AUTOLYCUS: Exploiting Explainable Artificial Intelligence (XAI) for Model Extraction Attacks against Interpretable Models

Abdullah Caglar Oksuz
Case Western Reserve University
Cleveland, Ohio, USA
abdullahcaglar.oksuz@case.edu

Anisa Halimi
IBM Research - Dublin
Dublin, Ireland
anisa.halimi@ibm.com

Erman Ayday
Case Western Reserve University
Cleveland, Ohio, USA
erman.ayday@case.edu

ABSTRACT

Explainable Artificial Intelligence (XAI) aims to uncover the decision-making processes of AI models. However, the data used for such explanations can pose security and privacy risks. Existing literature identifies attacks on machine learning models, including membership inference, model inversion, and model extraction attacks. These attacks target either the model or the training data, depending on the settings and parties involved.

XAI tools can increase the vulnerability of model extraction attacks, which is a concern when model owners prefer black-box access, thereby keeping model parameters and architecture private. To exploit this risk, we propose AUTOLYCUS, a novel retraining (learning) based model extraction attack framework against interpretable models under black-box settings. As XAI tools, we exploit Local Interpretable Model-Agnostic Explanations (LIME) and Shapley values (SHAP) to infer decision boundaries and create surrogate models that replicate the functionality of the target model. LIME and SHAP are mainly chosen for their realistic yet information-rich explanations, coupled with their extensive adoption, simplicity, and usability.

We evaluate AUTOLYCUS on six machine learning datasets, measuring the accuracy and similarity of the surrogate model to the target model. The results show that AUTOLYCUS is highly effective, requiring significantly fewer queries compared to state-of-the-art attacks, while maintaining comparable accuracy and similarity. We validate its performance and transferability on multiple interpretable ML models, including decision trees, logistic regression, naive bayes, and k-nearest neighbor. Additionally, we show the resilience of AUTOLYCUS against proposed countermeasures.

KEYWORDS

Model extraction attacks, explainable artificial intelligence, XAI, privacy attacks in machine learning, adaptive retraining

1 INTRODUCTION

Machine Learning (ML) has become a crucial tool for many businesses, enabling them to analyze data and make predictions with greater accuracy and efficiency. To make ML accessible to a wider range of businesses, cloud service providers offer cost-effective Machine Learning as a Service (MLaaS) platforms, providing customers with pretrained models or tools to deploy their own models.

As the adoption of MLaaS platforms grew, there has been a corresponding increase in the demand for tools that facilitate explainable Artificial Intelligence (XAI) [4]. These tools are crucial in providing users with transparency and a comprehensive understanding of how decisions are made by ML models [7]. The lack of interpretability and transparency in models that inherently operate as black boxes, such as neural networks, or interpretable models operating in black-box settings–which we specifically focus on in this paper– (i.e., where users have limited or no access to the internal workings of a model) can foster distrust and hinder their adoption.

Local Interpretable Model-agnostic Explanations (LIME) [41] and SHapley Additive exPlanations (SHAP) are two popular XAI techniques utilized by major MLaaS platforms such as Azure Machine Learning (Responsible AI) [2], Watson Machine Learning (OpenScale) [11], Amazon SageMaker (Clarify) [47], and Google Cloud (Vertex Explainable AI) [10]. LIME generates local explanations for any ML model prediction by creating a new model around the sample of interest. Whereas, SHAP computes global feature importance values using Shapley values [33] from cooperative game theory [37] and a local approximation of the model to compute feature importance values for the entire dataset.

The monetization of MLaaS and increasing reliance on the models provided therein have opened up new risks for businesses. One such risk is the threat of inference attacks including membership inference [51], model inversion [16], and model extraction [55]. In particular, threat of model extraction (or stealing) attacks proposed by Tramer et al. [55] on MLaaS platforms can result in theft of proprietary models, valuable intellectual property, and the loss of monetary gains. In such attacks, an adversary attempts to extract the target ML model, either in whole or in part, by training a surrogate model using a set of queries to the target model. In addition, an attacker can execute a model extraction attack more easily using the information provided by XAI tools. For example, Milli et al. [34] introduced the first model extraction attack that exploits explanations to solve the hyperplanes in neural network image classifiers with the help of gradient based explanations. Chandrasekaran et al. [8] demonstrated that the model extraction attacks can be formulated as active learning. In addition to privacy concerns, Rudin [44] proposed that the use of interpretable models should be increased instead of explaining black-box (non-interpretable) models for high-stake decisions. This is to ensure accountability and facilitating human understanding and trust in AI systems.

Although the exploitation of AI explanations for adversarial machine learning is a recent area of investigation, there have been several studies [24, 34, 35, 56, 59] conducting model extraction attacks with XAI guidance on neural networks and image classification tasks. These studies predominantly rely on image classifier-specific

Proceedings on Privacy Enhancing Technologies 2024(4), 684–699

https://doi.org/10.56553/popets-2024-0137

explanation tools (e.g., class activation maps (CAMs) [46, 63]), which provide richer, gradient-based information compared to model-agnostic explanations. While insightful, the exclusive focus on neural networks and image classifiers within these studies overlooks the potential insights and advancements that could be gained from exploring alternative model architectures and diverse data settings. Reliance solely on explanation methods suitable for neural networks may be incompatible with certain model types or inadvertently disclose excessive model information for the less complex interpretable models, effectively transforming the attacks into semi-white-box scenarios. In contrast, our objective is to conduct a model-agnostic attack with minimal interference across multiple interpretable models. Besides, the studies targeting interpretable models, which are still highly used in ordinary statistical modeling and data analysis due to their simplicity, are limited. Thus, exploring the vulnerabilities that AI explanations present to such models, especially under black-box settings, is an uncharted territory.

In this work, we propose a novel, model-agnostic, retraining (learning) and perturbation based model extraction attack framework called AUTOLYCUS that exploits XAI to target interpretable models in black-box access settings. As a prior-guided adversarial-example crafting method, AUTOLYCUS leverages model explanations on local decision boundaries and feature importances. SHAP offers feature importances while LIME offers both feature importances and decision boundaries. This can be exploited to generate membership queries.

AUTOLYCUS is influenced by the line-search retraining and adaptive retraining methods, designated as the main strategies in label-only scenarios, as proposed by Tramer et al. [55]. Line-search retraining sends adaptive membership queries to the vicinity of the decision boundaries of a model based on previous queries, contrary to the uniform queries of other label-only attacks [9, 38]. AUTOLYCUS differs from the aforementioned attacks in terms of how it utilizes the explanations. AUTOLYCUS does not directly seek decision boundaries. Instead, it utilizes line-search within the vicinity of explanations to generate samples that accurately reflect the decision boundaries during the retraining. While AUTOLYCUS can leverage external resources such as additional datasets or linkage data as prior information for the adversary, we limit access to black-box and constrain the information disclosed by explanations. While approaches like Equation Solving and Path-finding [55] suffer from auxiliary white-box model information and realism, our approach imposes lighter assumptions on the adversary compared to similar works [34, 38, 55] retaining practicality and realism.

We demonstrate the effectiveness of AUTOLYCUS on different ML datasets in terms of the accuracy of the constructed surrogate model and the similarity of the surrogate model to the target one. We show that AUTOLYCUS provides high accuracy and similarity for interpretable models like decision trees, logistic regression, naive bayes, and k-nearest neighbor. Additionally, we demonstrate that AUTOLYCUS outperforms the SOTA methods and remains effective even in the presence of countermeasures.

Contributions Main contributions of AUTOLYCUS are as follows.

- AUTOLYCUS can be initiated with minimal prerequisites (e.g., minimal hyperparameter knowledge and auxiliary data), making it an easily deployable and efficient attack.
- AUTOLYCUS can produce high-fidelity and high-accuracy reconstructions of the target model with fewer queries compared to the exact reconstruction attacks such as Equation-Solving [55], Path-Finding [55], and Lowd-Meek [31].
- AUTOLYCUS is a retraining based attack. It can create partially extracted surrogate models with comparable accuracy and similarity even when it is under hard query limitations.
- To the best of our knowledge, AUTOLYCUS is the first model extraction attack that exploits AI explanations for targeting interpretable models with black-box access.
- To the best of our knowledge, AUTOLYCUS is the first model extraction attack that exploits model agnostic AI explanation tools.

The rest of the paper is organized as follows. In the next section, we go over the related work. In Section 3, we provide background on XAI. In Section 4, we describe the considered system and threat models. In Section 5, we describe AUTOLYCUS system model in detail. In Section 6, we provide the evaluation results. In Section 7, we show the performance of AUTOLYCUS in the existence of potential countermeasures. In Section 8, we discuss potential extensions and limitations. Finally, in Section 9, we conclude the paper.

2 RELATED WORK

We review two primary lines of related research: (i) privacy attacks in machine learning and (ii) privacy risks due to model explanations.

Privacy attacks in ML. Research on machine learning privacy has identified several attacks, such as membership inference [9, 30, 45, 51] to determine whether a given user is part of the training dataset; attribute inference [18, 53] to infer additional private attributes of a user based on the observed ones; model inversion [6, 16, 61] to reconstruct data from a training dataset; and model extraction [55] to infer the model parameters/hyperparameters and/or the architecture of a model. In this work, we focus on model extraction (or stealing) attacks where an attacker tries to extract sensitive information or intellectual property from a trained model. The attacker typically has access to the model's input-output behavior and uses this information to build a replica of the model (functionality extraction) [36, 49] or extract other valuable information from it (attribute extraction) [13, 22, 55, 57]. Tramer et al. [55] propose the first model stealing attack considering both white-box and black-box access to ML models. Wang et al. [57] propose a model extraction attack that targets the hyperparameters of ML models. Papernot et al. [38] propose a model extraction attack against black-box DNN classifiers. Such attacks are effective against a variety of machine learning models, including decision trees, deep neural networks, and support vector machines [21, 34, 55, 57].

Privacy risk due to model explanations. While XAI techniques are useful for the interpretation of the prediction provided by the machine learning models, they can also be leveraged to enhance privacy attacks. Shokri et al. [50] study the privacy risk due to membership inference attacks based on feature-based model explanations and show how various explanation methods leak information about the training data. Zhao et al. [62] analyze how explanation methods enhance model inversion attacks. Their work focuses on image datasets like MNIST [29] and CIFAR-10 [28], and shows that the accuracy of the model inversion attacks increases significantly

when explanations are used. Kariyappa et al. [24], Milli et al, [34], Miura et al. [35], Truong et al. [56], and Yan et al. [59] propose various model extraction attacks by exploiting gradient based AI explanations (e.g., Saliency Maps [52], Class Activation Maps [63]) on black-box image classifiers. Aïvodji et al. [1] utilize counterfactual explanations for model extraction attacks. In this paper, in contrast to Shokri and Zhao's works, we focus on model extraction attacks exploiting explanations.

3 BACKGROUND ON EXPLAINABLE AI

Explainable AI (XAI) is an approach for developing artificial intelligence systems that are transparent, interpretable, and can provide clear explanations for their decision-making processes [12, 19]. The goal of XAI is to ensure that AI systems are trustworthy, fair, and accountable, and that they can be used to inform human decision-making in a reliable and understandable manner. Traditional AI algorithms, like deep learning neural networks, can be very effective at making decisions based on large amounts of data, but they are often viewed as "black boxes" because their internal workings are difficult to understand. This can lead to concerns about bias, discrimination, or errors in decision-making [44]. XAI aims to address these concerns by integrating techniques for transparency and interpretability into the design of AI systems, including those that are already deemed interpretable. This may involve using simpler models, providing visualizations or natural language explanations, or detecting anomalous behavior.

In the literature, various XAI techniques have been developed for improving the interpretability and transparency of machine learning models, particularly for neural network models in computer vision. Examples of such techniques include Layer-wise Relevance Propagation (LRP) [3], Deep Taylor Decomposition (DTD) [27], Pattern Difference Analysis (PDA) [64], Testing with Concept Activation Vectors (TCAV) [26], and Explainable Graph Neural Networks (XGNN) [60]. For interpretable models such as decision trees, XAI techniques such as Anchors [42], SHAP [33], and LIME [41] can be utilized to provide explanations.

For further information about the techniques utilized in major MLaaS platforms and why LIME and SHAP are primarily focused in this paper, please refer to the Appendix.

Local Interpretable Model-Agnostic Explanations (LIME) LIME [41] is a widely used XAI tool by model owners to provide an understanding of the rationale behind machine learning models predictions. This is accomplished by generating local approximations to the models around a specific instance of interest or a sample. For instance, To provide explanations for a sample X on model M, LIME creates a dataset of different $\hat{X}$s which are the perturbed versions of X. This perturbation is executed differently depending on the data type, such as through pixel masking in image data or word replacement in text data. Then, each sample in the perturbed dataset is classified by M. Following this, LIME fits a local linear model around X, using $\hat{X}$s and their classifications by M. The resulting linear model has its own local decision boundaries to explain both linear and non-linear boundaries of M, allowing samples to be created and exploited within its vicinity. The feature weights in the local linear model denote the relative importance of each feature in M's prediction, presented through means such as a heat-map or feature importance scores. It is important to note that the decision boundaries of the local linear model generated by LIME might not exactly match M's exact decision boundaries due to non-linearity. But, for interpretable models, these local explanations are more accurate and consistent compared to the explanations provided to black-box models like neural networks.

Shapley Values (SHAP) SHAP utilizes Shapley values [48] of a sample to determine the marginal contribution of each feature. These values are then presented as explanations of that particular sample on a given prediction. In predictions, information released by SHAP can vary drastically. For instance, it can either only release the corresponding Shapley values per class - which is the most realistic scenario considering black-box access - or it can provide a plot that displays every sample in the training dataset along with the particular sample. Needless to say, the latter leaks a lot of information about the model. In AUTOLYCUS, we assume the former. To integrate SHAP into AUTOLYCUS, we either need to transform Shapley values into decision boundaries (like LIME) or only computing the feature importance based on them. Given the undirected and scalar structure of SHAP explanations, decision boundaries can not be derived. Hence, we utilize feature importances of SHAP.

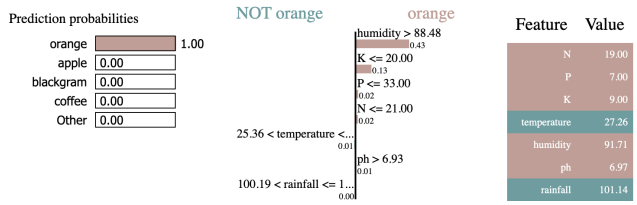

Figure 1: LIME explanation example from the Crop dataset

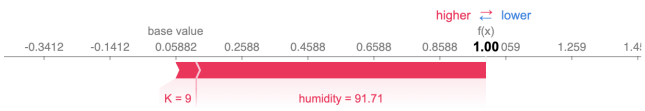

Figure 2: SHAP explanation example from the Crop dataset

We present illustrative examples of LIME and SHAP applied to a decision tree model trained on the Crop dataset [20]. Figure 1 displays the LIME explanation of a membership query which is classified as orange by the target model. On the left-hand side, the prediction probabilities for each class are presented. In the middle section of the figure, the local factors contributing to the model's decision are listed. These factors are ranked in decreasing order of their explanatory strength. Each local factor is composed of contributing features and their corresponding decision boundary intervals. On the right-hand side, the features and their values are listed, color coded to indicate whether they strengthen or weaken the model's prediction. Figure 2 displays the SHAP explanation of the same sample with feature importances. These examples are intended to demonstrate the functionality of LIME and SHAP and the type of information they provide for an interpretable model. Further details about the Crop dataset and the experimental setup are presented in Section 6.1 and subsequent sections.

4 SYSTEM AND THREAT MODELS

Consider a target model M that is trained on a dataset D_M and is deployed for use via an API (e.g., in an ML-as-a-Service (MLaaS) platform). Users can send queries to the target model M and receive as an output, the predicted label, and the corresponding explanation generated by the explainable AI (XAI) tools (as described in Section 3). For instance, a user sends a query for a given sample X_i to the target model M and receives an output including the predicted class y_i, and its corresponding explanations E_i provided by XAI (as shown in Figure 3). A model extraction attack occurs when an adversary attempts to learn a surrogate model S that closely approximates M by exploiting both the output of the predicted labels and the explanations returned by XAI tools. The goal of the attacker is to create a surrogate dataset that inherently reflects the decision boundaries, and hence the functionality of the target model M.

We assume that the attacker possesses black-box access to the target model M, including knowledge of the type -not architecture- of the machine learning model (i.e., decision tree, logistic regression). Even unknown, the model type can be inferred with reasonable accuracy by an individual possessing expertise in machine learning transferability [39] during retraining. Furthermore, the attacker is informed of the possible values that y_i may take, i.e., the class names, and is thus aware of the total number of classes t. The attacker has access to an auxiliary dataset D_A, comprising one or multiple samples to initialize the traversal algorithm. This is to prevent sending blind queries resulting in overhead to the query budget. The composition of D_A may range from a rudimentary collection of samples to comprehensive datasets. Note that in the absence of samples, a random query of default sample values can still be utilized to initiate the process. This can be utilized to simulate a scenario where the attacker lacks auxiliary information. However, the exploration capability with such limited samples can be restrictive, and a higher query budget is essential for an efficient model extraction attack compared to scenarios where a more extensive D_A is available. Lastly, the attacker knows the basic characteristics of the features of the dataset that is used to train model M, such as their type, domain, and lower and upper bounds (i.e., $0 \leq \text{pH} \leq 14$).

We direct the reader's attention to Table 1 for the symbols and notations that appear frequently in this paper.

Table 1: Frequently used symbols and notations.

M	Target Model
S	Surrogate Model
D_M	Training dataset of target model
D_S	Training dataset of surrogate model
D_A	Attacker's auxiliary dataset
D_E	Exploration dataset
Q	Query budget (number of queries)
X_i	Sample i
$\hat{X}_i$	Altered sample i
x_i^j	Sample i's j^{th} feature
$\hat{x}_i^j$	Altered feature j of sample i
y_i	Predicted class for sample i
E_i	Explanation for sample i
Exp_M	Explainer function of model M
k	Number of top features allowed to be explored
m	Number of total features
n	Number of samples per class
t	Number of total classes
l, u	Lower and upper bounds for the number of samples generated per class
db_i^j	Decision boundaries of the sample i's j^{th} feature
δ	Alteration coefficient(s)
TRAV-A	The traversal algorithm

5 PROPOSED WORK

In AUTOLYCUS, the attacker first needs to create a surrogate dataset D_S to learn a surrogate model S that closely approximates the target model M. We assume that the attacker has access to n samples per class in the auxiliary dataset $D_A = X_1, X_2, \ldots, X_{t*n}$ which may or may not have samples from the original dataset D_M. Here, $X_i = \{x_i^1, x_i^2, \ldots, x_i^m\}$, where x_i^j represents the value of feature j in sample X_i and m is the total number of features. We also assume that there exists a query budget Q which restricts the total number of queries that an attacker can send to the target model M. The proposed model extraction attack is depicted in Figure 3. In Step 1, the attacker sends a query using one of its local samples (from the auxiliary dataset) to generate additional samples. In Step 2, the validity of the query is confirmed. In Step 3, the ML model M predicts the class of the queried sample, and LIME or SHAP computes its explanation. In Step 4, the MLaaS platform returns the model's class prediction along with its explanation to the user. In the last step (Step 5), the attacker parses and uses the model explanations to generate altered samples that differ from their predecessors by a single or k features. SHAP returns the Shapley values of each feature for the class predicted by M and they are parsed as feature importances. This process is informed and guided by explanations as opposed to sending random queries. By increasing the number of queries sent, more samples labeled with the target model's predictions are retrieved. In any retraining attack, the samples utilized as the training dataset D_S for surrogate models S are intended to mimic the behavior of the target model. In the following, we describe the details of AUTOLYCUS.

5.1 Generating Candidate Samples

The goal of the attacker is to send informative queries to the target model M instead of blind queries. For that, depending on the XAI tool that is used by the target model, the attacker explores the decision boundaries returned by LIME explanations or feature importances returned by SHAP explanations. Assume the attacker sends a query for a given sample X_i (e.g., one of the samples in D_A) to the target model M. As discussed, the target model M sends the predicted class y_i and the corresponding explanation E_i. As discussed in Section 3, the explanation returned by LIME consists of the prediction probabilities (due to black-box access scenario, we assume only the top class) y_i and the decision boundaries db_i^j ordered by feature importances. Whereas, SHAP only returns the feature importances. To generate new and informative candidate samples, denoted as D_C in Algorithm 1, the attacker considers only the top k features of the sample X_i through its feature importances. There

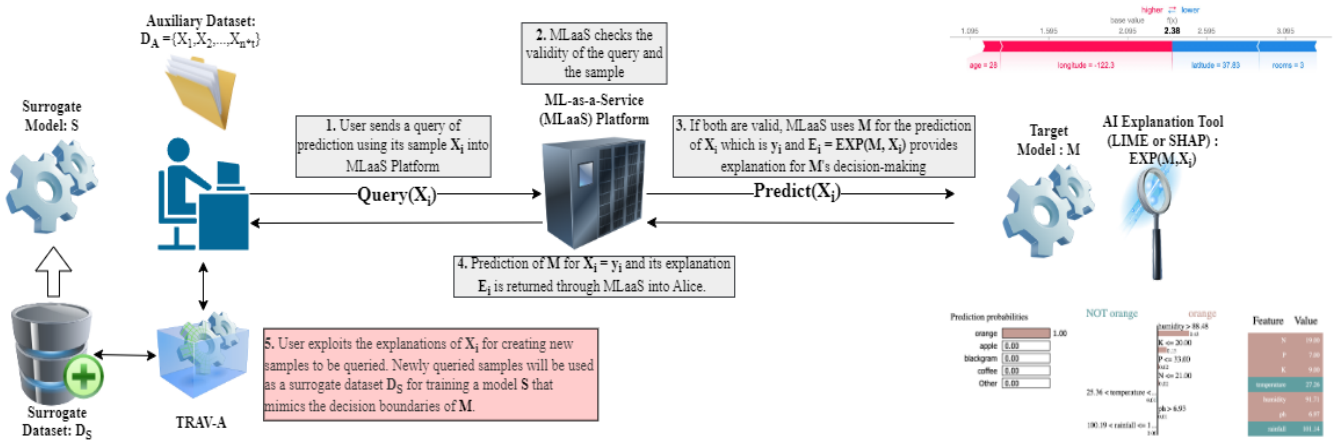

Figure 3: AUTOLYCUS system diagram consisting of the following steps: (1) a user sends a query to the MLaaS platform, (2) the MLaaS platform verifies the validity of the query such that no empty or incomplete queries are sent, (3) the ML model M predicts the class of the queried sample y_i and the explainer computes its explanation E_i, (4) the MLaaS platform returns the results to the user, and (5) in case of an adversarial user, they exploit explanations via TRAV-A algorithm (as described in Section 5.1) to extract the decision boundaries of the target model M.

are two strategies to follow here. In the first exploration procedure, the features are altered one by one resulting in a breadth-first tree search with the width and depth of the tree depending on the chosen value of k. In the second exploration procedure, the top k features are altered all at once. This results in a more diverse set of samples explored. This is an optimal strategy to follow if the size of D_A is small and the distribution of classes is close to uniform distribution. However, if the size of D_A is large and if there are rare classes present, the first procedure may provide better exploration results. In the first exploration procedure, a higher k value will result in a wider tree with low depth, implying that there are more features, and thus more variants of the same sample that need to be explored. This may lead to the exploration of numerous uninformative samples and ultimately exhaust the query budget. Conversely, a lower k value will yield a slender tree with high depth, indicating that more samples are examined with only the top features being prioritized. This results in the inadequate exploration of informative samples, potentially losing crucial information about the structure of the tree. The choice between a high or low value of k and altering them one-by-one or all-at-once as a viable strategy depends on the distribution of feature importances (if they are known or inferred).

For each of the top k features, the attacker generates candidate samples by analyzing the decision boundaries returned by LIME or the feature importances returned by SHAP. The attacker computes new values of feature(s) j, denoted as $\hat{x_i}^j$ by altering it into the decision boundary db_i^j in LIME or to the next available value in SHAP with the coefficient(s) denoted as δ_j. Formally, $\hat{x_i}^j$ is computed as: $\hat{x_i}^j = db_i^j \pm \delta_j$ (in LIME) or $\hat{x_i}^j = x_i{}^j \pm \delta_j$ (in SHAP), where j is the index of the feature that the attacker is aiming to modify and δ_j is the alteration coefficient. For LIME, δ is equal to 1 for categorical features to reflect encoding difference and to 0.01 (or lower) for continuous features. Since the perturbation is decided by the decision boundaries in LIME, δ has lower importance. On the other hand, it is very important in SHAP, since it determines the exploration difference between successive samples. A good rule of thumb is setting it close to the standard deviations if available or to the quarters of solution range. Depending on the intended alteration, δ_j can be manually configured to larger or lower values if necessary. The resulting candidate sample is obtained as $\hat{X_i} = \{x_i^1, x_i^2, \ldots, \hat{x_i}^{j_1}, \ldots, x_i^m\}$ or $\hat{X_i} = \{x_i^1, \hat{x_i}^2, \ldots, x_i{}^{m-2}, \hat{x_i}^{m-1}, x_i^m\}$.

Figure 1 shows $X_i = \{19, 7, 9, 27.26, 91.71, 6.97, 101.14\}$, a toy sample from the Crop dataset [20] and its corresponding explanation E_i. In this example, the top features for $k = 3$ are 'humidity', 'K', and 'P'. If we focus on 'humidity' and select $\delta = 0.01$, the new generated sample will be $\hat{X_i} = \{19, 7, 9, 27.26, \mathbf{88.49}, 6.97, 101.14\}$. Figure 2 shows the same example for SHAP. In this example, the only influential features in the model's decision are 'K' and 'humidity'. When both of these features are altered by 10 ($\delta_2 = 10$ and $\delta_4 = 10$), the new generated sample(s) will be $\hat{X_i} = \{19, 7, \mathbf{19}, 27.26, \mathbf{91.71} \pm \mathbf{10}, 6.97, 101.14\}$ ('K' $= 9 - 10 = -1$ is invalid and discarded)

5.2 Creating the Surrogate Dataset

In order to have a balanced dataset, the attacker aims to create a surrogate dataset D_S that contains similar and proportional number of samples per class to the target model's training dataset D_M. Thus, it needs to generate a minimum number of samples lower-bounded by a value denoted as l per class and $l \times t$ samples in total. This is to ensure that no prediction class is under-represented in D_S. In order to use the query budget Q efficiently, the attacker limits the number of samples generated per class to an upper bound value denoted as u. If the bounding values of l and u are not enforced, the more frequently predicted classes may dominate D_S during traversal, resulting in an imbalanced dataset. This imbalance may hinder the performance of resulting surrogate models S, leading to over-fitting in common classes and inadequate predictions for rarer classes. To address this, l and u values can be represented as scalar

Algorithm 1 Traversal algorithm used for generating samples that will train the surrogate model

```
function TRAV-A(D_A, M, Exp_M, t, n, l, u, Q, δ)
    D_E, D_S, y_S ← D_A, {}, {}
    visits ← {0, 0, ...0}                                  ▷ (length t)
    q ← 0
    while D_E ≠ ∅ and (Q ≥ q or ANY(visits)) ≤ l do
        q++
        X_i ← D_E.POP()
        y_i ← M.PREDICT(X_i)
        if visits[y_i] < u then
            visits[y_i]++
            D_S.ADD(X_i)
            y_S.ADD(y_i)
            E_i ← PARSE_EXP(X_i, M, Exp_M, t, n)
            D_C ← GENERATE_SAMPLES(X_i, E_i, n, δ)
            for i in D_C do
                if i ∉ D_E, D_S then
                    D_E.PUSH(i)
                end if
            end for
        end if
    end while
    return D_S, y_S
end function
```

values for an equal number of samples to be discovered in each class, or as lists allowing for the exploration of different numbers of samples for each class. We recommend setting l based on the frequencies of classes in D_M, if such information is available, and setting u by incrementing l with a desired margin of error. If the frequencies of classes are unknown, a good rule of thumb is setting l and u close to Q/t such that the balance of D_S is maintained for an accurate model training.

The attacker uses the traversal algorithm **TRAV-A** to create the surrogate dataset D_S. This algorithm is built upon running the line-search retraining algorithm proposed by Tramer et al. [55] on the model explanations. The detailed steps are described in Algorithm 1. Recall that the attacker has access to n samples per class from the auxiliary dataset D_A. Thus, initially, D_S is limited to D_A. Let D_E denote the dataset with the samples that need to be explored (initially, $D_E = D_A$). TRAV-A starts by exploring the samples in D_E. It selects the first sample X_i in dataset D_E and sends a query to the target model M. After receiving the predicted class y_i, TRAV-A checks if the maximum number of samples generated for this class has been reached. If that is the case, TRAV-A continues by sending a query for the next sample in D_E and checking if the above condition is met. Otherwise, TRAV-A adds X_i to the surrogate dataset D_S and generates new candidate samples. In Section 5.1, we described how to generate the candidate samples. The generated samples which have not been previously explored (i.e., is not a part of D_E), are added to D_E. In the next iteration, TRAV-A sends a query for the next sample in D_E. TRAV-A employs a breadth-first search strategy to traverse the candidate samples generated during the exploration of a specific sample. We adopt this approach to prevent the deep exploration and propagation of a single sample. This process continues until one of the following conditions is met: (i) there are no unexplored samples in D_E, (ii) the query budget Q has been exhausted, or (iii) the minimum number of samples l that needs to be generated per class has been reached. Note that the lower bound l should be picked close to the approximate query limit per class Q/t such that the budget is utilized as effectively as possible. It is noteworthy that the last condition (iii) is optional, but it complements the second condition (ii). This condition is included in scenarios where the query budget Q is exceeded without having explored a satisfactory number of samples from each class. Consequently, if the query budget is allowed to be exceeded, condition (iii) can substitute for it and ensure that an adequate number of samples from each class have been examined.

Using the generated surrogate dataset D_S, the attacker trains the surrogate model(s) S. Recall that the attacker knows the type of the target model M and can create multiple surrogate models under different hyperparameter settings. Then, the attacker can select the surrogate model with the most similarity without spending additional query budget. The attacker can calculate this similarity by refraining one-k^{th} of the queried samples from the surrogate model training for k-fold cross-validation.

6 EVALUATION

In this section, we describe the datasets, evaluation settings and metrics, and the obtained results.

6.1 Datasets

To evaluate the performance of AUTOLYCUS, we employ six common datasets, most of which are from UCI Machine Learning Repository [25]. For details about the datasets, please refer to Table 2.

Table 2: Dataset properties.

Dataset Name	# Samples	# Features(m)	# Classes(t)	Data Type
Iris [25]	150	4	3	Continuous
Crop [20]	1700	7	17	Continuous
Breast Cancer [25]	569	30	2	Continuous
Adult Income [25]	48,842	14	2	Hybrid
Nursery [25]	12,960	8	3	Categorical
Mushroom [25]	8,124	22	2	Categorical

6.2 Evaluation Settings and Metrics

Evaluation settings. We use scikit-learn [40] Python package for pre-processing, training, and evaluating the models. Prior to training the models, we apply pre-processing techniques to the datasets by eliminating any missing entries and encoding categorical variables using label encoders. Each dataset is split into three subsets: (i) the training dataset (75%), (ii) the test dataset (15%), and (iii) the auxiliary set (10%). The training dataset is used for training the target model M, the test dataset for evaluating the performance of the model extraction attacks, and the auxiliary set for creating the auxiliary dataset D_A. We randomly generate a subset of n samples per class from the auxiliary set, thereby creating a minimal dataset D_A. It is noteworthy that especially for models with a small number of classes, D_A can be manually generated by the attacker with

Table 3: Comparison with SOTA

Dataset	SOTA Attacks				AUTOLYCUS				
	Attack Name	Model	$1 - R_{test}$	Queries	Model	$1 - R_{test}$	Queries	n	XAI Tool
Iris	Equation Solving [55]	Logistic Regression	1	644	Logistic Regression	1	**100**	1	LIME
Iris	Path Finding [8, 55]	Decision Tree	1	246	Decision Tree	1	**10**	1	LIME
Iris	IWAL [8]	Decision Tree	1	361	Decision Tree	1	**10**	1	LIME
Breast Cancer	Equation Solving	Logistic Regression	1	[644,1485]	Logistic Regression	0.992	**100**	5	SHAP
Adult Income	Equation Solving	Logistic Regression	1	1485	Logistic Regression	0.998	**1000**	5	SHAP
Adult Income	Path Finding	Decision Tree	1	18323	Decision Tree	0.937	**1000**	5	SHAP
Adult Income	IWAL	Decision Tree	1	244188	Decision Tree	0.937	**1000**	5	SHAP

random sampling without any prior knowledge. This observation closely aligns with scenarios in which the attacker lacks auxiliary data. Then, the attacker uses D_A to initiate the traversal algorithm (as described in Section 5.2).

We conduct experiments on the following interpretable models; decision trees, logistic regression, naive bayes, k-nearest neighbor classifiers, and random forest. We also test the applicability of AUTOLYCUS against multilayer perceptrons and discuss its limitations in Section 8. To mitigate potential bias during each model training, we train 50 surrogate models on dataset D_S using distinct random states and generating 10 auxiliary datasets D_A to initialize the traversal. We repeat each experiment 10 times for each auxiliary dataset D_A and report the average of the results.

Evaluation metrics. The experiments aim to evaluate the performance of the constructed surrogate models in comparison to the target models using two distinct metrics. In particular, we employ the accuracy and similarity metrics to measure the efficacy of the constructed surrogate models against the test dataset. Accuracy represents the proportion of correct classifications relative to all predictions made by the surrogate model. In contrast, model similarity is the label agreement between the target model and the surrogate models against a neutral dataset, which is referred in the literature also as fidelity or $1 - R_{test}$ [55]. The formal definition of R_{test} over a test dataset D is $R_{test}(f, \hat{f}) = \sum_{(\boldsymbol{x},y)\in D} d(f(x), \hat{f}(x))/|D|$. This enables us to determine the extent to which the generated surrogate models are able to replicate the functionality of the target models.

Comparison with other techniques. We compare the performance of AUTOLYCUS with four other approaches.

Baseline attack. The surrogate model S is trained directly on the auxiliary dataset ($D_A = D_S$). In this scenario, the attacker does not send any queries ($Q = 0$) to the target model M.

Steal-ML attacks. Tramer et al. propose a "path-finding attack" to target decision tree models and an equation-solving attack to target logistic regression models for model extraction [55]. The path-finding attack[1] is a deterministic, top-to-bottom attack that explores all the nodes until an exact reconstruction is achieved. In [55], the splits in the tree path are exploited by extracting node predicates and creating alternate samples. Here, node predicates are conditions a sample must satisfy to end up in the same node ID. The downside of this attack is its strong assumptions like node IDs being readily available and incomplete queries being sent to avail exploration from the top of the tree to the bottom. "The equation-solving attack" is a technique employed against logistic regression models and neural networks. Its query results are converted into linear equations to be solved collectively. Depending on the setting, the extraction may require queries ranging from dozens to hundreds (can be millions for neural networks). This method uses confidence intervals as auxiliary model information, which inadvertently discloses model's internal mechanisms, thereby overlooking the context of black-box access scenarios.

IWAL attack. Chandrasekaran et al. [8] propose an active learning based model extraction attack IWAL to target tree structured models. IWAL is the importance weighted active learning algorithm of Beygelzimer et al. [5], which iteratively refines a tree in each query by minimizing the labeling error.

6.3 Experiments

We assess the impact of attack configuration parameters on the performance of AUTOLYCUS: (i) the query budget (Q) which is commonly referred to as the maximum number of queries that can be sent to the MLaaS platform, (ii) the number of samples per class (n) in the attacker's auxiliary dataset (D_A) and (iii) the method of explanation. Further details are provided in Figure 4 and 5. For each corresponding model type and dataset, we compare the similarity results and the query budget required for AUTLOYCUS to the ones required for the baseline attack, Steal-ML, and IWAL attack. We obtain the results for these attacks from their respective papers. If certain datasets and model types are not explicitly documented or addressed in the papers mentioned above, comparisons are made only when relevant information is available. There are other significant attack types like EAT [8], adaptive retraining [8], Lowd and Meek [31], and Counterfactuals [1, 58] (see Section 8) to consider in the literature. However, these attacks are incompatible with this study due to the model types (e.g., non-interpretable models) they consider. Further details about the comparison of our best results between SOTA attacks are provided in Table 3. It is crucial to highlight that, within our specific attack, LIME explanations offer significantly more information under the current assumptions. This disparity arises from SHAP's provision of feature importances without directional guidance for perturbations. It acts as a semi-blind querying method guided by the order of top features and careful selection of δ. The sole scenario in which attacks using SHAP surpasses attacks using LIME is when local decision boundaries inadequately convey global model behavior.

[1]Path-finding results are obtained from the reproduced results of Chandrasekaran et al. [8] and not from the original paper [55]

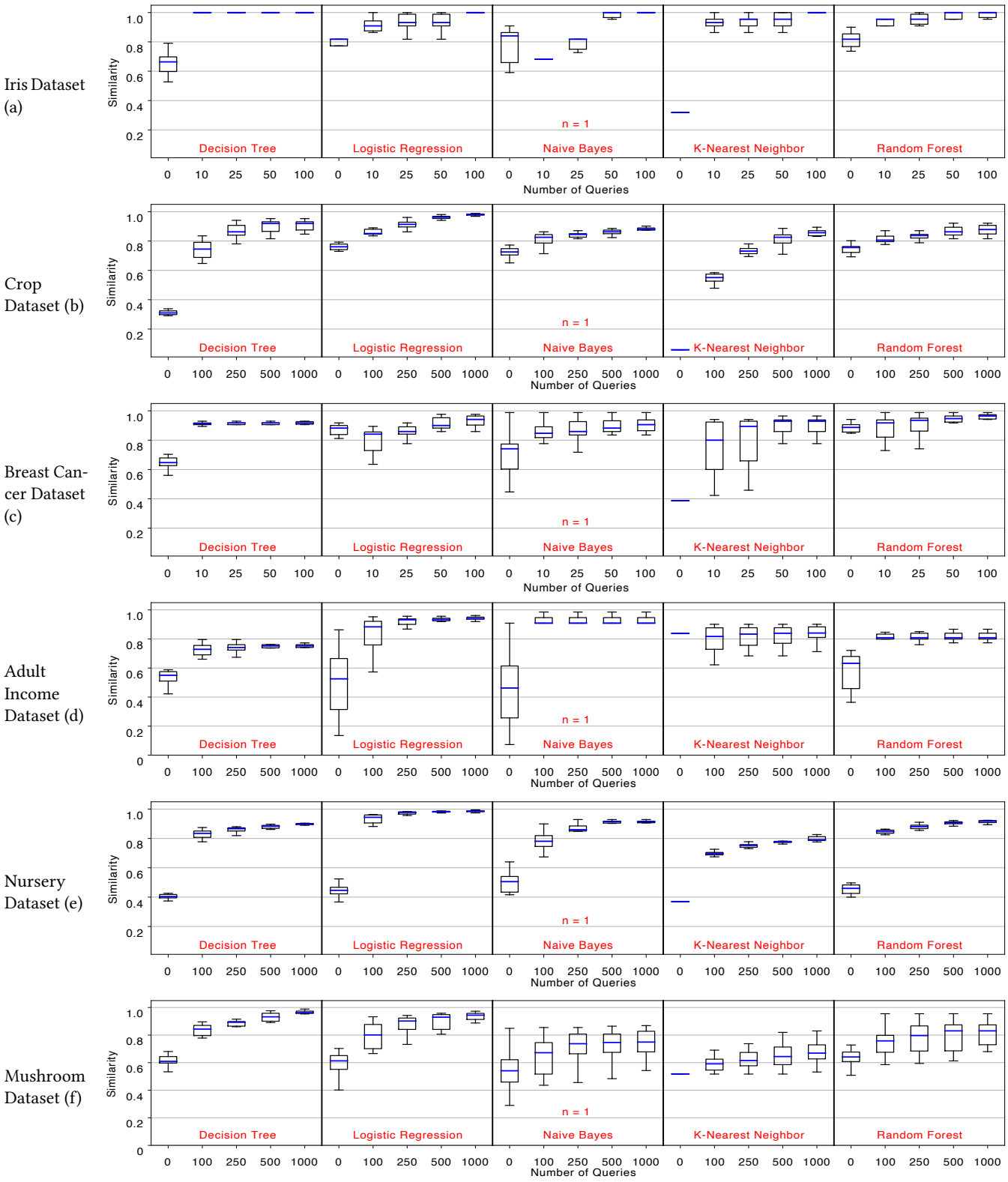

Figure 4: Impact of the number of queries (Q) on surrogate model similarity when LIME is used. ($k = 3, n = 1$)

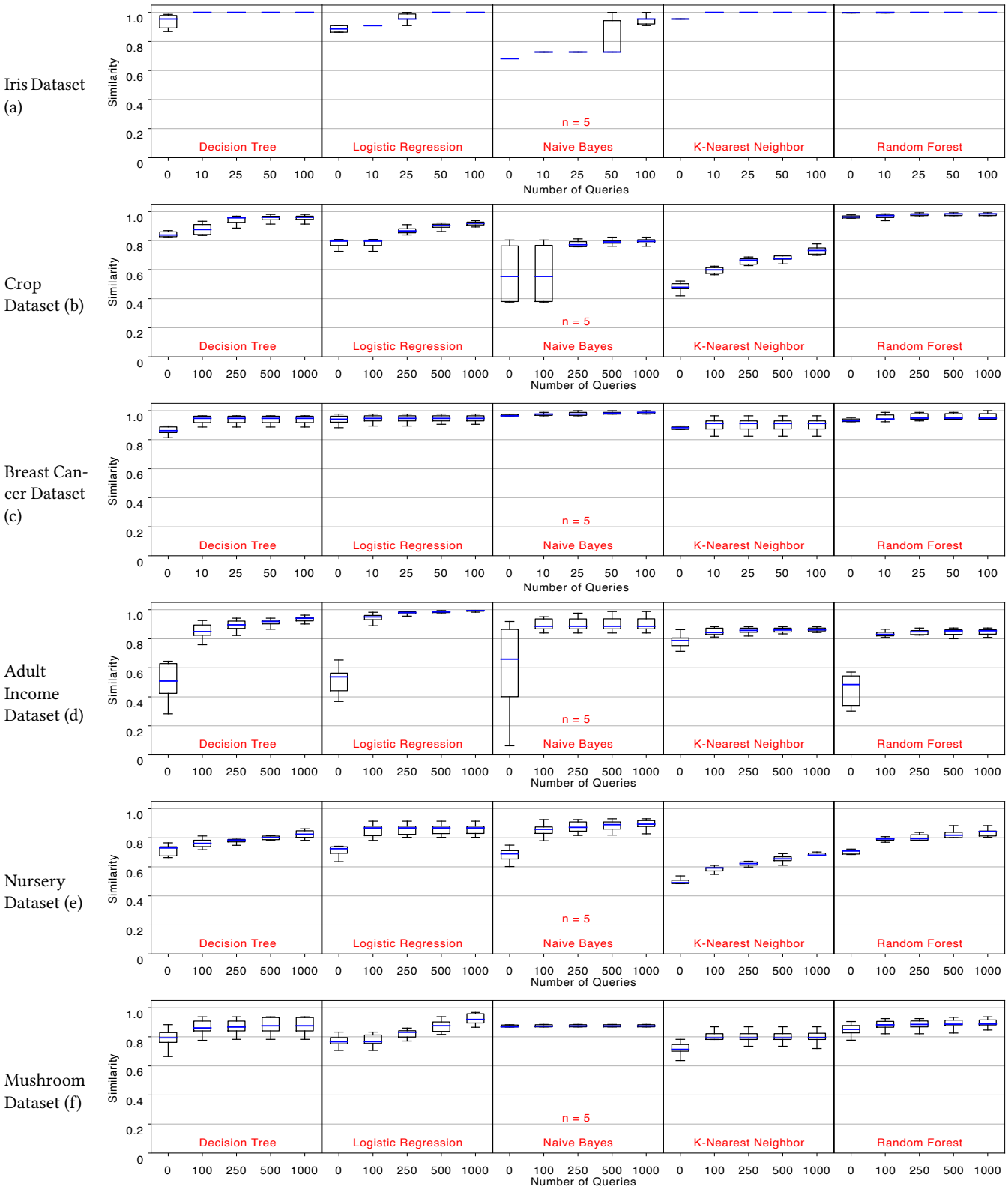

Figure 5: Impact of the number of queries (Q) on surrogate model similarity when SHAP is used. ($k = 3, n = 5$)

For Figures 4, 5, and Table 3, the number of top features allowed to be explored (k) is set to 3. The size of the auxiliary dataset per class (n) is set to 1 (for LIME) and 5 (for SHAP). This is a design choice to demonstrate the impact of the size of the auxiliary dataset while providing a slight leverage to SHAP considering the aforementioned shortcoming. In our additional experiments under similar n settings ($n = 1$ for SHAP or $n = 5$ for LIME), SHAP performed worse compared to LIME due to having less informativeness. Finally, δ of SHAP attacks are drawn from the standard deviations and solution space divisions observed in the auxiliary dataset.

Iris dataset. For all considered machine learning models, we observe that the target models reach an accuracy equal to 1. The attacker obtains a surrogate model S that has a similarity close to 1 to the target model M by sending 10 queries to most target models (as shown in Figures 4 and 5(a)). This is attributed to the simplicity of models trained on the Iris dataset. SHAP outperforms LIME due to its higher value of n. The proximity of query similarities to the baseline attack when $Q = 10$ in SHAP further confirms this. Except for naive bayes, when $Q < 50$, AUTOLYCUS significantly outperforms the baseline approach in both LIME and SHAP attacks. Compared to the number of queries reported in Table 3 for Steal-ML and IWAL attacks against decision tree and logistic regression models, AUTOLYCUS requires fewer queries, especially as the size of the auxiliary dataset increases.

Crop dataset. Crop dataset contains the highest number of classes (t) among the datasets we consider. The target models reach accuracies above 0.98 except for the naive bayes model which reaches an accuracy of 0.84. Due to having high t in the Crop dataset, across all models, the gradual increase of model similarity is more apparent as the number of queries increases. When $Q = 1000$, in Figure 4(b) the logistic regression model has a similarity close to 1. For Figure 5(b), decision tree and random forest models have such similarity. We make two observations by comparing these results with those obtained for the Iris dataset. Firstly, the performance of the model extraction attack is enhanced as the model complexity (architecture, number of features, and classes) decreases or model accuracy increases in accordance with the statement of Rigaki et al. [43]. Secondly, even with the extra dataset, LIME sometimes has an edge over SHAP due to the high informativeness of its explanations.

Breast Cancer dataset. For the Breast Cancer dataset, target model accuracies are between 0.86 to 1. In Figure 5(c), we observe that even a baseline attack can generate models with sufficient accuracy if enough real-life samples are known. In Figure 4(c), we observe that the overall similarity between target models and surrogate models consistently increases across different model types as the number of queries increases from 0 to 100. Except for decision tree models, with only 100 queries, there is always at least one surrogate model that achieves close to perfect similarity with target models. When both LIME and SHAP attacks are observed, the performance of the decision tree model does not increase as n increases from 1 to 5. Because the decision boundaries provided by explanations do not necessarily cover the entire solution space of target models and some classes or branches may remain unexplored. Our additional experiments on this dataset with higher k provided perfect extractions which can be explained with a more uniform distribution of top important features. Even though the equation solving attack was evaluated in this dataset, it was not provided by Tramer et al. [55] the number of queries needed to extract a logistic regression model trained on this dataset. Based on the performances against the Iris dataset, which results in a simpler model extracted with 644 queries, and the Adult Income dataset, which yields a more complex model extracted with 1485 queries, we can hypothesize that a logistic regression model trained on the Breast Cancer dataset might be extracted with a number of queries somewhere in between. Notably, AUTOLYCUS outperforms the Equation Solving attack by achieving extraction with 100 (6.4 times less) queries.

Adult Income dataset. For the Adult Income dataset in Figures 4(d) and 5(d), the target models have accuracies around 0.8. The Adult Income dataset consists of more than 100 features -when encoded- and generates complex tree models with a tree depth of more than 20. This is a critical dataset that highlights the importance of optimizing all attack configuration parameters for best performance. It requires a higher query limit, a higher number of features considered, a higher and diverse number of auxiliary samples, and an informed set of δ values provided for perturbation. These dependencies differ between model types. For logistic regression, surrogate models with 0.998 similarity to the target model can be obtained with 1000 queries by SHAP attacks compared to the 1485 queries of Equation Solving attack. For decision tree models, 0.937 similarity can be obtained in 1000 queries when $n = 5$ and δ set optimized in SHAP attack compared to the 18323 and 244188 queries of Path Finding and IWAL attacks. Given better configuration parameters, the performance can be further improved.

Nursery dataset. For the Nursery dataset, the target models have accuracies between 0.95 to 1 except for naive bayes. This dataset comprises eight non-binary (when encoded more than 20) categorical features. Overall similarity across different models is around 0.91 for LIME attacks (Figure 4(e)) and 0.81 for SHAP attacks (Figure 5(e)). In this dataset, LIME extractions provide consistently better results compared to SHAP. This is due to the uniform distribution of feature importances and LIME overcomes it with the more informed decision boundaries provided. When k is increased to values closer to m (number of total features), SHAP performs similarly to LIME even when n is as low.

Mushroom dataset. Much like the Nursery dataset, the target models of the Mushroom dataset demonstrate accuracies ranging from 0.95 to 1, except for naive bayes with accuracy of 0.8. The Mushroom dataset, along with the Adult Income dataset, boasts the highest number of features (when encoded), resulting in a substantial solution space. In the cases of decision trees and logistic regression, surrogate models with similarities close to or equal to 1 are achieved with 1000 queries in LIME attacks. However, similarities stagnate in SHAP attacks compared to the baseline attack across model types except for logistic regression. Naive Bayes and K-Nearest Neighbor model extractions improve slowly or stagnate compared to other models in both attacks. This indicates that similar to the Nursery dataset, when $k = 3$, it does not encompass all the significant features. thus necessitating an increase in k. Also in LIME attacks, there is a large variance in the attack's performance for different auxiliary datasets for a fixed n value. This is an indicator that the informativeness of auxiliary data can drastically reduce or improve the performance of extraction on these models.

Number of features explored. The number of features allowed for traversal (k), plays a pivotal role in influencing the performance

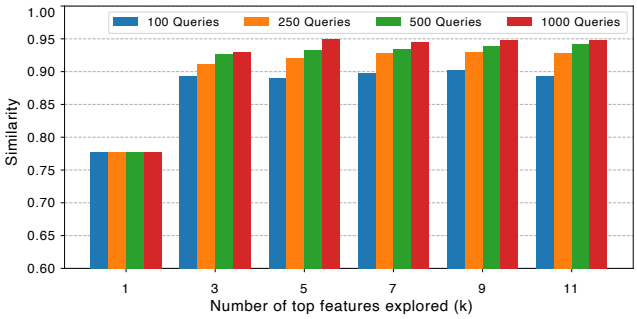

Figure 6: Impact of k on surrogate model similarity. $(n = 3, LIME)$

of AUTOLYCUS. In the experiments, we set $k = 3$ (as indicated in Figure 4 and 5). While this value may be adequate for models trained on datasets with a limited number of features, it might be insufficient for larger models, especially when a higher number of features are influential in model decision making.

We demonstrate the impact of k on a logistic regression model trained on the Adult Income dataset for a LIME attack. Note that the impact of k on SHAP attacks is similar to the LIME attacks since both attacks utilize feature importances. Figure 6 shows the similarity of the surrogate model to the target model for different numbers of queries when varying k. For $k = 1$, the exploration predominantly revolves around a single feature, thereby restricting the surrogate model S to a maximum similarity of 0.77. With an increase in k up to 5, surrogate models exhibit a top similarity of 0.95 when 1000 queries are generated. Beyond $k > 4$, the improvement in similarities becomes marginal, suggesting that the target model relies on approximately 5 to 6 crucial features.

7 COUNTERMEASURES

In this section, we explore five countermeasures that can be employed against AUTOLYCUS. Two of these are based on differential privacy [14]. These methods are influenced by Tramer et al.'s suggestions for prediction minimization [55].

One of the countermeasures explored in the experiments is differential privacy, which serves as a defense mechanism against the model extraction attack. We examine three types of differentially private methods. The first method involves adding noise to the training dataset of the target model M. As noted by Tramer et al., this method does not mitigate the model extraction attacks since the objectives do not align, and applying noise may only impede the learning of sensitive information about the training dataset without targeting the boundaries or parameters of models. We verify this by applying naive differential privacy with calibrated Laplacian noise using ϵ values between $[1, \ln(3)]$ to the training data. We observe that such application does not mitigate the model extraction attacks and has no significant impact on the decision boundaries. Therefore, as suggested by Tramer et al., we direct our efforts towards targeting the model boundaries.

The second differentially private method we explore is dynamic distortion. It involves applying noise directly to the decision boundaries of LIME explanations and changing these boundaries as new queries are sent to the model. In LIME, the decision boundaries of each feature are static sorted lists, meaning $DB^j = \{db^j 1, db^j 2, ...\}$ per model, where db_i^j is equal to the neighbors of x_i^j.[2] To implement this countermeasure, we first determine Q queries as the privacy budgets and removed Q samples from the training dataset of the target model. This way for each sample's decision region, we explore how much the decision boundaries of explanations deviate between any neighboring models M and M'. Note that these models are trained with $n * t$ and $n * t - Q$ samples, respectively. Considering a sample X_i and for all neighboring models M and M', the maximum difference any decision boundary d_i^j can have can be formalized as follows: $\sigma_i^j = max||M(d_i^j) - M'(d_i^j)||$. Then, we calculate a Laplacian distribution with deviation corresponding to σ as the sensitivity, and for varying ϵ values, we apply Laplacian noise directly to explanation decision boundaries such that they are changed in every query. The results for dynamic distortion when $\epsilon = 1$ are shown in Figures 7 (a). This method not only fails to mitigate the model extraction attack, but it also enhances the attack strength by allowing a more diverse and informative surrogate dataset to be explored.

The third method is static distortion. It's a generic random noise application, using the dynamic distortion method with $\epsilon = 1$. In this method though, the noise drawn from the Laplace distribution is unchanged across queries such that the model explanations seem like they are drawn from another model. The results for static distortion are shown in Figure 7 (b). We observe that this method also fails to mitigate the model extraction attack and, like the previous method, makes the attack even stronger by improving the accuracy and similarity of resulting surrogate models. However, the improvement is negligible compared to the second method.

Applying noise with sufficiently low ϵ values can distort (dynamic and static) the explanation boundaries to prevent model extraction attacks. However, the utility of model explanations is hindered more than the model extraction success, in such instances. Experiments with lower ϵ values, produced mathematically impossible decision boundaries due to excessive perturbation.

The fourth method involves adding noise to the feature importances from SHAP. These importances dictate the sequence of feature perturbations based on an attacker-defined δ_j value. While significant noise could disrupt this sequence and cause deviations in near to k^{th} feature perturbations, unless the value of k is sufficiently low, these deviations would normalize with increasing queries. However, this normalization would hinder the utility of explanations, potentially misguiding users about their predictions. Thus, using noise on SHAP explanations is unlikely to effectively counter model extraction attacks.

Another countermeasure suggested by Tramer et al. is rounding the confidence values obtained during predictions. This method is intuitively effective against models with confidence values and extraction attacks that deterministically solve boundaries (i.e., equation-solving attacks, path-finding attacks). Since AUTOLYCUS reflects the decision boundaries in the re-training process stochastically and only employs labels and the decision boundaries of explanations,

[2]Neighboring relationship between the states of features (x_i^j values) is based on the corresponding decision boundaries.

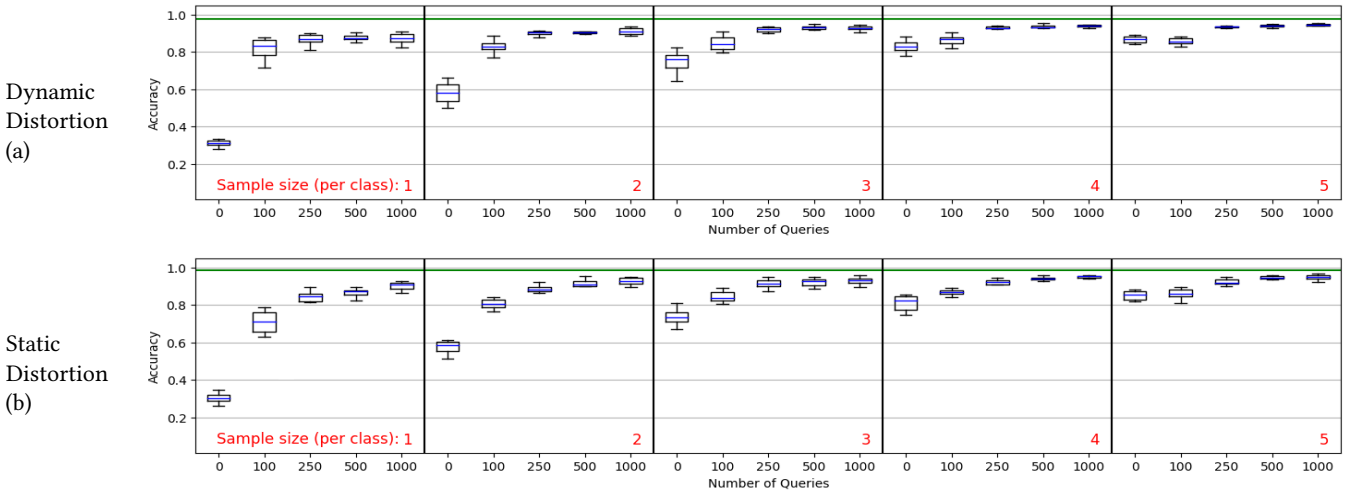

Figure 7: Impact of countermeasures on the Crop dataset.

methods such as rounding confidences or explanation strength do not have any impact on its performance.

Prediction minimization methods are prone to failure in AUTOLYCUS, but countermeasures like setting query budgets can slow down the extraction process. While not explored in this study, fingerprinting against malicious surrogate model sharing [32] is a potential future work.

8 DISCUSSION

Limitations. The primary limitation of AUTOLYCUS is the probabilistic reconstruction of models in comparison to the exact reconstruction by the white-box attacks (Equation Solving, Path-finding) of Tramer et al. [55]. AUTOLYCUS involves retraining models in a black-box setting, and therefore attack configuration parameters such as explanations, traversal, or auxiliary datasets are subject to limitations in some models.

For instance, outlier classifications that are not adequately represented in explanations or auxiliary datasets (e.g., edge cases in decision tree models) may remain unexplored during the traversal procedure. This limitation is particularly evident in complex models with a high number of top features, where some branches may not be properly represented in the surrogate model, reducing their similarity to the target model. An example limitation is observed in the experiments where LIME is used for targeting a decision tree model trained on the Adult Income dataset. Despite producing surrogate models with similar accuracy, the low accuracy and multiple influential features of this model made it difficult to adequately discover and functionally represent certain aspects of the model with the assumptions and small auxiliary datasets available.

Furthermore, AUTOLYCUS's effectiveness is limited to the type of data that target models are trained on. AUTOLYCUS performs well on tabular and mixed data suitable for interpretable models and decision boundaries. However, in image classification tasks, tabular LIME and SHAP explanations of single pixels have equivalent strengths, making these individually considered features and their boundaries uninformative, effectively rendering the attack useless. Since image classifications consider a collection of pixels or regions in figures, explanations, and traversal suited for such a task must be employed. Therefore, AUTOLYCUS should be employed against models with informative explanations.

In contrast with evaluations in Section 6.3, the performance of AUTOLYCUS is influenced by careful adjustment of the following attack configuration and model parameters: (i) ensuring sufficiently high number of queries are sent; (ii) selecting adequate number of features (k) for perturbation, guided by the distribution of feature importance; (iii) incorporating sufficient number of (iv) informative auxiliary samples; (v) considering complexity of the models -lower complexity models are extracted easier-, including any unknowns or parameters in neural network terminology; (vi) ensuring the accuracy of the model, (vii) optimizing selection of δ values especially when SHAP is used as the XAI tool. These factors collectively contribute to the efficacy of AUTOLYCUS in adversarial settings.

Other model-agnostic explainability tools. In the realm of alternative explanation techniques, the integration of counterfactuals offers an additional source of valuable information, with various documented attacks in the literature [1, 58]. Our goal with LIME and SHAP explanations is similar to replicating the utility of counterfactuals with additional steps. While providing counterfactuals directly would streamline our framework, it would also provide limited novelty. Moreover, due to documented privacy risks, MLaaS providers may refrain from using these explanations in commercial settings. Due to these concerns, we have opted not to explicitly include them in our framework.

Application to other machine learning models. After obtaining promising results against interpretable models, we investigate the effectiveness and transferability of AUTOLYCUS against black-box models. Our initial focus lies on neural networks. We conduct experiments on four datasets, with a configuration similar to the Steal-ML attack, where target models consist of multilayer perceptrons with a single hidden layer comprising 20 nodes. Only for the Breast Cancer dataset, hidden layers are increased to 100 nodes due to the model's low accuracy of 0.22 on 20 nodes. We set $k = 3$

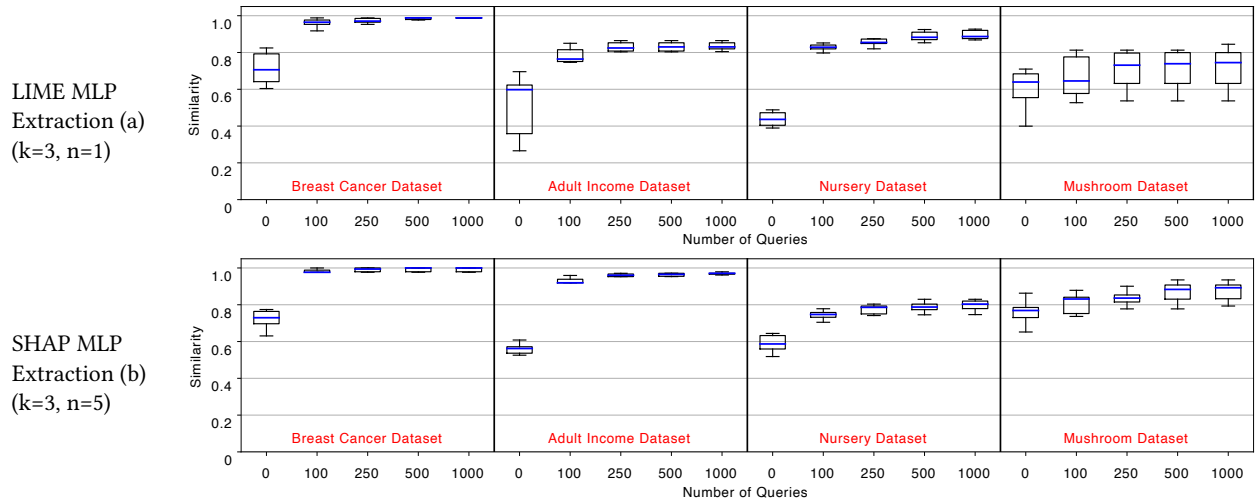

Figure 8: Multilayer perceptron surrogate model similarities across different datasets.

and $n = 1$ (for LIME) and $n = 5$ (for SHAP) following the setting in Section 6.3. Figure 8 shows the similarities for each dataset.

For the Breast Cancer dataset, the attacker obtains surrogate models exhibiting a similarity of up to 1 with 100 queries in both LIME and SHAP attacks. This outcome aligns with our expectations given the dataset's relatively low complexity, as demonstrated by the results shown in Figures 4 and 5(c). Furthermore, for both Adult Income and Nursery datasets, surrogate models achieve comparable performances when the target models are random forests and LIME is used. On the other hand, while the SHAP attack performs better than LIME attacks due to increased auxiliary dataset size, it is less efficient in extracting the neural network trained on the Nursery dataset compared to LIME attacks. This discrepancy is attributed to differences in explanation informativeness. LIME excels in explaining categorical solution spaces with fewer unknowns when k is low. As the solution space of the Nursery dataset is relatively small compared to Adult Income and Mushroom datasets, LIME explanations retain their efficacy against SHAP explanations even with a low value of n. However, with an increase in k, the performance of the SHAP attack begins to improve, similar to the behavior observed with LIME explanations. In the Mushroom dataset, an increased number of queries does not yield a desired improvement in the similarity of surrogate models for LIME attack. Experiments involving higher values of k also do not yield significant performance enhancement observed in the case of interpretable models. This limitation may be attributed to the scarcity of queries required to accurately represent the complex boundaries of neural networks and the number of features with high solution space further complicating the extraction on relatively low query budgets. In the SHAP attack, extraction is more successful owing to the larger number of available auxiliary samples (n), and this success tends to increase with higher numbers of queries. This suggests that a greater abundance of samples facilitates the exploration of rarer cases (which seems to be more abundant in the Mushroom dataset) more effectively. Additionally, in the process of extracting the complex boundaries of neural networks, LIME and SHAP explanations may exhibit occasional inconsistencies, which could constitute an additional factor, hindering performance. In conclusion, under comparable conditions with sufficiently large values of n and k, LIME mostly demonstrates superior performance in capturing similarities within neural networks compared to SHAP.

9 CONCLUSION

In this paper, we have proposed AUTOLYCUS, a novel model-agnostic extraction attack that exploits model-agnostic AI explanations to extract the decision boundaries of interpretable models. The empirical results on six different datasets have shown the efficacy of the attack. In particular, we have demonstrated that an attacker can exploit AI explanations to create fairly accurate surrogate models that have high similarity to the target models even under low query budgets. We have also demonstrated that AUTOLYCUS requires fewer queries for partial reconstructions with comparable accuracy and similarity than the state-of-the-art attacks that rely on exact reconstruction. Furthermore, we have explored potential countermeasures to mitigate this attack such as adding noise to the decision boundaries of explanations or adding noise to the training data. The empirical results have shown that these countermeasures are ineffective against AUTOLYCUS. As a future work, we plan to study the efficacy of multiple model explanations, potential increased risks due to explanations in black-box machine learning models such as convolutional neural networks, and work on new and effective countermeasures.

ACKNOWLEDGMENTS

For the research reported in this publication, Abdullah Caglar Oksuz and Erman Ayday was supported in part by the National Science Foundation (NSF) under grant number OAC-2112606. Anisa Halimi was partly supported by the European Union's Horizon 2020 research and innovation programme under grant number 951911 – AI4Media. The content is solely the responsibility of the authors and does not necessarily represent the official views of the agencies funding the research.

REFERENCES

[1] Ulrich Aïvodji, Alexandre Bolot, and Sébastien Gambs. 2020. Model extraction from counterfactual explanations. arXiv:2009.01884 https://arxiv.org/abs/2009.01884

[2] Microsoft Azure. 2024. Model interpretability - Azure Machine Learning. https://learn.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability?view=azureml-api-2 [Accessed 01-Mar-2024].

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015), e0130140. https://doi.org/10.1371/journal.pone.0130140

[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

[5] Alina Beygelzimer, Daniel Hsu, John Langford, and Tong Zhang. 2010. Agnostic active learning without constraints. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1* (Vancouver, British Columbia, Canada) *(NIPS'10)*. Curran Associates Inc., Red Hook, NY, USA, 199–207. https://proceedings.neurips.cc/paper/2010/hash/00411460f7c92d2124a67ea0f4cb5f85-Abstract.html

[6] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, Nadia Heninger and Patrick Traynor (Eds.). USENIX Association, Santa Clara, CA, USA, 267–284. https://www.usenix.org/conference/usenixsecurity19/presentation/carlini

[7] IBM Thomas J Watson Research Center. 2019. Explainable Artificial Intelligence (XAI). https://www.ibm.com/watson/explainable-ai

[8] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. 2020. Exploring Connections Between Active Learning and Model Extraction. In *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, Srdjan Capkun and Franziska Roesner (Eds.). USENIX Association, USA, 1309–1326. https://www.usenix.org/conference/usenixsecurity20/presentation/chandrasekaran

[9] Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. 2021. Label-Only Membership Inference Attacks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, USA, 1964–1974. http://proceedings.mlr.press/v139/choquette-choo21a.html

[10] Google Cloud. 2021. Vertex Explainable AI. https://cloud.google.com/vertex-ai/docs/explainable-ai/overview [Accessed 24-Feb-2024].

[11] IBM Cloud. 2023. Watson Openscale. https://www.ibm.com/docs/en/cloud-paks/cp-data/4.8.x?topic=models-configuring-model-evaluations [Accessed 24-Feb-2024].

[12] Filip Karlo Dosilovic, Mario Brcic, and Nikica Hlupic. 2018. Explainable artificial intelligence: A survey. In *41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018, Opatija, Croatia, May 21-25, 2018*, Karolj Skala, Marko Koricic, Tihana Galinac Grbac, Marina Cicin-Sain, Vlado Sruk, Slobodan Ribaric, Stjepan Gros, Boris Vrdoljak, Mladen Mauher, Edvard Tijan, Predrag Pale, and Matej Janjic (Eds.). IEEE, USA, 210–215. https://doi.org/10.23919/MIPRO.2018.8400040

[13] Vasisht Duddu, Debasis Samanta, D. Vijay Rao, and Valentina Emilia Balas. 2018. Stealing Neural Networks via Timing Side Channels. arXiv:1812.11720 http://arxiv.org/abs/1812.11720

[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. 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 (Lecture Notes in Computer Science, Vol. 3876)*, Shai Halevi and Tal Rabin (Eds.). Springer, USA, 265–284. https://doi.org/10.1007/11681878_14

[15] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. 2019. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *J. Mach. Learn. Res.* 20 (2019), 177:1–177:81. http://jmlr.org/papers/v20/18-760.html

[16] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. Association for Computing Machinery, New York, NY, USA, 1322–1333. https://doi.org/10.1145/2810103.2813677

[17] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 5 (2001), 1189 – 1232. https://doi.org/10.1214/aos/1013203451

[18] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 619–633. https://doi.org/10.1145/3243734.3243834

[19] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek. 2020. Explainable AI Methods - A Brief Overview. In *xxAI - Beyond Explainable AI - International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers (Lecture Notes in Computer Science, Vol. 13200)*, Andreas Holzinger, Randy Goebel, Ruth Fong, Taesup Moon, Klaus-Robert Müller, and Wojciech Samek (Eds.). Springer, USA, 13–38. https://doi.org/10.1007/978-3-031-04083-2_2

[20] Atharva Ingle, N Nithin Srivatsav, and Swathi N Shayana. 2020. Crop Recommendation Dataset. https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

[21] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High Accuracy and High Fidelity Extraction of Neural Networks. In *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, Srdjan Capkun and Franziska Roesner (Eds.). USENIX Association, USA, 1345–1362. https://www.usenix.org/conference/usenixsecurity20/presentation/jagielski

[22] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. 2019. PRADA: Protecting Against DNN Model Stealing Attacks. In *IEEE European Symposium on Security and Privacy, EuroS&P 2019, Stockholm, Sweden, June 17-19, 2019*. IEEE, USA, 512–527. https://doi.org/10.1109/EUROSP.2019.00044

[23] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda B. Viégas, and Michael Terry. 2019. XRAI: Better Attributions Through Regions. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, USA, 4947–4956. https://doi.org/10.1109/ICCV.2019.00505

[24] Sanjay Kariyappa and Moinuddin K. Qureshi. 2020. Defending Against Model Stealing Attacks With Adaptive Misinformation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, USA, 767–775. https://doi.org/10.1109/CVPR42600.2020.00085

[25] Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. 2023. UCI Machine Learning Repository. https://archive.ics.uci.edu

[26] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, USA, 2668–2677. https://proceedings.mlr.press/v80/kim18d.html

[27] Pieter-Jan Kindermans, Kristof T. Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. 2018. Learning how to explain neural networks: PatternNet and PatternAttribution. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, USA. https://openreview.net/forum?id=Hkn7CBaTW

[28] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2010. Cifar-10 (canadian institute for advanced research). http://www.cs.toronto.edu/~kriz/cifar.html

[29] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. http://yann.lecun.com/exdb/mnist/

[30] Zheng Li and Yang Zhang. 2021. Membership Leakage in Label-Only Exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, Republic of Korea) *(CCS '21)*. Association for Computing Machinery, New York, NY, USA, 880–895. https://doi.org/10.1145/3460120.3484575

[31] Daniel Lowd and Christopher Meek. 2005. Adversarial learning. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett (Eds.). ACM, USA, 641–647. https://doi.org/10.1145/1081870.1081950

[32] Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. 2021. Deep Neural Network Fingerprinting by Conferrable Adversarial Examples. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, USA. https://openreview.net/forum?id=VqzVhqxkjH1

[33] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4768–4777. https://dl.acm.org/doi/pdf/10.5555/3295222.3295230

[34] Smitha Milli, Ludwig Schmidt, Anca D. Dragan, and Moritz Hardt. 2019. Model Reconstruction from Model Explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (Atlanta, GA, USA) *(FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 1–9. https://doi.org/10.1145/3287560.3287562

[35] Takayuki Miura, Satoshi Hasegawa, and Toshiki Shibahara. 2021. MEGEX: Data-Free Model Extraction Attack against Gradient-Based Explainable AI. arXiv:2107.08909 https://arxiv.org/abs/2107.08909

[36] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff Nets: Stealing Functionality of Black-Box Models. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, USA, 4954–4963. https://doi.org/10.1109/CVPR.2019.00509

[37] Guillermo Owen. 2013. Applications of Game Theory to Economics. *IGTR* 15, 3 (2013), 1340011. https://doi.org/10.1142/S0219198913400112

[38] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17)*. Association for Computing Machinery, New York, NY, USA, 506–519. https://doi.org/10.1145/3052973.3053009

[39] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. 2016. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. arXiv:1605.07277 http://arxiv.org/abs/1605.07277

[40] 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, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. https://www.jmlr.org/papers/v12/pedregosa11a.html

[41] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[42] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, USA, 1527–1535. https://doi.org/10.1609/AAAI.V32I1.11491

[43] Maria Rigaki and Sebastian García. 2024. A Survey of Privacy Attacks in Machine Learning. *ACM Comput. Surv.* 56, 4 (2024), 101:1–101:34. https://doi.org/10.1145/3624010

[44] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215. https://doi.org/10.1038/s42256-019-0048-x

[45] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, USA. http://dx.doi.org/10.14722/NDSS.2019.23119

[46] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, USA, 618–626. https://doi.org/10.1109/ICCV.2017.74

[47] Amazon Web Services. 2023. Amazon SageMaker Clarify. https://aws.amazon.com/sagemaker/clarify/ [Accessed 24-Feb-2024].

[48] Lloyd S. Shapley. 1951. *Notes on the N-Person Game - I: Characteristic-Point Solutions of the Four-Person Game.* RAND Corporation, Santa Monica, CA. https://doi.org/10.7249/RM0656

[49] Yi Shi, Yalin Sagduyu, and Alexander Grushin. 2017. How to steal a machine learning classifier with deep learning. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, USA, 1–5. https://doi.org/10.1109/THS.2017.7943475

[50] Reza Shokri, Martin Strobel, and Yair Zick. 2021. On the Privacy Risks of Model Explanations. In *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, Marion Fourcade, Benjamin Kuipers, Seth Lazar, and Deirdre K. Mulligan (Eds.). ACM, USA, 231–241. https://doi.org/10.1145/3461702.3462533

[51] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, USA, 3–18. https://doi.org/10.1109/SP.2017.41

[52] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034 [cs.CV]

[53] Congzheng Song and Vitaly Shmatikov. 2020. Overlearning Reveals Sensitive Attributes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, USA. https://openreview.net/forum?id=SJeNz04tDS

[54] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, USA, 3319–3328. http://proceedings.mlr.press/v70/sundararajan17a.html

[55] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *Proceedings of the 25th USENIX Conference on Security Symposium* (Austin, TX, USA) *(SEC'16)*. USENIX Association, USA, 601–618. https://dl.acm.org/doi/10.5555/3241094.3241142

[56] Jean-Baptiste Truong, Pratyush Maini, Robert J. Walls, and Nicolas Papernot. 2021. Data-Free Model Extraction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, USA, 4771–4780. https://doi.org/10.1109/CVPR46437.2021.00474

[57] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing Hyperparameters in Machine Learning. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, USA, 36–52. https://doi.org/10.1109/SP.2018.00038

[58] Yongjie Wang, Hangwei Qian, and Chunyan Miao. 2022. DualCF: Efficient Model Extraction Attack from Counterfactual Explanations. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*. ACM, USA, 1318–1329. https://doi.org/10.1145/3531146.3533188

[59] Anli Yan, Teng Huang, Lishan Ke, Xiaozhang Liu, Qi Chen, and Changyu Dong. 2023. Explanation leaks: Explanation-guided model extraction attacks. *Information Sciences* 632 (2023), 269–284. https://doi.org/10.1016/j.ins.2023.03.020

[60] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. 2020. XGNN: Towards Model-Level Explanations of Graph Neural Networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, USA, 430–438. https://doi.org/10.1145/3394486.3403085

[61] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, USA, 250–258. https://doi.org/10.1109/CVPR42600.2020.00033

[62] Xuejun Zhao, Wencan Zhang, Xiaokui Xiao, and Brian Y. Lim. 2021. Exploiting Explanations for Model Inversion Attacks. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, USA, 662–672. https://doi.org/10.1109/ICCV48922.2021.00072

[63] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning Deep Features for Discriminative Localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, USA, 2921–2929. https://doi.org/10.1109/CVPR.2016.319

[64] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. 2017. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, USA. https://openreview.net/forum?id=BJ5UeU9xx

A XAI IN MLAAS PLATFORMS

Major MLaaS platforms such as Azure Machine Learning (Responsible AI) [2], Amazon SageMaker (Clarify) [47], Watson Machine Learning (OpenScale) [11] and Google Cloud (Vertex Explainable AI) [10] offer various explanation techniques. Some of the provided techniques in these platforms are included but not limited to SHAP [33], LIME [41], Permutation Feature Importance (PFI) [15], Partial Dependence Plots (PDP) [17], Integrated Gradients [54] and XRAI [23]. However most explainers offered by MLaaS providers [2, 10, 47] are predominantly white-box, intended primarily for model developers. They provide detailed insights into the inner workings of the models, contrasting with the simpler, black-box interfaces often provided to the querying users. These use cases include, but are not limited to, detecting bias, assessing fairness, detecting outliers, and dimensionality reduction. Consequently, for commercialized black-box models, explainers should provide utility for a given sample but don't reveal the entire inner workings of the models. Thus, explainers employed must be local, preferably model-agnostic, and post-hoc. The family of explainers that satisfy these criteria for the models we consider is limited. It consists of SHAP, LIME, counterfactuals/what-ifs (which are discussed in Section 8) and feature importances (already returned by LIME and SHAP).