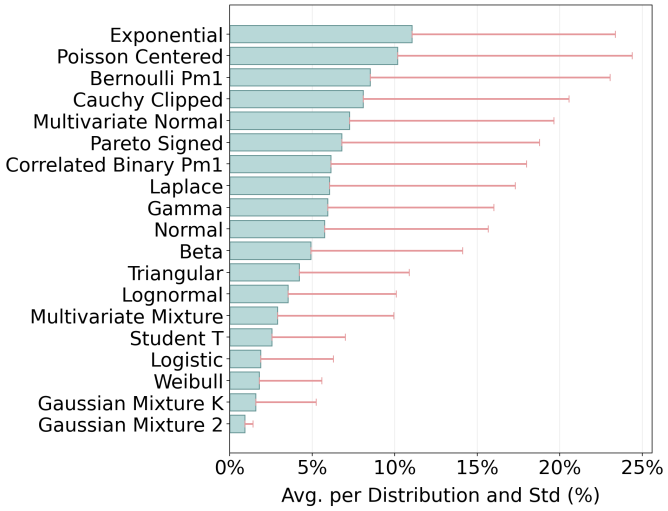


Fig. 3. Accepted synthetic records by distribution for E25. Thick bars report the average proportion of accepted records contributed by each candidate distribution across data sets, while thin bars denote one std error.

TABLE II
GRID SEARCH PARAMETERS FOR ECHOFOREST

Parameter	Values
`bootstrap`	{True, False}
`max_depth`	{3, 5, 10, 20, 30, 50}
`max_features`	{3, 5, 10, 15, `'sqrt'`}
`min_samples_leaf`	{50, 100, 200, 300}
`n_estimators`	{20, 50, 100, 200, 300}
`criterion`	{`'gini'`, `'entropy'`}

model's predictions on original data rather than synthetic data. In the following, we refer to this property as *fidelity*. For each tuple (*data set*, *generation strategy*), we perform a grid search over the random forest hyper-parameters, reported in Table II. Formally, given a synthetic data set D_i^{synth} and the labels $\hat{Y}_{bb} = bb(D_i^{\text{synth}})$ predicted by the opaque NN, we split the synthetic data into training and validation subsets and train the surrogate as $\text{ECHOFOREST}_i = \text{train}(D_i^{\text{synth, train}}, \hat{Y}_{bb})$, after the hyper-parameter selection through grid-search.

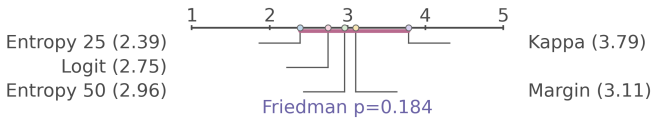


Fig. 4. Critical Difference plot comparing all ECHOFOREST variants in terms of Macro F-1 ranking.

We then evaluate the RF on the original test set by comparing the surrogate predictions against the predictions produced by the opaque model. Figure 4 reports the critical difference (CD) analysis across ECHOFOREST methods. The ENTROPY, LOGIT, and MARGIN configurations achieve the best average ranks, while KAPPA obtains the lowest rank. However, the Friedman test ($p = 0.184$) does not indicate statistically significant differences among our filtering criteria, and all pairwise

TABLE III
MEAN AND STANDARD DEVIATION ACROSS THE ECHOFOREST CONFIGURATIONS AND THE COMPETITOR METHODS.

	E25	E50	MARGIN	LOGIT	KAPPA	AIM	MST	PRIVBAYES	DP-CTGAN
Mean	0.92	0.93	0.91	0.92	0.84	0.81	0.48	0.48	0.46
Std	0.08	0.07	0.09	0.08	0.22	0.22	0.17	0.16	0.20

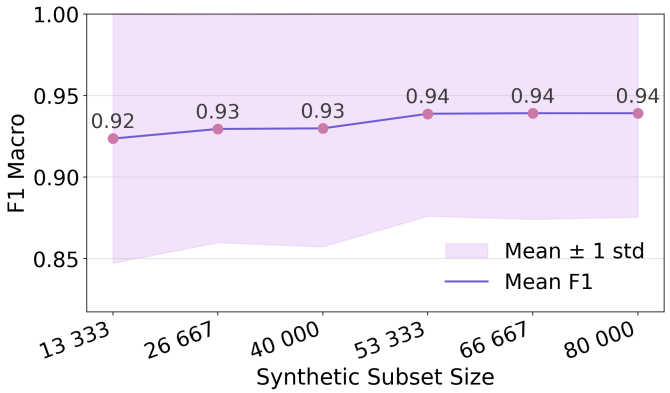


Fig. 5. Ablation study (on INCOME, ACTIVITY, SPOTIFY, ELEC, POL, WAVE M, LANDSAT M, 3-fold cross validation) to evaluate the impact of the number of synthetic records generated, in terms of fidelity to the original model.

rank differences remain below the CD threshold. While this does not allow us to conclude that the filtering criteria are equivalent, it suggests that the behavior of ECHOFOREST is driven primarily by the exploration-exploitation process and is not strongly dependent on a particular filtering strategy. Table III extends the comparison to the competitor methods. We note that the reported metric is surrogate fidelity, i.e., the agreement between the surrogate predictions and those of the NN. While all synthetic data are labeled using the same NN predictions, methods such as DP-CTGAN, MST, AIM, and PRIVBAYES are designed to reproduce the statistical properties of the original data rather than the behavior of a target model. The Table III shows that E25, E50, MARGIN, and LOGIT achieve the highest fidelity scores, all exceeding 0.90 on average. Among the competitors, AIM achieves the strongest performance (0.81), while the others obtain lower scores. The lower performance of DP-CTGAN should also be interpreted considering the privacy budget of $\epsilon = 0.5$. Finally, the ECHOFOREST variants exhibit lower variability across datasets, with standard deviations between 0.07 and 0.09. This result is interesting because it highlights that *our method, despite relying on fewer assumptions than its competitors, performs better or on a par with them in terms of fidelity.*

D. Ablation study on the size of the synthetic data sets.

Figure 5 reports the effect of varying the number of synthetic records generated on the final surrogate performance. In these experiments, we evaluated synthetic data set sizes ranging from approximately 13000 to 80000 samples. The results show that performance stabilizes after approximately 50000 samples, suggesting that the surrogate has converged largely. For this reason, we selected 80000 as a common upper budget for the main experiments, providing a reasonable trade-off between performance and computational cost.

E. Privacy: analysis and protection

This section analyzes the privacy of ECHOFOREST from two perspectives. First, we evaluate its empirical resistance to Membership Inference Attacks (MIAs), assessing whether the released surrogates expose information about the training data. Second, we investigate the effect of Laplace perturbation during synthetic data generation and study the resulting trade-off between perturbation strength and surrogate fidelity.

1) Membership inference attacks: Although the basic version of ECHOFOREST does not provide a formal privacy guarantee, it is important to evaluate whether the generated surrogates expose information about training data in practice. To this end, we consider MIAs against the original opaque model and the corresponding ECHOFOREST surrogates [21], [28], [19], [15]. The goal is to assess whether an attacker can determine if a given record was part of the training set.

We assume a standard machine-learning-as-a-service setting in which the training data remain private, but the attacker can query the model and observe prediction probability vectors. In our setting, we consider a *statistical* adversary who does not have access to the original records and can only rely on aggregate information to construct auxiliary data when needed.

We evaluate *Loss*, following Yeom et al. [28]; *Confidence*, following the opaque model attack setting of Shokri et al. [21]; and *Entropy* and *Margin*, following the score-based evaluation framework of Song and Mittal [22]. These attacks infer membership from statistics of the prediction vector. In addition, we also consider the LiRA attack, using shadow models, trained on statistical data, to obtain a different attack setting.

Figure 6 reports the AUC obtained by the *Loss* attack on both the original opaque model and the corresponding ECHOFOREST. Although we evaluated multiple MIAs, for brevity we only report the results for the Loss attack, which are representative of the overall trends observed. We further restrict the figure to datasets exhibiting a substantial privacy breach in the original opaque model. In all reported cases, the attack becomes less effective against the surrogate, with AUC values moving closer to the random baseline. This suggests that the surrogate attenuates the membership signal of the original model while preserving its predictive behavior. We stress that this should be interpreted as an empirical robustness result and not a formal privacy guarantee.

2) Laplace output perturbation: To evaluate the impact of the proposed perturbation mechanism, we experimented with different privacy budgets ($\epsilon \in \{0.1, 0.5, 1.0, 5.0\}$) (Section III). The results are reported in Figure 7. As expected, stronger output perturbation (smaller ϵ) reduces the utility of the resulting surrogate. The average fidelity Macro-F1 increases from 0.58 at $\epsilon = 0.1$ to 0.72 at $\epsilon = 0.5$, 0.83 at $\epsilon = 1.0$, and 0.86 at $\epsilon = 5.0$. At the same time, the standard deviation decreases as ϵ increases, indicating that stronger output perturbation affects data sets differently; some data sets already achieve satisfactory fidelity at $\epsilon = 0.5$, while others

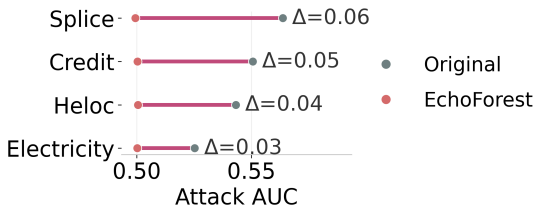


Fig. 6. We report the AUC of *Loss* attacks on ECHOFOREST and on the original opaque model. On ECHOFOREST, AUC drops near 0.50 or below, indicating random performance, and thus good protection.

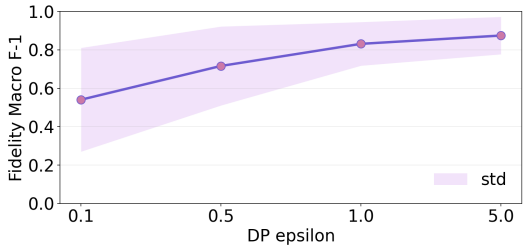


Fig. 7. Laplace perturbation during synthetic data generation. Smaller ϵ values correspond to larger perturbations of the outputs. The results show a clear trade-off: fidelity improves as ϵ increases and stabilize around $\epsilon = 1.0$. Nevertheless, several data sets already achieve satisfactory performance at $\epsilon = 0.5$, visible from the std.

require $\epsilon = 1.0$ to transfer the behavior of the original opaque model after the perturbation applied to its output probabilities. This behavior is expected, as stronger perturbation makes the opaque model harder to mimic. Therefore, ECHOFOREST already reduces direct dependence on the training data by operating without access to the real records or their statistics, while Laplace perturbation further limits the information that can be extracted from the queried probability vectors. Overall, the results suggest that the privacy-utility trade-off becomes favorable between ϵ in $[0.5, 1.0]$ while the values beyond $\epsilon = 1.0$ indicates the beginning of a performance plateau.

F. Explanation extraction

After training ECHOFOREST, we also evaluated its explanatory interface. Explanations extracted from the RF correspond to the actual decision paths followed by instances within the ensemble. In our setting, a natural baseline explanation consists of collecting, for a target prediction, all the rules originating from trees whose predicted class matches the final majority vote of the random forest. This baseline is faithful by construction, as each collected rule contributes to the final decision. However, our empirical analysis shows that this raw set of supporting rules is often too large to be practically useful: for the considered data sets, a single prediction is supported by an average of 92.1 rules, corresponding to 1410.7 premises per explained instance.

To address this issue, we apply the strategy of Section III-C, obtaining a compact and diverse subset of high-quality simplified supporting rules. Since the *top-k* mode requires the user to specify the desired number of rules, in Figure 8 we report the results obtained with the *coverage-driven* criterion, which automatically adapts the number of selected rules to the explained instance. Applying this approach gives more compact explanations: the final explanation contains, on average, 3.74 rules and 45.87 premises per instance, corresponding to reductions of approximately 95.9% in the number of rules and 96.7% in the number of premises compared to the full set. Despite this compression, the selected rules maintain high purity (highly homogeneous splits) indicating that the retained explanation remains strongly aligned with the RF prediction.

Overall, the rule extraction component serves two complementary purposes: on the one hand, it provides faithful local evidence for the forest prediction; on the other hand, it highlights the need for a post-processing step that summarizes such evidence into a smaller and more usable subset of rules. Since these rules are extracted directly from the released surrogate, they expose its internal decision logic and are intrinsically faithful to it. An interactive demonstration of our modular explainer is available at Link.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed ECHOFOREST, a methodology that mimics the behavior of an AI system by providing a surrogate classifier that is both intrinsically interpretable and detached from the original training data. This detachment is obtained through an *ad hoc* synthetic data generation procedure capable of producing novel data for training the surrogate classifier. Our synthetic generation process exploits the behavior of prediction probability vectors, particularly the peaks that occur when the AI system has already seen a record.

We empirically demonstrate that our methodology achieves strong results in terms of fidelity with respect to the original opaque model. Furthermore, our approach achieves results that are better than or on a par with those of our competitors, despite the fact that the competitors use more information about the original dataset than we do.

In future work, we plan to extend ECHOFOREST beyond Random Forests and investigate alternative surrogates, analyzing their trade-offs between fidelity and interpretability. We also intend to further study the explanation component. Our approach reduces explanation complexity, but remains unclear whether the summaries align with user preferences. We therefore plan to evaluate alternative abstraction mechanisms through structural quality indicators, stability criteria, as well as task-specific constraints, as a cost-effective complement to human-subject evaluations.

ACKNOWLEDGMENT

Partial support was received from the Government of Catalonia under the ICREA Acadèmia Prize awarded to J. Domingo-Ferrer, the Agencia Española de Investigación under project "CLEARING-IT", PID2024-157271NB-I0, and the European Union under projects "SoBigDataIP", HORIZON-101292277, and "TANGO", Grant Agreement No. 101120763.

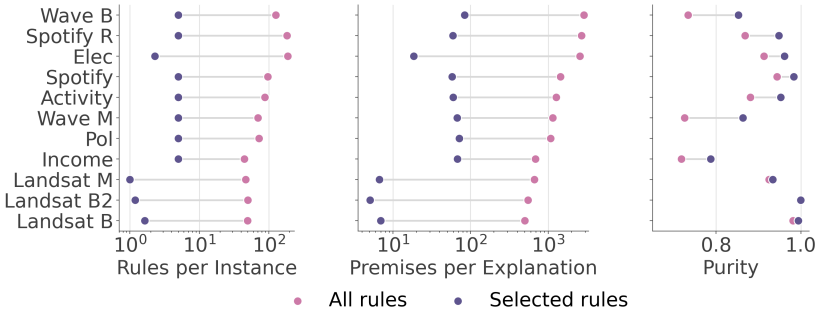


Fig. 8. Rules per instance, premises per explanation, and purity before and after our rule selection procedure.

REFERENCES

[1] Emanuele Albini, Jason Long, Danial Dervovic, and Daniele Magazzeni. Counterfactual shapley additive explanations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1054–1070, 2022.

[2] Mohammed Binjubeir, Abdulghani Ali Ahmed, Mohd Arfian Bin Ismail, Ali Safaa Sadiq, and Muhammad Khurram Khan. Comprehensive survey on big data privacy protection. *IEEE Access*, 8:20067–20079, 2019.

[3] Alberto Blanco-Justicia, David Sánchez, Josep Domingo-Ferrer, and Krishnamurty Muralidhar. A critical review on the use (and misuse) of differential privacy in machine learning. *ACM Computing Surveys*, 55(8):1–16, 2022.

[4] Jeremiah Blocki and Ryan Williams. Resolving the complexity of some data privacy problems. In *Automata, Languages and Programming: 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part II 37*, pages 393–404. Springer, 2010.

[5] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, 37(5):1719–1778, 2023.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[7] Mei Ling Fang, Devendra Singh Dhami, and Kristian Kersting. DP-CTGAN: differentially private medical data generation using ctgans. In *Artificial Intelligence in Medicine, AIME 2022*, volume 13263 of *Lecture Notes in Computer Science*, pages 178–188. Springer, 2022.

[8] Rami Haffar, Josep Domingo-Ferrer, and David Sánchez. Explaining misclassification and attacks in deep learning via random forests. In *Modeling Decisions for Artificial Intelligence: 17th International Conference, MDAI 2020, Sant Cugat, Spain, September 2–4, 2020, Proceedings 17*, pages 273–285. Springer, 2020.

[9] Rami Haffar, Francesca Naretto, David Sánchez, Anna Monreale, and Josep Domingo-Ferrer. Glor-flex: Local to global rule-based explanations for federated learning. In *Proceedings of the 2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2024.

[10] Rami Haffar, David Sanchez, and Josep Domingo-Ferrer. Explaining predictions and attacks in federated learning via random forests. *Applied Intelligence*, 53(1):169–185, 2023.

[11] Mona Khalil, Alberto Blanco-Justicia, Najeeb Jebreel, and Josep Domingo-Ferrer. Membership inference attacks beyond overfitting, 2025.

[12] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *Acm transactions on knowledge discovery from data (tkdd)*, 1(1):3–es, 2007.

[13] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *CoRR*, abs/2108.04978, 2021.

[14] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. Aim: an adaptive and iterative mechanism for differentially private synthetic data. *Proc. VLDB Endow.*, page 2599–2612, 2022.

[15] Anna Monreale, Francesca Naretto, and Simone Rizzo. Agnostic label-only membership inference attack. In *17th International Conference on Network and System Security*. Springer, 2023.

[16] Francesca Naretto, Anna Monreale, and Fosca Giannotti. Evaluating the privacy exposure of interpretable global and local explainers. *Transactions on Data Privacy (TDP)*, 2025.

[17] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410. IEEE, 2016.

[18] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " 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*, pages 1135–1144, 2016.

[19] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

[20] Reza Shokri, Martin Strobel, and Yair Zick. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021.

[21] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[22] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[23] Liwei Song, Reza Shokri, and Prateek Mittal. Membership inference attacks against adversarially robust deep learning models. In *2019 IEEE Security and Privacy Workshops (SPW)*, 2019.

[24] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[25] UE. The artificial intelligence act. https://artificialintelligenceact.eu/the-act/, 2023. Accessed: 2023-11-28.

[26] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[27] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems 32, NeurIPS 2019*, pages 7333–7343, 2019.

[28] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018.

[29] Van der Schaar M. Yoon J, Drumright LN. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE J Biomed Health Inform.*, 24(8), 2020.

[30] Jun Zhang, Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes : private data release via bayesian networks. *ACM Transactions on Database Systems*, 2017.