DualCF: Efficient Model Extraction Attack from Counterfactual Explanations

Yongjie Wang
Nanyang Technological University
Singapore
yongjie002@e.ntu.edu.sg

Hangwei Qian
Nanyang Technological University
Singapore
QIAN0045@e.ntu.edu.sg

Chunyan Miao
Nanyang Technological University
Singapore
ascymiao@ntu.edu.sg

ABSTRACT

Cloud service providers have launched Machine-Learning-as-a-Service (MLaaS) platforms to allow users to access large-scale cloud-based models via APIs. In addition to prediction outputs, these APIs can also provide other information in a more human-understandable way, such as counterfactual explanations (CF). However, such extra information inevitably causes the cloud models to be more vulnerable to extraction attacks which aim to steal the internal functionality of models in the cloud. Due to the black-box nature of cloud models, however, a vast number of queries are inevitably required by existing attack strategies before the substitute model achieves high fidelity. In this paper, we propose a novel simple yet efficient querying strategy to greatly enhance the querying efficiency to steal a classification model. This is motivated by our observation that current querying strategies suffer from decision boundary shift issue induced by taking far-distant queries and close-to-boundary CFs into substitute model training. We then propose DualCF strategy to circumvent the above issues, which is achieved by taking not only CF but also counterfactual explanation of CF (CCF) as pairs of training samples for the substitute model. Extensive and comprehensive experimental evaluations are conducted on both synthetic and real-world datasets. The experimental results favorably illustrate that DualCF can produce a high-fidelity model with fewer queries efficiently and effectively.

CCS CONCEPTS

• **Security and privacy → Software and application security**; • **Computing methodologies → Artificial intelligence**; **Neural networks**; **Machine learning algorithms**; **Reasoning about belief and knowledge**.

KEYWORDS

Counterfactual Explanations, Model Extraction Attack, Decision Boundary Shift, Model Security and Privacy

1 INTRODUCTION

Abundant machine learning models are deployed with automated decision-making ability in various fields such as computer vision [17], medical diagnosis [20], recommender systems [27], language translation [46], healthcare [36] and finance [6]. Due to the model/data privacy and computational capacity, the trained models are usually deployed in the cloud via MLaaS platforms, with only public Application Programming Interfaces (APIs) for remote access on a pay-per-query basis. Inevitably, there exists a tension between public accessibility and model confidentiality. On the one hand, the open APIs should be publicly accessible everywhere and anytime. On the other hand, both the datasets and models are intellectual properties of the owners and should be kept private and confidential since (1) model training requires expensive cost on human power, data collection and computation resource; (2) individual's privacy rights should be protected from potential reveals and attacks. Even so, the great commercial value of cloud model steers adversaries to conduct model extraction attack, i.e., to steal the internal functionality of cloud model to construct a substitute model without expensive cost, which facilitates further data tampering to bypass monitoring [50] and stronger attacks, e.g., adversarial attack [33], model inversion attack [11] and membership inference attack [40].

Model extraction attack aims to obtain a functionally equivalent or near-equivalent machine learning model, which achieves high agreement (up to 100%) with the cloud model with as fewer queries as possible. We consider real-life scenarios for model extraction attack where service providers (e.g., banks, health centers) deploy machine learning services in the cloud and allow remote access via open APIs. Users can query the API with an input to obtain the corresponding prediction. We assume that (1) the training details, architectures and parameters of the cloud model are invisible to users; (2) users can obtain the data format, the number of classes from the public profile; and (3) users can collect a set of samples to query multiple times. As the API is often on pay-per-query basis and the service provider monitors the abnormal data traffic, fewer queries are preferred.

Among all these attacks, we focus on the model extraction attack with counterfactual explanations (CF). Counterfactual explanations answer "what minimum changes are needed for an input instance to alter the current prediction to a particular different one" [48]. Especially, for a given input and a pretrained model, counterfactual explanation methods find explanations that have minimal cost to convert the current prediction to a different particular prediction, usually from an undesirable prediction to a desirable one, subject
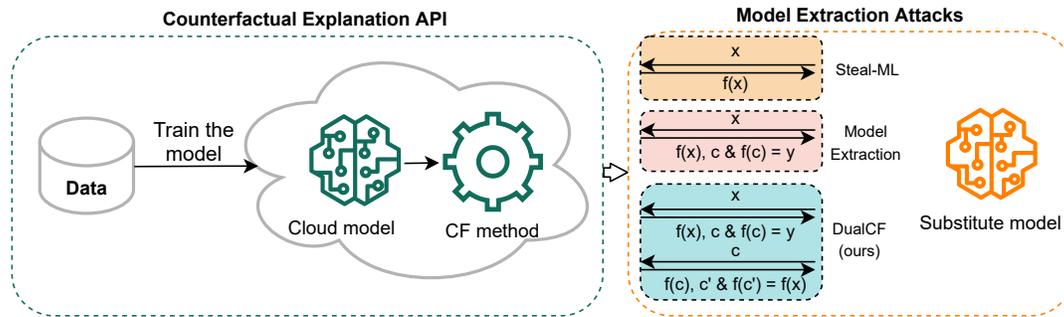
Figure 1: The workflow of model extraction attacks with counterfactual explanations. Service providers train the cloud model on their private data and deploy the counterfactual explanation service in the cloud. Adversaries aim to construct a substitute model by leveraging the information provided by APIs. Three types of model extraction attacks are illustrated. *Steal-ML* [44] only uses the prediction output of the query x. *Model Extraction* [3] considers both the query x and CF c while our proposed DualCF takes CF c and CCF c' into training. In Model Extraction Attacks module, "←" denotes uploading a query to the API and "→" denotes receiving outcomes from the API.

to specified constraints. Note that our proposed approach also uses counterfactual explanation methods to flip desired predictions to undesired ones. This is uncommon for general propose, but such counterfactual explanation methods are capable of flipping predictions of any input. Counterfactual explanations from a desirable prediction to undesirable one reveal the corresponding actions the subject should avoid to prevent the situation from turning worse. As counterfactual explanations offer suggestions with minimum cost to flip the current prediction, therefore, they have broad applications in healthcare (altering an unhealthy situation to a health one), finance (improving the loan approval rate), school admission (obtaining a school offer), paper review (minor revision for paper acceptance), and custom service recovery (regaining the loyalty of unhappy customers). To better illustrate how counterfactual explanations work, we take the loan application as an example: an applicant seeks a mortgage from a loan-granting bank. The applicant submits his/her related information (including age, education, salary, credit score) to the bank. The bank deploys a machine learning model with the binary classifier and then denies this loan application due to the submitted attributes of low salary and poor credit score. Naturally, the individual seeks to know the reason behind the rejection and further to know the necessary changes required before the loan can be approved. The decision-making system equipped with counterfactual explanations is able to provide constructive suggestions in a human-understandable way such as "increasing the salary by 500 and increasing the credit score by 100" to the applicant, showing the minimal improvement required before the application can be approved. Hence, different from original cloud models merely making opaque algorithmic decisions, cloud models equipped with CF provide additional information to better reflect the underlying key factors that explain the outcome of cloud models. Both attacks are illustrated in Fig. 1.

While much research [10, 23, 29, 45] studies how to improve the explanations to meet user requirements, few works study the security and privacy issues of counterfactual explanations, which are essential for safety-critical applications. Despite the intuition that such additional information would naturally enhance the risk

of model leakage, up to now, researchers have not reached a consensus on the security and privacy issues of model extraction attacks with CF. Some works [16, 48] claim that CF allows individuals to receive explanations without conveying the internal logic of the algorithmic black box since it only conveys a limited set of dependencies on an input, while other researchers [3, 42] do not agree with the above viewpoint. Recently, [3] experimentally verifies that adversaries can extract a high-fidelity model with counterfactual explanations leaking the information of decision boundary of cloud model. Nevertheless, due to the black-box property of the cloud model, adversaries need to continuously query the API to collect sufficient information before the substitute model can mimic the cloud model with a high agreement. With this in mind, we seek to propose a simple but efficient querying strategy, named *DualCF*, which greatly reduces the number of required queries, and further lowers the overall cost.

Our querying strategy *DualCF* is motivated from the *decision boundary shift issue* in existing *Model Extraction* method [3]. This issue refers to the decision boundary of the substitute model shifting away from the decision boundary of the cloud model when *Model Extraction* method takes CFs and queries into substitute model training. As it is challenging and almost impossible to obtain the ground-truth decision boundary and training data distribution of the cloud model in advance, the random queries may be far from the decision boundary of cloud model. However, counterfactual explanations are close to the decision boundary regardless of whatever probability threshold of target class is selected for the stop condition. This is because the cloud model tends to assign higher probabilities to all training samples to minimize the training loss. As queries and counterfactual explanations have different predictions, substitute model tends to move towards the middle area between them to separate them confidently. Note that, however, the decision boundary of cloud model is actually close to counterfactual explanations but far from queries. We illustrate this issue in Fig. 2(a) in two-dimensional data space. As queries are randomly selected at each time, the decision boundary of substitute model may shift to different locations, as shown in Fig. 2(a) and Fig. 2(b). To mitigate
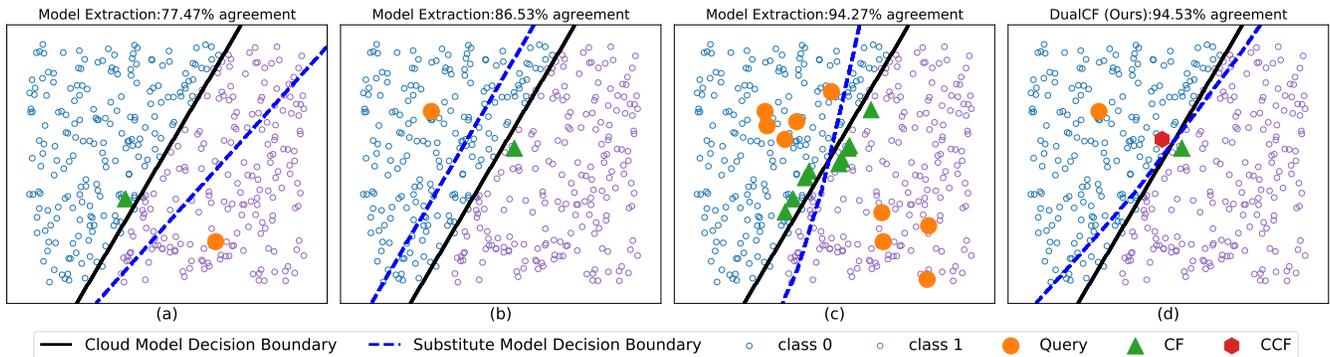
Figure 2: The illustration of decision boundary shift issue. (a) and (b) demonstrate that substitute model decision boundary shifts away from the ground truth due to far-distant queries in existing method *Model Extraction* [3]. (c) shows that *Model Extraction* has to use more queries to mitigate this issue. Our method *DualCF* can achieve comparable agreement with only one pair CF and CCF, as shown in (d), which favorably illustrates the efficacy of our method.

this issue, *Model Extraction* method has to require more queries at a higher attack cost and hence lower efficiency, as shown in Fig. 2(c).

To reduce the number of queries resulting from the decision boundary shift issue, we propose a simple yet efficient method *DualCF*. The gist of our method is to find proper queries with similar distances to decision boundary of cloud model. Note that explicitly selecting the close-to-boundary queries is quite difficult without decision boundary information of cloud model beforehand. To achieve so, our method feeds the current counterfactual explanation (CF) of a query into the open API, and obtains the counterfactual explanation of CF (named CCF for abbreviation). The workflow differences between our DualCF and existing methods (*Steal-ML* [44] and *Model Extraction* [3]) are shown in Fig. 1. Note that CF and CCF are counterfactual explanations of query and CF respectively, and they tend to locate in the region close to the decision boundary of cloud model but with different predictions. Lastly, we train the substitute model using pairs of CF and CCF. As shown in Fig. 2(d), DualCF with only a pair of CF and CCF achieves comparable agreements as *Model Extraction*. Due to the fact that CF and CCF have the similar distances to the decision boundary of the cloud model and have opposite predictions, the decision boundary shift issue in substitute model training is mitigated. CF and CCF work similarly to support vectors in SVM [9] to help infer the decision boundary. For any query, CF and CCF are always located in the close-to-boundary region, a subset of full data space. Therefore, our method is less sensitive to sampling procedure of queries from full data space.

Our proposed DualCF greatly reduces the queries theoretically and experimentally for model extraction attack, which consolidates the viewpoint that counterfactual explanations cause the model leakage. Our paper triggers an alarm about the privacy and security of counterfactual explanation service in the cloud and we hope it motivates the corresponding protection strategy in the future. The main contributions of this paper are three-folds:

- We dive into the fundamental mechanism of counterfactual explanation generation, and bridge the model extraction attack and counterfactual explanation theoretically.

- We observe that the bottleneck of existing attacks come from decision boundary shift issue, which is caused by training on far-distant queries and CF together. To enhance the efficiency, we propose a simple method DualCF that leverages both the CF and the counterfactual explanation of CF (CCF) into the substitute model training. As CF and CCF have the similar distances to the the decision boundary of cloud model, the boundary shift issue is reduced.

- We conduct extensive experiments on synthetic and real-life datasets. Our study shows that our proposed method can extract the high-fidelity model efficiently and effectively compared with existing methods.

2 RELATED WORK

Counterfactual Explanations. More complex models are deployed in the cloud for automatic decisions. Due to the black box nature of cloud models, attempts to explain the internal process for a prediction are required to enhance the model trust and rectify negative decisions. Counterfactual explanations [48] provide a way to understand the reasons on certain predictions and advice how to make smallest changes to receive a desired prediction. The desired prediction contrasts with the fact (current prediction), which is regarded "counterfactual". Counterfactual explanations mainly serve the following purposes: helping users understand why a prediction is made; detecting the model bias for algorithmic fairness [24]; and providing suggestions to receive a desired result from current decision model [22, 45]. This problem is also studied under other research terminologies like recourse [45], inverse classification [26], and contrastive explanation [10]. Following [48], many studies focus on how to model the practical and case-related requirements into mathematical constraints and then solving problems with proper solvers.

Considering some features are immutable (e.g., race, gender), [45] introduces the actionability constraint by freezing the immutable features among improved instance and original query. [19] requires the probability of counterfactual explanations to follow the data

distribution should be large enough to ensure plausible explanations. Diverse explanations are possible to cover multiple choices for each query. Based on this, [29, 38] incorporate diversity constraints on the generated set. As sparser explanations are easier to be adopted for users, [10] enforces sparsity property by adding L_0 or L_1 loss to penalize the changes over many entries of features. In practice, changing a certain feature (e.g., education) may implicitly affect other features (e.g., salary). Hence, it is proper to consider relations [19, 23] or joint effect [34] between feature subsets into modeling. As some features are incomparable, [49] returns the skyline of non-dominated counterfactual explanations. Once the objective and constraints are determined, it is crucial to develop proper solvers to find satisfied solutions. Typically, different solvers are required for different models and data properties. Besides, varying confidential levels of cloud models require different search process of solvers. For example, integer programming [45] or mixed integer programming [38] solvers are used for linear models with integer or categorical features. If a model is differentiable, gradient descent can be adopted [48]. Similarly, model-agnostic method *growing spheres* [26] searches the closest counterfactual explanation from the growing sphere around the query. Dijkstra's algorithm [35] is used for finding a feasible path from the query and the closest counterfactual explanation. Feature tweaking [43] is designed to retrieve the sub-path leading to target prediction from the current prediction path and select the minimum perturbations for a decision tree model.

Model Extraction Attacks. Model extraction attack intends to train a substitute model that approximates the cloud model well in terms of accuracy and fidelity. The accuracy-oriented methods [25, 31] aim to create a substitute model that has similar or better performance on a task as the cloud model, while the fidelity-oriented methods [12, 18, 21] target to reconstruct a high-fidelity substitute model that approximates the decision boundary of the cloud model as faithfully as possible. In addition to the attacks based on counterfactual explanations, it is also possible to conduct attack with *prediction outputs*, and *gradients*. *Prediction outputs* are the most common outcomes from remote APIs, which consist of the discrete predicted class, predicted class probability and probability vector. Adversaries design various attack algorithms [13, 21, 32, 33, 44, 51] to extract a high-fidelity or high-accuracy model from remote predictions. The essential step for these methods is using active learning [7] strategies to either generate informative synthetic points [21, 33, 44, 51] or select max-coverage points [32]. The "informative points" represent the data points close to the decision boundary (i.e., adversarial examples [14]) of the cloud model, and the "max-coverage points" mean the data points should be far distant between each other (i.e., core-set [39]). *Gradients* help explain the model behavior upon an infinitesimal perturbation [41], a.k.a. the sensitivity in the neighborhood. As such, gradients can be used to explain the feature importance for a prediction of a given input. [28] finds that gradient-based feature importance methods can easily expose the cloud model to adversaries. In particular, the gradient of a arbitrary instance is the model weights for a linear model.

Despite the fact that counterfactual explanations do not disclose the cloud model in its entirety, the security and privacy of CFs have largely been overlooked [5, 24]. Some works [16, 48] claim that

counterfactual explanations cannot expose the internal algorithmic logic except a limited set of dependencies on a single instance. Even though such limited information is trivial for extraction, gathering sufficient information with more queries is prominent to conduct model extraction. [42] points out that counterfactual explanations disclose more secrets of cloud model and can enhance the attack efficiency. A recent study [3] firstly conducts the model extraction attack on counterfactual explanations by viewing these close-to-boundary explanations as additional training instances, but we observe that it suffers from the decision boundary shift issue caused by far-distant queries especially when the query size is small. Therefore, it requires higher attack cost to query more times. Our proposed DualCF mitigates the boundary shift issue by introducing a novel querying strategy, which also greatly reduces the querying cost.

3 PRELIMINARIES

A cloud model $f : \mathcal{X} \in \mathbb{R}^d \to \mathcal{Y} \in \mathbb{R}$ takes an arbitrary input query $\boldsymbol{x} \in \mathcal{X}$ as the input, and predicts the output $\in \mathcal{Y}$. In this paper, we assume the cloud model is a pretrained neural network model for classification, parametrized by frozen weights . The counterfactual explanation method $: f \times \mathcal{X} \to \mathcal{X}$ generates a minimal perturbed instance $\boldsymbol{c} \in \mathcal{X}$ for the input instance $\boldsymbol{x}$ such that $f(\boldsymbol{c})$ has the desirable prediction. Formally, searching for the counterfactual explanation $\boldsymbol{c}$ can be framed as the following mathematical formulation,

$$\arg\min_{\boldsymbol{c}} \; d(\boldsymbol{x}; \boldsymbol{c}) \tag{1}$$
$$s\text{:}t\text{:} \; f(\boldsymbol{c}) = \text{ and } f(\boldsymbol{c}) , f(\boldsymbol{x})$$

where $d(\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is a distance (cost) metric measuring the changes between the input $\boldsymbol{x}$ and $\boldsymbol{c}$, and is the desirable target which is different from the original prediction $f(\boldsymbol{x})$. That is, we seek to find counterfactual explanations that belong to a target class while still remains proximal to the original instance. In addition to the constraints in Eq (1), more constraints such as sparsity, feasibility and diversity can be added according to task-specific requirements [47], which are left for future work.

In this paper, we focus on the high-fidelity extraction attack on counterfactual explanations which aims to extract a functional equivalent or near-equivalent model h that behaves very similarly to the model f. It can be formulated as the following mathematical problem: for a set of queries $\mathcal{D}$ and a set of corresponding counterfactual explanations, it finds a substitute model h that performs equivalently on an evaluation set $\mathcal{T}$.

$$\max \; \sum_{\boldsymbol{x}_i \in \mathcal{T}} \mathbb{1}_{f(\boldsymbol{x}_i)=h(\boldsymbol{x}_i)} \tag{2}$$
$$s\text{:}t\text{:} \; h(\boldsymbol{x}) = f(\boldsymbol{x}); \boldsymbol{x} \in \mathcal{D} \tag{3}$$
$$\boldsymbol{c} = (f; \boldsymbol{x}); \boldsymbol{x} \in \mathcal{D} \tag{4}$$
$$h(\boldsymbol{c}) = f(\boldsymbol{c}) \tag{5}$$

The existing method [3] feeds a query $\boldsymbol{x}$ into the API and then obtains the prediction $f(\boldsymbol{x})$ and counterfactual explanation $\boldsymbol{c}$. The pairs of $(\boldsymbol{x}; f(\boldsymbol{x}))$ and $(\boldsymbol{c}; f(\boldsymbol{c}))$ are treated as the training set without discrimination to learn the substitute model h.

4 PROPOSED APPROACH

4.1 Relation between Model Extraction Attack and Counterfactual Explanations

In this section, we observe that counterfactual explanations explicitly reveal not only the decision boundary location of cloud model but also important features of the cloud model. Such information is favorable for model extraction attack.

Here, we illustrate the information leakage through construction of counterfactual explanations. Let $\{x_1; x_2; ::: ; x_d\}$ denote d-dimensional features of an instance $\boldsymbol{x}$, where each feature x_i is associated with a feature importance value w_i w.r.t. cloud model's prediction, e.g., w_i is i-th coefficient if f is a linear model. A larger w_i leads to a larger prediction change if we tweak the feature x_i by the same amount. For simplicity, we consider the scenario with 2 features ($d = 2$) and $w_1 > w_2$. In Eq. (1), note that the prediction change $\Delta f = f(c) - f(\boldsymbol{x})$ from current prediction $f(\boldsymbol{x})$ to desired prediction $f(c)$ should be strictly satisfied, and the explanations with minimal cost (measured by some distance metrics) are preferred. We write the objective on 2d example as,

$$\arg\min_{c} ||(c_1 - x_1; c_2 - x_2)||_p; \text{ s.t. } (c_1 - x_1; c_2 - x_2)(w_1; w_2)^T = \Delta f:$$

As w_1 is larger in our assumption, smaller change $(c_1 - x_1)$ can satisfy the prediction change in the constraint and then associates with a smaller cost in the objective. Therefore, the ideal counterfactual explanation should change the feature x_1 first. From the above analysis, we can conclude that counterfactual explanation methods naturally give high priority to change the features that have larger importance to the desirable prediction.

Essentially, model extraction attack seeks to infer the decision boundary of cloud model from the parameter space. With the favorable information from counterfactual explanations besides prediction output, the search process of substitute model is accelerated. As counterfactual explanations and queries have different predictions, there must exist a decision boundary to separate them. In addition, the feature with minor change corresponds to the important features in the cloud model. Therefore, we can infer the full or partial ranking of important features, which also help enhance the attack efficiency. The closeness of counterfactual explanations to decision boundary of cloud model should be also unearthed for efficient attacks.

## 4.2 Decision Boundary Shift Issue by Far-distant Queries

The straightforward method takes counterfactual explanations and queries as training samples of the substitute model, as pointed in [3]. However, we observe that this method suffers from the decision boundary shift issue, i.e., the substitute model's decision boundary shifts away from the ground truth. The issue is more severe especially when query size is smaller. To relieve the issue, existing methods have to adopt more queries, which result in higher querying cost. Here, we illustrate where this issue comes from and how it affects the attack efficiency.

In the beginning, adversaries know nothing about the decision boundary of cloud model and therefore the queries to upload may be far from the decision boundary of cloud model. However, counterfactual explanations produced by cloud model are usually close to decision boundary, and in the other side of the decision boundary. When we train the substitute model with counterfactual explanations and queries, the decision boundary of the substitute model tends to move to the middle area of the query and CF to separate them confidently. This deviates from the fact that decision boundary of the cloud model is close to counterfactual explanations and may be far from queries. As shown in Fig. 2(a) and (b), the substitute model's decision boundary is distorted by far-distant queries. Due to the sampling variance in queries, the substitute model's decision boundary may move to different regions. This results in a unstable substitute model for an attack method. That is why existing methods need to use more queries to achieve high fidelity and algorithm robustness.

For probabilistic models, the constraint $f(c) =$ in Eq. (1) becomes $p(|c) >$, meaning that the counterfactual explanation method searches the closest explanations above a given probability threshold of target class. As shown in Fig. 3(a), regions closer to decision boundary correspond to probability around $0:5$, wherein the model is uncertain about its predictions. Regions far away from decision boundary, on the contrary, correspond to either smaller or larger values, and the values indicate more confident predictions on class 0 and 1, respectively. In this case, could a higher probability threshold push counterfactual explanations away from the decision boundary to reduce the issue? The answer is yes, but the fact is that even if we set a higher probability threshold, counterfactual explanations are still close to the decision boundary. We observe that this issue roots in the over-confident predictions phenomenon in cloud models. That is, deep neural networks tend to assign high probability to all training instances, resulting in that instances with high probability can be close to the decision boundary, as shown in Fig. 3(d). We discuss this phenomenon in the following.

For a general neural network model for classification, the probability of predicted label belonging to class $k \in [1; K]$ is computed as $p(k) = \frac{\exp(z_k)}{\sum_{i=1}^{K} \exp(z_i)}$ or $p(k) = (z_k)$, where z_i, z_k represent the logits in the last layer of the neural network. The cross-entropy loss for classification is defined as $` = - \sum_{k=1}^{K} q(k) \log(p(k))$, where q is a one-hot vector for ground truth. Minimizing this loss function is equivalent to maximize the likelihood of ground truth, in other words, $z_k >> z_i$ for i, k is desired if the ground truth class is k. As all training points are undifferentiated during training, the model tends to assign higher probability on actual class entry for each instance in order to minimize the loss function. This brings the over-confident prediction problem: the probability increases quickly and flattens if we gradually move an instance away from the decision boundary. We illustrate this with a binary classification model on a synthetic dataset in Fig. 3. The figure shows that more and more instances (i.e., instances in the left side of $< 10\%$ contour and instances in the right side of $> 90\%$ contour) have high probability ($> 90\%$ for class 1 or $< 10\%$ for class 0) and hence the probability density contour lines become denser and denser during training. Therefore, simply adjusting the probability threshold cannot ease the issue since counterfactual explanations are consistently close to the decision boundary due to the over-confident prediction problem.
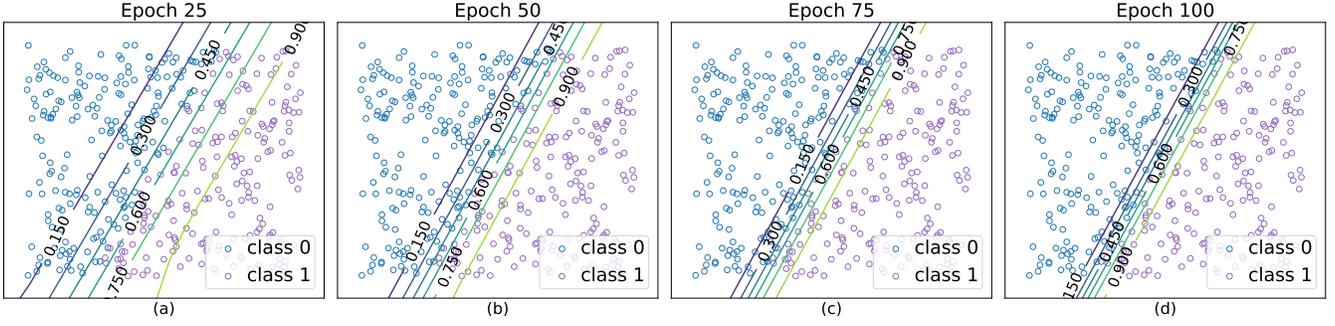
Figure 3: The probability density contour lines during training stage (at epoch 25, 50, 75, 100) of the cloud model trained on a synthetic dataset for binary classification task. The blue and purple dots are sampled points from training set. The values on the contour lines represent the probability calculated by the sigmoid function with range being $[0, 1]$. We can see the model tends to assign high probabilities to more samples during training.

4.3 DualCF

To overcome the decision boundary shift issue resulting from far-distant queries, we propose our algorithm *DualCF* by considering both CF and the counterfactual explanation of CF (CCF) as pairs of training samples for substitute model. As discussed previously, the decision boundary of the substitute model is distorted by taking far-distant queries and close-to-boundary counterfactual explanations into training. Our intuition is that, if training samples with different classes have similar distances to the decision boundary of cloud model, then this issue can be alleviated. As counterfactual explanations locate in the close-to-boundary region, could we search close-to-boundary queries directly? Actually, we do not know the decision boundary of cloud model in the beginning. Without decision boundary information, it is impossible to find the close-to-boundary queries directly. However, we can treat the counterfactual explanation of an arbitrary query as another query to the cloud API and obtain a CCF with the same prediction of original query. Since both CF and CCF are produced by the same counterfactual explanation API, CF and CCF have the similar distances to the decision boundary of the cloud model naturally. In addition, CCF and CF reside in opposite regions of the decision boundary due to their different predictions of classes. As such, the decision boundary shift issue is mitigated by adopting CF and CCF into substitute model training.

The proposed *DualCF* is listed in Algorithm 1. Our method first uploads a query x into the counterfactual explanation API and obtain an explanation c (line 5). Secondly, our method treats current explanation c as another query and obtains a CCF c' (line 6). Similarly, we generate the pair of CF and CCF for each of N queries and denote them as $\mathcal{S}$. Finally, we train the substitute model h with $\mathcal{S}$. It needs to be emphasized that c and its corresponding c' should be used in the same batch in order to mitigate the decision boundary shift issue. Hence, our proposed DualCF is a simple yet efficient querying strategy. The major difference between DualCF and recent studies [3, 44] is how to construct the training set for h. Steal-ML [44] collects $(x, f(x))$, Model Extraction [3] adopts $(c, f(c))$ and $(x, f(x))$ while our proposed DualCF uses $(c, f(c))$ and $(c', f(c'))$. Note that the initial query x is not used as training


Algorithm 1 DualCF

1: **Input**: Queries $\{x_i\}_{i=1}^{N}$, cloud model f, counterfactual explanation API $\mathcal{A}(\cdot)$.
2: **Output**: Substitute model h.
3: Initialize empty training set for substitute model $\mathcal{S} = \{\}$.
4: **for** i< N **do**
5:     $c = \mathcal{A}(f, x)$
6:     $c' = \mathcal{A}(f, c)$
7:     $\mathcal{S} = \mathcal{S} \cup \{(c, f(c)), (c', f(c'))\}$
8: **end for**
9: cur_iter = 0
10: **while** cur_iter < max_iter **do**
11:     Minimize the training loss of h on dataset $\mathcal{S}$.
12:     cur_iter++.
13: **end while**
14: **return** h.

---

set in our DualCF. We also introduce a variant of our proposed method, denoted as DualCFX, which takes x as training data as well, i.e., $(c, f(c)), (c', f(c'))$, and $(x, f(x))$.

We visualize the differences between existing attack methods and DualCF in Fig. 4 on a binary dataset. Two classes (shown as blue and purple dots) are separated by the model f. Three orange dots are three sampled queries, red hexogons and green triangles are CFs and CCFs respectively. The queries, corresponding CFs and CCFs are numbered by $0, 1, 2$. Steal-ML [44] with prediction output only achieves lowest agreement. Model Extraction adds counterfactual explanations into training and achieves higher agreement. We also notice the decision boundary are pulled away from the ground truth by far-distant queries in Fig. 4(c). With CFs and CCFs as training pairs, our method obtains a substitute model with the highest agreement.

From the above analysis, we can see the advantages of DualCF. CF c and CCF c' are both close to decision boundary and have different predictions, not only reducing the boundary shift issue but also bringing a tighter space for inferring the decision boundary of cloud model. The CF c and CCF c work as the support vectors in
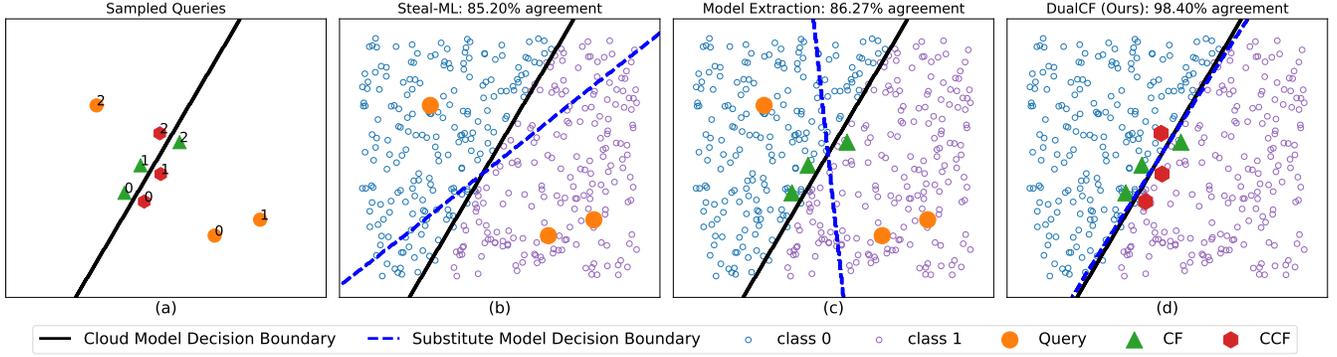
Figure 4: The toy examples on a binary synthetic dataset. We show 3 **queries and corresponding CFs and CCFs (orange dots, green triangles, and red hexagons) in figure (a). Figure (b), (c) and (d) illustrate the substitute model's decision boundary produced by 3 attack methods. Steal-ML uses prediction outputs of queries, Model Extraction applies both queries and CFs, and our proposed DualCF considers CFs and CCFs. Our proposed *DualCF* achieves the highest agreement.**

SVM algorithm [9]. Secondly, the number of each class is balanced favoring the learning process. Thirdly, our method is less sensitive to the sampling procedure of the queries because CF and CCF locate in the denser region close to the decision boundary.

4.4 DualCF for A Linear Model

Here, we illustrate how our DualCF extracts a functionally equivalent model from only a pair of CF and CCF for a linear model. Suppose we have a binary linear model $f = (\langle \ ; x \rangle + b)$, where the decision boundary is determined by the parameter and is the sigmoid function. For a query x, its counterfactual explanation c can be optimally found by searching along the direction of gradient $(f'(x) = f(x) * (1 - f(x)) * \ )$, stopping when it reaches a certain probability belonging to the target class. Using the API, we obtain the CF and CCF of a query.

LEMMA 4.1. *For a binary linear model f , we can extract a substitute model h with* 100% *agreement, from a pair of CF c and CCF c' given an input x.*

PROOF. As the model is linear, the CF and CCF can be optimally found along or against the gradient direction. We draw a straight line through the input point x which is perpendicular to decision boundary. The intersection point of the straight line and decision boundary is denoted as x_0. Then, the CF and CCF can be written as

$$c = x_0 + a \qquad (6)$$
$$c' = x_0 - a \qquad (7)$$

respectively, where a depends on the probability threshold for a counterfactual explanation. As we have c and c' on hand, we can obtain a point $x_0 = \frac{c+c'}{2}$ on the decision boundary. Besides, from any two of x, c and c', we can obtain the slope . With slope and a point in the line, we can obtain the decision boundary without any training. The lemma holds. □

Although extending the lemma to a nonlinear model remains unsolved, this linear scenario verifies that our proposed method can replicate the cloud model efficiently. In experiment section, we

evaluate proposed DualCF in more general situations where the cloud model is nonlinear and complex.

5 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the proposed **DualCF** and its variant **DualCFX** on both synthetic and real-life datasets to compare with state-of-the-art methods.

5.1 Baselines

We employ the following state-of-the-art methods to verify the performance of our method.

- **Steal-ML** [44]. This method first labels the queries with the cloud model f , and then trains the model h on the labelled dataset. We randomly select queries for this method.
- **Steal-ML (CoreSet)** [32]. As random selection may choose redundant or similar samples that do not bring more useful information for extraction, we consider the second baseline that leverages the CoreSet algorithm in [12] to select the most distant samples to query the cloud model.
- **Model Extraction** [3]. This method is proposed to extract a substitute model for counterfactual explanation methods. It utilizes the prediction of queries and counterfactual explanations from the cloud model to train the substitute model.

5.2 Implementation Details

We first train a Multilayer Perceptron (MLP) model f as the cloud model for each dataset. For the counterfactual explanation method, we use the existing algorithm [48] implemented in DiCE [37] (without diverse requirement) for the cloud model. The default objective function $d(\cdot)$ is the $L2$ distance metric on normalized features. We display counterfactual explanations from DiCE on synthetic datasets in Fig.11 in Appendix A.1. After that, we train another MLP model h to approximate cloud model f . For simplicity, cloud model and substitute model have the same architecture, but we train both models with different random initializations and different data. In the following experiments, we also study the influence of substitute models with different capacities from cloud model. Adam

optimizer is used for minimizing the binary cross-entropy loss of cloud model and substitute model. We stop the substitute model training until it reaches a maximum epoch. For fair comparison, all substitute models for all baselines and proposed methods on each dataset have the same training settings and architectures. We introduce more details of substitute model training in Appendix A.2.

5.3 Evaluation Metrics

As we target to build a high-fidelity model h that behaves as similar to f as possible, we firstly define the **agreement** which measures the prediction difference between the f and h on a evaluation set of size n,

$$\text{Agreement} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{f(\boldsymbol{x}_i)=h(\boldsymbol{x}_i)}. \tag{8}$$

A higher agreement is better for substitute models on the same queries. To show how the query size influences the algorithm behavior, we gradually select increasing number of queries and report the agreement on them. A higher agreement curve is better. To reduce the experiment variance due to random sampling of queries and random initialization in model training, we compute the average agreement over 100 runs for a fixed query size. We also compare the **standard deviation (std)** of 100 runs to measure the algorithmic stability. The smaller standard deviation means that the attack method is more stable to the sampling procedure and model initialization. If two model extraction methods have the same agreement, then lower standard deviation is preferred.

5.4 Datasets

We consider the following 5 datasets, which are widely used in research on counterfactual explanations for evaluation.

- **Synthetic Dataset**. We generate two 2-dimensional datasets where all data points are uniformly sampled from a close interval $[0, 6]$. We draw a straight line on the first dataset and a S-curve on the second dataset, to separate data points of each dataset into two parts and assign each part with a label. The first dataset is named **Syn-Linear** and the second dataset is named **Syn-Nonlinear** respectively.
- **Give Me Some Credit (GMSC)** [1]. This dataset is collected for predicting whether someone will experience finance distress in the next two years by his/her financial and demographic information (10 numerical features). The full dataset contains $150, 000$ applicants where $139, 974$ applicants are labeled as "good" and $10, 026$ applicants are labeled as "bad". We select the first $10, 026$ "good" records and all "bad" records to form the final dataset for balance. We follow the pre-processing procedure in [30] to fill the missing values, remove outliers and delete irrelevant features, etc.
- **Heloc Dataset** [8]. It is used for predicting whether an individual will repay the Heloc account in two years by 23 numerical features describing the personal information. The target variable "RiskPerformance" is binary, 5000 "good" records indicating no default , 5459 "bad" records indicating the opposite. We keep the top-10 important features based on the analysis [2] of the IBM team, which is the champion of FICO

Heloc challenge. We adopt the same pre-processing in [2] to remove abnormal values.
- **Boston Housing Dataset** [15]. The dataset has 506 records, where each record has 12 features for predicting the house price in Boston. We follow the pre-processing of the alibi tutorial [4], which firstly transforms the continuous labels into binary classes based on whether the house price is above the median or not, and then removes three categorical features.

We split these datasets into 3 disjointed sets: training, query and evaluation sets at the ratio of 50%, 25%, 25%. We describe them in detail as follows: (1) **training set** (50%) is used for training the cloud model f; (2) **query set** (25%) is used to feed into counterfactual explanation API $(\cdot)$ for obtaining counterfactual explanations, and we train the substitute model h on them. Note that we merely upload a subset of queries to counterfactual explanation API in our experiments; (3) **evaluation set** (25%) is for evaluating how faithful the substitute model h is. As we target to build a high-fidelity model, we use the predicted label instead of the true label on the evaluation set. A validation set for tuning the cloud model f is omitted because we assume the cloud model is given and frozen. Therefore, we directly use the architecture and hyper-parameters from public to train the cloud model f. For the cloud model, we normalize the features by standard normalization where the mean value and variance are estimated on the training set. For the substitute model, we compute the mean value and variance on the full query set for standard normalization.
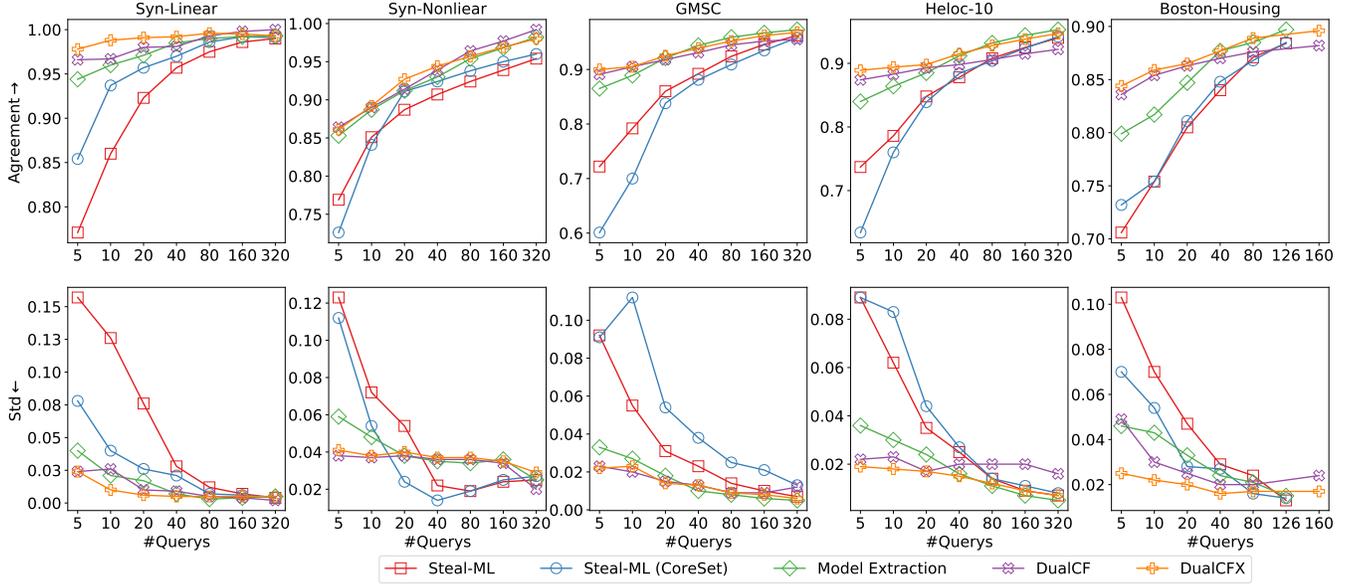
5.5 Experimental Results

We report competitive experimental results on synthetic and real-life datasets in Fig. 5. The x-axis represents the size of queries in each run. The y-axis of the upper figures represents the average agreement of 100 runs while the y-axis of the bottom figures describes the standard deviation of agreements of 100 runs. From them, we can see (1) Extraction methods on counterfactual explanations achieve better results that those merely on prediction output. This is consistent with our previous analysis where counterfactual explanations leak the boundary information of the cloud model which can help the model extraction; (2) DualCF and DualCFX achieve the best agreement on five datasets when the query size is small; (3) DualCFX will improve the DualCF slightly because it takes the queries into training. More training data will result in better performance. Therefore, we suggest adopting the DualCFX as the default method; (4) The standard deviation of agreements of our proposed methods is smaller than baseline methods in general, which means our methods are less sensitive to the sampling procedure; (5) When the query size increases, the gaps between all algorithms decrease because too many predictions from the cloud model are enough to train a good substitute model.

5.6 Ablation Studies

Next, we perform several ablation studies to investigate the effects of some experiment factors on extraction performance.

Model Capacity. We first study how capacity of the substitute model affects the extraction results. The baseline substitute model is a MLP classifier, and we consider three variants of the baseline architecture: removing 50% nodes of last layer, adding 50% nodes to

Figure 5: Experiment results on synthetic and real-life datasets. The upper figures report the average agreement while the bottom figures report the standard deviation of agreements of 100 runs for a fixed size of queries. ↑/↓ mean the higher/lower results are better.

the last layer, and adding one layer. We detail the baseline architectures and their variants on each dataset in Table 2 in Appendix A.3. We train these models on the same query set and counterfactual explanations, and then report the agreement in Fig. 6. We can see the differences between substitute models with different capacities are inconsiderable, implying that the model capacity does not play a key role in the extraction attack. A similar conclusion can be found in [3, 33]. As the architectures of the cloud model are often unknown to adversaries, selecting the best model is challenging. Adversaries can select a substitute model based on prior knowledge, e.g., the state-of-the-art models for the same or similar task.

Threshold. Counterfactual explanation methods search an explanation above a probability threshold of target class. Here we empirically investigate the influence of different thresholds. We set the threshold from 0.6 to 0.9 by 0.1 and generate different counterfactual explanations for each threshold with the same method. Then, we train a substitute model with DualCF on CFs and CCFs for each threshold respectively. The experiment results are shown in Fig. 7. The agreement differences between different thresholds are slight on the five datasets. This is consistent to our discussion in Section 4.2. In the training stage, the cloud model tends to assign high probability to the training points to minimize the training loss. With a higher probability threshold, CF and CCF are still close to the decision boundary of cloud model.

The Distance Metric $d(\cdot)$. A proper distance metric is essential for finding meaningful proximal instances. Here, we investigate the efficacy of the following three commonly used distance metrics, i.e., 1) $L1$: $\sum_{i=1}^{d}(x_i - c_i)^2$; 2) $L2$: $\sum_{i=1}^{d}|x_i - c_i|$; and 3) L1/MAD: $\sum_{i=1}^{d}\frac{|x_i - c_i|}{MAD_i}$ with MAD_i is the median absolute deviation of feature i over the training set.

As shown in Fig. 8, our proposed model is consistently robust to the choice of distance metric on synthetic datasets. On the rest datasets, however, such robustness does not stand when the number of queries is small, i.e., L2 distance metric outperforms $L1$ and $L1/MAD$ when the queries are limited. We conjecture that it is due to the sparsity nature of $L1$-based metrics, where such sparse explanations do not fit the high-dimensional input space.

Imbalanced Dataset. This experiment was conducted to illustrate that performance of attack methods based on counterfactual explanations are more stable on the imbalanced dataset. In this experiment, we choose an imbalanced GMSC query dataset where the "Good" applicants are 5 times more than "Bad" applicants. The evaluation set is kept unchanged. This is the general case in some applications like fraud detection, healthcare. We report experiment results on imbalanced query set in Fig. 9. Compared with the results on balanced datasets in column 3 in Fig. 5, we can see the variance of attack methods based on cloud model's prediction are larger. However, attack methods for counterfactual explanations are more stable due to the balanced classes during the substitute model training.

Shuffling CFs and CCFs. We claim that CF and its CCF should be in the same batch during training. Fig. 10 reports the experiment difference between with/without shuffling CFs and CCFs. Experiment results reveal that training with pairs of CFs and CCFs in the same batch achieves better results. The shuffle operation will destroy the propose of our method that we should train on pairs of instances with similar distance to distance boundary of cloud model and with different predictions.
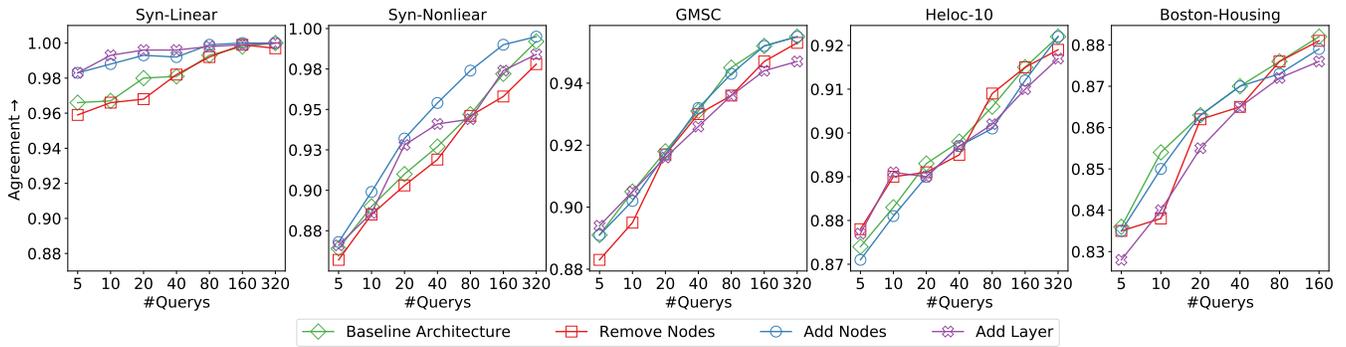
Figure 6: Experiment comparisons on different model capacities in proposed DualCF. "Remove Nodes", "Add Nodes", and "Add Layer" represent three variants of baseline architectures.
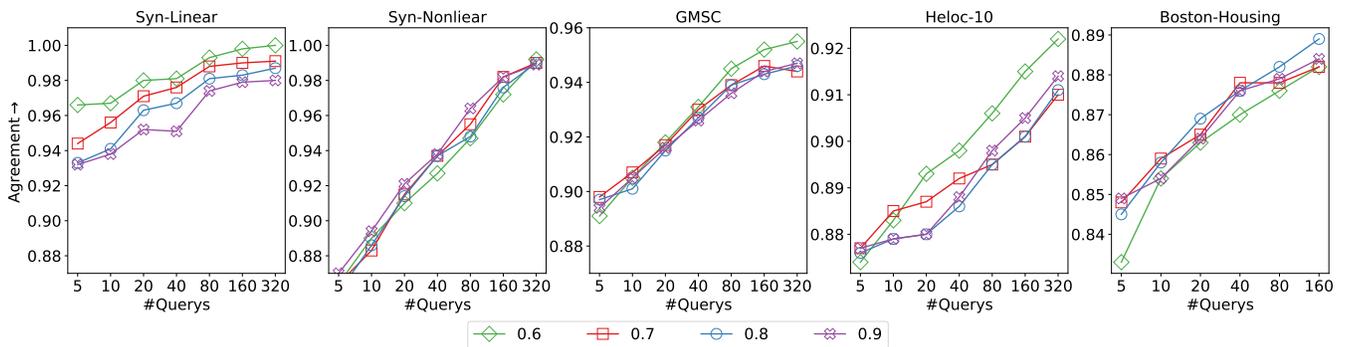
Figure 7: Experimental comparisons on different thresholds in counterfactual explanation generation in our proposed DualCF. We select different thresholds {0.6, 0.7, 0.8, 0.9} as the prediction confidence to generate different CFs and CCFs.
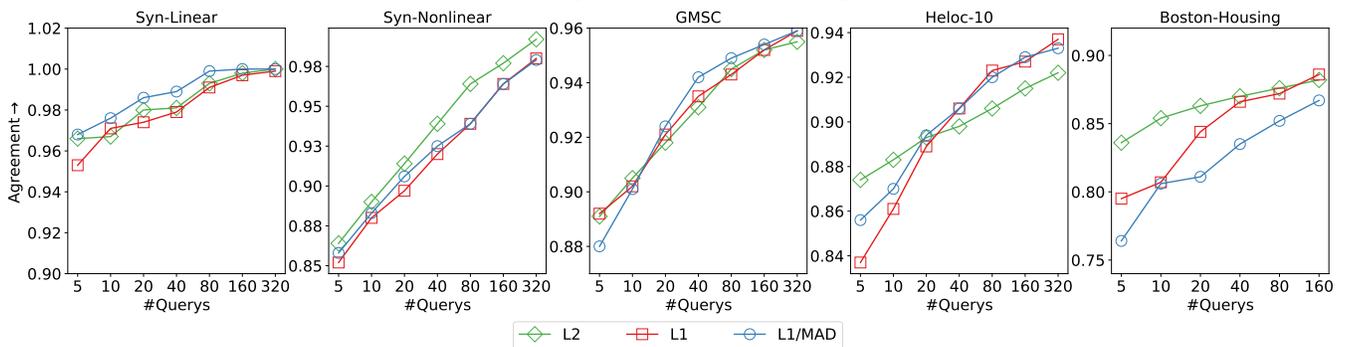
Figure 8: Experimental comparisons on different distance metrics in our proposed DualCF.

6 CONCLUSION

Counterfactual explanation method searches the minimum perturbations for an input to achieve a particular different prediction, which has broad applications. However, the boundary information leaked from counterfactual explanations is prone to model extraction attacks. In this paper, we propose a simple yet efficient method DualCF that mitigates the decision boundary shift issue in existing methods. Extensive experiments demonstrate that our method can achieve a high-fidelity model with much fewer queries. Our work raises the awareness of the privacy and security issues of counterfactual explanations, and further motivates the countermeasures (e.g., monitoring malicious queries, adding superfluous features on model training and counterfactual explanations for misleading adversaries, restrict one-way generation of counterfactual explanations.) to protect the cloud model. In the future, we will work on more general attacks and defenses on multi-class classification models and regression models, and explore the influence of different practical constraints on the security of counterfactual explanations.
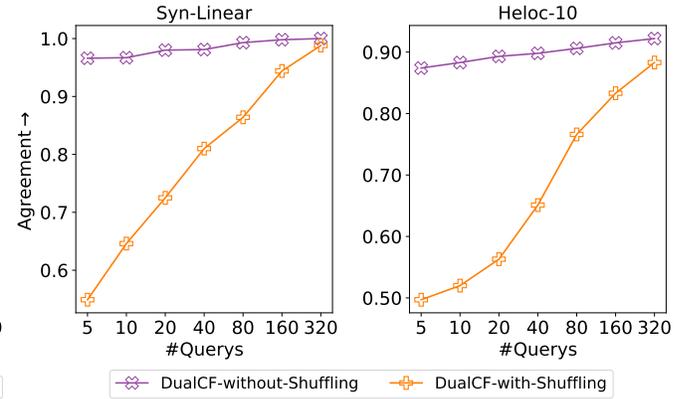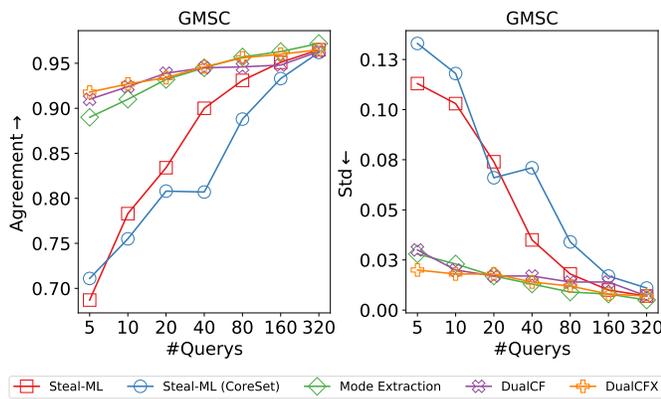
Figure 9: Experiment results on imbalanced GMSC query set.

Figure 10: Experiment results on with/without shuffling for DualCF.

ACKNOWLEDGMENTS

REFERENCES

[1] [n. d.]. *Give Me Some Credit Dataset.* https://www.kaggle.com/c/GiveMeSomeCredit/overview

[2] IBM AI Explainability 360. [n. d.]. *Credit Approval.* https://nbviewer.org/github/IBM/AIX360/blob/master/examples/tutorials/HELOC.ipynb

[3] Ulrich Aïvodji, Alexandre Bolot, and Sébastien Gambs. 2020. Model extraction from counterfactual explanations. *arXiv preprint arXiv:2009.01884* (2020).

[4] SeldonIO alibi. [n. d.]. *Counterfactuals guided by prototypes on Boston housing dataset.* https://docs.seldon.io/projects/alibi/en/latest/examples/cfproto_housing.html

[5] Solon Barocas, Andrew D. Selbst, and Manish Raghavan. 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020,* Mireille Hildebrandt, Carlos Castillo, L. Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna (Eds.). ACM, 80–89. https://doi.org/10.1145/3351095.3372830

[6] Luyang Chen, Markus Pelger, and Jason Zhu. 2020. Deep learning in asset pricing. *Available at SSRN 3350138* (2020).

[7] David Cohn, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine learning* 15, 2 (1994), 201–221.

[8] FICO Community. [n. d.]. *Explainable Machine Learning Challenge.* https://community.fico.com/s/explainable-machine-learning-challenge

[9] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[10] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. 2018. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems.* 592–603.

[11] 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.* 1322–1333.

[12] Xueluan Gong, Yanjiao Chen, Wenbin Yang, Guanghao Mei, and Qian Wang. 2021. InverseNet: Augmenting Model Extraction Attacks with Training Data

Inversion. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21,* Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2439–2447. https://doi.org/10.24963/ijcai.2021/336 Main Track.

[13] Xueluan Gong, Qian Wang, Yanjiao Chen, Wang Yang, and Xinchang Jiang. 2020. Model Extraction Attacks and Defenses on Cloud-Based Machine Learning Models. *IEEE Communications Magazine* 58, 12 (2020), 83–89.

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[15] David Harrison Jr and Daniel L Rubinfeld. 1978. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management* 5, 1 (1978), 81–102.

[16] Masoud Hashemi and Ali Fathi. 2020. PermuteAttack: Counterfactual Explanation of Machine Learning Credit Scorecards. *arXiv preprint arXiv:2008.10138* (2020).

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[18] 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 20).* 1345–1362.

[19] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. 2019. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615* (2019).

[20] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873 (2021), 583–589.

[21] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N Asokan. 2019. PRADA: protecting against DNN model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P).* IEEE, 512–527.

[22] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. 2021. Algorithmic Recourse: from Counterfactual Explanations to Interventions. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021,* Madeleine Clare Elish, William Isaac, and Richard S. Zemel (Eds.). ACM, 353–362. https://doi.org/10.1145/3442188.3445899

[23] Amir-Hossein Karimi, Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. 2020. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *Advances in Neural Information Processing Systems* 33 (2020).

[24] Atoosa Kasirzadeh and Andrew Smart. 2021. The Use and Misuse of Counterfactuals in Ethical Machine Learning. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021,* Madeleine Clare Elish, William Isaac, and Richard S. Zemel (Eds.). ACM, 228–236. https://doi.org/10.1145/3442188.3445886

[25] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2019. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In *International Conference on Learning Representations.*

[26] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2017. Inverse classification for comparison-based interpretability in machine learning. *arXiv preprint arXiv:1712.08443* (2017).

[27] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. 2015. Recommender system application developments: a survey. *Decision Support*

Systems 74 (2015), 12–32.

[28] 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*. 1–9.

[29] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.

[30] Nicholas. [n. d.]. *EDA - Credit Scoring, Top 100 on Leaderboard*. https://www.kaggle.com/nicholasgah/eda-credit-scoring-top-100-on-leaderboard

[31] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4954–4963.

[32] Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. 2020. Activethief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 865–872.

[33] 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*. 506–519.

[34] Neel Patel, Martin Strobel, and Yair Zick. 2021. High Dimensional Model Explanations: An Axiomatic Approach. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, Madeleine Clare Elish, William Isaac, and Richard S. Zemel (Eds.). ACM, 401–411. https://doi.org/10.1145/3442188.3445903

[35] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 344–350.

[36] Hangwei Qian, Sinno Jialin Pan, Bingshui Da, and Chunyan Miao. 2019. A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition.. In *IJCAI*. 5614–5620.

[37] Chenhao Tan Ramaravind K. Mothilal, Amit Sharma. [n. d.]. *Diverse Counterfactual Explanations (DiCE) for ML*. https://github.com/interpretml/DiCE

[38] Chris Russell. 2019. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 20–28.

[39] Ozan Sener and Silvio Savarese. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*.

[40] 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)*. IEEE, 3–18.

[41] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).

[42] Kacper Sokol and Peter A Flach. 2019. Counterfactual explanations of machine learning predictions: opportunities and challenges for AI safety. In *SafeAI@ AAAI*.

[43] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. 2017. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 465–474.

[44] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 601–618.

[45] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 10–19.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[47] Sahil Verma, John Dickerson, and Keegan Hines. 2020. Counterfactual Explanations for Machine Learning: A Review. *arXiv preprint arXiv:2010.10596* (2020).

[48] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31 (2017), 841.

[49] Yongjie Wang, Qinxu Ding, Ke Wang, Yue Liu, Xingyu Wu, Jinglong Wang, Yong Liu, and Chunyan Miao. 2021. The Skyline of Counterfactual Explanations for Machine Learning Decision Models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2030–2039.

[50] Weilin Xu, Yanjun Qi, and David Evans. 2016. Automatically evading classifiers. In *Proceedings of the 2016 network and distributed systems symposium*, Vol. 10.

[51] Honggang Yu, Kaichen Yang, Teng Zhang, Yun-Yun Tsai, Tsung-Yi Ho, and Yier Jin. 2020. CloudLeak: Large-Scale Deep Learning Models Stealing Through Adversarial Examples.. In *NDSS*.