Fingerprinting Classifiers With Benign Inputs

Thibault Maho, Teddy Furon, *Senior Member, IEEE*, and Erwan Le Merrer

***Abstract*— Recent advances in the fingerprinting of deep neural networks are able to detect specific instances of models, placed in a black-box interaction scheme. Inputs used by the fingerprinting protocols are specifically crafted for each precise model to be checked for. While efficient in such a scenario, this nevertheless results in a lack of guarantee after a mere modification of a model (*e.g.*finetuning, quantization of the parameters). This article generalizes fingerprinting to the notion of model families and their variants and extends the task-encompassing scenarios where one wants to fingerprint not only a precise model (previously referred to as a *detection* task) but also to identify which model or family is in the black-box (*identification* task). The main contribution is the proposal of fingerprinting schemes that are resilient to significant modifications of the models. We achieve these goals by demonstrating that benign inputs, that are unmodified images, are sufficient material for both tasks. We leverage an information-theoretic scheme for the identification task. We devise a greedy discrimination algorithm for the detection task. Both approaches are experimentally validated over an unprecedented set of more than 1,000 networks.**

***Index Terms*— Fingerprinting, deep neural networks, information theory.**

I. Introduction

FINGERPRINTING classifiers aims at deriving a signature uniquely identifying a machine learning model, like the human fingerprint's minutiae in biometry. This is essentially a black-box problem: the classifier to be identified is in a black-box in the sense that one can just make some queries and observe the resulting model outputs. For instance, this is the case when the model is embedded in a chip, or accessible via an API.

The main application that related works [1], [2], [3], [4], [5] target is the proof of ownership. An accurate deep neural network is a valuable industrial asset due to the know-how for training it, the difficulty of gathering a well-annotated training dataset, and the required computational resources to learn its parameters. In this context, the entity identifying a black-box wants to *detect* whether it is not a stolen model of her.

Another at least as critical application is information gain. For instance, an attacker willing to delude the classifier first gains some knowledge about the remote model, or a company wants to determine which model is in use in a competitor's production system. This application has been left aside as of today, and we tackle it under the notion of the fingerprinting *identification* task.

For clarity, we name Alice the entity willing to identify the model that Bob has embedded in the black-box.

A. Challenges

We hereafter name a *model* a reference architecture, together with its set of hyperparameters tuned by its designers. When any of these components are modified, we coin the resulting model a *variant*. The biggest difficulty is that there exist plenty of ways to modify a model while maintaining its intrinsic good accuracy. These procedures simplify a network (quantization of the weights and/or activations, pruning, see *e.g.* [6]), or make it more robust (preprocessing of the input, adversarial re-training [7]). These mechanisms were not a priori designed to make fingerprinting harder but they leave room for Bob to tamper with the fingerprint of a model. We assume that Alice also knows some of these procedures. Yet, they are often defined by many parameters and among them scalars so that there is virtually an infinity of variants. Like in biometry, the fingerprint should be discriminative enough to be unique per model but also sufficiently robust to identify a variant.

The approaches in the literature have two common pillars. They use the boundaries in the input space drawn by a classifier as the fingerprint, *i.e.*the unique signature identifying the model [2], [4], [5]. Two neural networks sharing the same architecture, the same training set and procedure are different because the training is stochastic (like the Stochastic Gradient Descent). This causes their boundaries in the input space not to overlap fully. Most of the papers in the literature are looking for discriminative deviations of these boundaries. Second, the key task is detection: Alice makes a guess about the model in the black-box and then she sends specific queries to test whether her hypothesis holds [2], [4], [5], [8].

B. Our Rationale and Contributions

Our work differs from related works on two key aspects: *i)* we do not forge any specific input but use regular benign inputs, and *ii)* we directly identify models using their intrinsic classification behavior.

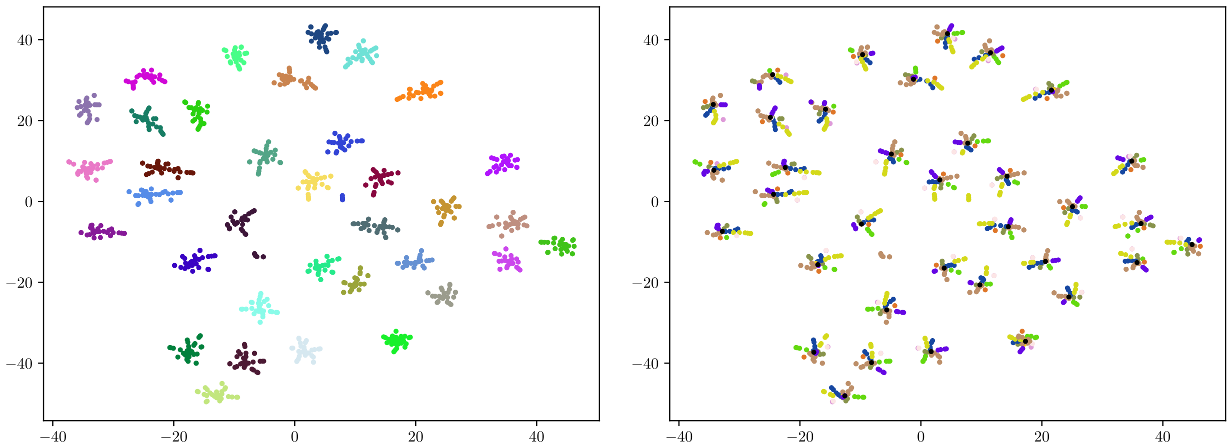
We thoroughly investigate the use of benign inputs for fingerprinting models contrary to the previous works crafting specific inputs. We thus do not need to probe the input space to discover the decision boundaries. Benign inputs constitute a certain advantage, as it removes the need of often

Manuscript received 30 August 2022; revised 20 June 2023; accepted 18 July 2023. Date of publication 2 August 2023; date of current version 8 September 2023. This work was supported by Chaire Security of Artificial Intelligence for Defense Applications (SAIDA) through Agence Nationale de la Recherche (ANR) and French Agence Innovation pour la Défense (AID) agencies. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rafael F. Schaefer. *(Corresponding author: Thibault Maho.)*

The authors are with the CNRS, IRISA, Inria, University of Rennes 1, 35000 Rennes, France (e-mail: thibault.maho@inria.fr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2023.3301268, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2023.3301268



(a) Color by vanilla model

(b) Color by variation

Fig. 1. A t-SNE representation of the pairwise distances of 1081 different models: 10 types of variation applied on 35 off-the-shelves vanilla models for ImageNet with different parameters (listed in App. X).

complex crafting procedures. It is less prone to defenses being implemented on Bob's side (*e.g.*rejection based on the distance to the decision frontier [9]).

The second salient observation is the restriction of previous works on the detection task. The more general possibility to identify a model or a family of models inside the black-box remains unstudied.

In a nutshell, when Bob has picked a model among a set of networks known by Alice, then our solution is essentially deterministic: Alice has to find a sequence of inputs of minimum length to identify the black-box. We apply a greedy algorithm that carefully selects the input to iteratively narrow down the set of suspects, *i.e.*candidate models, until it becomes a singleton. Approximation theory tells that this is suboptimal but we report that in practice many networks are indeed identified within less than three input queries.

When Bob has made a variant of a model, its output may not match the output of any known model by Alice. We then use C.E. Shannon's information theory to measure the statistical similarity between the outputs of two models. This approach is common in the field of Information Forensics and Security, especially in biometry [10], [11], PUF [12], content identification [13], [14], or traitor tracing [15], [16].

As an appetizer, Figure 1 depicts the t-SNE representation from the pairwise distances within a set of 35 vanilla models and their variants. The model families are well clustered in the sense that variants are closer to their original network than any other model. Alice may not identify precisely the variant of the model but at least she can accurately identify its family, *i.e.*infer which was the original vanilla model and even which kind of variation Bob applied.

Our contribution is fourfold.

1) We demonstrate that the mere use of benign images is enough to accomplish high success rates for fingerprinting modern classification models. This is to be opposed to the computationally demanding task of crafting inputs for that same goal.
2) The fingerprinting detection task, introduced by state-of-the-art works, is complemented with the introduction of the identification task. We frame the latter as an information theoretical problem.
3) We present a distance based on the empirical Mutual Information, gauging how close two models are. This distance permits generalizing the notion of modifications (also coined as *attacks*) on models through the concept of model families and variants[1].
4) We perform extensive experimentation by considering more than 1,000 classification models on ImageNet. A head-to-head comparison with the two related works reports significant improvements w.r.t. accuracy in the detection task.

Section II is a threat analysis listing all the working assumptions in our work. The next two sections deal both with detection (Alice verifies her hypothesis about the black-box) and identification (Alice discovers which model is in the black-box) but under two scenarios: Section III builds on the fact that Bob has picked a model among the set known by Alice, whereas Section IV assumes that the black-box may be an unknown model. Both sections contain experimental results. Section VI is devoted to the related works and benchmarking with state-of-the-art detection schemes. A summary of notations is provided in Table I.

II. Threat Model

This section details the goals of Alice and Bob.

A. Bob: Keeping his Model Anonymous

1) Goals: Bob is playing first by secretly selecting a model and putting it in the black-box under scrutiny. This model can be a vanilla model or a variant of a known model. A variant is created by applying on a given vanilla model m the procedure V parametrized by $\theta \in \Theta$ which describes the type of modification and the associated parameters. This can be thought of as an attack by Bob on the vanilla model to harden identification. We denote such a variant by $\mathsf{v} = \mathsf{V}(\mathsf{m}, \theta)$.

The goal of Bob is to offer an accurate black-box classifier while maintaining the 'anonymity' of the model in use. The

[1]Our code is attached to this submission, and will be open-sourced should the article be accepted

TABLE I
NOTATIONS

D	The task of detection
I	The task of identification
m	A vanilla model as listed in Tab. **??**
$\mathsf{V}(\mathsf{m},\theta)$	Variant obtained by applying procedure θ on m
Θ	Set of all variation procedures and parameters
$\mathsf{acc}(\mathsf{m})$	Accuracy of model m
b	The model in the black-box
z	Top-k output of the black-box
$\mathcal{C}$	The set of classes labeled from 1 to C
$\mathcal{Z}_k$	The set of top-k outcomes
$(C)_k$	Falling factorial $(C)_k := C(C-1)\dots(C-k+1)$
$\mathcal{B}$	Set of models Bob can create as defined in (2)
$\mathcal{A}$	Set of models known by Alice
$\mathcal{P}$	Set of public vanilla models listed in Tab. **??**
$\mathcal{F}(\mathsf{m})$	Family spanned by vanilla model m (3)
$\mathcal{F}(\mathsf{m},\Psi)$	Family spanned by m and variation $\Psi\subset\Theta$ (4)
$\mathcal{X}$	A collection of benign inputs
N	Cardinality of $\mathcal{X}$
$q_{1:\ell}$	An ordered list of indices in $[\![N]\!]$
$[\![N]\!]$	Integers from 1 to N
$\mathcal{D}(x)$	Set of outputs given by models in $\mathcal{D}$ for input x
$\mathcal{M}(x,y,\mathcal{D})$	Subset of models of $\mathcal{D}$ giving y for input x
$(\mathcal{A}\backslash\mathcal{F})^{(\ell)}$	Subset of candidate models at step ℓ (6)
$s^{(\ell+1)}(x)$	Score of input x at step $\ell+1$ when $\mathcal{A}=\mathcal{B}$
$L^{\mathsf{pos}}, L^{\mathsf{neg}}$	Nb. of queries for a positive / negative detection
S_k	Surjection from $\mathcal{Z}_k$ to $\mathcal{S}_k := \{0,1,\dots,k\}$
z,y	Top-k output of black-box b or of model m
Z,Y	Random top-k output when the input is random
$\tilde{z},\tilde{Z},\tilde{y},\tilde{Y}$	Similar outputs after the sujection
W_θ	$(k+1)\times(k+1)$ Transition matrix
$\hat{I}(\tilde{Z},\tilde{Y})$	Empirical mutual information in bits
$\hat{H}_{\tilde{Z}}(\tilde{z})$	Empirical entropy in bits
$D_L(\mathsf{m}_1,\mathsf{m}_2)$	Distance between models with L queries
$D_L(\mathsf{b},\mathcal{F})$	Distance of the black-box from family $\mathcal{F}$

first requirement is that a small loss in the model performance is tolerated by Bob. If a variant does not comply with this criterion then Bob cannot consider it as an option. In classification, the performance of a model m is often gauged by the top-1 accuracy, denoted $\mathsf{acc}(\mathsf{m})$. We formalize this requirement as

$$\frac{\mathsf{acc}(\mathsf{m})-\mathsf{acc}(\mathsf{V}(\mathsf{m},\theta))}{\mathsf{acc}(\mathsf{m})}<\eta, \tag{1}$$

where $\eta>0$ is the tolerance (15% in our experimental work).

We also assume that the black-box performs the same classification task. As far as we know, fingerprinting is not possible between two networks performing different tasks if only top-k output is available. Transfer learning is therefore not considered as in previous works [2], [3], [17].

2) Resources: The second requirement is more subtle. We first need to limit the power of Bob. If Bob creates an accurate model *ex nihilo*, then Alice can pursue neither detection nor identification. We assume that Bob cannot train such a model from scratch because he lacks good training data, expertise in machine learning, or computing resources. This also means that Bob can retrain a model only up to a limited extent (typically using a small amount of new data). In other words, the complexity of the procedure creating $\mathsf{v}=\mathsf{V}(\mathsf{m},\theta)$ ought to be much smaller than the effort spent at training the original model m. Our experimental work considers two kinds of procedures.

a) Modification of the input: $\mathsf{v}(x)=\mathsf{m}(\mathsf{T}(x,\theta))$. Classifiers are robust to light input modifications. For images, the transformation T can be JPEG compression, posterizing, blurring, *etc*. In the same spirit, *randomized smoothing* [18] consists in adding noise to the input and aggregating the predicted classes into one single output.

b) Modification of the model: $\mathsf{v}(x)=\mathsf{T}(\mathsf{m},\theta)(x)$. The transform T slightly changes the model weights by for instance quantization, pruning, adversarial retraining or finetuning. Some of these procedures require small retraining with few resources so as not to lose too much accuracy.

In the sequel, the model in the black-box is denoted by b and $\mathcal{B}$ is the set of all possibilities, defined as:

$$\mathcal{B} := \{\mathsf{v}=\mathsf{V}(\mathsf{m},\theta):\mathsf{m}\in\mathcal{P},\theta\in\Theta,\mathsf{acc}(\mathsf{v})>(1-\eta)\mathsf{acc}(\mathsf{m})\}, \tag{2}$$

where $\mathcal{P}$ is a set of vanilla models and Θ a set of transformations (encompassing the identity $\mathsf{v}=\mathsf{m}$).

B. Alice: Disclosing the Remote Model

1) Goals: The task of Alice is to disclose which model is in the black-box. This has two flavours: detection or identification.

Detection (denoted by D) means that Alice performs a hypothesis test. She first makes a hypothesis about the black-box, then makes some queries, and finally decides whether the hypothesis holds based on the outputs of the black-box. The outcome of the detection is thus binary: Alice's hypothesis is deemed correct or not. This is the nominal use case in the related works [1], [2], [3], [4], [5].

Identification (denoted by I) means that Alice has no prior about the model in the black-box. She makes queries and processes the outputs to finally make a guess. The outcome is either the name of a model she knows, or the absence of a decision if she has not enough evidence.

2) Knowledge About the Black-Box: The second crucial point is her knowledge about the black-box. Alice can only detect or identify a relation to a model she knows: it means she has an implementation of this model, which she can freely test. We denote the set of models known by Alice by $\mathcal{A}$.

As by the very definition of a *variant*, Alice may know some of them but not all of them. For instance, some procedures V admit a real number as a parameter. Therefore, there is virtually an infinite number of variants. This leads to the convenient notion of a model *family*, we now introduce under three flavours:

- $\mathcal{F}(\mathsf{m})$: This family is the set of all variants made from the original vanilla model m:

$$\mathcal{F}(\mathsf{m}) := \{v=\mathsf{V}(\mathsf{m},\theta):\theta\in\Theta\}. \tag{3}$$

- $\mathcal{F}(\mathsf{m},\Psi)$: This family is the set of all variants made from the original vanilla model m by a specific procedure:

$$\mathcal{F}(\mathsf{m},\Psi) := \{v=\mathsf{V}(\mathsf{m},\theta):\theta\in\Psi\subset\Theta\}, \tag{4}$$

  where Ψ denotes the subset of parameters related to this specific procedure.

- $\mathcal{F}(\mathsf{m}, \{\theta\})$: This family is a singleton composed of a particular variant:

$$\mathcal{F}(\mathsf{m}, \{\theta\}) := \{v = \mathsf{V}(\mathsf{m}, \theta)\}. \quad (5)$$

With these definitions in mind, detection is based on the hypothesis that the black-box belongs to a given family, while Identification looks for the family the black-box belongs to.

3) Resources: A third element is the resources of Alice. We already mention the set $\mathcal{A}$ containing some vanilla models and few variants of theirs. She also has a collection of typical inputs, *i.e.*a testing dataset. We suppose that these inputs are statistically independent and distributed as the data in the training set of the models. In the sequel, the collection of inputs is denoted $\mathcal{X} = \{x_1, \ldots, x_N\}$.

In the end, be it for detection or identification, Alice selects some elements of $\mathcal{X}$ for querying the black-box. We denote this by an ordered list of indices: $q_{1:\ell} = (q_1, \ldots, q_\ell) \in [\![N]\!]^\ell$, where $[\![N]\!] := \{1, \ldots, N\}$. This means that Alice first queries x_{q_1}, and then x_{q_2} and so forth. The outputs of the black-box are denoted as $z_{1:\ell} = (z_{q_1}, \ldots, z_{q_\ell})$, with $z_{q_i} = \mathsf{b}(x_{q_i})$.

C. The Classifier in the Black-Box

The black-box works as any classifier. We denote the set of possible classes $\mathcal{C}$. The output $z = \mathsf{b}(x)$ for input x is the first k classes ordered by their predicted probabilities (*i.e.*the top-k). It means that z is an ordered list in $\mathcal{C}^k$: $z = (c_1, \ldots, c_k)$. The set $\mathcal{Z}_k$ of possible outcomes has a size as big as $(C)_k := C(C-1)\ldots(C-k+1)$. The black-box only discloses the top-k classes (*i.e.*this work does not build on the associated predicted probabilities). In the experimental work, the size of $\mathcal{C}$ is $C = 1,000$ (ImageNet) and $k \in \{1, 3, 5\}$ which is usual in several image classification APIs. We assume that the considered models and variants have an accuracy which is not perfect; the typical accuracy of ImageNet classic models ranges from 70% to 85%.

D. Summary

This paper considers scenarios which are labeled as $(\mathsf{Task}, \mathcal{F}, \mathcal{A}, k)$ where $\mathsf{Task} \in \{\mathsf{D}, \mathsf{I}\}$ (Detection or Identification), $\mathcal{F}$ is the kind of family that will be inferred by Alice, $\mathcal{A}$ is the set of models known by Alice, and k indicates that the output of the black-box is the top-k classes. There is a clear cut between the following two cases:

- Walled garden: $\mathcal{A} = \mathcal{B}$. We impose that the black-box is one of the networks known by Alice.
- Open world: $\mathcal{A} \subsetneq \mathcal{B}$. The black-box may not be a model known by Alice. This is the case when Bob uses an unknown variant, for instance.

This distinction drives the structure of the next sections because our solutions are of different nature.

III. Walled Garden the Black-Box Is a Known Model

Under the assumption that $\mathcal{A} = \mathcal{B}$, Alice achieves her goal when she correctly guesses which family the black-box belongs to. The alternative is to fail to gather enough evidence to make a decision. For the sake of clarity, we explain our procedure for a given family $\mathcal{F} \subset \mathcal{A}$, which can be one of the three types of families presented in Sect. II-B2.

Alice has a set of models composed of some vanilla models $\mathcal{P} = \{\mathsf{m}_1, \ldots, \mathsf{m}_M\}$ and some variants of theirs. Alice also has the collection of benign inputs $\mathcal{X} = (x_1, \ldots, x_N)$. Offline, she creates a database of $|\mathcal{A}|N$ outputs $(\mathsf{m}(x_j))_{\mathsf{m}\in\mathcal{A}, j\in[\![N]\!]}$.

Let $\mathcal{D}$ be a subset of $\mathcal{A}$. We define by $\mathcal{D}(x) := \{\mathsf{m}(x) : \mathsf{m} \in \mathcal{D}\}$ the set of labels predicted by the models in $\mathcal{D}$ for input x. With abuse of notations, $\mathcal{D}(x_{q_{1:\ell}})$ is the set of the concatenation of labels predicted by the models in $\mathcal{D}$ for the entries $(x_{q_1}, \ldots, x_{q_\ell})$. Conversely, $\mathcal{M}(x, y, \mathcal{D}) := \{\mathsf{m} : \mathsf{m} \in \mathcal{D}, \mathsf{m}(x) = y\}$ lists the models in $\mathcal{D}$ predicting y for input x.

A. Detection $(\mathsf{D}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$

Alice first makes a hypothesis about a family $\mathcal{F}$, and her goal is to discover whether the outcome is *positive* ($\mathsf{b} \in \mathcal{F}$) or *negative* ($\mathsf{b} \in \mathcal{A}\backslash\mathcal{F}$). We assume that Alice is convinced about her hypothesis and that she hopes for a *positive*. Our procedure thus focuses on reducing the number of models in $\mathcal{A}\backslash\mathcal{F}$ likely to be the black-box.

Alice uses a greedy algorithm which leverages the information about the black-box retrieved from the previous queries. According to the outputs of the black-box, several models can be discarded. At step ℓ, $(\mathcal{A}\backslash\mathcal{F})^{(\ell)}$ (resp. $\mathcal{F}^{(\ell)}$) denote the subset of models in $\mathcal{A}\backslash\mathcal{F}$ (resp. $\mathcal{F}^{(\ell)}$) which agree with the previous outputs. These are candidates in the sense that they could be the black-box model.

$$(\mathcal{A}\backslash\mathcal{F})^{(\ell)} := \bigcap_{i=1}^{\ell} \mathcal{M}(x_{q_i}, \mathsf{b}(x_{q_i}), \mathcal{A}\backslash\mathcal{F}). \quad (6)$$

Initially, all the models are candidates: $(\mathcal{A}\backslash\mathcal{F})^{(0)} = \mathcal{A}\backslash\mathcal{F}$ and $\mathcal{F}^{(0)} = \mathcal{F}$. At step $\ell+1$, the greedy algorithm sorts the inputs that have not yet been queried according to a score. This score $s^{(\ell+1)}(x)$ reflects how much the set of candidates $(\mathcal{A}\backslash\mathcal{F})^{(\ell)}$ reduces if input x is submitted next. We propose the expectation of the number of candidate models outside the family after querying input x assuming that the black-box is randomly picked in $\mathcal{F}^{(\ell)}$. This average is weighted by the number of models in $\mathcal{F}^{(\ell)}$ predicting a particular label:

$$s^{(\ell+1)}(x) = \sum_{y\in\mathcal{F}^{(\ell)}(x)} \left|\mathcal{M}(x, y, (\mathcal{A}\backslash\mathcal{F})^{(\ell)})\right| \frac{|\mathcal{M}(x, y, \mathcal{F}^{(\ell)})|}{|\mathcal{F}^{(\ell)}|} \quad (7)$$

Alice then submits one of the inputs with the lowest score:

$$q_{\ell+1} \in \arg\min_k s^{(\ell+1)}(x_k). \quad (8)$$

Our procedure stops after L iterations when meeting one of the three stopping criteria:

- $(\mathcal{A}\backslash\mathcal{F})^{(L)} = \emptyset$: The detection result is *positive*. No model outside the family responds like the black-box. The black-box is in the family since we assume it belongs to $\mathcal{A}$.
- $\mathcal{F}^{(L)} = \emptyset$: The detection result is *negative*. The responses of the black-box are different from the ones of the models in the family $\mathcal{F}$.
- $\min_k s^{(L+1)}(x_k) = |(\mathcal{A}\backslash\mathcal{F})^{(L)}|$: The detection failed. All the remaining models in $(\mathcal{A}\backslash\mathcal{F})^{(L)}$ and in $\mathcal{F}^{(L)}$ produce

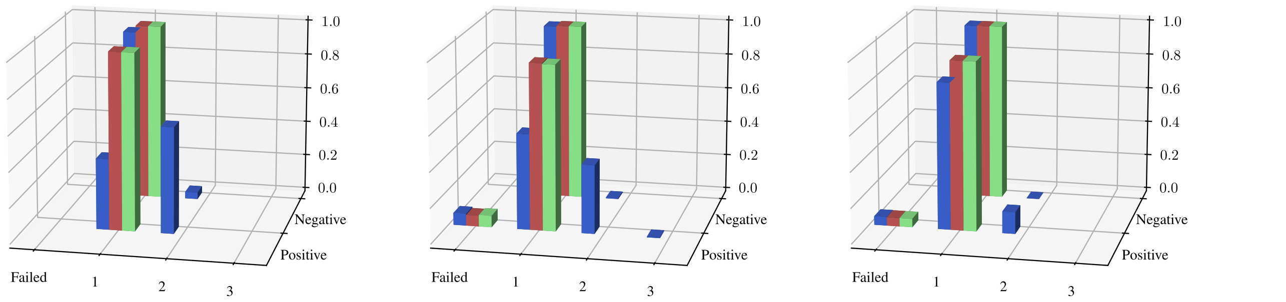


Fig. 2. Probability distribution of the number of queries for $(\mathsf{D}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ when the black box returns top-k classes with $k = 1$ (*blue*), $k = 3$ (*red*) or $k = 5$ (*green*). Family considered from left to right: $\mathcal{F}(\mathsf{m})$ (3), $\mathcal{F}(\mathsf{m}, \Psi)$ (4), and $\mathcal{F}(\mathsf{m}, \{\theta\})$ (5).

the same prediction no matter which input is submitted. It is therefore impossible to discern them.

B. Identification $(\mathsf{I}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$

Identification means that Alice makes a partition of her set of models in disjoint families: $\mathcal{A} = \cup_{i=1}^{n_\mathcal{F}} \mathcal{F}_i$. Her goal is to identify which family the black-box belongs to.

It is easy to base a verification procedure onto a detection scheme assuming there is no failure. Alice arbitrarily orders the families, and sequentially tests the hypotheses until she finds a match. The expected number of queries is given by (see the proof in App. C):

$$\mathbb{E}(L) = \frac{1}{n_\mathcal{F}} \sum_{j=1}^{n_\mathcal{F}} \mathbb{E}\left(L_j^{\mathsf{pos}}\right) + \frac{n_\mathcal{F} - 1}{2 n_\mathcal{F}} \sum_{j=1}^{n_\mathcal{F}} \mathbb{E}\left(L_j^{\mathsf{neg}}\right) \quad (9)$$

where $(\mathbb{E}(L_j^{\mathsf{pos}}), \mathbb{E}(L_j^{\mathsf{neg}}))$ are the expected number of queries necessary for taking a positive or negative decision about the *detection* of the hypothesis $\mathcal{F}_j$.

We propose a better approach based on a greedy algorithm similar to the detection one. Suppose that Alice has already submitted ℓ queries to the black-box. By comparing the outputs of the black-box and of the models she knows, she is able to distinguish models which are not in the black-box from models likely to be in the black-box. This list of remaining models is denoted $\mathcal{A}^{(\ell)}$. The goal of Alice is to reduce the set of candidates to a single family $\mathcal{F}_i$, not knowing in advance the model Bob placed in the black-box:

$$\exists i, \mathcal{F}_i \subset \mathcal{A}^{(\ell)}. \quad (10)$$

In the beginning, all the models are possibly in the black-box, *i.e.* $\mathcal{A}^{(0)} = \mathcal{A}$. At step $\ell+1$, the greedy algorithm chooses the best input to query next knowing $\mathcal{A}^{(\ell)}$.

Alice may resort to the following heuristics. She supposes that the black-box is randomly chosen uniformly in the set of remaining models $\mathcal{A}^{(\ell)}$. For any input x not queried yet, she computes the expectation of the number of remaining families if x were selected next, *i.e.*$|\{\mathcal{F}_i : \mathcal{F}_i \cap \mathcal{A}^{(\ell+1)} \neq \emptyset\}|$. She randomly chooses among the inputs minimizing this figure:

$$s^{(\ell+1)}(x) = \sum_{y \in \mathcal{A}^{(\ell)}(x)} \left(\sum_{i=1}^{n_\mathcal{F}} \delta_{\left[\mathcal{M}(x,y,\mathcal{F}_i^{(\ell)}) \neq \emptyset\right]} \right) \frac{|\mathcal{M}(x, y, \mathcal{A}^{(\ell)})|}{|\mathcal{A}^{(\ell)}|},$$

where $\delta_{[\mathcal{E}]}$ is the indicator function of event $\mathcal{E}$. The input to be submitted is sampled among the ones with the lowest score:

$$q_{\ell+1} \in \arg\min_k s^{(\ell+1)}(x_k). \quad (11)$$

TABLE II

PERCENTAGE OF INPUTS IN $\mathcal{X}$ CONCLUDING DETECTION $(\mathsf{D}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ WITHIN A SINGLE QUERY

Family	top-1	top-3	top-5
Vanilla $\mathcal{F}(\mathsf{m})$ (3)	0.28%	1.2%	**12.5%**
Variation $\mathcal{F}(\mathsf{m}, \Psi)$ (4)	0.31%	4.8%	**21.0%**
Singleton $\mathcal{F}(\mathsf{m}, \{\theta\})$ (5)	0.37%	8.3%	**31.4%**

C. Experimental Work

1) Detection: A first experimental work measures the number of queries needed for detection with the three types of family defined in Eq. (3), (4) and (5). It considers two cases: Alice's hypothesis is correct (positive case) or incorrect (negative case). The combinations are not analyzed exhaustively. For example, in the negative case for a singleton family (*i.e.* Alice is wrong to suspect that the black-box is $\mathcal{F}(m, \{\theta\})$), there are $|\mathcal{A}|(|\mathcal{A}| - 1)$ possible combinations, *i.e.*more than a million. Instead, the experiment randomly picks 1,000 positive and 1,000 negative among all these cases. Figure 2 shows the results obtained. As a side-product, Table II gives the percentage of inputs in $\mathcal{X}$ answering the detection problem under the best case, *i.e.*when one unique query is sufficient.

a) Few queries are enough: When the detection succeeds (be it a positive or negative decision about the hypothesis), at most three queries are needed, and in most cases, only one is sufficient. This holds although the greedy algorithm is known to be suboptimal. When the greedy algorithm needs three inputs, another algorithm may only need two queries. Yet, when the greedy algorithm needs two, no algorithm can do better because the greedy would have found an unique input if existing. Positive and negative conclusions are roughly drawn within the same number of queries, although our algorithm is designed to quickly prove positive detections.

b) Few failures: The inability to detect the model in the black-box as part of the family $\mathcal{F}$ happens when:

$$\exists \mathsf{m} \in \mathcal{F}, \exists \mathsf{m}' \in \mathcal{A} \backslash \mathcal{F}, \forall x \in \mathcal{X}, \quad \mathsf{m}(x) = \mathsf{m}'(x). \quad (12)$$

The failures occur when the family corresponds to a set of variations (4) or an exact model (5). It happens that the

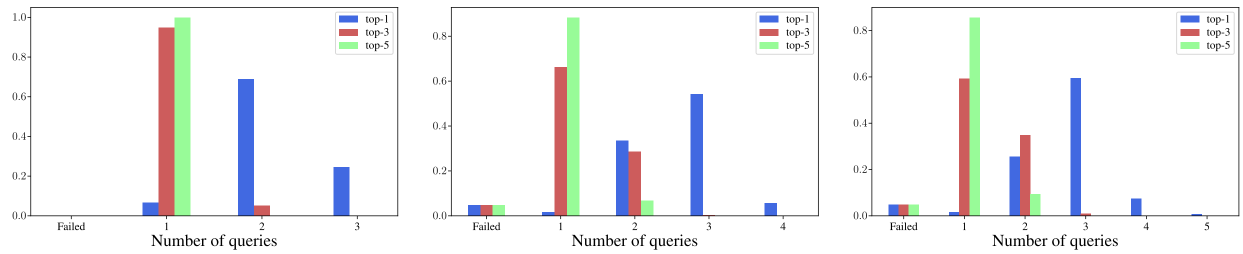


Fig. 3. Probability distribution of the number of queries for $(\mathsf{I}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ when the black box returns top-k classes with $k = 1$ (*blue*), $k = 3$ (*red*) or $k = 5$ (*green*). Family considered from left to right: $\mathcal{F}(\mathsf{m})$ (3), $\mathcal{F}(\mathsf{m}, \Psi)$ (4), and $\mathcal{F}(\mathsf{m}, \theta)$ (5).

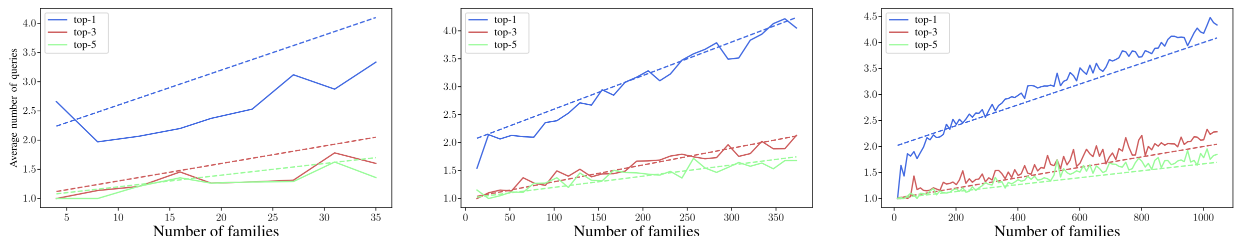


Fig. 4. Average number of queries as a function of the number of families $n_{\mathcal{F}}$ for $(\mathsf{I}, \mathcal{F}, \mathcal{A} = \mathcal{B}, k)$ with the expectation score (7) and when the black-box returns top-k classes with $k = 1$ (*blue*), $k = 3$ (*red*) or $k = 5$ (*green*). The dotted lines represent the linear regressions (13). Family considered from left to right: $\mathcal{F}(\mathsf{m})$ (3), $\mathcal{F}(\mathsf{m}, \Psi)$ (4), and $\mathcal{F}(\mathsf{m}, \{\theta\})$ (5).

algorithm cannot distinguish a few pairs of different variations issued from the same vanilla model. This is the only possible explanation: Otherwise, *i.e.*the indistinguishable models come from two different vanilla networks, a failure would also occur when detecting families spanned from a vanilla model (5), but this is not reported in Fig. 2. In other words, Alice can always guess that the black-box is a variation of a given vanilla model, and rarely she cannot guess which variation it is exactly.

On the other hand, failures should also happen in the negative case. None is reported in Fig. 2 because they are statistically rare. For a given family $\mathcal{F}$, suppose that the models m and m' in (12) are both unique. A failure happens in the positive case if Bob puts model m in the black-box. This happens with probability $1/|\mathcal{F}|$. A failure happens in the negative case if Bob puts model m' in the black-box. This happens with probability $1/|\mathcal{A}\backslash\mathcal{F}| < 1/|\mathcal{F}|$.

Experimentally, the number of queries to end up in a failure is similar to the number of queries for getting a positive outcome.

c) A bigger top-k is better: When the output of the black-box is rich, *i.e.*top-k classes with $k > 1$, one unique input is sufficient. Moreover, Table II shows that there are more of these unique inputs in $\mathcal{X}$. In this case, Alice no longer needs a large collection of benign inputs.

d) A bigger family is harder to detect: Families of type (3) are bigger than families (4) which are bigger than the singleton (5). Ignoring the failure case, Figure 2 and Table II show that it is harder to detect a large family. It is more frequent that some model members take different outputs in large families. On the contrary, we observe that the variants of the same model with the same variation but with different parameters often share the same output.

2) Identification: The protocol is similar to the previous one for detection. Figure 3 shows the results.

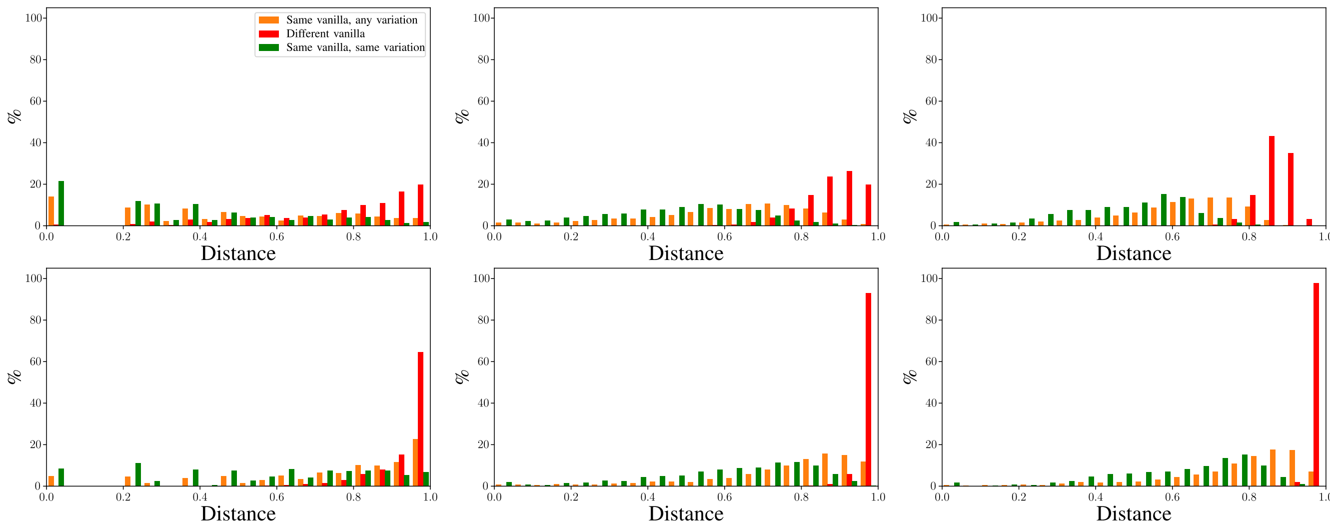
a) Identification vs. detection: Comparing Fig. 2 and 3, two times more queries are necessary for identifying a family rather than detecting it. It is possible to identify a model quickly with at most five benign queries which are a lot less than the sequential procedure (9). Identification is a harder task than detection to a small extent.

The biggest difference is under the top-1 scenario where a unique query is rarely sufficient. The 35 vanilla models considered here were trained on the same dataset. They have good accuracy ($> 70\%$). If many unique inputs to identify existed, this would mean that for any of these inputs, the 35 models give 35 different top-1 predictions. Assuming that one of these models makes a correct classification, the other 34 models are wrong. If a lot of these inputs existed, this would imply models with low accuracy. In other words, these inputs are necessarily rare, or even non-existing.

b) A bigger top-k is better: In contrast to detection, the gain of information provided by top-3 and top-5 is substantial. When the top-5 is returned, 90% of the families are identified within one query. The supervised training of the vanilla models only focuses on the top-1 s.t. it agrees with the ground truth class. For $k > 1$, the top-k is almost specific of the model. This explains the big improvement from top-1 to top-k.

c) Number of families: Figure 4 represents the evolution of the average number of queries to identify one out of $n_{\mathcal{F}}$ families. The more families, the bigger the number of queries on average. But this number also depends on the size of the families and the top-k. We observe that the increase is roughly linear (see dashed lines in Fig. 4). As a rule of thumb, we observe that the expectation of the number of queries roughly follows the empirical law:

$$\mathbb{E}(L) \approx 0.002 \times \frac{\mathbb{E}(|\mathcal{F}|) n_{\mathcal{F}}}{k} + \beta(k), \tag{13}$$



(a) $L = 20$ Images (b) $L = 100$ Images (c) $L = 1000$ Images

Fig. 5. Histogram of the distance $D_L(\mathsf{m}_1, \mathsf{m}_2)$ when $(\mathsf{m}_1, \mathsf{m}_2) \in \mathcal{F}^2(\mathsf{m})$ (orange), $(\mathsf{m}_1, \mathsf{m}_2) \in \mathcal{F}^2(\mathsf{m}, \Psi)$ (green), or m_1 and m_2 are variants of different vanilla models (red). Inputs randomly sampled in $\mathcal{X}$ (*top*) or in $\mathcal{X}'$ -Entropy Sect. IV-B2- (*bottom*).

where $\mathbb{E}(|\mathcal{F}|)$ is the average number of elements in the family. This is a major improvement w.r.t. (9). For instance, for singleton family, $\mathbb{E}(|\mathcal{F}|) = 1$ and the rate equals 0.002 under top-1, whereas the rate in (9) cannot be lower than 0.5 since we need at least one query to discard a hypothesis, *i.e.*$\mathbb{E}(L_j^{\mathsf{neg}}) \geq 1$.

IV. Open World:The Black-Box Is an Unknown Model

This section assumes that $\mathcal{A} \subsetneq \mathcal{B}$ because $\mathcal{B}$ contains models or variants of models unknown by Alice.

A. Modeling

1) Assumptions: Our working assumptions are the following: When queried by random inputs, a variant $\mathsf{V}(\mathsf{m}, \theta)$ produces outputs statistically

- independent from the outputs of a different model m'.
- dependent from the outputs of the original model m.

We consider a particular procedure for generating a variant as being like a transmission channel. The output Z of the variant $\mathsf{V}(\mathsf{m}, \theta)$ is as if the output Y of the original model m were transmitted to Alice through a noisy communication channel parametrized by θ. Like in C.E. Shannon's information theory of communication, we model this channel by the conditioned probabilities $W_\theta(z, y) = \mathbb{P}(Z = z|Y = y), \forall (z, y) \in \mathcal{Z}_k$.

2) Surjection: One difficulty of this context is the big size of the set $\mathcal{Z}_k$ of outcomes under the top-k assumption: $|\mathcal{Z}_k| = (C)_k$. It is then difficult to establish reliable statistics about the transition matrix W_θ which is as large as $(C)_k \times (C)_k$.

When working with top-k outputs, Alice resorts to a surjection $\mathsf{S}_k : \mathcal{Z}_k \mapsto \mathcal{S}_k$ with $\mathcal{S}_k := \{0, 1, \ldots, k\}$. This greatly reduces the set of outcomes. We denote $\tilde{z} = \mathsf{S}_k(z)$ and $\tilde{y} = \mathsf{S}_k(y)$. We choose a function S_k slightly more complex than suggested by this simple notation. Indeed, for any input x, we assume that Alice has a reference class $c(x) \in \mathcal{C}$. It is the ground truth class for annotated data. Otherwise, Alice computes the top-1 output of all the models she knows, and takes a majority vote to decide on $c(x)$. For this piece of data, a model gives $\mathsf{m}(x) = (c_1, \ldots, c_k)$ and the surjection makes:

$$\mathsf{S}_k(\mathsf{m}(x)) = \begin{cases} j & \text{if } \exists j : c_j = c(x) \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

In words, $\mathsf{S}_k(\mathsf{m}(x))$ is the rank of the reference class in the top-k output or 0 if the reference class is not returned. In the end, Alice uses a transmission matrix $(W_\theta(\tilde{z}, \tilde{y}))$ which is only $(k + 1) \times (k + 1)$.

B. Detection $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$

For the detection task, Alice first makes the following hypothesis: The black-box is a variant of the vanilla model $\mathsf{m} \in \mathcal{A}$. This variant may be the identity ($\mathsf{b} = \mathsf{m}$), or a variant she knows, or a variant she does not know.

Contrary to the previous section, Alice randomly chooses L inputs $(X_1, \ldots, X_L) \subset \mathcal{X}$ to query the black-box and compares the observations $(\tilde{Z}_1, \ldots, \tilde{Z}_L)$ to the outputs she knows $(\tilde{Y}_1, \ldots, \tilde{Y}_L)$, with $\tilde{Z}_\ell := \mathsf{S}_k(\mathsf{b}(X_\ell))$, $\tilde{Y}_\ell := \mathsf{S}_k(\mathsf{m}(X_\ell)), \forall \ell \in [\![L]\!]$. We use capital letters here to outline that these are random variables since Alice randomly chooses the inputs.

There are two difficulties: i) to gauge the distance between the outputs observed from the black-box and from model m (see Sect. IV-B1) and ii) to randomly sample informative inputs from the set $\mathcal{X}$ (see Sect. IV-B2).

1) Discriminative Distance: Alice tests two hypothesis:

- $\mathcal{H}_1$: The black-box is a variant of model m. There is a dependence between $\tilde{Z}$ and $\tilde{Y}$ which is captured by the statistical model of the variant:

$$\mathbb{P}_1(\tilde{Z} = \tilde{z}, \tilde{Y} = \tilde{y}) := W_\theta(\tilde{z}, \tilde{y})\mathbb{P}(\tilde{Y} = \tilde{y}).$$

- $\mathcal{H}_0$: The black-box is not a variant of model m. There is no statistical dependence and

$$\mathbb{P}_0(\tilde{Z} = \tilde{z}, \tilde{Y} = \tilde{y}) := \mathbb{P}(\tilde{Z} = \tilde{z})\mathbb{P}(\tilde{Y} = \tilde{y}).$$

The well-celebrated Neyman-Pearson test is the optimal score for deciding which hypothesis holds. For L independent observations, it writes as

$$s = \sum_{j=1}^{L} \log \frac{\mathbb{P}_1(\tilde{Z} = \tilde{z}_j, \tilde{Y} = \tilde{y}_j)}{\mathbb{P}_0(\tilde{Z} = \tilde{z}_j, \tilde{Y} = \tilde{y}_j)} = \sum_{j=1}^{L} \log \frac{W_\theta(\tilde{z}_j, \tilde{y}_j)}{\mathbb{P}(\tilde{Z} = \tilde{z}_j)}. \quad (15)$$

We introduce the empirical joint probability distribution defined by

$$\hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) := L^{-1}|\{j \in [\![L]\!] : \tilde{z}_j = \tilde{z} \text{ and } \tilde{y}_j = \tilde{y}\}| \quad (16)$$

in order to rewrite (15) as

$$s = L \sum_{(\tilde{z},\tilde{y}) \in \mathcal{S}_k^2} \hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) \log \frac{W_\theta(\tilde{z}, \tilde{y})}{\mathbb{P}(\tilde{Z} = \tilde{z})}. \quad (17)$$

This formalization is not tractable because W_θ is not known: Alice does not know which variant θ is in the black-box, and indeed it might be an unknown variant. Yet, (17) guides us to a more practical score function, the empirical mutual information:

$$\hat{I}(\tilde{Z}, \tilde{Y}) := \sum_{(\tilde{z},\tilde{y}) \in \mathcal{S}_k^2} \hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) \log \frac{\hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y})}{\hat{P}_{\tilde{Z}}(\tilde{z}) \hat{P}_{\tilde{Y}}(\tilde{y})}, \quad (18)$$

with the empirical marginal probabilities:

$$\hat{P}_{\tilde{Z}}(\tilde{z}) := \sum_{\tilde{y} \in \mathcal{S}_k} \hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}), \quad \hat{P}_{\tilde{Y}}(\tilde{y}) := \sum_{\tilde{z} \in \mathcal{S}_k} \hat{P}_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}). \quad (19)$$

In words, the model of the distributions $(\mathbb{P}_0, \mathbb{P}_1)$ is replaced with empirical frequencies learned on the fly. Resorting to the empirical mutual information to decode transmitted messages in digital communication is known as Maximum Mutual Information (MMI), recently proven universally optimal [19].

The empirical mutual information is a kind of similarity (the bigger, the more $\tilde{Z}$ looks like $\tilde{Y}$). Its value lies in the interval $[0, \min(\hat{H}(\tilde{Z}), \hat{H}(\tilde{Y}))]$ with the empirical entropy given by:

$$\hat{H}(\tilde{Z}) := -\sum_{\tilde{z}} P_{\tilde{Z}}(\tilde{z}) \log P_{\tilde{Z}}(\tilde{z}). \quad (20)$$

We prefer dealing with a normalized distance and we introduce:

$$D_L(\mathsf{b}, \mathsf{m}) := 1 - \frac{\hat{I}(\tilde{Z}; \tilde{Y})}{\hat{H}(\tilde{Y}, \tilde{Z})} \in [0, 1]. \quad (21)$$

This defines the Rajski distance [20] between the models b and m respectively producing $\tilde{Z}$ and $\tilde{Y}$.

The distances between all the pairs of models is shown in App. B in Fig. 9. The block diagonal shows that the distances between variants of the same vanilla models are small. We clearly see the cluster of variants centered on each vanilla model. Indeed, Figure 1 in the introduction is a t-SNE representation extracted from such pairwise distances between models in $\mathcal{B}$.

For a given model m, let us consider two extreme scenarios:

- The model m is in the black-box so that $\tilde{z}_j = \tilde{y}_j$, $\forall j \in [\![L]\!]$. Then $P_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) = 1$ if $\tilde{z} = \tilde{y}$, and 0 otherwise, producing $D_L(\mathsf{b}, \mathsf{m}) = 0$.
- The black-box and model m yield independent outputs so that $P_{\tilde{Z},\tilde{Y}}(\tilde{z}, \tilde{y}) = P_{\tilde{Z}}(\tilde{z}) P_{\tilde{Y}}(\tilde{y})$, then $D_L(\mathsf{b}, \mathsf{m}) = 1$.

In the end, Alice deemed the hypothesis $\mathcal{H}_1$ as being true when the distance is small enough: $D_L(\mathsf{b}, \mathsf{m}) < \tau \rightarrow \mathcal{H}_1$ is true. Alice makes two kinds of errors:

- False positive: $D_L(\mathsf{b}, \mathsf{m}) < \tau$ whereas $\mathcal{H}_1$ is false.
- False negative: $D_L(\mathsf{b}, \mathsf{m}) \geq \tau$ whereas $\mathcal{H}_1$ is true.

Alice sets the threshold τ such that the probability of false positive is lower than a required level α. The converse, *i.e.*controlling the probability of false negative, is an illusion. Appendix D shows for instance that there is no way to theoretically upper bound the distance between a variant and its original model, even if both of them share good accuracy. Our working assumption is that this mutual information is indeed large enough for a reliable hypothesis test and the experimental work confirms this in Sect. IV-D.

2) Selection of Inputs: The empirical mutual information is a consistent estimator of the mutual information which depends on the channel transition matrix W_θ and the input probability distribution $P_{\tilde{Y}}$. A result of the theory of communication is that for a given transmission channel, there is an input probability which maximises the mutual information. This is of utmost importance to design a communication system achieving the channel capacity as defined by C.E. Shannon. In our framework, this would make the distance between a model and its variant closer to 0 likely avoiding a false negative.

However, this idea is not applicable to our scheme because Alice may suspect a plurality of variants, each of them leading to a different optimal input distribution. The black-box may also contain an unknown variant excluding any optimization.

Yet, when Alice chooses random inputs, she has the feeling that these inputs must not be too easy to be classified otherwise any model outputs the same prediction. This is not discriminative of a given model in the black-box and it may lead to a false positive. On the other hand, these inputs must not be too hard to be classified neither otherwise the prediction tends to be random, destroying the correlation between a model and its variant. This may lead to a false negative.

Our experimental work investigates several selection mechanisms of the inputs. All of them amount to randomly pick inputs from a subset $\mathcal{X}'$ of $\mathcal{X}$.

- All. There is indeed no selection and $\mathcal{X}' = \mathcal{X}$.
- 50/50. Alice's hypothesis concerns a family of variants derived from a vanilla model m. $\mathcal{X}'$ is composed of 50% of inputs well classified by m (*i.e.*$\mathsf{m}(x) = c(x)$), 50% inputs for which $\mathsf{m}(x) \neq c(x)$.
- 30/70. The same definition but with 30% well classified and 70% wrongly classified by m.
- Entropy. $\mathcal{X}'$ is composed of the inputs whose top-1 predictions are highly random. For a given input, Alice computes the predictions from all the models in $\mathcal{A}$ and measures the empirical entropy of these predicted labels. She then sorts the inputs of $\mathcal{X}$ by their entropy, and $\mathcal{X}'$ contains the head of this ranking.

The second and third options are dedicated to the detection task since they only need the vanilla model m at the root of Alice's hypothesis. The last selection mechanism demands a long preprocessing step depending on how big the set of models $\mathcal{A}$ is. It is dedicated to the identification task.

In the worst-case, a model consistently predicts the ground truth at the same top-k position for all submitted images. This situation results in $P_{\tilde{Y}}(\tilde{y} = k) = 1$ leading to a null entropy and an undefined distance D_L. To mitigate this problem, Alice adopts a strategy where she ensures that at least one correct and one incorrect classification are selected for all models, following the introduced selection process. It highlights the limitation of our method for fingerprinting models achieving perfect accuracy.

C. Identification ($\mathsf{I}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k$)

The identification task is nothing more than an extension of the detection. Instead of a binary hypothesis, Alice is now facing a multiple hypotheses test with $M + 1$ choices:

- $\mathcal{H}_i$: The black-box is a variant of vanilla model m_i, with $1 \leq i \leq M$,
- $\mathcal{H}_0$: The black-box is a variant of an unknown model.

The usual way is to compute distance $D_L(\mathsf{b}, \mathsf{m}_i)$ per vanilla model $\mathsf{m}_i \in \mathcal{A}$, and to decide for model $i^\star = \arg\min_{1 \leq i \leq M} D_L(\mathsf{b}, \mathsf{m}_i)$, if $D_L(\mathsf{b}, \mathsf{m}_{i^\star})$ is lower than a threshold, otherwise Alice chooses hypothesis $\mathcal{H}_0$. If a known model is in the black-box, only three events may occur:

- Alice makes a correct identification,
- Alice can not make any decision. She deems $\mathcal{H}_0$ as true.
- Alice makes a wrong identification.

Again, by fine-tuning the threshold, Alice controls the probability of the last event. Note that the probability of success is expected to be smaller than for the previous task. Identification is more difficult since several hypotheses are competing.

1) Compound Model: Information theory helps Alice again thanks to an analogy with the communication over a compound channel. In this communication problem, a message m_i has been emitted and transmitted through a channel W_θ. The receiver knows a compound channel, *i.e.*a set of channels $\{\theta_j\}_{j=1}^V \subset \Theta$. It knows that the received signal has gone through one of them, but it does not know which one. There exists an optimal decoder for each channel in the set. The receiver just does not know which one to use. A theoretically grounded decoder is to decode the signal with each decoder and to aggregate this decoding with a min operator [21].

The analogy is the following: the inputs go through all the models $\{\mathsf{m}_i\}$ known by Alice, and the outputs are like messages. Bob has chosen one model, *i.e.*one of these messages. Yet, Bob uses a variant which emits noisy outputs observable to Alice. Now, suppose that Alice knows a set of variants in a given family: $\{\mathsf{V}(\mathsf{m}_i, \theta_j)\}_j \subset \mathcal{F}$. She uses these variants for computing distances $D_L(\mathsf{b}, \mathsf{V}(\mathsf{m}_i, \theta_j))$ that she aggregates into one distance w.r.t. the family:

$$D_L(\mathsf{b}, \mathcal{F}) := \min_j D_L(\mathsf{b}, \mathsf{V}(\mathsf{m}_i, \theta_j)). \tag{22}$$

Intuitively, the black-box might be a very degraded version of a model which is indeed 'closer' to a milder variant than to the original model m_i.

D. Experimental Work

The previous experimental work in Sect. III-C considers three kinds of family concerning the black-box as defined in Sect. II-B2. When the family is a singleton, because $\mathcal{F} = \mathcal{F}(\mathsf{m}, \{\theta\})$ or $\mathcal{F} = \mathcal{F}(\mathsf{m}, \Psi)$ and $|\Psi| = 1$, then the distance between the black-box and this unique model is exactly zero. This easy case is now excluded to focus on cases where Alice does not know the variant in the black-box.

Contrary to Sect. III-C, Alice now resorts to statistical tests. Any distance between models is a random value since the queries are randomly selected. Our experimental protocol makes 20 measurements of any considered distance thanks to 20 independent inputs samples.

1) Assumptions About the Statistical Model: Section IV-A1 makes two assumptions about the statistical dependence between the predictions of models in the same family $\mathcal{F}$ and independence when coming from different families. Figure 5 experimentally verifies these working assumptions.

The distances between two models $\mathsf{V}(\mathsf{m}, \theta)$ and $\mathsf{V}(\mathsf{m}', \theta')$ for two vanilla models m and m' and any variants $(\theta, \theta') \in \Theta \times \Theta$ are computed. This sums up to 583,740 combinations. Figure 5 shows the histogram of these distance values over 20 bins in red. A high number L of queries makes the measured distance more precise. The selection of the inputs has a major impact. When sampled on $\mathcal{X}$ (first row), the distance rarely values the maximum showing imperfect independance. This phenomenon has been revealed in [22]. Yet, when sampled on $\mathcal{X}'$ containing more inputs hardly correctly classified (second row), the distances are closer to one. The models tend to be independent when queried with a good selection of inputs.

Figure 5 also shows the histogram of distances between models belonging to the same family spanned by a vanilla model m, be it $\mathcal{F}(\mathsf{m}, \Psi)$ (same type of variation) or $\mathcal{F}(\mathsf{m})$ (any kind of variation). It is not possible to get a non-trivial upper bound of the distance in this case (explained in App. D). We empirically observe that two models from the same type of variation are usually closer. It is therefore easier to detect or identify families $\mathcal{F}(\mathsf{m}, \Psi)$ than $\mathcal{F}(\mathsf{m})$.

2) Detection ($\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k$)*:* The experiment considers all combinations of hypothesis and model put in the black-box. There are 35 vanilla models and 1046 variants. This makes 35 families of type $\mathcal{F}(\mathsf{m})$ with an average of 30 members per family. This represents 1081 positive cases and 36,754 negative cases. There are 377 families of type $\mathcal{F}(\mathsf{m}, \Psi)$ of which 203 with a size bigger than 1. This makes 907 positive cases and 218,536 negative cases. To assess the detection performance, Alice leverages the negative cases to determine the threshold τ as the empirical α-quantile of the False Positive Rate (FPR) (see Sect. IV-B1).

a) Selection of inputs: Table III shows the TPR obtained when the black-box returns only *top-1* decisions. As expected, the performances for the families $\mathcal{F}(\mathsf{m}, \Psi)$ are higher. The selection Entropy is clearly the best option. Its drawback is that it needs statistics about the predictions of many vanilla models. As far as the detection task is concerned, the other

TABLE III

TRUE POSITIVE RATE FOR $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, 1)$ WITH $L = 100$ QUERIES SAMPLED IN $\mathcal{X}'$ (SEE SECT. IV-B2). THE DELEGATE MODEL IS THE CLOSEST TO m. FALSE POSITIVE RATE IS SET TO 5%

	All	50/50	30/70	Entropy
$\mathcal{F}(\mathsf{m})$	83.4 ± 1.4	92.6 ± 1.0	92.8 ± 0.8	$\mathbf{94.7 \pm 0.7}$
$\mathcal{F}(\mathsf{m}, \Psi)$	86.9 ± 0.9	95.3 ± 0.9	95.6 ± 0.8	$\mathbf{97.2 \pm 0.5}$

TABLE IV

TRUE POSITIVE RATE FOR $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$ AND DIFFERENT DELEGATES WITH $L = 100$ QUERIES IN 30/70, FPR = 5%

Delegate		Close	Median	Far
$\mathcal{F}(\mathsf{m})$	top-1	$\mathbf{92.8 \pm 0.8}$	91.3 ± 1.1	31.6 ± 2.8
	top-3	$\mathbf{94.2 \pm 0.9}$	93.5 ± 0.2	36.8 ± 4.9
	top-5	$\mathbf{93.2 \pm 0.7}$	92.9 ± 0.8	35.4 ± 3.5
$\mathcal{F}(\mathsf{m}, \Psi)$	top-1	95.6 ± 0.8	$\mathbf{96.3 \pm 0.7}$	82.2 ± 1.2
	top-3	96.3 ± 0.6	$\mathbf{97.7 \pm 0.37}$	86.1 ± 1.3
	top-5	96.0 ± 0.6	$\mathbf{97.6 \pm 0.4}$	85.7 ± 0.9

selections are to be preferred. They only require the predictions of the suspected vanilla model. In the sequel, the selection 30/70 is used for further experiments on the detection task.

b) The delegate model: Alice measures a single distance in between the black-box and a delegate model of the hypothesis' family $\mathcal{F}$. Which member of the family is the best delegate? Three choices are proposed based on the distance to the vanilla model spanning the family: *Close*, *Median*, and *Far*. For instance, the *Close* option means that the delegate is the closest member in the family to the vanilla model:

$$\mathsf{m}_d = \arg\min_{\mathsf{m}' \in \mathcal{F}} D_L(\mathsf{m}, \mathsf{m}'). \tag{23}$$

In the case where $\mathcal{F} = \mathcal{F}(\mathsf{m})$, the closest member is m. It is not the case when $\mathcal{F} = \mathcal{F}(\mathsf{m}, \Psi)$, because the vanilla model m is not in this family. Recall that the intersection between two families has to be the empty set, otherwise Alice could not distinguish them.

Table IV evaluates the three options. Only the 180 families with more than 3 members are considered here. For smaller families, the three options would give the same delegate.

The delegate greatly influences the results. The best choice is to select the delegate as lying at the 'center' of the family. It means the *Close* option for the family $\mathcal{F}(\mathsf{m})$, which is indeed the vanilla model m, or the *Median* option for family $\mathcal{F}(\mathsf{m}, \Psi)$.

c) Top-k observations: The detection is evaluated for top-k outputs in Fig. 6. The best results are surprisingly obtained for $k = 1$ in Tab. V for a few queries. As the number of queries increases, the performance of $k = 3$ surpasses it and all top-k values converge to the same score after 500 queries. Our explanation is the following. The bigger k the richer the model. Yet, the empirical mutual information is calculated from $(k+1)^2$ estimated probabilities. For a given number of queries, the fewer estimations the more accurate they are. The top-1 is faster to estimate and reach good results quickly. Once the number of queries is enough, the top-3 takes the lead. They quickly get very good results close to 100%, simultaneously as top-5.

To summarize, the TPR reaches 95% for 150 queries under top-1, 110 under top-3, and 140 for top-5.

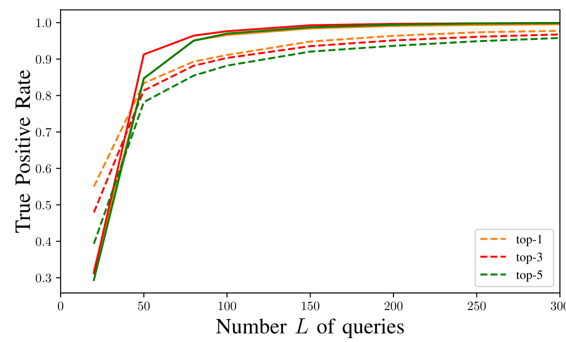


Fig. 6. True Positive Rate for $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$ function of the number of queries randomly selected in 30/70, FPR = 5%, best delegate options for $\mathcal{F}(\mathsf{m})$ (dash) and $\mathcal{F}(\mathsf{m}, \Psi)$ (plain).

TABLE V

TRUE POSITIVE RATE FOR $(\mathsf{D}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$ WITH RANDOM QUERIES SELECTED WITH *30/70*, FPR = 5%

Number of queries		$L = 20$	$L = 50$	$L = 100$	$L = 500$
$\mathcal{F}(\mathsf{m})$	top-1	**79.7**	86.9	92.8	**99.4**
	top-3	77.3	**88.0**	**94.2**	99.3
	top-5	76.8	87.3	93.2	99.3
$\mathcal{F}(\mathsf{m}, \Psi)$	top-1	83.1	91.5	96.3	**99.7**
	top-3	**84.3**	**94.1**	**97.7**	**99.7**
	top-5	83.6	94.0	97.6	99.6

TABLE VI

CORRECT IDENTIFICATION RATE FOR $(\mathsf{I}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$ WITH RANDOM QUERIES SELECTED WITH ENTROPY

Number of queries		$L = 50$	$L = 100$	$L = 500$
$\mathcal{F}(\mathsf{m})$ delegate = {close}	top-*1*	**67.1**	**80.0**	**98.6**
	top-*3*	49.1	57.8	85.3
	top-*5*	48.4	55.7	80.4
$\mathcal{F}(\mathsf{m}, \Psi)$ delegate = {median}	top-*1*	65.8	68.3	74.1
	top-*3*	58.2	64.5	71.4
	top-*5*	52.7	57.2	69.2
$\mathcal{F}(\mathsf{m}, \Psi)$ delegate = {close, median}	top-*1*	**73.1**	**77.2**	**83.6**
	top-*3*	61.8	70.0	80.2
	top-*5*	60.4	66.3	78.5

3) Identification $(\mathsf{I}, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$: All conclusions obtained in the previous section are kept. Alice now has for delegate the vanilla model m for $\mathcal{F}(\mathsf{m})$ and the *Median* model for $\mathcal{F}(\mathsf{m}, \Psi)$. Images are sampled with Entropy as defined in Sect. IV-B2.

a) Experimental protocol: We divide the identification task into three steps, each of them being prone to errors.

In the first step, Alice decides whether to abstain or proceed with identification. In the negative case where $\mathsf{b} \in \mathcal{F}(\mathsf{m}')$ but $\mathsf{m}' \notin \mathcal{A}$, the correct answer is to abstain and to consider the null hypothesis $\mathcal{H}_0$. If b belongs to $\mathcal{F}(\mathsf{m})$ and $\mathsf{m} \in \mathcal{A}$, the correct answer is to move to the next step of identification. We set the probability of error in the negative cases to 5% by controlling the threshold τ. Alice abstains if all distances are above the threshold. For this purpose, $\mathcal{A}$ consists of 30 models, while the remaining 5 models are used to generate the negative cases. Alice computes the distances between b and the 30 vanilla models in $\mathcal{A}$. This process is repeated 20 times, with a random selection of 5 excluded models from $\mathcal{P}$.

Once Alice decides that the black box is identifiable, the second step is to disclose the family $\mathcal{F}(\mathsf{m}_i)$. She decides for

TABLE VII
FALSE POSITIVE RATE FOR THE $(\mathsf{D}, \mathcal{F} = \{\mathsf{m}\}, \mathcal{A} = \mathcal{B}, 1)$ TASK

Fingerprinting scheme		Finetuning		Histogram	Randomized	Prune			Posterize	Half	JPEG
		All	Last		Smoothing	All	Last	Filter		Precision	
Sensitive Ex. [8] $L = 20$	$\epsilon = 8/255$	18.1	20.3	0	35.7	15.4	31.5	57.4	77.0	100	100
	$\epsilon = 16/255$	21.4	1.8	0	31.0	10.0	21.4	48.5	63.4	97.1	97.1
FBI with $L = 2$		**0**	**0**	**0**	**0**	**0**	**16.9**	**0**	**0**	**0**	**0**

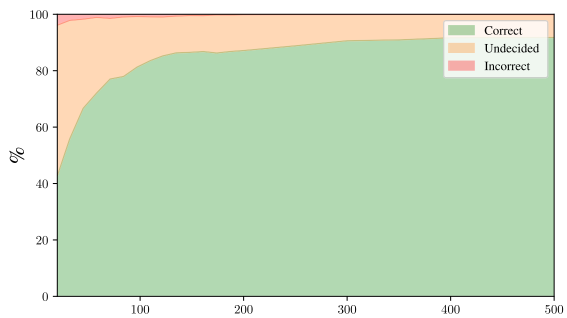


Fig. 7. Probability distribution for $(\mathsf{I}, \mathcal{F}(\mathsf{m}), \mathcal{A} \subsetneq \mathcal{B}, 1)$ vs. number L of queries. Threshold set to have a maximum 5% errors in negative cases.

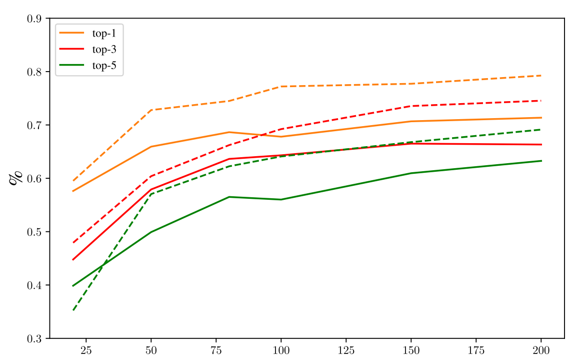


Fig. 8. Correct Identification Rate for $\mathcal{F}(\mathsf{m}, \Psi)$ as a function of the number of queries. One (plain) or two (dashed) delegates per family.

the hypothesis $\mathcal{H}_i$ minimizing the distance. When multiple models achieve this minimum distance, Alice is unable to make a decision and chooses to abstain. This conservative choice is more likely to occur when few images are submitted.

Finally, Alice identifies the variation, knowing she has made a correct identification of the global family $\mathcal{F}(\mathsf{m}_i)$. In this case, Alice has to identify the correct variation among 6 families $\{\mathcal{F}(\mathsf{m}, \Psi_j)\}_{j=1:6}$: randomized smoothing, pruning (filter, all, last), JPEG, posterize (See App. A). Alice thus computes 6 distances based on their delegates and identifies the family $i^\star = \arg\min_j D_L(\mathsf{b}, \mathcal{F}(\mathsf{m}, \Psi_j))$. No thresholding is needed here. For each family, 20 variants with random parameters and complying with (1) are created. This leads to 700 new models tested in the black-box, different from the 1,081 models considered so far.

b) Identifying $\mathcal{F}(\mathsf{m})$: Alice almost surely identifies the family $\mathcal{F}(\mathsf{m})$ of the black-box as shown in Fig. 7 and Tab. VI. She reaches her maximum success rate at around 300 queries. After 200 queries, no incorrect identification is made but 10% of abstention remains. This is due to the thresholding which

TABLE VIII
TRUE POSITIVE RATE FOR $(\mathsf{D}, \mathcal{F}(\mathsf{m}), \mathcal{A} \subsetneq \mathcal{B}, 1)$, FPR SET TO 5%

Fingerprinting scheme	Parameter	Number of queries	
		$L = 100$	$L = 200$
IP-Guard [2]	BP [28] & 50 iter.	66.9	72.7
FBI	Random	83.4	95.0
	30/70	92.8	97.8
	Entropy	**94.7**	**98.0**

prevents Alice from misclassification in the negative case. If no thresholding is done, the success rate reaches 94.7% within 100 queries and 99.1% at 500.

The number of queries is higher than for detection. For equivalent performance, 4 times more queries are necessary for identification than for detection. Nevertheless, identification proceeded by sequential detection would take on average 3.000 queries (24 times more w.r.t. detection) as foreseen by (9).

c) Identifying $\mathcal{F}(\mathsf{m}, \Psi)$: With a single delegate, Table VI and Figure 8 show a rather difficult identification. Variants far from the vanilla model are correctly identified. The main difficulty comes from the variations that slightly modify the model. These variants are close to m, which is the center of the cluster $\mathcal{F}(\mathsf{m})$ (see Fig. 1), therefore it is hard to distinguish them. The compound (22) with the median and the close delegates yields a boost if L is large enough.

d) Top-k observations: The best results are obtained for $k = 1$ in Tab. VI on every task, like for detection. For the family $\mathcal{F}(\mathsf{m})$, the information gained by top-k needs too many queries to catch up with the top-1. For family $\mathcal{F}(\mathsf{m}, \Psi)$, the difference is smaller. Indeed, top-k with $k \leq 3$ gives slightly better results from $\approx 1,000$ queries and above.

V. STATE-OF-THE-ART BENCHMARK

A. Previous Works

Since the work of IP-Guard [2], all the fingerprinting papers leverage adversarial examples. They start with a small collection of benign inputs (except [23] starting from random noise images) and apply a white-box attack like CW [24]. It forges adversarial examples that lie close to the decision boundaries, which are the signatures of a model.

Two trends are connected to two applications. The first one deals with the integrity of the model. In this scenario, Alice makes sure that Bob placed her model in the black-box without any alteration. The goal is to sense a *fragile fingerprint* such that any modification of the vanilla model is detectable because it changes the fingerprint. In that light, methods in [8] and [25] create sensitive examples which are triggered only by modifications of the vanilla model.

TABLE IX

TRUE POSITIVE RATE PER VARIATION UNDER $(\mathsf{D}, \mathcal{F}(\mathsf{m}), \mathcal{A} \subsetneq \mathcal{B}, 1)$.FALSE POSITIVE RATE SET TO 5% AND $L = 100$ QUERIES

Method	Parameter	Finetuning		Half Precision	Histogram	JPEG	Posterize	Prune			Randomized Smoothing
		All	Last					All	Filter	Last	
IP-Guard [2]	BP [29] 50 iterations	0.5	92.3	**100**	27.3	**100**	9.2	72.7	89.2	**100**	26.1
FBI	Random	85.6	91.5	**100**	64.2	**100**	88.3	65.0	87.8	87.3	60.0
	30/70	**94.5**	**97.3**	**100**	89.4	**100**	95.9	87.4	97.3	97.9	78.1
	Entropy	**94.5**	**97.3**	**100**	**92.0**	**100**	**98.5**	**91.9**	**99.5**	**98.9**	**85.5**

The second application is *robust fingerprint* as considered so far in this paper. The followers of IP-Guard [2] forge adversarial examples which are more robust in the sense that they remain adversarial for any variation of the model while being more specific to the vanilla model. Paper [3] proposes to use the universal adversarial perturbations of the vanilla model. Paper [26] introduces the concept of conferrable examples, *i.e.*adversarial examples which only transfer to the variations of the targeted model. AFA [5] activates dropout as a cheap surrogate of variants when forging adversarial examples. TAFA [4] extends this idea to other machine learning primitives.

Our take in this article is that using benign images is sufficient, and we addressed the fingerprinting problem without the need to rely on adversarial examples or any other technique to alter images to get them nearby the boundaries. Indeed, crafting adversarial examples is rather simple but forging them with extra specificities (fragile or robust to variation) is complex. It happens that all above-mentioned papers consider small input dimensions like MNIST or CIFAR (32×32 pixel images); none of them use ImageNet (224×224) except IP-Guard [2]. Also, no paper considers that the inputs can be reformed by a defense (in order to remove an adversarial perturbation before being classified) or detected as adversarial [27].

B. Fragile Fingerprinting

The application considered in [8] imagines that Alice wants to detect whether the black-box is exactly m and not a variant. This corresponds to our scenario $(\mathsf{D}, \mathcal{F}(\mathsf{m}, \{\theta\}), \mathcal{A} = \mathcal{B}, 1)$ where θ is the identity variation, and $\mathcal{A} = \mathcal{F}(\mathsf{m})$.

We create $L = 20$ sensitive examples per model with 200 iterations and two distortion budgets ($\epsilon = 8/255$ and $16/255$) using the code[2] released by [8]. It happens that its performance on ImageNet (reported in Tab. VII) is lower than the one reported in [8] on small input size datasets (like CIFAR). Especially, this scheme can not distinguish the vanilla model and its variants 'JPEG' or 'Half precision' even with a number of queries ($L = 20$) bigger than the one recommended ($L = 8$) in [8]. Our scheme needs no more than *two* queries and perfectly accurate, except when pruning the last layer for five out of thirty-five models.

C. Robust Fingerprinting

This application is related to our scenario $(\mathsf{D}, \mathcal{F}(\mathsf{m}), \mathcal{A} \subsetneq \mathcal{B}, k)$. IP-Guard [2] is the only work showing to be tractable and effective on large input size like in ImageNet. It leverages several white-box attacks to create adversarial examples. The best results demonstrated in the paper are with the attack CW [24]. We instead use BP [28] because it exhibits similar performances while being much faster (only 50 iterations). The BP implementation is from GitHub.[3]

Table VIII compares the performances under 100 and 200 queries and top-1 observations. Any selection of the inputs beats IP-Guard [2] . Detailed results are reported in Table IX. Some variations are easier to detect ('precison', 'pruning') and the two methods are on par. On the contrary, randomized smoothing which is a popular variation yet never considered in the literature, is more difficult. IP-Guard [2] is based on crafting adversarial examples close to the decision boundaries which are greatly crumpled by randomized smoothing. Not relying on adversarial examples seems to be a clear advantage in this case. Our method offers more stability in the results: No variation pulls the TPR below 85%.

VI. RELATED WORK

Having reviewed the previous works dedicated to the fingerprinting task in the section above, we here review another closely related domain: the *watermarking* of models.

Watermarking is the active counterpart of fingerprinting: Instead of relying on specifics of a fixed model to devise its unique fingerprint, watermarking modifies the model for which ownership must be proven. While watermarking is a common practice for decades in the field of image processing [30], it has just recently been incepted into the machine learning domain. Uchida et al. [31] first proposed to watermark a deep neural network by embedding it into the weights and biases of the model. Quickly after this initial proposal, works instead focused on a black-box model, where the presence of a watermark can be assessed by Alice from remote interaction with the suspected deep neural network, just like for fingerprinting. In [32], authors insert information by altering the decision boundaries through finetuning. In [33], authors also retrain the model to obtain the wrong labels for a so-called *trigger set* of inputs, that constitutes the watermark. Please refer to [34] for a complete overview of the domain.

Some papers claim that robust fingerprinting could replace watermarking with the clear advantage that no modification of the model is needed [5]. We strongly disagree. No fingerprinting scheme, including ours, brings any formal guarantee on what is in the black-box. It is indeed the primary goal of watermarking to guarantee a certain level of trust.

[2]Sensitive Examples' GitHub: https://github.com/zechenghe/Sensitive_Sample_Fingerprinting

[3]Boundary Projection's GitHub: https://github.com/hanwei0912/walking-on-the-edge-fast-low-distortion-adversarial-examples

VII. Conclusion

The problem of accurate and efficient fingerprinting of valuable models is salient. This paper demonstrates that such a demand can be fulfilled by solely using benign inputs, in not only the classic detection task, but also in the novel identification task we have introduced. This has the important implication that we no longer need models in white-box access to compute their fingerprints.

We provide the following takeaways.

i) In the walled garden setup of Sect. III, less than ten inputs are needed but these are sequentially and carefully selected among a large collection depending on the previous outputs of the black-box. In other words, the key is the interaction between the greedy algorithm and the black-box. Observing top-1, top-3 or top-5 makes a difference. It is easier to spot inputs that single out a model with richer outputs.

ii) In the open-world setup of Sect. IV, hundreds of inputs are necessary but the scheme is not iterative and selection is less crucial. Surprisingly, observing richer outputs does not yield any gain in this setup.

iii) The identification task is merely more complex than detection. Our identification schemes are much more efficient than the naive sequential search.

iv) Bob's best defenses in our experimental protocol are randomized smoothing for robust fingerprinting and pruning the last layer for fragile fingerprinting. It means that the former reduces the statistical dependence of the outputs, while the latter hardly perturbs the outputs given by the vanilla model.

One limitation of our work is that it cannot handle classifiers whose accuracy is almost perfect. This would happen for too easy classification setups where the value of models is lower, and fingerprinting is less critical. We nevertheless expect future models and applications to continue to be complex tasks, where reaching high accuracy levels will remain a struggle for both the academia and industry.

References

[1] S. Wang and C.-H. Chang, "Fingerprinting deep neural networks—A DeepFool approach," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[2] X. Cao, J. Jia, and N. Z. Gong, "IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, May 2021, pp. 14–25.

[3] Z. Peng, S. Li, G. Chen, C. Zhang, H. Zhu, and M. Xue, "Fingerprinting deep neural networks globally via universal adversarial perturbations," 2022, *arXiv:2202.08602*.

[4] X. Pan, M. Zhang, Y. Lu, and M. Yang, "TAFA: A task-agnostic fingerprinting algorithm for neural networks," in *Proc. Eur. Symp. Res. Comput. Secur.* Cham, Switzerland: Springer, 2021, pp. 542–562.

[5] J. Zhao, Q. Hu, G. Liu, X. Ma, F. Chen, and M. M. Hassan, "AFA: Adversarial fingerprinting authentication for deep neural networks," *Comput. Commun.*, vol. 150, pp. 488–497, Jan. 2020.

[6] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," 2015, *arXiv:1510.00149*.

[7] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, 2016.

[8] Z. He, T. Zhang, and R. Lee, "Sensitive-sample fingerprinting of deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4724–4732.

[9] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 135–147.

[10] M. P. Yankov, M. A. Olsen, M. B. Stegmann, S. S. Christensen, and S. Forchhammer, "Fingerprint entropy and identification capacity estimation based on pixel-level generative modelling," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 56–65, 2020.

[11] J. Daugman, "Information theory and the IrisCode," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 400–409, Feb. 2016.

[12] R. van den Berg, B. Skoric, and V. van der Leest, "Bias-based modeling and entropy analysis of PUFs," in *Proc. 3rd Int. Workshop Trustworthy Embedded Devices*, New York, NY, USA, Nov. 2013, pp. 13–20.

[13] H. Yu and P. Moulin, "Regularized AdaBoost learning for identification of time-varying content," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1606–1616, Oct. 2014.

[14] S. Voloshynovskiy, T. Holotyak, and F. Beekhof, "Soft content fingerprinting with bit polarization based on sign-magnitude decomposition," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2033–2047, Oct. 2015.

[15] T. Furon and L. Pérez-Freire, "EM decoding of Tardos traitor tracing codes," in *Proc. 11th ACM Workshop Multimedia Secur.*, Sep. 2009, pp. 99–106.

[16] Y.-W. Huang and P. Moulin, "On the saddle-point solution and the large-coalition asymptotics of fingerprinting games," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 160–175, Feb. 2012.

[17] S. Wang, X. Wang, P.-Y. Chen, P. Zhao, and X. Lin, "Characteristic examples: High-robustness, low-transferability fingerprinting of neural networks," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1–8.

[18] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. ICML*, 2019, pp. 1310–1320.

[19] R. Tamir and N. Merhav, "The MMI decoder is asymptotically optimal for the typical random code and for the expurgated code," 2020, *arXiv:2007.12225*.

[20] C. Rajski, "A metric space of discrete probability distributions," *Inf. Control*, vol. 4, no. 4, pp. 371–377, 1961.

[21] E. Abbe and L. Zheng, "Linear universal decoding for compound channels," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 5999–6013, Dec. 2010.

[22] H. Mania, J. Miller, L. Schmidt, M. Hardt, and B. Recht, "Model similarity mitigates test set overuse," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–10.

[23] S. Wang et al., "Intrinsic examples: Robust fingerprinting of deep neural networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2021, pp. 1–23.

[24] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[25] E. L. Merrer and T. Gilles, "TamperNN: Efficient tampering detection of deployed neural nets," in *Proc. IEEE 30th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2019, pp. 424–434.

[26] N. Lukas, Y. Zhang, and F. Kerschbaum, "Deep neural network fingerprinting by conferrable adversarial examples," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–18.

[27] A. Kherchouche, S. A. Fezza, W. Hamidouche, and O. Déforges, "Detection of adversarial examples in deep neural networks with natural scene statistics," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7.

[28] B. Bonnet, T. Furon, and P. Bas, "Generating adversarial images in quantized domains," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 373–385, 2022.

[29] H. Zhang, Y. Avrithis, T. Furon, and L. Amsaleg, "Walking on the edge: Fast, low-distortion adversarial examples," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 701–713, 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02931493

[30] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking," in *Proc. 3rd IEEE Int. Conf. Image Process.*, Sep. 1996, pp. 211–214.

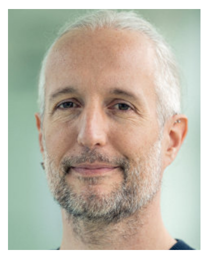
[31] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proc. ACM Int. Conf. Multimedia Retr.*, Jun. 2017, pp. 269–277.

[32] E. Le Merrer, P. Pérez, and G. Trédan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Comput. Appl.*, vol. 32, no. 13, pp. 9233–9244, Jul. 2020.

[33] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 1615–1631.
[34] Y. Li, H. Wang, and M. Barni, "A survey of deep neural network watermarking techniques," *Neurocomputing*, vol. 461, pp. 171–193, Oct. 2021.
[35] U. Shaham et al., "Defending against adversarial images using basis functions transformations," 2018, *arXiv:1803.10840*.
[36] E. Raff, J. Sylvester, S. Forsyth, and M. McLean, "Barrage of random transforms for adversarially robust defense," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6521–6530.
[37] A. Pedraza, O. Deniz, and G. Bueno, "Approaching adversarial example classification with chaos theory," *Entropy*, vol. 22, no. 11, p. 1201, Oct. 2020.
[38] R. Wightman. (2019). *PyTorch Image Models*. [Online]. Available: https://github.com/rwightman/pytorch-image-models
[39] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch," in *Proc. 18th ACM Int. Conf. Multimedia*, Oct. 2010, pp. 1485–1488.
[40] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.
[41] N. Das et al., "SHIELD: Fast, practical defense and vaccination for deep learning using JPEG compression," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 196–204.
[42] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.
[43] T. Maho, T. Furon, and E. L. Merrer, "Randomized smoothing under attack: How good is it in practice?" in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 3014–3018.
[44] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 656–672.
[45] B. Li, C. Chen, W. Wang, and L. Carin, "Second-order adversarial attack and certifiable robustness," in *Proc. ICLR*, 2019, pp. 1–15.
[46] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient ConvNets," 2016, *arXiv:1608.08710*.

Thibault Maho received the M.Sc. degree in signal and image processing from Institut Polytechnique de Grenoble, France, in 2018. He is currently pursuing the Ph.D. degree in security for artificial intelligence applications with Linkmedia Team, Inria, Rennes, France.

Teddy Furon (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in signal processing from Telecom ParisTech, in 1998 and 2002, respectively. He has worked in industry (Thomson and Technicolor) and academia (Université catholique de Louvain, Belgium, and the Linkmedia Team, Inria, Rennes, France). He co-founded the company Imatag protecting rights of photo agencies. His research interests include the security related to multimedia, signal processing, and artificial intelligence. He has been named the AID chair in security of artificial intelligence. He has been an Associate Editor of four journals, including IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.



Erwan Le Merrer worked as a Senior Scientist at Technicolor from 2008 to 2019. He is currently a Researcher at Inria, France. He is also leading the Scientific Council of the Société Informatique de France. He is involved in the algorithmic audit of AI-based decision-making algorithms.