The Limits of Provable Security Against Model Extraction

1^{st} Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Can we hope to provide *provable* security against model extraction attacks? As a step towards a theoretical study of this question, we unify and abstract a wide range of "observational" model extraction defense mechanisms — roughly, those that attempt to detect model extraction using a statistical analysis conducted on the distribution over the adversary's queries. To accompany the abstract observational model extraction defense, which we call OMED for short, we define the notion of *complete* defenses – the notion that benign clients can freely interact with the model – and *sound* defenses – the notion that adversarial clients are caught and prevented from reverse engineering the model. We then propose a system for obtaining provable security against model extraction by complete and sound OMEDs, using (average-case) hardness assumptions for PAC-learning.**

Our main result nullifies our proposal for provable security, by establishing a computational incompleteness theorem for the OMED: any efficient OMED for a machine learning model computable by a polynomial size decision tree that satisfies a basic form of completeness cannot satisfy soundness, unless the subexponential Learning Parity with Noise (LPN) assumption does not hold. To prove the incompleteness theorem, we introduce a class of model extraction attacks called *natural Covert Learning attacks* based on a connection to the Covert Learning model of Canetti and Karchmer (TCC '21), and show that such attacks circumvent *any* defense within our abstract mechanism in a black-box, nonadaptive way.

Finally, we further expose the tension between Covert Learning and OMEDs by proving that the existence of Covert Learning algorithms *requires* the nonexistence of provable security via efficient OMEDs. Therefore, we observe a "win-win" result, by obtaining a characterization of the existence of provable security via efficient OMEDs by the nonexistence of natural Covert Learning algorithms.

***Index Terms*—Model Extraction, Model Stealing, Covert Learning, Adversarial Machine Learning, Provable Security.**

I. Introduction

In a *model extraction attack*, an adversary maliciously probes an interface to a machine learning model in an attempt to extract the machine learning model itself. In many cases, preventing model extraction helps increase security and privacy, especially with respect to model inversion and adversarial example attacks (see e.g. [1] and references therein). Additionally, in Machine Learning as a Service (MLaaS), the model is considered confidential as the server usually operates with a pay-per-query scheme. Therefore, maintaining the secrecy of ML models and finding effective model extraction defense mechanisms is paramount. Indeed, the problem of how to defend against model extraction has been considered from a practical perspective previously (e.g. [2]–[6]).

Most proposed model extraction defenses (MEDs) in the literature belong to two types (except a few notable exceptions, see e.g. [7]). The first type aims to limit the amount of information revealed by each client query. One intuitive proposal for this type of defense is to add independent noise (i.e. respond an incorrect prediction independently with some probability) or even deliberately modify the underlying model. This type of solution is not a focus of this work, because it necessarily sacrifices predictive accuracy of the ML model, and is therefore not an option for many ML systems where accuracy is critical such as autonomous driving, medical diagnosis, or malware detection.

The second type of MED that has been proposed aims to separate "benign" clients — those that want to obtain predictions but will not attempt to extract the model — and "adverse" clients — clients that aim to extract the model. This type of "observational" defense is the focus of the present work. A common implementation of the observational defense involves so-called "monitors" that, receive as input a batch of queries submitted by the client, and compute some statistic meant to measure the likelihood of adversarial behavior, with the goal of rejecting a client's requests when the queries pass a certain threshold on the statistic (e.g. [2], [5], [6]). Essentially, observational defenses aim to control the *distribution* of the client's queries, by classifying any clients that fail to conform to the appropriate distributions as adverse, and then prohibiting them from accessing the model. To date, the choice of such appropriate distributions have been made heuristically, for instance, in [5], an appropriate distribution is one with the property that the distribution over hamming distances between independent samples is normally distributed.

However, no formal definitions of security against model extraction have been suggested, and there has also not been much formal work done in an effort to understand the theoretical underpinnings of the proposed observational defenses, in particular. This fact is highlighted by Vaikuntanathan as an open problem in [8]. As a result, a "cat-and-mouse" progression of attacks and defenses has developed, while no satisfying guarantees have been discovered (for neither cat nor mouse).

A. Our Contributions

In this work, we study the landscape of model extraction attacks and defenses from a theoretical perspective. We seek an answer to the following three-part question:

> Can we provide *provable security* against model extraction attacks, for any ML model, using an observational defense? If so, can it be efficiently implemented? In fact, how do we even define provable security?

We propose a broad method for obtaining cryptographic-strength provable security via an observational defense, and then provide a negative answer to the second part of the question, via the following program:

- We formally define a class of abstract MEDs by unifying the common observational defense technique seen in the literature.
- We formalize the concepts of *complete* and *sound* defenses, namely, the provable guarantees that benign clients are accepted and may interact with the machine learning interface, and adverse clients are rejected. We show how our formalisms initiate a method for obtaining a theory of provable security against efficient model extraction acttacks by relying on the computational hardness of PAC-learning.
- Then, via a connection to the Covert Learning model of [9], we give a method for generating provably good and efficient attacks on the abstract defense, granted that the defense is efficient and it satisfies a basic form of completeness. We then obtain an attack on decision tree models protected by the abstract class of MEDs by an existing algorithm of [9]. The attack relies on the subexponential Learning Parity with Noise (LPN) assumption, and to the best of our knowledge, constitutes the first provable and efficient attack on any large class of MEDs, for a large class of ML models.
- Using the existence of the attack, we prove our main result: informally, every efficient defense mechanism (within the abstract class of MEDs) for decision tree models which satisfies a basic notion of completeness does not satisfy soundness, even for efficient attackers. This result essentially prevents instantiating the OMED method for provable security.

Additionally, we further develop the relationship between Covert Learning and OMED defenses. As a consolation to the negative result, we prove that "either way science wins": provable security by OMEDs exist *if and only if* Covert Learning algorithms do not exist.

B. Technical Overview and Approach

Let us describe our approach in more detail.

1) The Observational Defense and Provable Security: First, we informally describe our abstract observational model extraction defense mechanism (OMED). The OMED can be described in the following simple terms: it analyzes the requested queries, and decides if the queries are distributed in an acceptable manner. It is generally said that this is meant to decide if the client's actions are consistent with a benign client, or an adversarial client (i.e. one who is attempting to perform model extraction).[1]

a) Observational Model Extraction Defense: An OMED for an ML model is an algorithm which takes as input a sequence of queries and outputs either accept or reject.

In other words, the OMED makes a decision about the nature of the client which depends solely on the client's query selection. As noted, this technique captures one large camp of the candidate MEDs seen in the literature, and is purposely defined as broadly as possible so as to strengthen our negative result. In Section III-B, we highlight and discuss how three recent proposals (Extraction Monitor [2], PRADA [5], and VarDetect [6]) implement special cases of the OMED.

Towards obtaining a theory of provable security against model extraction, we argue that, intuitively, a good OMED (and any MED) should satisfy:

b) MED Completeness: For any benign client, the defense mechanism does not reject the client and allows the benign client to continue to interact with the model, with high probability.

A completeness requirement on a MED can be interpreted as formalizing the *usefulness* of the defense. In other words, the defense at the very least provides the opportunity for benign clients to interact with the model. On the other hand, the MED should provide some nontrivial security guarantee. We consider *soundness* of a MED:

c) MED Soundness: For any adverse client that has attempted extraction, the defense mechanism rejects the client, with high probability.

A soundness requirement can be interpreted as formalizing the *security* of the defense. That is, attackers will not be able to deceive the OMED into granting interaction with the ML model in such a way that allows extraction.

d) Cryptographically-Hard Model Extraction: In Section III-A, we show how to combine completeness, soundness, and hardness assumptions for average-case or heuristic PAC-learning (see e.g. [10], [11]) to obtain cryptographic-strength security against all efficient model extraction attacks. Informally, we show,

Theorem I.1. *(Informal version of Theorem III.10) Let $\mathcal{M}$ be any OMED satisfying completeness and soundness, for a class of ML models $\mathcal{F}$. Then, if $\mathcal{F}$ has no efficient average-case PAC-learning algorithm, then there exists a large subset S of ML models within $\mathcal{F}$ such that any efficient client cannot extract a good approximation to any $f \in S$, except with negligible probability.*

For an in-depth discussion on the road to provable security for OMEDs, see Section I-B4. We define average-case PAC learning in Section II.

[1] In Section I-B4 we argue that the analysis of adverse vs. benign clients that is prevalent in the literature implicitly employs a simulation-based definition of security against model extraction.

2) Generation of Attacks: Next, we introduce a way to generate model extraction attacks that are effective in *efficiently* performing high fidelity model extraction in the presence of *any* efficient and complete OMED. A model extraction attack has high fidelity if, with respect to the underlying model and some fixed loss function, the extracted model achieves a low loss.

The attack operates in the following setting. Let $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ be an ensemble of function classes, with input space $\mathcal{X}_n = \{0,1\}^n$ and output space $\mathcal{Y}_n = \{-1,1\}$ (we use the labels $\{-1,1\}$ for mathematical convenience). An entity learns a proprietary model $f \in \mathcal{F}_n$. The entity then provides an oracle to f accessible by clients, monitored by and OMED $\mathcal{M}$, which is itself a (possibly randomized) algorithm. $\mathcal{M}$ acts as an external decision making process for preventing model extraction (see Section III for a graphical depiction). Using the recently introduced notion of Covert Learning algorithms [9] (see Section I-B5 for a more detailed overview, and Section IV-A for a formal treatment), we show:

Theorem I.2. *(informal version of Theorem IV.3). Assuming the existence of a natural Covert Learning algorithm, there exists a probabilistic polynomial time (p.p.t.) model extraction attack $\mathcal{A}$ such that $\mathcal{A}$ has high fidelity and any complete OMED $\mathcal{M}$ outputs accept when given a batch of queries from $\mathcal{A}$, with high probability.*

We stress that the class of attacks is *nonadaptive*, in the sense that it does not rely on any knowledge of the specific implementation of the OMED, nor the underlying model. Moreover, we call the class of model extraction attacks *natural Covert Learning attacks*, as they rely on natural Covert Learning algorithms — special cases of the Covert Learning model of [9].

Finally, we instantiate a concrete universal model extraction attack, by invoking an algorithm of [9]. The algorithm of [9], called CLDT, learns a decision tree $f : \{0,1\}^n \to \{-1,1\}$ in the PAC with membership queries model — with the added constraint that the membership queries are drawn from a distribution which is cryptographically pseudo-random. The algorithm relies on the subexponential hardness of the standard Learning Parity with Noise (LPN) problem (to be defined formally in Section IV-C).

We say that an OMED satisfies uniform completeness if it outputs accept on a batch of uniformly distributed queries, with high probability. We prove:

Theorem I.3. *(informal version of Theorem IV.6). Assuming subexponential hardness of the standard LPN problem, there exists a p.p.t. model extraction attack $\mathcal{A}$ for decision tree classifiers such that $\mathcal{A}$ has high fidelity and any OMED that satisfies uniform completeness must output accept when given a batch of queries from $\mathcal{A}$, with high probability.*

We note that, in contrast to typical PAC-learning model extraction attacks (outlined in Section I-B4), the attack is *provably efficient*; it runs in time polynomial in n.

We then use Theorem I.3 to prove our main result:

Corollary I.4. *(informal version of Corollary IV.7). Assuming subexponential hardness of the standard LPN problem, any efficient OMED (for a decision tree classifier) which satisfies uniform completeness cannot satisfy soundness, even against efficient adversaries.*

The main result follows from Theorem I.3 because of the nature of the attack. In short, the adversary poses as a benign client by requesting queries according to a distribution which is computationally indistinguishable from that which would be requested by a truly benign client. Since the completeness requirement of the OMED dictates that the honest client would "pass" the OMED, we may conclude that the adversary — which operates in a way computationally indistinguishable from honest — must also pass (with high probability).

3) A Characterization of Feasible OMEDs by Nonexistence of Covert Learning: As consolation for the main result, we show that non existence of efficient OMEDs is sufficient to imply Covert Learning algorithms. This is a "win-win" dynamic, in that it gives a proof of:

Corollary I.5. *(informal version of Corollary V.2) The following statements are equivalent.*

1) *There exists an OMED (for a class of ML models $\mathbb{C}$) that is complete and sound.*
2) *There does not exist a natural Covert Learning algorithm for $\mathbb{C}$.*

4) Discussion: Towards Complexity-Based Provable Security of Observational Defenses: Inspired by Modern Cryptography, a lofty goal behind developing a theory of security against model extraction would be to ultimately obtain *provable* security guarantees. For example, an initial attempt could attempt to leverage zero-knowledge style ideas, to obtain a guarantee that a client learns nothing about the ML model than they could have learned prior to the interaction. However, this is likely too strong of a goal, because at the very least the client will learn some queried examples.[2]

What kind of guarantees could we feasibly hope to obtain, then? One possible revised goal, could be to guarantee that a client learns only as much as possible from some *random examples* from the model (perhaps using some simulation-based security definition). In fact, this notion of security appears to be implicitly behind existing observational defenses. The literature on practical observational defenses tends to cite the goal of *detecting* model extraction, but the downstream effect is that the observational defenses seek to *exactly confine* the examples obtained by the client to some *specific distributions* (by enforcing a particular benign behavior). Hence, the idea of only serving clients confined to these benign example

[2] For example, in the setting of Machine Learning as a Service (MLaaS), the client must be granted "in good faith" at least some ability to learn information, since otherwise the client may take business elsewhere.

distributions undoubtedly assumes that whatever the benign client can learn about the model is indeed "secure."

To unpack this, let us focus on the case of observational defenses for binary classifiers. At first glance, the beautiful learning theory of Vapnik and Chervonenkis — which tells us that a number of samples proportional to the VC dimension of the hypothesis class suffices for PAC learning — seems to dash the hopes of using this model of security to obtain any meaningful protection. Indeed, an adversary could simply query the model a sufficient number of times according to one of the appropriate distributions of random examples, and then apply a PAC-learning algorithm. The output of the algorithm would be a function which would be a strong approximation to the underlying ML model with high confidence.

However, this view does not account for the *complexity* of such attacks. Indeed, for many important families of classifiers (e.g. boolean decision trees), no efficient (i.e., polynomial time) PAC-learning algorithms are known despite intense effort from the learning theory community.[3] In fact, no efficient algorithms are known even when the examples are restricted to being uniformly distributed, and the classifier itself is drawn from some kinds of distributions (i.e., in an average-case way, see e.g. [10]). Hence, this lends credence to the idea that the revised model of security might actually effective in preventing unwanted model extraction by *computationally bounded clients*, for instance by forcing the client to interact with the query interface in a way that mimics uniformly random examples, or some other hard example distribution. In this way, security against model extraction could be *provable* in a complexity-theoretic way: one could give a reduction from PAC-learning to model extraction in the presence of observational defenses. In other words, one could hope to prove a theorem that says "any efficient algorithm to learn an approximation of a proprietary ML model when constrained by an observational defense yields a distribution-specific PAC-learning algorithm (that is currently beyond all known techniques)."

One potential pitfall of the preceding discussion of provable security is that due to the worst-case guarantees for PAC-learning, the described reduction would not rule out the useless case that a single model is hard to extract in the presence of observational defenses, but all others are easy. However, even in an average-case or heuristic PAC-learning setting, where the concept itself is drawn from a distribution (see [10], [11]), there is still a conjectured cryptographic hardness of learning for sufficiently complex classes of concepts and concept distributions. Therefore, we can continue to envision a reduction from average-case learning to model extraction for *most* underlying ML models (provided they are sufficiently complex to begin with). Such provable security against efficient model extraction adversaries would be a significant development in finding the theory behind observational defenses.

5) Discussion: Relating Covert Learning to Model Extraction in the Presence of Observational Defenses: Our results draw heavily from the work of [9] and thus they require familiarity with the Covert Learning model. Let us first give a brief overview of the necessary ideas in this section,[4] and illuminate the connection between Covert Learning and our setting of Model Extraction.

The Covert Learning model — a variant of Valiant's PAC model in the agnostic learning with membership queries setting — formalizes a new type of privacy in learning theory. Specifically, the Covert Learning algorithms provide the guarantee that the membership queries leak very little information about the concept or the learner's hypothesis class with respect to a computationally bounded passive adversary. In other words, the learner can PAC-learn the concept in question (using knowledge of secret internal randomness), while the adversary remains nearly completely "in the dark" with respect to the concept and the learner's hypothesis, even when it views the entire transcript of membership queries and oracle responses (but is not privy to the secret randomness). At its heart, the Covert Learning model uses the foundational simulation paradigm of cryptography to achieve these goals. Roughly, any Covert Learning algorithm must have an accompanying simulator that emulates the membership queries (the "real" learner) in a computational indistinguishable way — using nothing but random examples (the "ideal" learner).[5]

Originally, the Covert Learning model was introduced along an application to the secure outsourcing of automated scientific experiments, and a brief note regarding the possibilities of model extraction attacks against MLaaS. The crux of this work thus formalizes this connection between the adversary in Covert Learning — a *distinguisher* that attempts to differentiate between a "real" learner and and "ideal" learner — and the "adversary" in a model extraction attack — an OMED. In particular we consider the abstract OMED, and show that "natural" Covert Learning algorithms can fool the OMED in the same way that they fool the Covert Learning adversary (roughly, a Covert Learning algorithm is "natural" if the membership queries are pseudo-random).

Thus, our attacks work by leveraging the Covert Learning guarantees to generate a distribution of queries which is *computationally indistinguishable* from a distribution which will be considered benign by the OMED. Still, by the guarantees of Covert Learning, the responses to the queries given by the server allow the client to extract the underlying model. Hence, we can show that a single Covert Learning attack can achieve high fidelity while "fooling" any OMED. That is, any OMED will output "accept" with high probability even though an extraction attack is being performed. Since the attack is completely black-box with respect to the implementation

[3] We note that there exist efficient learning algorithms for polynomial size decision trees that use *correlated queries* [12] [13]. Therefore, these families of classifiers are at least efficiently learnable by a server who had this type of data access, so the setting is still relevant.

[4] We refer the reader to [9] for technical details, and a deeper conceptual exposition.

[5] To elaborate a little, the simulator functions *without* access to the underlying concept (or knowledge of the hypothesis class, in the case of agnostic learning). Instead, the simulator accesses, for instance uniformly random examples (which for many interesting hypothesis classes makes the learning requirement hard, hence the use of the *real/ideal* nomenclature).

of the OMED (it only requires that the OMED is *efficient* and satisfies the basic uniform completeness condition), the existence of this attack demonstrates the *incompleteness* of the OMED.

C. Related Work

a) Existing OMEDs: We point the reader to Section III-B for a detailed discussion on some of the practical OMEDs proposed in the literature.

b) Secure inference for MLaaS: A somewhat related approach to improving the privacy of Machine Learning as a Service (MLaaS) termed "secure inference" has been proposed. This approach borrows from ideas in the field of Secure Function Evaluation (where parties can securely compute a function without revealing their inputs), and makes use of garbled circuits [14] or fully homomorphic encryption [15]. However, the principle guarantee of the "secure inference" approach only provides hiding of information about the model beyond what can be deduced from the query and the model's output. Hence, a secure inference approach to security against model extraction would implicitly assume (incorrectly) that total leakage from the predictions is little, and that recovering the model from its predictions would be infeasible or impossible. Therefore, the "secure inference" approach does not properly prevent model extraction when considering clients who repeatedly interact with the service.

c) More natural Covert Learning attacks: The works of [16] and [17] introduce a method for sampling pairs of matrices (A, T) with entries in $\mathbb{Z}_q$, such that A is statistically close to a uniformly random matrix, while M_2 is a low-norm, full-rank trapdoor matrix such that $A \cdot T$ is the all zero matrix. In [8], Vaikuntanathan notes that this sampling algorithm gives an easy, yet somewhat contrived model extraction attack. In fact, it is a natural Covert Learning attack as well. More specifically, for any ML model that is essentially a linear function over $\mathbb{Z}_q$ with added Gaussian noise $e \in \mathbb{Z}_q^m$,[6] the linear function (denoted $s \in \mathbb{Z}_q^n$) can be extracted by querying $sA + e$, and then taking $(sA + e)T = eT$, which can then be used to extract e via Gaussian elimination (T is full rank). Then, given e, s is easily recoverable. However, A is *statistically* close to uniformly random, so the queries are impossible to distinguish from uniformly random queries with any significant advantage. This statistical Covert Learning attack could rule out even *unbounded* OMEDs, however it only works for the very narrow class of noisy linear models, which do not appear frequently in practice. The work of [9] discusses how to similarly sample trapdoors for the low-noise LPN problem [18].[7] The techniques gives rise to another natural Covert Learning attack for an LPN variant of the above setting, however the queries are only *computationally* close to uniform. Elaborating on this techinque in the present paper would be quite time-consuming, hence we refer the reader to [9].

d) Related Formalisms: We note that the formalisms in this work are inspired by the field of Interactive Proofs [19]. Also, the work of [20], who work on protocols for verifying forecasting algorithms, inspired the drive to prove computational incompleteness theorems in this setting.

II. Technical Preliminaries

a) Basic Notation and Terminology: Let $\mathcal{X}$ be a set, and define the ensemble $\{\mathcal{X}_n\}_{n\in\mathbb{N}}$ where $\mathcal{X}_n$ is the n-wise direct product of $\mathcal{X}$. Let $\Delta(\mathcal{X}_n)$ be the convex polytope of all distributions over $\mathcal{X}_n$. We denote by $\mathcal{P}_n$ a property of distributions, where $\mathcal{P}_n \subseteq \Delta(\mathcal{X}_n)$. For an input space $\mathcal{X}_n$ and output space $\mathcal{Y}_n$, let $\mathcal{F}_n \subseteq \{f : \mathcal{X}_n \to \mathcal{Y}_n\}$ be a class of functions with domain $\mathcal{X}_n$ and codomain $\mathcal{Y}_n$, and the corresponding ensemble $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$.

We will use the following standard of computational indistinguishability.

Definition II.1. *(Computational Indistinguishability) Let $\{X_n\}, \{Y_n\}$ be sequences of distributions with X_n, Y_n ranging over $\{0,1\}^{m(n)}$ for some $m(n) = n^{O(1)}$. $\{X_n\}$ and $\{Y_n\}$ are computationally indistinguishable if for every polynomial time algorithm A and sufficiently large n,*

$$|\Pr[A(1^n, X_n) = 1] - \Pr[A(1^n, Y_n) = 1]| \leq \mathrm{negl}(n)$$

Often, n is clear from the context, so the subscript is omitted.

We define the following learning models that are considered in this work. Define the oracle $\mathrm{Ex}(c, D)$ as one which samples independently $x \sim D$ (for a distribution D) and returns $(x, f(x))$.

Definition II.2. *(PAC Learning) We say that a concept class $\mathbb{C}$ is PAC-learnable with respect to a distribution class $\mathcal{D}$ if there exists an algorithm $\mathcal{A}$ such that for any distribution $D \in \mathcal{D}$, concept $c \in \mathbb{C}$, $\mathcal{A}$ and given as input $n \in \mathbb{N}, \epsilon, \delta > 0$, outputs a function h such that*

$$\Pr_{\mathcal{A}}\left[\Pr_{x\sim D}\left[c(x) \neq h(x)\right] < \epsilon\right] \geq 1 - \delta$$

We say that $\mathbb{C}$ is efficiently PAC-learnable with respect to $\mathcal{D}$ if $\mathcal{A}$ runs in time polynomial in n, ϵ, δ.

The following definition of heuristic PAC learning due to Nanashima [11] can be seen as a variant of many existing average-case learning models, where the distribution over concepts is fixed to be uniform. In his original work, Nanashima defines a distribution over *representation* strings over concepts, but in this work it suffices to consider a distribution over actual concepts.

Definition II.3. *(Heuristic PAC-learning — adapted from [11]) Let $\mathbb{C}$ be a concept class, and let $\mathcal{U}$ be the uniform distribution over $\mathbb{C}$. We say that the concept class $\mathbb{C}$ is heuristically PAC-learnable with respect to the distribution class $\mathcal{D}$ if there exists an algorithm $\mathcal{A}$ that for any $D \in \mathcal{D}$, and given as input*

[6] This setting is a bit contrived, since the typical ML models would rarely resemble such a noisy inner product mod q.

[7] These techniques closely resembled that of the seminal work of Alekhnovich [18].

$n \in \mathbb{N}, \epsilon, \delta, \eta > 0$, and access to $\mathrm{E}\textsc{x}(c, D)$ for some $c \sim \mathcal{U}$, outputs a function h such that

$$\Pr_{c\sim\mathcal{U}}\left[\Pr_{\mathcal{A}}\left[\Pr_{z\sim\mathcal{D}}\Big[c(x) \neq h(x)\Big] \leq \epsilon\right] \geq 1-\delta\right] \geq 1-\eta$$

We say that $\mathbb{C}$ is efficiently (η', ϵ')-heuristically PAC-learnable with respect to $\mathcal{D}$ if $\mathcal{A}$ runs in time polynomial in $n, \eta, \epsilon, \delta$, and η, ϵ are fixed to η', ϵ'.

III. The Abstract Observational Model Extraction Defense

In this section, we formally introduce the OMED as a unifying abstraction for the current state of the art MEDs. Before we introduce the OMED formally, let us describe the model extraction setting in detail.

The setting begins by an ML model $f : \mathcal{X} \to \mathcal{Y}$ being adversarially chosen (potentially from some restricted class of functions). Then, we consider the case that a probabilistic polynomial time algorithm $\mathcal{C}$ (the client), can interact via an oracle to the ML model f. We denote this oracle by $\mathcal{O}_f$. The goal of the benign client is to obtain some predictions $f(x)$ for queries $x \in \mathcal{X}$. On the other hand, the goal of the adversarial client is to output an approximate of f, $\hat{f}$ that approximately minimizes a loss function.

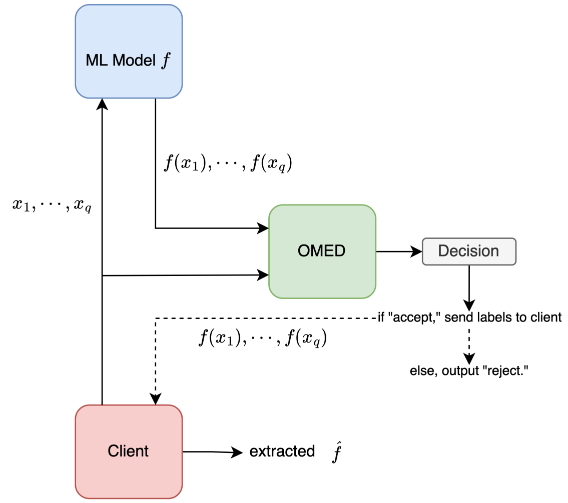

Fig. 1. A depiction of the extraction setting. The adverse client queries the model f, attempting to extract an approximation $\hat{f}$. The OMED watches over the interaction and outputs a decision to accept (and forward the labels) or reject the client based on whether or not it is deemed adverse or benign.

However, the adverse client must be able to perform the extraction in the presence of the OMED. In particular, the OMED is able to view all the queries made by $\mathcal{C}$ (see Figure 1), and the labels that would be returned. The OMED then outputs a decision "accept" or "reject," which essentially mean that the client is benign or adverse, respectively. Again, the point is that the client only receives labels on its queries when the OMED outputs "accept."

With this in mind, let us now formally define the OMED. As noted in the introduction and in [21], defense mechanisms for model extraction have mostly split into two tribes: reducing the information gained per client query, and differentiating malicious extraction adversaries from benign users. The OMED mechanism abstracts the latter approach; the implementation is left unspecified. Hence, we define an OMED as follows.

Definition III.1. *(OMED) A p.p.t. algorithm $\mathcal{M}$ is a $(T(n), q(n))$-OMED for a class of ML models $\mathcal{F}$ if for every $n \in \mathbb{N}, f \in \mathcal{F}_n$, $\mathcal{M}$ runs in time $T(n)$ and takes as input a list of $q(n)$ examples $S = [(x_1, f(x_1)), \cdots (x_{q(n)} f(x_{q(n)}))] \subseteq (\mathcal{X}_n \times \mathcal{Y}_n)^{q(n)}$ and outputs $\sigma \in \{\text{accept}, \text{reject}\}$.*

The definition of the OMED is defined as generally as possible from the perspective of the defense, but makes one important restriction on the client: the examples are requested in large batches, rather than as an adaptive sequence. This nonadaptive setting (with respect to the query selection) can be viewed as unnecessarily restrictive on the client. However, since we prove negative results on the possibility of MEDs via OMEDs, the restriction on the client actually *strengthens* our results. Furthermore, the defense could be expanded to a multi-client setting, where all clients must submit their batches simultaneously. This is again a restriction on the power of the client(s), and thus strengthens our negative results.

A. How to Obtain Provable Security

The above definition of an OMED makes no claims about desirable properties given by the OMED. Thus, what properties should we expect from the OMED? As mentioned in Section I-B4, the goal of the OMED is not just to classify the behavior of the clients, but to actually *confine* the clients to certain predefined benign behaviors. However, it is not enough to simply define security as the event that the client behavior is benign, because this actually needs to be detected and then enforced.

Hence, it should be that a good OMED guarantees that (with high probability) any benign client is accepted, while any adverse client is rejected (and thus prevented from reverse engineering the underlying model). Naturally, the former requirement resembles completeness in an interactive proof system while the latter requirement resembles soundness. Through this lens of Interactive Proofs, we will formalize a notion of completeness and soundness.

First, let us explain how we model a client's behavior in the context of the model extraction setting depicted above. It has been noted in the model extraction literature (e.g. [5]) that defenses should consider how a client's queries relate to each other, rather than how they look individually. This idea is implemented by assuming that a client's queries follow a *distribution*[8]. We adopt a similar idea in this work: we assume that a client's queries follow a distribution P over $\mathcal{X}$. Now, we may nail down how adverse and benign clients behave, and subsequently the OMED. Informally, the idea is that all benign

[8] This model for the basic behavior of a client has appeared previously in the literature (e.g. [5], [6]).

client request examples according to distributions that share some abstract property. We may formalize this as follows:

Definition III.2. *We denote by $\mathcal{P}_n$ a property of distributions, where $\mathcal{P}_n \subseteq \Delta(\mathcal{X}_n \times \mathcal{Y}_n)$.*

Thus intuitively,

Definition III.3. *We say that a set of examples $Q \subseteq \mathcal{X}_n \times \mathcal{Y}_n$ is $\mathcal{P}_n$-benign if $Q \sim P$ for some $P \in \mathcal{P}_n$.*

and,

Definition III.4. *We say that a set of examples $Q \subseteq \times\mathcal{Y}_n$ is $\mathcal{P}_n$-adverse if $Q \sim P$ for some $P \notin \mathcal{P}_n$.*

Moreover, we will informally refer to a client as benign (adverse) if he always requests $\mathcal{P}_n$-benign ($\mathcal{P}_n$-adverse) sets (and it will be the case that $\mathcal{P}_n$ is clear from the context).

1) Completeness: The definition of completeness is straightforward:

Definition III.5. *(MED completeness) We say that $\mathcal{M}$ is a $(T(n), q(n))$-OMED is δ-complete with respect to the property $\mathcal{P}_n$ if for any client $\mathcal{C}$ who requests examples $S_{\mathcal{C}}$ which are $\mathcal{P}_n$-benign, it holds that*

$$\Pr_{\mathcal{M}, S_{\mathcal{C}}}\Big[\mathcal{M}(S_{\mathcal{C}}) = \text{accept}\Big] \geq 1 - \delta$$

2) Provable Security for Benign Clients: The definition of completeness implicitly assumes that, in the case that the client is classified as benign, they are essentially free to interact with the model. Thus, the underlying assumption is that by virtue of queries being benign, the ML model is not considered at risk for being extracted by the server. Therefore, the choice of the benign property is of utmost importance. As discussed in the introduction (see Section I-B4), this leaves room for a theory of provable security. For example, under our framework, a solid choice for a property would be the $\mathcal{P}_n = \mathcal{U}_n$ (i.e., the uniform distribution over examples). This is because many interesting classes of models are thought to be hard to learn from uniformly random examples, even for most models in the class (that is, in the heuristic PAC-learning case, rather than just in the worst-case).

Indeed, a reduction from extracting the ML model (in the average-case over the uniform distribution) using $\mathcal{U}_n$-benign queries to heuristic PAC-learning with respect to the uniform distribution nearly writes itself. To formalize this intuition, we prove the following lemma, which essentially says that if PAC learning is impossible for a large fraction of the class, then, given that the OMED accepted a client that used $\mathcal{U}_n$-benign queries, the probability that most models could be extracted (with arbitrarily high fidelity) is negligible. We will later use the lemma to show how to obtain a formal notion of security against model extraction by all clients, whether adverse or benign.

We first need to establish what constitutes a successful model extraction. Similarly to [4], we define the following extraction experiment. Let $\mathcal{F}_n \subseteq \{f : \mathcal{X}_n \to \mathcal{Y}_n\}$ and $\mathcal{A}$ be a probabilistic polynomial time adversary. We fix a loss function $\mathcal{L}_{D,f} : \mathcal{F}_n \to [0, 1]$ parametrized by an ML model $f \in \mathcal{F}_n$, and a distribution D_n over $\mathcal{X}_n$.

Definition III.6. *(Extraction experiment) Let $\mathsf{Exp}_{\mathcal{C},D_n,f,q(n),\epsilon}$ be defined as the output of the following process.*

1) *$\mathcal{C}$ interacts with $\mathcal{O}_f$ by requesting the labels of a set S of $q(n)$ queries.*
2) *Using $[x, \mathcal{O}_f(x)]_{x \in S}$, $\mathcal{C}$ outputs a candidate model $\hat{f}$.*
3) *If $\mathcal{L}_{D,f}(\hat{f}) < \epsilon$, output* "extracted," *else output* "unextracted."

Lemma III.7. *(Provable security against clients behaving benignly) Let $\mathcal{M}$ be any p.p.t. OMED satisfying δ-completeness for any $\delta \geq 1 - 1/\text{poly}(n)$, with respect to a property of distributions $\mathcal{P}_n$. Then, if a class of ML models $\mathcal{F}_n$ is not efficiently (η, ϵ)-heuristic PAC-learnable with respect to any distribution $D \in \mathcal{P}_n$, then there exists $F \subseteq \mathcal{F}_n$ of size at least $\eta \cdot |\mathcal{F}_n|$, such that for any p.p.t. client $\mathcal{C}$ which requests a set of $q(n)$ examples $S_{\mathcal{C}}$ that is $\mathcal{P}_n$-benign, and for all $f \in S, D_n \in \mathcal{P}_n$,*

$$\Pr_{\mathcal{M}, S_{\mathcal{C}}}\Big[\mathsf{Exp}_{\mathcal{C},D_n,f,q(n),\epsilon} = \text{extracted} \;\Big|\; \mathcal{M}(S_{\mathcal{C}}) = \text{accept}\Big] \leq \text{negl}(n)$$

Proof. We show the contrapositive. Let E, A be the events that $\mathsf{Exp}_{\mathcal{C},D_n,f,q(n),\epsilon}$ = extracted and $\mathcal{M}(S_{\mathcal{C}})$ = accept, respectively. Thus, suppose that we have a p.p.t. $\mathcal{C}$ such that for some $D_n \in \mathcal{P}_n$, $\epsilon > 0$, and a δ-complete $\mathcal{M}$,

$$\Pr_{f \sim \mathcal{F}_n}\left[\Pr_{\mathcal{M}, S_{\mathcal{C}}}\big[E \mid A\big] \geq \frac{1}{\text{poly}(n)}\right] \geq 1 - \eta$$

This implies that

$$\Pr_{f \sim \mathcal{F}_n}\left[\frac{\Pr_{\mathcal{M}, S_{\mathcal{C}}}\big[E \;\wedge\; A\big]}{\Pr_{\mathcal{M}, S_{\mathcal{C}}}\big[A\big]} \geq \frac{1}{\text{poly}(n)}\right] \geq 1 - \eta$$

which is equivalent to

$$\Pr_{f \sim \mathcal{F}_n}\left[\Pr_{\mathcal{M}, S_{\mathcal{C}}}\big[E \;\wedge\; A\big] \geq \frac{1 - \delta}{\text{poly}(n)}\right] \geq 1 - \eta$$

since $\mathcal{M}$ is δ-complete. We now may observe that latest equation is the guarantee that $\mathcal{C}$ (η, ϵ)-heuristically PAC learns $\mathcal{F}_n$ in time $\text{poly}(n)$ with accuracy ϵ and confidence $(1 - \delta)/\text{poly}(n))$, for some $D_n \in \mathcal{P}_n$.

Then, it follows that $\mathcal{F}$ is (η, ϵ)-heuristically PAC-learnable with respect to D_n by running $\mathcal{C}$ $\text{poly}(n)$ times to produce many hypotheses, testing each by random sampling, and outputting the most accurate hypothesis. This works as long as $\delta \geq 1 - 1/\text{poly}(n)$.

□

3) Soundness: The notion of provable security from Lemma III.7 only deals with clients that request sets of examples that are $\mathcal{P}_n$-benign. Thus, we need to provide some guarantees when this is not the case. To this end, we also formalize soundness using the above extraction experiment.

Definition III.8. *(MED soundness) We say that $\mathcal{M}$ is a $(T(n), q(n))$-OMED for $\mathcal{F}_n$ is δ-sound with respect to the property $\mathcal{P}_n$ if for any client $\mathcal{C}^*$ who requests examples $S_{\mathcal{C}^*}$ which are $\mathcal{P}_n$-adverse, and for any $f \in \mathcal{F}_n, D_n \in \mathcal{P}_n$, $\epsilon > 0$, it holds that*

$$\Pr_{\mathcal{M}, S_{\mathcal{C}^*}}\left[\mathcal{M}(S_{\mathcal{C}^*}) = \text{accept} \;\middle|\; \mathsf{Exp}_{\mathcal{C}^*, D_n, f, q(n), \epsilon} = \text{extracted}\right] < \delta$$

The definition mirrors soundness from an Interactive Proof, where for any adversary that does not use benign queries, and given that the adversary would have extracted the model, the probability that the OMED $\mathcal{M}$ errs by accepting the adversary is low. In this cryptographic setting, the OMED would want to set δ to a negligible function of n.

4) Cryptographically-Hard Model Extraction Against All Clients: To tie it all together, we argue that combining completeness, soundness, and hardness assumptions for heuristic PAC-learning light the way to providing full security against model extraction by an OMED. Our notion of security against model extraction (defined below) constitutes bounding the probability that a client wins the extraction game by a negligible function, in this case of n (the size of the learning problem).

Definition III.9. *(Security against Model Extraction) We say that an OMED $\mathcal{M}$ for $\mathcal{F}$ is (η, ϵ)-secure against model extraction if for sufficiently large n, there exists $S \subseteq \mathcal{F}_n$ of size at least $\eta \cdot |\mathcal{F}_n|$, such that for any p.p.t. client $\mathcal{C}$ which requests a set of $q(n)$ examples $\mathcal{S}_{\mathcal{C}}$ and for all $f \in S, D_n \in \mathcal{P}_n$,*

$$\Pr_{\mathcal{M}, S_{\mathcal{C}}}\left[\mathsf{Exp}_{\mathcal{C}, D_n, f, q(n), \epsilon} = \text{extracted}\right] \leq \text{negl}(n)$$

Thus, we prove that, essentially, for any complete, sound, OMED $\mathcal{M}$ for a class of ML models $\mathcal{F}_n$, if $\mathcal{F}_n$ has no η-heuristic PAC-learning algorithm, then at least η fraction of models in $\mathcal{F}_n$ can not be extracted by any client, no matter how it behaves, except with negligible probability.

Theorem III.10. *(Provable Security from Complete and Sound OMEDs) Let $\mathcal{M}$ be any p.p.t. OMED satisfying δ-completeness and γ-soundness for any $\delta \geq 1-1/\text{poly}(n), \gamma < 1/\text{poly}(n)$, with respect to a property of distributions $\mathcal{P}_n$. Then, if a class of ML models $\mathcal{F}_n$ has no p.p.t. (η, ϵ)-heuristic PAC-learning algorithm with respect to any distribution $D_n \in \mathcal{P}_n$, then $\mathcal{M}$ is (η, ϵ)-secure against model extraction.*

Proof. By definition, the examples requested by $\mathcal{C}$ are either $\mathcal{P}_n$-benign or $\mathcal{P}_n$-adverse. In the former case, Lemma III.7 implies that

$$\Pr_{\mathcal{M}, S_{\mathcal{C}}}\left[\mathsf{Exp}_{\mathcal{C}, D_n, f, q(n), \epsilon} = \text{extracted}\right] \leq \text{negl}(n)$$

In the latter case, we have the guarantee from Definition III.8 that

$$\Pr_{\mathcal{M}, S_{\mathcal{C}^*}}\left[\mathcal{M}(S_{\mathcal{C}^*}) = \text{accept}\right] < \gamma$$

where γ is a quantity bounded above by a negligible function of n. Therefore the statement follows. □

We remark that the level of security is essentially determined by the strength of the hardness assumption on heuristic PAC-learning. To be specific, an assumption of hardness of (η, ϵ)-heuristic PAC-learning maps to an (η, ϵ)-secure OMED. This dynamic mirrors, but formalizes, the efforts of practical OMEDs in the literature, which implicitly make an underlying assumption that certain query distributions make extraction hard. In other words, we phrase the implicit assumptions as heuristic PAC-learning hardness assumptions.

B. Defense Proposals As Special Cases of the OMED

Our proposed OMED technique for provable security against model extraction is purposely defined as generally as possible. However, we view it beneficial to discuss the relation to some concrete MEDs which have been proposed.

In this section, we will review three MEDs, [2], [5] and [6], demonstrating that each are special cases of an OMED (with unproved completeness and soundness guarantees). We start each example with a direct quote from the original paper so as to directly demonstrate the relevance to our OMED framework.

1) Extraction Monitors: Information Gain and Feature Space Coverage: The work of [2] proposes two different strategies for detecting model extraction attacks. Both strategies "[quantify] the extraction status of models by continually observing the API query and response streams of users" and provide a warning when a certain extraction status is reached. This is indeed the paradigm outlined by the OMED.

The first proposal of [2] seeks to continuously train a "proxy model" for each client, where the client queries are used to train the model. The function of the proxy model is to estimate the information/knowledge gained by a client with respect to a validation set which is given by the server (and when this information reaches some threshold the client is flagged). The distribution of this validation set mimics the training set of the underlying model. It is noted that it may require significant computational resources to train and update the proxy model (for every user and each incoming query), and thus [2] propose to use a lightweight decision tree proxy model.

In the second proposal, the observational keeps a short description of client queries, and estimates the client's learning rate (of the extraction attack) by analyzing the feature space covered by these queries (as they relate to to the class boundaries of the underlying model. It is noted that a drawback of this proposal is that the class boundaries of certain complex models (e.g. neural networks) are not easily found. Thus, it is proposed that the owner of the underlying model uploads a "surrogate" decision tree which has high fidelity with respect to the complex model (class boundaries of decision trees are easily interpreted by their leaf nodes).

2) PRADA: The MED known as PRADA [5] "analyzes the distribution of consecutive API queries and raises an alarm when this distribution deviates from benign behavior" [5]. Immediately, it is clear that the PRADA method is a candidate for being identified as a special case of the OMED. The defense works under the observation that queries requested by an adversarial client are likely to have a distribution that differs from the characteristic distribution of queries from a benign client. In PRADA, this benign characteristic was chosen to be the property that the distribution over hamming distances between each query in the requested batch should be normally distributed. This choice is backed by observational evidence that certain popular attacks such as the attack of [22] *do not* satisfy this condition. Hence, PRADA tries to satisfy completeness and soundness with respect to the property of all distributions that have a pairwise hamming distance normally distributed (e.g. the uniform distribution over $\{0,1\}^n$).

3) VarDetect: The work of [6] proposes a MED called VarDetect which is designed "to continuously observational the distribution of queries to [the model] from each user" [6]. Specifically, VarDetect trains a Variational Autoencoder (VAE) to map the "problem domain" (PD) dataset distribution (the PD distribution mimics the distribution of data that was used to train the underlying model) and the adversarial "outlier" (O) data distribution (the distribution of attacker queries) to distinct regions in latent space. Benign clients are assumed to query from the PD distribution while adverse clients are assumed to query from an O distribution. VarDetect purports to separate these two by computing the maximum mean discrepancy (MMD) between the latent mapping of the client's queries and that of the PD distribution (the MMD test flags the client if the result is above a certain threshold).

IV. Attacks on Efficient OMEDs from Natural Covert Learning

In this section, we will consider the question:

> Can we efficiently realize the provable security guarantees outlined in the previous section?

Towards a negative answer, we will introduce an attack on the OMED technique for provable security, via a connection to Covert Learning [9]. Our attack will generate a distribution of examples which is *computationally indistinguishable* from a distribution in the property that is accepted by the OMED. In other words, the attacker operates (computationally) indistinguishably from a benign client, in the eyes of the OMED. Still, the labelled queries allow the algorithm to extract a model with high fidelity.

a) Notation: We briefly recall some standard terminology and notation from learning theory which we use throughout the remainder of the work. A concept f is a function over an input domain $\mathcal{X}_n$ and label domain $\mathcal{Y}_n$. A concept class $\mathbb{C} = \{\mathbb{C}_n\}_{n\in\mathbb{N}}$ is a sequence of sets of functions $\mathbb{C}_n = \{f : \mathcal{X}_n \to \mathcal{Y}_n\}$. We call a pair $(x, y) \in \mathcal{X}_n \times \mathcal{Y}_n$ an *example*, where x is the *input* and y is the *label*. In the rest of this work, we use $\mathcal{X}_n = \{0,1\}^n$, and $\mathcal{Y}_n = \{-1, 1\}$. A membership oracle $\mathcal{O}_f$ for a concept f is an oracle with the property that on query $z \in \{0,1\}^n$, $\mathcal{O}_f(z) = f(z)$. Let $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ be a distribution ensemble over $\mathcal{X}_n$. Define the distribution $\text{Ex}(f, D_n, q)$ as the output of sampling independently $x_1, \cdots x_q \sim D_n$ and returning $[(x_1, f(x_1)), \cdots (x_q, f(x_q))]$. We denote by $\mathcal{U}_n$ the uniform distribution over $\{0,1\}^n$. Finally, let $\mathcal{L}_{D,f} : \mathbb{C} \to [0,1]$ be a loss function parameterized by a concept f and distribution D over inputs.

A. What is Natural Covert Learning?

We will focus on a special case of Covert Learning, which we call *natural Covert Learning*. A natural Covert Learning algorithm, essentially, is a membership query learning algorithm that satisfies the normal PAC-learning guarantees with respect to an example distribution D, with the added property that distribution over the membership queries and labels is computationally indistinguishable from examples sampled according to D.

More formally:

Definition IV.1. *(Natural Covert Learning) Let $\mathbb{C}$ be a boolean concept class, let $\mathcal{D}$ be a distribution ensemble, and fix a loss function $\mathcal{L}$. We say that $\mathcal{A}$ is a $q(n)$-natural Covert Learning algorithm for $\mathbb{C}$ with respect to $\mathcal{D}$ and $\mathcal{L}$ if for every $n \in \mathbb{N}, f \in \mathbb{C}_n, \epsilon, \delta > 0$, $\mathcal{A}$ satisfies the following:*

- *Completeness. For the random variable $h = \mathcal{A}^{\mathcal{O}_f}(n, \epsilon, \delta)$ satisfies*
$$\Pr_h\left[\mathcal{L}_{D_n,f}(h) \le \epsilon\right] \ge 1 - \delta$$
- *Privacy. For every p.p.t. adversary* $\texttt{Adv}$*, and $S \sim$* $\text{Ex}(f, D_n, q(n))$*,*
$$\left|\Pr_{\texttt{Adv},S}\left[\texttt{Adv}(S) = 1\right] - \Pr_{\texttt{Adv},\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})}\left[\texttt{Adv}(\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})) = 1\right]\right| \le \text{negl}(n)$$
where $\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})$ denotes the distribution over the queries made by $\mathcal{A}$ and the responses by the oracle.

B. Natural Covert Learning Attack

In this section, our goal is to show that the existence of a natural Covert Learning algorithm for a particular concept class implies inadequacy of a polynomial time OMED for a class of models equal to the concept class. More specifically, we prove that satisfying soundness for an OMED is impossible, for any reasonable completeness parameter. This suffices to rule out obtaining provable security against model extraction by an OMED by instantiating Theorem III.10.

Theorem IV.2. *Suppose that there exists a $q(n)$-natural Covert Learning algorithm $\mathcal{A}$ for a hypothesis class $\mathbb{C}$, with respect* *to $\mathcal{D}$ and $\mathcal{L}$. Then there exists a client $\mathcal{C}$ such that for any δ-complete OMED $\mathcal{M}$ (with respect to $\mathcal{P}_n$) for $\mathbb{C}$, it holds that for any $n \in \mathbb{N}, \epsilon, \delta_{\mathcal{A}} > 0, f \in \mathbb{C}_n$,*

$$\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathcal{M}(S_{\mathcal{C}}) = \text{accept} \;\wedge\; \mathsf{Exp}_{\mathcal{C},D_n,f,q(n),\epsilon} = \text{extracted}\Big] \geq (1-\delta)(1-\delta_{\mathcal{A}}) - \text{negl}(n)$$

where $\text{negl}(n)$ *denotes a negligible function of* n *and* $\delta_{\mathcal{A}}$ *is the failure probability of* $\mathcal{A}$.

Proof. Let $S_{\mathcal{A}}$ denote the set of $q(n)$ examples which are queried by $\mathcal{A}$. Let $S \sim \text{Ex}(f, D_n, q(n))$. Furthermore, define E to be the event that $\mathcal{M}(S_{\mathcal{A}}) = \text{accept}$ and that $\mathsf{Exp}_{\mathcal{A},D_n f,q(n),\epsilon} = \text{extracted}$.

Using the fact that by our Covert Learning assumption we have that for every p.p.t. adversary $\mathtt{Adv}$,

$$\left|\Pr_{\mathtt{Adv},S}\Big[\mathtt{Adv}(S) = 1\Big] - \Pr_{\mathtt{Adv},S_{\mathcal{A}}}\Big[\mathtt{Adv}(S_{\mathcal{A}}) = 1\Big]\right| \leq \text{negl}(n) \tag{1}$$

and $\mathcal{M}, f$ are polynomial time computable, we then get that

$$\Pr_{\mathcal{M},S_{\mathcal{A}},\mathcal{A}}\Big[E\Big] \geq \Pr\Big[\mathcal{M}(S) = \text{accept} \;\wedge \;\mathsf{Exp}_{\mathcal{A},D_n,f,q(n),\epsilon} = \text{extracted}\Big] - \text{negl}(n)$$

Then, we can conclude by independence of events

$$\Pr_{\mathcal{M},S_{\mathcal{A}},\mathcal{A}}\Big[E\Big] = \Pr\Big[\mathcal{M}(S) = \text{accept}\Big] \wedge \;\Pr\Big[\mathsf{Exp}_{\mathcal{A},D_n,f,m(n),\epsilon} = \text{extracted}\Big] - \text{negl}(n) \geq (1-\delta)(1-\delta_{\mathcal{A}}) - \text{negl}(n)$$

The last equation follows by the δ-completeness property of $\mathcal{M}$. Therefore, the statement is proved by allowing $\mathcal{C}$ to be identical to $\mathcal{A}$.

□

From this attack, a generic incompleteness theorem follows, essentially because the attack works against *any* efficient OMED.

Corollary IV.3. *Suppose that there exists a $q(n)$-natural Covert Learning algorithm $\mathcal{A}$ for a hypothesis class $\mathbb{C}$ with respect to $\mathcal{D}$ and $\mathcal{L}$. Then if $\mathcal{M}$ is a* $(\text{poly}(q(n)), q(n))$*-OMED for $\mathbb{C}$, and if $\mathcal{M}$ is δ-complete with respect to $\mathcal{D}$, then it is not* $(1-\delta-\gamma)$*-sound with respect to $\mathcal{D}$, when* $q(n) = \text{poly}(n)$ *and* $\gamma \geq 1/\text{poly}(n)$.

Proof. By Theorem IV.2 there exists a client $\mathcal{C}$ that uses examples $S_{\mathcal{C}}$ which are D_n-adverse such that for any δ-complete OMED $\mathcal{M}$ (with respect to $\mathcal{D}$) for $\mathbb{C}$, it holds that for any $f \in \mathbb{C}, \epsilon > 0$,

$$\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathcal{M}(S_{\mathcal{C}}) = \text{accept} \;\wedge\; \mathsf{Exp}_{\mathcal{C},D_n,f,q(n),\epsilon} = \text{extracted}\Big] \geq (1-\delta)(1-\delta_{\mathcal{C}}) - \text{negl}(n)$$

where $\text{negl}(n)$ is a negligible function of n and $\delta_{\mathcal{C}}$ is the failure probability of $\mathcal{C}$. Then, we can deduce that

$$\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathcal{M}(S_{\mathcal{C}}) = \text{accept} \;\Big|\; \mathsf{Exp}_{\mathcal{C},D_n,f,q(n),\epsilon} = \text{extracted}\Big] = \frac{\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[E\Big]}{\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathsf{Exp}_{\mathcal{C},D_n f,q(n),\epsilon} = \text{extracted}\Big]} \geq \frac{(1-\delta)(1-\delta_{\mathcal{C}}) - \text{negl}(n)}{1-\delta_{\mathcal{C}}}$$

where E is defined as the event that $\mathcal{M}(S_{\mathcal{A}}) = \text{accept}$ and that $\mathsf{Exp}_{\mathcal{A},D_n f,q(n),\epsilon} = \text{extracted}$. Thus by taking an appropriately large n, $\mathcal{M}$ cannot be $(1-\delta-\gamma)$-sound for a reasonable choice of $\delta_{\mathcal{C}}$ such as 0.99. □

C. Concrete Attack and Incompleteness Theorem

In this section, we show that under the subexponential hardness assumption on the standard LPN problem, there exists an attack of the type outlined in the previous section. Let us first formally introduce our assumption.

Definition IV.4. ***Search LPN assumption.*** *For* $\mu \in (0, 0.5), n \in \mathbb{N}$, *the* $(m(n), T(n))-\mathsf{SLPN}_{\mu,n}$ *search assumption states that for every inverter* $\mathbb{I}$ *running in time* $T(n)$,

$$\Pr_{s,\boldsymbol{A},e}[\mathbb{I}(\boldsymbol{A}, \boldsymbol{A}s \oplus e) = s] \leq \frac{1}{T(n)}$$

where $s \overset{\$}{\leftarrow} \mathbb{Z}_2^n, \boldsymbol{A} \overset{\$}{\leftarrow} \mathbb{Z}_2^{m(n)\times n}, e \overset{\$}{\leftarrow} \beta_\mu^{m(n)}$.

Thus, the assumption we adopt is the $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ assumption. The following theorem is implicit in [9]. Let $\mathsf{DT}[\text{poly}(n)]$ be the set of all decision trees of size $\text{poly}(n)$.

Theorem IV.5. *(Agnostic Covert Learning of decision trees from [9]) Given query access to a function* $f : \{0,1\}^n \to \{-1,1\}$, *there exists an algorithm $\mathcal{A}$ running in time* $\text{poly}(s, 1/\epsilon, \log(1/\delta))$ *and making* $q(n) = \text{poly}(n, 1/\epsilon, \log(1/\delta))$ *query accesses such that, unless the* $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ *assumption does not hold,*

1) *(Completeness) $\mathcal{A}$ outputs $h : \{0,1\}^n \to \{-1,1\}$ such that*

$$\Pr_{x\sim\mathcal{U}_n}\Big[h(x) \neq f(x)\Big] \leq \min_{g\in\mathsf{DT}[s]} \Pr_{x\sim\{0,1\}^n}\Big[g(x) \neq f(x)\Big] + \epsilon$$

with probability $1-\delta$.

2) *(Privacy) The distribution over examples requested by $\mathcal{A}$ is computationally indistinguishable, but statistically distinguishable, from* $\text{Ex}(f, \mathcal{U}_n, q(n))$.

We may now proceed to combine Theorem IV.2 and Theorem IV.5 to obtain a concrete natural Covert Learning attack on models implemented by decision tree classifiers of polynomial size. Let $\mathbb{U} = \{\mathcal{U}_n\}_{n\in\mathbb{N}}$, and let $\mathcal{L}_{D,f}(h) = \Pr_{x\sim D}[h(x) \neq f(x)]$.

Theorem IV.6. *Under the* $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ *assumption, there exists a client* $\mathcal{C}$ *that requests examples* $S_{\mathcal{C}}$ *that are* $\mathcal{U}_n$*-adverse such that for any* δ*-complete OMED* $\mathcal{M}$ *(with respect to* $\mathbb{U}$*) for* $\mathsf{DT}[\text{poly}(n)]$*, it holds that for any* $f \in \mathsf{DT}[\text{poly}(n)]$*,*

$$\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathcal{M}(S_{\mathcal{C}}) = \text{accept} \;\wedge\; \mathsf{Exp}_{\mathcal{C},\mathcal{U}_n f,q(n),\epsilon} = \text{extracted}\Big] \geq (1-\delta)(1-\delta_{\mathcal{C}}) - \text{negl}(n)$$

where $\text{negl}(n)$ *is a negligible function of* n *and* $\delta_{\mathcal{C}}$ *is the failure probability of* $\mathcal{C}$*.*

Proof. Observe that the algorithm described in Theorem IV.5 constitutes a natural Covert Learning algorithm for $\mathbb{C}$ with respect to $\mathbb{U}$, and loss function $\mathcal{L}_{D,f}(h) = \Pr_x[h(x) \neq f(x)]$. Thus, the statement follows directly from Theorem IV.5 and Theorem IV.2. □

1) Incompleteness Theorem: The previous theorem can be interpreted as the existence of a universal attack against any OMED for decision tree classifiers with δ-completeness for the uniform property (up to SLPN assumptions). Hence, the main result now follows.

Corollary IV.7. *(Main result.) Under the* $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ *assumption, if* $\mathcal{M}$ *is a* $(\text{poly}(q(n)), q(n))$*-OMED for* $\mathsf{DT}[\text{poly}(n)]$ *and* $\mathbb{U}$*, then if* $\mathcal{M}$ *is* δ*-complete it is not* $(1-\delta-\gamma)$*-sound, when* $q(n) = \text{poly}(n)$*, and* $\gamma \geq 1/\text{poly}(n)$*.*

Proof. By Theorem IV.6 there exists a client $\mathcal{C}$ such that for any δ-complete OMED $\mathcal{M}$ (with respect to $\mathbb{U}$) for $\mathsf{DT}[\text{poly}(n)]$, it holds that for any $f \in \mathsf{DT}[\text{poly}(n)], \epsilon > 0$,

$$\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathcal{M}(S_{\mathcal{C}}) = \text{accept} \;\wedge\; \mathsf{Exp}_{\mathcal{C},\mathcal{U}_n,f,q(n),\epsilon} = \text{extracted}\Big] \geq (1-\delta)(1-\delta_{\mathcal{C}}) - \text{negl}(n)$$

where $\text{negl}(n)$ is a negligible function of n and $\delta_{\mathcal{C}}$ is the failure probability of $\mathcal{C}$. Then, we can deduce that

$$\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathcal{M}(S_{\mathcal{C}}) = \text{accept} \;\Big|\; \mathsf{Exp}_{\mathcal{C},\mathcal{U}_n,f,q(n),\epsilon} = \text{extracted}\Big] = \frac{\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[E\Big]}{\Pr_{\mathcal{M},S_{\mathcal{C}}}\Big[\mathsf{Exp}_{\mathcal{C},\mathcal{U}_n f,q(n),\epsilon} = \text{extracted}\Big]} \geq \frac{(1-\delta)(1-\delta_{\mathcal{C}}) - \text{negl}(n)}{1-\delta_{\mathcal{C}}}$$

Thus by taking an appropriately large n, $\mathcal{M}$ cannot be $(1-\delta-\gamma)$-sound for a reasonable choice of $\delta_{\mathcal{C}}$ such as 19/20. □

As an example, our result shows that for any reasonable choice of δ for uniform completeness (e.g. 0.1, meaning that at most 0.1 honest clients are incorrectly rejected), then it is impossible to obtain any even remotely useful soundness guarantee. Using $\delta = 0.1$, our result shows that not even 0.11 fraction of extraction attempts will be detected by the OMED.

More generally, let us recall the provable security theorem, restated below.

Theorem IV.8. *(Restated Theorem III.10) Let* $\mathcal{M}$ *be any p.p.t. OMED satisfying* δ*-completeness and* γ*-soundness for any* $\delta \geq 1 - 1/\text{poly}(n), \gamma < 1/\text{poly}(n)$*, with respect to a property of distributions* $\mathcal{P}_n$*. Then, if a class of ML models* $\mathcal{F}_n$ *has no p.p.t.* (η, ϵ)*-heuristic PAC-learning algorithm with respect to any distribution* $D_n \in \mathcal{P}_n$*, then* $\mathcal{M}$ *is* (η, ϵ)*-secure against model extraction.*

The theorem, to be instantiated, requires that the completeness parameter and soundness parameter are noticeably apart.

Theorem IV.9. *Under the* $(2^{\omega(n^{\frac{1}{2}})}, 2^{\omega(n^{\frac{1}{2}})})-\mathsf{SLPN}_{\mu,n}$ *assumption, provable security against model extraction via efficient OMED and Theorem III.10 cannot be achieved.*

Proof. The statement follows immediately from Corollary IV.7 and inspection of Theorem III.10. □

V. Provable Security from the Impossibility of Covert Learning

In this section, we show the inverse of Corollary IV.3. That is, if natural Covert Learning algorithms do not exist, then we can obtain good OMEDs needed for security via Theorem III.10.

Theorem V.1. *If there does not exist a* $q(n)$*-natural Covert Learning algorithm for a concept class* $\mathbb{C}_n$ *with respect to* D_n*, then there exists* $\mathcal{M}$ *which is a* $(\text{poly}(q(n)), q(n) \cdot O(\log(\eta^{-1})))$*-OMED for a concept class* $\mathbb{C}_n$ *that is* δ*-complete and* η*-sound with respect to the property* $\{D_n\}$ *for* $\eta \leq \text{negl}(n)$*.*

Proof. Since there does not exist a $q(n)$-natural Covert Learning algorithm for the concept class $\mathbb{C}$ with respect to D_n, then it means that for every learning algorithm $\mathcal{A}$ such that for every $f \in \mathbb{C}, n \in \mathbb{N}, \epsilon, \gamma > 0$ we have that with probability $1-\gamma$, $\mathsf{Exp}_{\mathcal{A},D_n,f,q(n),\epsilon} = \text{extracted}$, then distribution over queries made by $\mathcal{A}$ is efficiently distinguishable from D_n. In other words, there exists p.p.t. `Adv` such that for $S \sim \text{Ex}(f, D_n, q(n))$,

$$\left|\Pr_{\texttt{Adv},S}\Big[\texttt{Adv}(S) = 1\Big] - \Pr_{\texttt{Adv},\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})}\Big[\texttt{Adv}(\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})) = 1\Big]\right| = \mu \tag{2}$$

$$\geq 1/\text{poly}(n) \tag{3}$$

where $\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})$ denotes the distribution over the queries made by $\mathcal{A}$ and the responses by the oracle. This implies that we can produce the desired OMED $\mathcal{M}$ by using `Adv` as follows:

1) Estimate $\Pr_{\texttt{Adv},S}[\texttt{Adv}(S) = 1]$ within an additive factor of $\mu/4$ with probability at least $1 - \eta/2$.
2) Using segments of submitted query requests $S_{\mathcal{C}}$, estimate $\Pr_{\texttt{Adv},S_{\mathcal{C}}}[\texttt{Adv}(S_{\mathcal{C}}) = 1]$ within an additive factor of $\mu/4$

(here $S_{\mathcal{C}}$ are segments of input queries from a client $\mathcal{C}$), with probability at least $1 - \eta/2$.

3) Output “reject” if the two estimates are not within $\mu/2$, and “accept” otherwise.

We analyze this process to prove the desired statement, starting with completeness and soundness. Without loss of generality, assume that $\Pr_{\texttt{Adv},S}[\texttt{Adv}(S) = 1] \geq 1 - \delta$. Then, it follows that $\mathcal{M}$ is δ-complete with respect to $\mathcal{P}_n$. Now, by (2), it must be the case that

$$\Pr_{\texttt{Adv},\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})}\left[\texttt{Adv}(\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})) = 1\right] \leq 1 - \delta - \mu$$

or

$$\Pr_{\texttt{Adv},\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})}\left[\texttt{Adv}(\mathsf{T}(\mathcal{A}^{\mathcal{O}_f})) = 1\right] \geq 1 - \delta + \mu$$

Hence, the estimation procedure is guaranteed to distinguish between the two cases with probability least $1 - \eta$ (using a union bound). Using this fact and inspecting step 3 gives η-soundness.

To see that $\mathcal{M}$ is an $(\mathrm{poly}(q(n)), \mathrm{poly}(q(n)))$-OMED specifically, observe that $\texttt{Adv}$ is efficient (and therefore runs in time $\mathrm{poly}(q(n))$, and consider that the estimation procedure requires $q(n) \cdot O(\log(\eta^{-1}))$ examples which are given by the client to estimate within the desired factor and with probability $1 - \eta$ as needed. □

We remark that in the above proof, it is important that the OMED has access to $q(n) \cdot O(\log(\eta^{-1}))$ examples from the client. Ideally, we can reduce this number, as clients may be able to perform model extraction using less queries. Nonetheless, it allows us to obtain (along with Corollary IV.3) the following “dichotomy” between the possibility of OMEDs and Covert Learning.

Corollary V.2. *(OMED/Covert Learning Dichotomy) The following statements are equivalent.*

1) *There exists an* $(\mathrm{poly}(q(n)), q(n) \cdot O(\log(\eta^{-1})))$*-OMED* $\mathcal{M}$ *for a concept class* $\mathbb{C}_n$ *which is* δ*-complete and* $\mathrm{negl}(n)$*-sound with respect to* D_n*.*
2) *There does not exist a* $q(n)$*-natural Covert Learning algorithm for a concept class* $\mathbb{C}_n$ *with respect to* D_n*.*

Proof. Corollary IV.3 gives $1 \to 2$, while Theorem V.1 gives $2 \to 1$. □

Acknowledgements

I would like to thank Ran Canetti for several illuminating conversations about this work.

References

[1] N. Carlini, M. Jagielski, and I. Mironov, “Cryptanalytic extraction of neural network models,” *arXiv preprint arXiv:2003.04884*, 2020.

[2] M. Kesarwani, B. Mukhoty, V. Arya, and S. Mehta, “Model extraction warning in mlaas paradigm,” in *Proceedings of the 34th Annual Computer Security Applications Conference*, pp. 371–380, 2018.

[3] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, “Stealing machine learning models via prediction apis,” in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pp. 601–618, 2016.

[4] V. Chandrasekaran, K. Chaudhuri, I. Giacomelli, S. Jha, and S. Yan, “Exploring connections between active learning and model extraction,” in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pp. 1309–1326, 2020.

[5] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, “Prada: protecting against dnn model stealing attacks,” in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 512–527, IEEE, 2019.

[6] S. Pal, Y. Gupta, A. Kanade, and S. Shevade, “Stateful detection of model extraction attacks,” *arXiv preprint arXiv:2107.05166*, 2021.

[7] A. Dziedzic, M. A. Kaleem, Y. S. Lu, and N. Papernot, “Increasing the cost of model extraction with calibrated proof of work,” *arXiv preprint arXiv:2201.09243*, 2022.

[8] V. Vaikuntanathan, “Secure computation and ppml: Progress and challenges.” https://www.youtube.com/watch?v=y2iYEHLY2xEab$_channel = TheIACR$, 2021.

[9] R. Canetti and A. Karchmer, “Covert learning: How to learn with an untrusted intermediary.” Cryptology ePrint Archive, Report 2021/764, 2021.

[10] A. Blum, M. Furst, M. Kearns, and R. J. Lipton, “Cryptographic primitives based on hard learning problems,” in *Annual International Cryptology Conference*, pp. 278–291, Springer, 1993.

[11] M. Nanashima, “A theory of heuristic learnability,” in *Conference on Learning Theory*, pp. 3483–3525, PMLR, 2021.

[12] O. Goldreich and L. A. Levin, “A hard-core predicate for all one-way functions,” in *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pp. 25–32, 1989.

[13] E. Kushilevitz and Y. Mansour, “Learning decision trees using the fourier spectrum,” *SIAM Journal on Computing*, vol. 22, no. 6, pp. 1331–1348, 1993.

[14] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar, “Chameleon: A hybrid secure computation framework for machine learning applications,” in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pp. 707–721, 2018.

[15] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, “Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy,” in *International Conference on Machine Learning*, pp. 201–210, PMLR, 2016.

[16] M. Ajtai, “Generating hard instances of lattice problems,” in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 99–108, 1996.

[17] C. Gentry, C. Peikert, and V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions,” in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 197–206, 2008.

[18] M. Alekhnovich, “More on average case vs approximation complexity,” in *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pp. 298–307, IEEE, 2003.

[19] S. Goldwasser, S. Micali, and C. Rackoff, “The knowledge complexity of interactive proof systems,” *SIAM Journal on computing*, vol. 18, no. 1, pp. 186–208, 1989.

[20] K.-M. Chung, E. Lui, and R. Pass, “Can theories be tested? a cryptographic treatment of forecast testing,” in *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pp. 47–56, 2013.

[21] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, “High accuracy and high fidelity extraction of neural networks,” in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pp. 1345–1362, 2020.

[22] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.