



**HAL**  
open science

## Empreinte de réseaux avec des entrées authentiques

Thibault Maho, Teddy Furon, Erwan Le Merrer

► **To cite this version:**

Thibault Maho, Teddy Furon, Erwan Le Merrer. Empreinte de réseaux avec des entrées authentiques. CAID 2022 - Conference on Artificial Intelligence for Defense, DGA Maîtrise de l'Information, Nov 2022, Rennes, France. hal-03879849

**HAL Id: hal-03879849**

**<https://hal.science/hal-03879849>**

Submitted on 30 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

# Empreinte de réseaux avec des entrées authentiques

Thibault MAHO  
Univ. Rennes, Inria, CNRS  
IRISA, Rennes, France  
thibault.maho@inria.fr

Teddy FURON  
Univ. Rennes, Inria, CNRS  
IRISA, Rennes, France  
teddy.furon@inria.fr

Erwan LE MERRER  
Univ. Rennes, Inria, CNRS  
IRISA, Rennes, France  
erwan.lemerrer@inria.fr

**Abstract**—Les avancées récentes dans le domaine des empreintes de réseaux profonds détectent des instances de modèles placées dans une boîte noire. Les entrées utilisées en tant qu’empreintes sont spécifiquement conçues pour chaque modèle à vérifier. Bien qu’efficace dans un tel scénario, il en résulte néanmoins un manque de garantie après une simple modification (e.g. réentraînement, quantification) d’un modèle.

Cet article s’attaque aux défis de proposer i) des empreintes qui résistent aux modifications significatives des modèles, en généralisant la notion de familles de modèles et leurs variantes, ii) une extension de la tâche d’empreinte à des scénarios où l’on souhaite un modèle précis (précédemment appelé tâche de *detection*), mais aussi d’identifier la famille de modèles qui se trouve dans la boîte noire (tâche d’*identification*).

Nous atteignons ces deux objectifs en démontrant que des entrées authentiques (non modifiées) sont un matériau suffisant pour les deux tâches. Nous utilisons la théorie de l’information pour la tâche d’identification et un algorithme glouton pour la tâche de détection. Les deux approches sont validées expérimentalement sur un ensemble inédit de plus de 1 000 réseaux.

**Index Terms**—Empreinte, Réseaux profonds, Théorie de l’information

## I. INTRODUCTION

L’empreinte est une signature identifiant de manière unique un modèle, comme les minuties de l’empreinte digitale en biométrie. Il s’agit essentiellement d’un problème en boîte noire: le classifieur à identifier se trouve dans une boîte noire dans le sens où l’on peut uniquement faire quelques requêtes et observer ses résultats. Par exemple, le modèle est intégré dans une puce ou nous pouvons l’interroger via une API (MLaaS).

La principale application visée par les travaux actuels [1]–[5] est la preuve de la propriété. Un réseau de neurones profonds précis est un actif industriel précieux en raison du savoir-faire nécessaire à son entraînement, de la difficulté de rassembler un ensemble de données bien annoté, et des ressources nécessaires à l’apprentissage de ses paramètres. Le coût de GPT-3, l’un des modèles NLP les plus grands et les plus précis est estimé à 4,6 millions de dollars<sup>1</sup>. Dans ce contexte, l’entité qui identifie une boîte noire veut *detecter* s’il ne s’agit pas d’un de ses modèles volés.

Une autre application critique est le gain d’informations. Par exemple, un attaquant désireux de tromper le classifieur gagne d’abord des connaissances sur le modèle distant. Une entreprise peut également déterminer quel modèle est utilisé

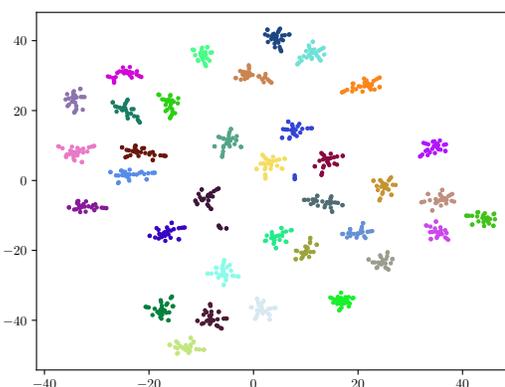


Fig. 1: Une représentation t-SNE des distances par paire de 1081 modèles différents: 10 types de variation appliqués sur 35 modèles vanilla prêts à l’emploi pour ImageNet avec différents paramètres. Ce travail exploite la séparabilité claire (groupes de couleurs) observée dans les décisions de ces modèles.

dans le système de production d’un concurrent. Cette application a été laissée de côté jusqu’à présent, et nous l’aborderons sous la notion d’*identification*.

Pour plus de clarté, nous nommons Alice l’entité voulant identifier le modèle que Bob a intégré dans la boîte noire.

a) *Challenges*: Il existe de nombreuses façons de modifier un modèle tout en conservant sa précision. Ces procédures permettent de simplifier un réseau (quantification des poids, pruning, voir e.g. [6]), ou le robustifier (prétraitement de l’entrée, réentraînement [7]). Nous nommons un modèle modifié une *variante*. Ces mécanismes n’ont pas été conçus a priori pour rendre l’empreinte plus difficile, mais ils laissent la possibilité à Bob d’altérer l’empreinte d’un modèle. Nous supposons qu’Alice connaît également certaines de ces procédures. Or, elles sont souvent définies par de nombreux paramètres, et parmi eux des scalaires, qui peuvent donner une infinité de variantes. Comme en biométrie, l’empreinte doit être suffisamment discriminante pour être unique par modèle mais aussi suffisamment robuste pour identifier une variante.

Les approches existantes reposent sur deux piliers. Elles utilisent les frontières de décision du classifieur comme empreinte [2], [4], [5]. Deux réseaux avec la même architecture et les mêmes ensemble et procédure d’apprentissage sont différents car l’apprentissage est stochastique. Ainsi, leurs

<sup>1</sup><https://lambdalabs.com/blog/demystifying-gpt-3/>

frontières dans l'espace d'entrée ne se chevauchent pas. La plupart des articles de la littérature recherchent des déviations discriminantes de ces frontières. Deuxièmement, la tâche clé est la détection: Alice fait une supposition sur le modèle dans la boîte noire et l'interroge ensuite avec des requêtes spécifiques pour valider son hypothèse [2], [4], [5], [8].

*b) Justification:* Nous constatons que l'utilisation d'entrées non modifiées n'a pas été étudiée en profondeur. Elle constitue un avantage certain, car elle supprime la nécessité de concevoir des moyens complexes pour les créer. Elle est moins sujette à l'utilisation de défenses du côté de Bob (e.g. rejet basé sur la distance à la frontière [9]). Les travaux précédents se restreignent à la tâche de détection. La possibilité plus générale d'identifier un modèle à l'intérieur de la boîte noire n'a pas été étudiée. Notre travail diffère donc des travaux précédents sur ces deux aspects: *i)* nous ne forgeons pas d'entrées spécifiques mais utilisons des entrées non modifiées (l'espace d'entrée n'est pas sondé pour découvrir les frontières de décision), et *ii)* nous identifions directement les modèles en utilisant leurs difficultés de classification sur une poignée d'entrées. Cette méthode est plus efficace que la détection séquentielle de modèles.

En résumé, quand Bob a choisi un modèle parmi un ensemble de réseaux connus par Alice, notre solution est essentiellement déterministe: Alice doit trouver la plus petite séquence d'entrées pour identifier la boîte noire. Nous appliquons un algorithme gourmand qui sélectionne soigneusement l'entrée pour réduire itérativement l'ensemble des suspects. La théorie de l'approximation indique que cette méthode est sous-optimale, mais nous constatons qu'en pratique, de nombreux réseaux sont identifiés en moins de trois requêtes.

Lorsqu'une attaque de Bob a créé une variante d'un modèle, la sortie de celui-ci peut ne pas correspondre à la sortie d'un modèle connu. Nous utilisons alors la théorie de l'information de Shannon pour mesurer la dépendance statistique entre les sorties des deux modèles. La Figure 1 est la représentation t-SNE à partir des distances par paire dans un ensemble de 35 modèles vanilla et de leurs variantes. Les familles de modèles sont bien regroupées dans le sens où les variantes sont plus proches de leur réseau d'origine que des autres modèles. Alice peut ne pas identifier précisément la variante du modèle mais elle peut identifier sa famille, déduire quel était le modèle vanilla d'origine et même quel type de variation a été appliqué.

*c) Contributions:* Notre contribution est quadruple. 1) Nous démontrons que la simple utilisation d'images non modifiées est suffisante pour atteindre des taux de réussite élevés pour l'empreinte des modèles. 2) La tâche de détection, introduite par l'état de l'art, est complétée par l'introduction de la tâche d'identification. Nous considérons cette dernière comme un problème de théorie de l'information. 3) Nous présentons une distance basée sur l'information mutuelle empirique, évaluant la proximité de deux modèles. Cette distance permet de généraliser la notion d'attaques sur les modèles à travers la notion de familles de modèles et de variantes. 4) Nous effectuons une expérimentation approfondie en considérant plus de 1 000 modèles de classification sur

ImageNet. Elle apporte des améliorations significatives en termes de précision dans la tâche de détection.

La section II est une analyse de menace listant toutes les hypothèses de travail. La section III s'intéresse à la détection (Alice vérifie son hypothèse sur la boîte noire) puis l'identification (Alice découvre quel modèle se trouve dans la boîte noire). Elle contient des résultats expérimentaux. La section IV est consacrée aux travaux apparentés et le benchmark avec les détections de l'état de l'art.

## II. MODÈLE DE MENACE

Cette section détaille les objectifs d'Alice et de Bob.

### A. Bob: garder son modèle anonyme

*1) Objectifs:* Bob joue en premier en choisissant secrètement un modèle et en le mettant dans la boîte noire. Ce modèle peut être un modèle classique ou une variante d'un modèle connu. Une variante est créée en appliquant sur un modèle vanilla  $m$  donné, la procédure  $V$  paramétrée par  $\theta \in \Theta$  qui décrit le type de modification et les paramètres associés. Cela peut être considéré comme une attaque de Bob sur le modèle vanilla pour renforcer l'identification. Nous désignons une telle variante par  $v = V(m, \theta)$ .

L'objectif de Bob est d'offrir un classifieur en boîte noire précis en conservant l'anonymat du modèle utilisé. Seule une petite perte de performance est tolérée. Pour la classification, les performances d'un modèle  $m$  sont évaluées par la précision du top-1, notée  $\text{acc}(m)$ . Nous formalisons cette exigence par

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

où  $\eta > 0$  est la tolérance (fixée à 15% dans nos expériences).

*2) Ressources:* La deuxième exigence est plus délicate. Nous devons limiter les capacités de Bob. Si Bob crée un modèle précis *ex nihilo*, alors Alice ne peut poursuivre ni détection ni identification.

Nous supposons que Bob ne peut pas créer un tel modèle à partir de zéro. Il ne dispose pas de bonnes données d'entraînement, d'expertise en apprentissage automatique ou de ressources suffisantes. Cela signifie aussi que Bob ne peut pas réentraîner un modèle, ou seulement avec des capacités limitées. En d'autres termes, la complexité de la procédure de création de  $v = V(m, \theta)$  doit être beaucoup plus faible que l'effort consacré à l'entraînement du modèle original  $m$ .

Notre travail expérimental considère deux procédures:

1) modification de l'entrée:  $v(x) = m(T(x, \theta))$ . Les classifieurs sont robustes aux transformations bénignes de l'entrée. En ce qui concerne les images, la transformation  $T$  peut être une compression JPEG, une égalisation d'histogramme, etc. Dans le même esprit, le random smoothing ajoute du bruit en entrée et agrège les classes prédites en une seule sortie.

2) modification du modèle:  $v(x) = T(m, \theta)(x)$ . La transformée  $T$  modifie les poids du modèle par quantification, fine-tuning... Certaines procédures requièrent un petit réentraînement avec peu de ressources pour limiter la chute de précision.

Dans la suite, le modèle en boîte noire est noté  $b$  et  $\mathcal{B}$  est l'ensemble des possibilités:  $b \in \mathcal{B}$ . Il est défini comme:

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

où  $\mathcal{P}$  est un ensemble de modèles vanilla et  $\Theta$  un ensemble de transformations (englobant l'identité  $v = m$ ).

### B. Alice: révéler le modèle à distance

1) *Objectifs*: La tâche d'Alice est de révéler quel modèle se trouve dans la boîte noire. Cette tâche peut prendre deux formes: détection ou identification.

*Détection* (noté D) signifie qu'Alice effectue un test d'hypothèse. Elle fait une hypothèse sur la boîte noire, puis effectue des requêtes pour valider l'hypothèse en se basant sur les sorties de la boîte noire. Le résultat de la détection est binaire: l'hypothèse d'Alice est jugée correcte ou non. Il s'agit de la tâche commune à tous les précédents travaux: [1]–[5].

*Identification* (noté I) signifie qu'Alice n'a pas a priori sur le modèle dans la boîte noire. Elle effectue des requêtes et traite les résultats pour finalement faire une supposition. Le résultat est soit le nom d'un modèle qu'elle connaît, ou l'absence de décision si elle n'a pas assez de preuves.

2) *Connaissance sur la boîte noire*: Le deuxième point crucial est sa connaissance sur la boîte noire. Alice peut uniquement détecter ou identifier un modèle qu'elle connaît. Elle a donc une implémentation de ce modèle qu'elle peut tester librement en boîte blanche. Nous désignons l'ensemble des modèles connus par Alice par  $\mathcal{A}$ .

Par la définition même d'une *variante*, Alice ne les connaît pas toutes. Par exemple, certaines procédures  $V$  admettent un nombre réel comme paramètre, et donc virtuellement un nombre infini de variantes existe. Cela conduit à la notion de *famille*, que nous présentons maintenant sous deux formes:

- $\mathcal{F}(m)$ : Cette famille est l'ensemble de toutes les variantes réalisées à partir du modèle vanilla  $m$ :

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

- $\mathcal{F}(m, \Psi)$ : Cette famille est l'ensemble de toutes les variantes réalisées à partir du modèle vanilla  $m$  par une procédure spécifique:

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

où  $\Psi$  désigne le sous-ensemble de paramètres liés à cette procédure spécifique.

Compte tenu de ces définitions, la détection repose sur l'hypothèse que la boîte noire appartienne à une famille donnée, alors que l'identification recherche la famille à laquelle la boîte noire appartient.

3) *Ressources*: Nous avons déjà mentionné l'ensemble  $\mathcal{A}$  contenant des modèles vanillas et des variantes de ceux-ci. Elle dispose aussi d'une collection d'entrées typiques, à savoir un ensemble de données de test. Nous supposons qu'elles sont statistiquement indépendantes et distribuées comme les données d'apprentissage des modèles. Dans la suite, la collection d'entrées est désignée par  $\mathcal{X} = \{x_1, \dots, x_N\}$ .

Au final, que ce soit pour la détection ou l'identification, Alice sélectionne des éléments dans  $\mathcal{X}$  pour interroger la boîte noire. Nous désignons cela par une liste ordonnée d'indices:  $q_{1:\ell} = (q_1, \dots, q_\ell) \in \llbracket N \rrbracket^\ell$ , où  $\llbracket N \rrbracket := \{1, \dots, N\}$ . Cela signifie qu'Alice soumet d'abord  $x_{q_1}$ , puis  $x_{q_2}$  et ainsi de suite. Les sorties de la boîte noire sont désignées par  $z_{1:\ell} = (z_{q_1}, \dots, z_{q_\ell})$ , avec  $z_{q_i} = b(x_{q_i})$ .

### C. Le classifieur dans la boîte noire

La boîte noire fonctionne comme n'importe quel classifieur. Nous désignons l'ensemble des classes possibles par  $\mathcal{C}$ . La sortie  $z = b(x)$  pour l'entrée  $x$  est les  $k$  premières classes ordonnées par leurs probabilités prédites (le top- $k$ ). Cela signifie que  $z$  est une liste ordonnée dans  $\mathcal{C}^k$ :  $z = (c_1, \dots, c_k)$ . L'ensemble  $\mathcal{Z}_k$  des résultats possibles a une taille aussi grande que  $(\mathcal{C})_k := \mathcal{C}(\mathcal{C} - 1) \dots (\mathcal{C} - k + 1)$ . La boîte noire ne divulgue que ces classes, et non les probabilités prédites associées. Dans le travail expérimental, la taille de  $\mathcal{C}$  est de 1 000 (ImageNet) et  $k \in \{1, 3, 5\}$  ce qui est habituel dans plusieurs API de classification d'images. Nous supposons que les modèles et variantes considérés ont une précision qui n'est pas parfaite. La précision du top-1 des modèles classiques d'ImageNet varie entre 70% et 85%.

## III. ALGORITHME

### A. Modélisation

1) *Hypothèses*: Nos hypothèses de travail sont les suivantes: lorsqu'elle est interrogée par des entrées aléatoires, une variante  $V(m, \theta)$  produit des sorties statistiquement:

- indépendantes des sorties d'un modèle différent  $m' \neq m$ .
- dépendantes des sorties du modèle original  $m$ .

Nous considérons qu'une procédure particulière de génération d'une variante s'apparente à un canal de transmission. La sortie  $Z$  de la variante  $V(m, \theta)$  est comme si la sortie  $Y$  du modèle original  $m$  était transmise à Alice à travers un canal de communication bruyant paramétré par  $\theta$ . Comme dans la théorie de l'information de Shannon, ce canal est modélisé par les probabilités conditionnelles  $W_\theta(z, y) = \mathbb{P}(Z = z | Y = y), \forall (z, y) \in \mathcal{Z}_k$ .

2) *Surjection*: Une des difficultés de ce contexte est la grande taille de l'ensemble  $\mathcal{Z}_k$  des résultats sous l'hypothèse du top- $k$ :  $|\mathcal{Z}_k| = (\mathcal{C})_k$ . Il est alors difficile d'établir des statistiques fiables sur la matrice de transition  $W_\theta$  qui est aussi grande que  $(\mathcal{C})_k \times (\mathcal{C})_k$ .

Lorsqu'elle travaille avec le top- $k$ , Alice a recours à une surjection  $S_k : \mathcal{Z}_k \mapsto \mathcal{S}_k$  avec  $\mathcal{S}_k := \{0, 1, \dots, k\}$ . Cela réduit considérablement l'ensemble des résultats. Nous désignons  $\tilde{z} = S_k(z)$  et  $\tilde{y} = S_k(y)$ . La fonction  $S_k$  est légèrement plus complexe que ne le suggère cette notation simple. En effet, pour toute entrée  $x$ , on suppose qu'Alice possède une classe de référence  $c(x) \in \mathcal{C}$ . Si Alice n'a pas la vérité de  $x$ , elle procède à un vote majoritaire sur tous les modèles qu'elle connaît pour décider de  $c(x)$ . Un modèle  $m(x) = (c_1, \dots, c_k)$  et la surjection donne:

$$S_k(m(x)) = \begin{cases} j & \text{si } \exists j : c_j = c(x) \\ 0 & \text{sinon.} \end{cases} \quad (5)$$

En d'autres termes,  $S_k(m(x))$  est le rang de la classe de référence dans la sortie top- $k$  ou 0 si la classe de référence n'est pas retournée.

### B. Detection

Pour la tâche de détection, Alice émet d'abord l'hypothèse suivante: La boîte noire est une variante du modèle vanilla  $m \in \mathcal{A}$ . Cette variante peut être l'identité ( $b = m$ ), ou une variante qu'elle connaît ou ne connaît pas.

Alice choisit aléatoirement  $L$  entrées  $(X_1, \dots, X_L) \subset \mathcal{X}$  pour interroger la boîte noire et compare les observations  $(\tilde{Z}_1, \dots, \tilde{Z}_L)$  aux sorties qu'elle connaît  $(\tilde{Y}_1, \dots, \tilde{Y}_L)$ , avec  $\tilde{Z}_\ell := S_k(b(X_\ell))$ ,  $\tilde{Y}_\ell := S_k(m(X_\ell))$ ,  $\forall \ell \in \llbracket L \rrbracket$ . Nous utilisons des majuscules ici pour souligner qu'il s'agit de variables aléatoires puisqu'Alice choisit aléatoirement les entrées.

Il y a deux difficultés : i) jauger la distance entre les sorties observées de la boîte noire et du modèle  $m$  (voir Sect. III-B1), et ii) échantillonner aléatoirement des entrées informatives à partir de l'ensemble  $\mathcal{X}$  (voir Sect. III-B2).

1) *Distance discriminante*: Alice teste en effet deux hypothèses :

- $\mathcal{H}_1$ : la boîte noire est une variante du modèle  $m$ . Il existe une dépendance entre  $\tilde{Z}$  et  $\tilde{Y}$  qui est capturée par le modèle statistique de la variante :

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

- $\mathcal{H}_0$ : la boîte noire n'est pas une variante du modèle  $m$ . Il n'y a pas de dépendance statistique et

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

Le célèbre test de Neyman-Pearson est le score optimal pour décider de l'hypothèse retenue. Pour  $L$  observations indépendantes, il s'écrit comme suit

$$s = \sum_{j=1}^L \log \frac{\mathbb{P}_1(\tilde{Z} = \tilde{z}_j, \tilde{Y} = \tilde{y}_j)}{\mathbb{P}_0(\tilde{Z} = \tilde{z}_j, \tilde{Y} = \tilde{y}_j)} = \sum_{j=1}^L \log \frac{W_\theta(\tilde{z}_j, \tilde{y}_j)}{\mathbb{P}(\tilde{Z} = \tilde{z}_j)}. \quad (6)$$

Nous introduisons la probabilité conjointe empirique:

$$\hat{P}_{\tilde{Z}, \tilde{Y}}(\tilde{z}, \tilde{y}) := L^{-1} |\{j \in \llbracket L \rrbracket : \tilde{z}_j = \tilde{z} \text{ and } \tilde{y}_j = \tilde{y}\}| \quad (7)$$

afin de réécrire (6) comme

$$s = L \sum_{(\tilde{z}, \tilde{y}) \in \mathcal{S}_k^2} \hat{P}_{\tilde{Z}, \tilde{Y}}(\tilde{z}, \tilde{y}) \log \frac{W_\theta(\tilde{z}, \tilde{y})}{\mathbb{P}(\tilde{Z} = \tilde{z})}. \quad (8)$$

Cette formalisation n'est pas réalisable car  $W_\theta$  n'est pas connu: Alice ne sait pas quelle variante de  $\theta$  se trouve dans la boîte noire, et il pourrait en réalité s'agir d'une variante inconnue. Pourtant, elle nous guide vers une fonction de score plus pratique, l'information mutuelle empirique :

$$\hat{I}(\tilde{Z}, \tilde{Y}) := \sum_{(\tilde{z}, \tilde{y}) \in \mathcal{S}_k^2} \hat{P}_{\tilde{Z}, \tilde{Y}}(\tilde{z}, \tilde{y}) \log \frac{\hat{P}_{\tilde{Z}, \tilde{Y}}(\tilde{z}, \tilde{y})}{\hat{P}_{\tilde{Z}}(\tilde{z})\hat{P}_{\tilde{Y}}(\tilde{y})}, \quad (9)$$

avec les probabilités marginales empiriques :

$$\hat{P}_{\tilde{Z}}(\tilde{z}) := \sum_{\tilde{y} \in \mathcal{S}_k} \hat{P}_{\tilde{Z}, \tilde{Y}}(\tilde{z}, \tilde{y}), \quad \hat{P}_{\tilde{Y}}(\tilde{y}) := \sum_{\tilde{z} \in \mathcal{S}_k} \hat{P}_{\tilde{Z}, \tilde{Y}}(\tilde{z}, \tilde{y}). \quad (10)$$

En d'autres termes, le modèle des distributions  $(\mathbb{P}_0, \mathbb{P}_1)$  est remplacé par des fréquences empiriques apprises à la volée. Le recours à l'information mutuelle empirique pour décoder les messages transmis dans les communications numériques est connu sous le nom de Maximum Mutual Information (MMI), qui s'est récemment avéré universellement optimal [10].

L'information mutuelle empirique est une sorte de similarité. Sa valeur est comprise dans  $[0, \min(\hat{H}(\tilde{Z}), \hat{H}(\tilde{Y}))]$  avec l'entropie empirique donnée par:

$$\hat{H}(\tilde{Z}) := - \sum_{\tilde{z}} P_{\tilde{Z}}(\tilde{z}) \log P_{\tilde{Z}}(\tilde{z}). \quad (11)$$

Une distance normalisée est préférée et nous introduisons:

$$D_L(b, m) := 1 - \frac{\hat{I}(\tilde{Z}, \tilde{Y})}{\min(\hat{H}(\tilde{Y}), \hat{H}(\tilde{Z}))} \in [0, 1]. \quad (12)$$

Cela définit la distance entre les modèles  $b$  et  $m$  produisant respectivement  $\tilde{Z}$  et  $\tilde{Y}$ . En effet, la figure 1 de l'introduction est une représentation graphique t-SNE extraite de telles distances par paires entre les modèles dans  $\mathcal{B}$ . Par exemple, considérons deux scénarios extrêmes :

- Le modèle  $m$  est dans la boîte noire de sorte que  $\tilde{z}_j = \tilde{y}_j$ ,  $\forall j \in \llbracket L \rrbracket$ . Alors  $P_{\tilde{Z}, \tilde{Y}}(\tilde{z}, \tilde{y}) = 1$  si  $\tilde{z} = \tilde{y}$ , et 0 sinon, ce qui produit  $D_L(b, m) = 0$ .
- La boîte noire et le modèle  $m$  produisent des sorties indépendantes de sorte que  $P_{\tilde{Z}, \tilde{Y}}(\tilde{z}, \tilde{y}) = P_{\tilde{Z}}(\tilde{z})P_{\tilde{Y}}(\tilde{y})$ , alors  $D_L(b, m) = 1$ .

Au final, Alice a jugé l'hypothèse  $\mathcal{H}_1$  comme étant vraie lorsque la distance est suffisamment petite:  $D_L(b, m) < \tau \rightarrow \mathcal{H}_1$  vrai. Alice fait deux types d'erreurs:

- Faux positif:  $D_L(b, m) < \tau$  alors que  $\mathcal{H}_1$  est faux.
- Faux négatif:  $D_L(b, m) \geq \tau$  alors que  $\mathcal{H}_1$  est vrai.

Alice fixe le seuil  $\tau$  de telle sorte que la probabilité de faux positifs soit inférieure à un niveau requis  $\alpha$ . Il n'y a aucun moyen de limiter théoriquement la distance entre une variante et son modèle original, même si les deux partagent une bonne précision. Notre hypothèse de travail est que cette information mutuelle est en effet suffisamment importante pour un test d'hypothèse fiable et le travail expérimental le confirme dans la section III-D.

2) *Sélection des entrées*: L'information mutuelle empirique est un estimateur cohérent de l'information mutuelle qui dépend de la matrice de transition du canal  $W_\theta$  et de la distribution de probabilité d'entrée  $P_{\tilde{Y}}$ . Un résultat de la théorie de l'information est que pour un canal de transmission donné, il existe une probabilité d'entrée qui maximise l'information mutuelle. Ceci est très important pour concevoir un système de communication atteignant la capacité du canal telle que définie par Shannon. Dans notre cadre, cela rendrait la distance entre un modèle et sa variante plus proche de 0, ce qui éviterait probablement un faux négatif.

Cependant, cette idée n'est pas applicable à notre schéma car Alice peut connaître une pluralité de variantes, toutes menant à une distribution optimale différente des entrées.

Pourtant, lorsqu'Alice choisit aléatoirement les entrées, elle a le sentiment qu'elles ne doivent pas être trop faciles à

classer, sinon tout modèle produit la même prédiction. Cela ne discrimine pas un modèle donné dans la boîte noire et peut conduire à un faux positif. D'autre part, si ces entrées sont trop difficiles à classer, la prédiction tend à être aléatoire. La corrélation entre un modèle et sa variante est détruite et conduit à un faux négatif.

Le travail expérimental étudie plusieurs mécanismes de sélection des entrées. Ils reviennent tous à choisir aléatoirement des entrées dans un sous-ensemble  $\mathcal{X}'$  de  $\mathcal{X}$ .

- Aléatoire. Il n'y a en effet aucune sélection et  $\mathcal{X}' = \mathcal{X}$ .
- 50/50. L'hypothèse d'Alice concerne une famille de variantes dérivées d'un modèle vanilla  $m$ .  $\mathcal{X}'$  est composé de 50% d'entrées bien classées par  $m$  (c'est-à-dire  $m(x) = c(x)$ ), 50% d'entrées pour lesquelles  $m(x) \neq c(x)$ .
- 30/70. Même définition mais avec 30% de bien classés et 70% de mal classés par  $m$ .
- Entropie.  $\mathcal{X}'$  est composé d'entrées dont les prédictions top-1 sont hautement aléatoires. Pour une entrée donnée, Alice calcule les prédictions de tous les modèles de  $\mathcal{A}$  et mesure l'entropie empirique de ces étiquettes prédites. Elle trie ensuite les entrées de  $\mathcal{X}$  par leur entropie, et  $\mathcal{X}'$  contient la tête de ce classement.

Les deuxième et troisième options ne nécessitent que le modèle vanilla  $m$ . Elles sont dédiées à la tâche de détection. La dernière option exige une étape de prétraitement en fonction de la taille de l'ensemble  $\mathcal{A}$ . Elle est dédiée à la tâche d'identification.

### C. Identification

La tâche d'identification est une extension de celle de détection. Au lieu d'une hypothèse binaire, Alice est maintenant confrontée à un test d'hypothèses multiples avec  $M + 1$  choix:

- $\mathcal{H}_i$ : La boîte noire est une variante du modèle vanilla  $m_i$ , avec  $1 \leq i \leq M$ ,
- $\mathcal{H}_0$ : La boîte noire est une variante d'un modèle inconnu.

La manière habituelle est de calculer la distance  $D_L(\mathbf{b}, m_i)$  par modèle vanilla  $m_i \in \mathcal{A}$ , et de décider pour le modèle  $i^* = \arg \max_{1 \leq i \leq M} D_L(\mathbf{b}, m_i)$ , si  $D_L(\mathbf{b}, m_{i^*})$  est inférieur à un seuil, sinon Alice choisit l'hypothèse  $\mathcal{H}_0$ . Si un modèle connu se trouve dans la boîte noire, seuls trois événements peuvent se produire:

- Alice fait une identification correcte,
- Alice ne peut pas prendre de décision ( $\mathcal{H}_0$  vrai).
- Alice fait une mauvaise identification.

En réglant finement le seuil, Alice contrôle la probabilité du dernier événement. Notez que la probabilité de succès devrait être plus faible que pour la tâche précédente. L'identification est plus difficile car plusieurs hypothèses sont en concurrence.

1) *Modèle composé*: La théorie de l'information aide à nouveau Alice grâce à une analogie avec la communication sur un canal composé. Dans ce problème de communication, un message  $m_i$  a été émis et transmis par un canal  $W_\theta$ . Le récepteur connaît un canal composé, c'est-à-dire un ensemble de canaux  $\{\theta_j\}_{j=1}^V \subset \Theta$ . Il sait que le signal reçu est passé

par l'un d'eux, mais il ne sait pas lequel. Il existe un décodeur optimal pour chaque canal de l'ensemble. Le récepteur ne sait simplement pas lequel utiliser. Un décodeur fondé en théorie consiste à décoder le signal reçu avec chacun des décodeurs et à agréger ce décodage avec un opérateur  $\min$  [11].

L'analogie est la suivante: les entrées passent par tous les modèles  $\{m_i\}$  connus par Alice, et les sorties sont ses messages. Bob a choisi un modèle, *i.e.* un de ces messages. Mais Bob utilise une variante qui émet des sorties bruitées. Maintenant, supposons qu'Alice connaisse un ensemble de variantes dans une famille donnée:  $\{V(m_i, \theta_j)\}_j \subset \mathcal{F}$ . Elle utilise ces variantes pour calculer des distances  $D_L(\mathbf{b}, V(m_i, \theta_j))$  qu'elle agrège en une distance par rapport à la famille:

$$D_L(\mathbf{b}, \mathcal{F}) := \min_j D_L(\mathbf{b}, V(m_i, \theta_j)). \quad (13)$$

### D. Expériences

Toute distance entre les modèles est une variable aléatoire puisque les requêtes sont choisies aléatoirement. Notre protocole effectue 20 mesures de toute distance considérée grâce à 20 échantillons d'entrées indépendantes.

1) *Hypothèses sur le modèle statistiques*: La section III-A1 fait deux hypothèses sur la dépendance statistique entre les prédictions des modèles de la même famille  $\mathcal{F}$  et l'indépendance s'ils proviennent de familles différentes. La figure 2 vérifie expérimentalement ces hypothèses de travail.

Les distances entre deux modèles  $V(m, \theta)$  et  $V(m', \theta')$  pour deux modèles vanillas  $m$  et  $m'$  et toutes les variantes  $(\theta, \theta') \in \Theta \times \Theta$  sont calculées. On obtient ainsi 583 740 combinaisons.

La figure 2 montre l'histogramme de ces valeurs de distance sur 20 bins en rouge. Une indépendance statistique parfaite implique une distance égale à 1. Un faible nombre d'entrées interrogées fait que la distance mesurée s'étend sur une large plage. La sélection des entrées a un impact majeur. Lorsqu'elle est échantillonnée sur  $\mathcal{X}$  (première ligne), la requête peut être une entrée 'facile' correctement classée par n'importe quel modèle. Cela nuit à l'indépendance statistique. Lorsqu'on échantillonne sur  $\mathcal{X}'$  contenant des entrées difficilement classées (deuxième ligne), les distances sont plus proches de un.

La figure montre également l'histogramme des distances entre les modèles appartenant à la même famille couverte par un modèle vanilla  $m$ , que ce soit  $\mathcal{F}(m, \Psi)$  (même type de variation) ou  $\mathcal{F}(m)$  (tout type de variation). Nous observons que deux modèles issus du même type de variation sont généralement plus proches. Il est donc plus facile de détecter ou d'identifier les familles  $\mathcal{F}(m, \Psi)$  que  $\mathcal{F}(m)$ .

2) *Détection*: L'expérience considère toutes les combinaisons d'hypothèses et de modèles mis dans la boîte noire. Il y a 35 modèles vanillas et 1046 variantes. Cela donne 35 familles de type  $\mathcal{F}(m)$  avec une moyenne de 30 membres par famille. Cela représente 1081 cas positifs et 36 754 cas négatifs. Il y a 377 familles de type  $\mathcal{F}(m, \Psi)$  dont 203 avec une taille supérieure à 1, soit 907 cas positifs et 218 536 cas négatifs. Les performances de détection sont évaluées par le taux de vrais positifs (TVP) lorsque le seuil  $\tau$  est fixé pour obtenir un taux de faux positifs (TFP) de 5%.

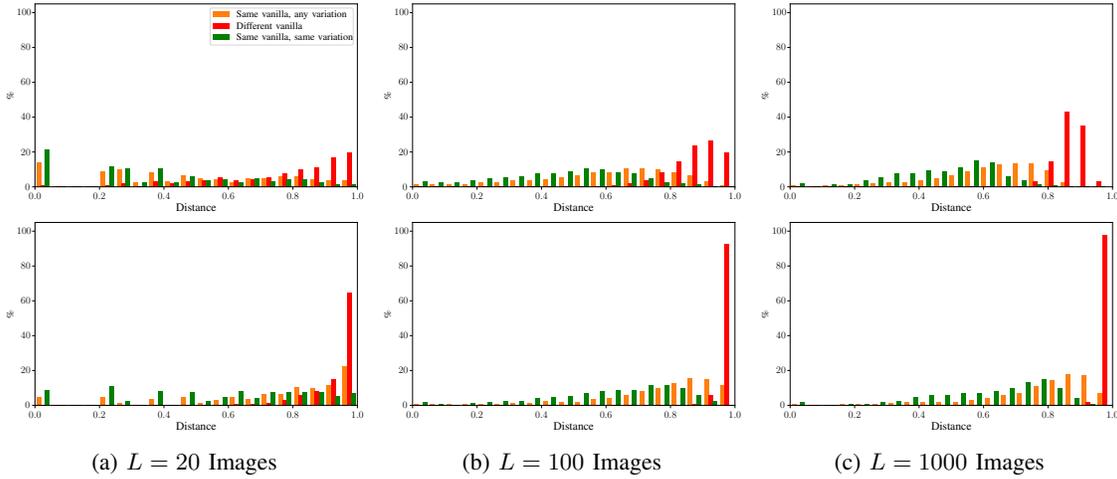


Fig. 2: Histogramme de la distance  $D_L(m_1, m_2)$  si  $(m_1, m_2) \in \mathcal{F}^2(m)$  (orange),  $(m_1, m_2) \in \mathcal{F}^2(m, \Psi)$  (vert), ou  $m_1$  et  $m_2$  sont issus de différents modèles vanillas (rouge). Entrées échantillonnées dans  $\mathcal{X}$  (*haut*) ou dans  $\mathcal{X}'$ -Entropie (*bas*).

a) *Sélection des entrées*: Le tableau I montre le taux de vrais positifs obtenu lorsque la boîte noire ne renvoie que le top-1. Comme prévu, les performances des familles  $\mathcal{F}(m, \Psi)$  sont plus élevées. La sélection Entropie est la meilleure option. Son inconvénient est qu'elle nécessite des statistiques sur les prédictions de nombreux modèles vanillas. En ce qui concerne la tâche de détection, les autres sélections sont à préférer car elles ne nécessitent rien d'autre que les prédictions du modèle vanilla suspecté. Dans la suite, la sélection 30/70 est utilisée pour de nouvelles expériences sur la tâche de détection.

b) *Le modèle délégué*: Alice mesure une distance unique entre la boîte noire et un modèle délégué de la famille  $\mathcal{F}$ . Trois choix de délégué sont proposés en fonction de la distance avec le modèle vanilla de la famille : *Proche*, *Médian*, et *Eloigné*. Par exemple, l'option *Proche* signifie que le délégué est le membre de la famille le plus proche du modèle vanilla:

$$m_d = \arg \min_{m' \in \mathcal{F}} D_L(m, m'). \quad (14)$$

Dans le cas où  $\mathcal{F} = \mathcal{F}(m)$ , le membre le plus proche est  $m$ . Ce n'est pas le cas lorsque  $\mathcal{F} = \mathcal{F}(m, \Psi)$ , car le modèle vanilla  $m$  n'est pas dans cette famille. Rappelons que l'intersection entre deux familles doit être l'ensemble vide.

Le tableau II évalue les trois options. Seules les 180 familles de plus de 3 membres sont considérées ici. Pour les familles plus petites, les trois options donneraient le même délégué.

Le délégué influence grandement les résultats. Le meilleur choix est de sélectionner le délégué comme se trouvant au

TABLE I: TVP pour détecter avec 100 requêtes aléatoires sélectionnées dans  $\mathcal{X}'$ . Le délégué sélectionné est le plus proche de  $m$ . TFP fixé à 5%.

	Aléatoire	50/50	30/70	Entropie
$\mathcal{F}(m)$	$79.4 \pm 2.1$	$89.2 \pm 1.3$	$91.1 \pm 1.5$	<b><math>95.2 \pm 0.5</math></b>
$\mathcal{F}(m, \Psi)$	$85.4 \pm 0.9$	$94.1 \pm 0.7$	$96.6 \pm 0.5$	<b><math>99.8 \pm 0.1</math></b>

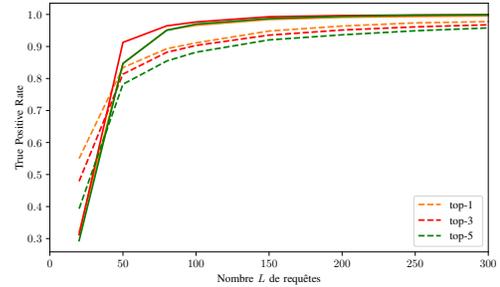


Fig. 3: TVP pour  $(D, \mathcal{F}, k)$  en fonction du nombre de requêtes sélectionnées aléatoirement dans 30/70, TFP = 5%, meilleures options de délégation pour  $\mathcal{F}(m)$  (tiret) et  $\mathcal{F}(m, \Psi)$  (plein).

'centre' de la famille. Cela signifie l'option *Proche* pour la famille  $\mathcal{F}(m)$  et l'option *Médian* pour la famille  $\mathcal{F}(m, \Psi)$ .

c) *Observation du Top-k*: La détection est évaluée pour le top- $k$  dans la figure 3. Les meilleurs résultats sont étonnamment obtenus pour  $k = 1$  dans le tableau III. Notre explication est la suivante. Plus  $k$  est grand, plus l'information est riche. Or, l'information mutuelle empirique est calculée à partir de  $(k+1)^2$  probabilités estimées. Pour un nombre donné de requêtes, moins il y a d'estimations, plus elles sont précises. Le top-1 obtient rapidement de très bons résultats proches de 100%, de sorte que les informations plus riches mais moins bien estimées pour les plus grands  $k$  ne sont pas compétitifs.

En résumé, le TVP atteint 95% pour 160 requêtes dans le top-1, 200 dans le top-3 et 250 dans le top-5.

3) *Identification*: Toutes les conclusions obtenues dans la section précédente sont conservées. Alice a maintenant pour délégué le modèle vanilla  $m$  pour  $\mathcal{F}(m)$  et le modèle *Médian* pour  $\mathcal{F}(m, \Psi)$ . Les images sont échantillonnées avec Entropie tel que défini dans la Sec. III-B2.

a) *Protocole*: Nous divisons la tâche d'identification en deux parties. Premièrement, Alice identifie une famille  $\mathcal{F}(m)$ . Comme expliqué précédemment, elle peut procéder à une

TABLE II: TVP pour détecter et différentes options de délégués avec  $L = 100$  requêtes aléatoires dans 30/70. TFP= 5%.

Délégué	$\mathcal{F}(m)$			$\mathcal{F}(m, \Psi)$		
	top-1	top-3	top-5	top-1	top-3	top-5
Proche	<b>91.1</b> $\pm$ 1.0	<b>90.3</b> $\pm$ 1.2	<b>88.2</b> $\pm$ 0.9	95.5 $\pm$ 0.7	94.4 $\pm$ 0.6	92.8 $\pm$ 0.7
Médian	79.5 $\pm$ 0.3	80.2 $\pm$ 0.2	78.3 $\pm$ 0.2	<b>96.9</b> $\pm$ 0.5	<b>97.8</b> $\pm$ 0.4	<b>96.9</b> $\pm$ 0.5
Eloigné	28.4 $\pm$ 2.3	33.8 $\pm$ 2.7	33.7 $\pm$ 3.5	84.8 $\pm$ 1.3	88.4 $\pm$ 1.0	97.3 $\pm$ 0.9

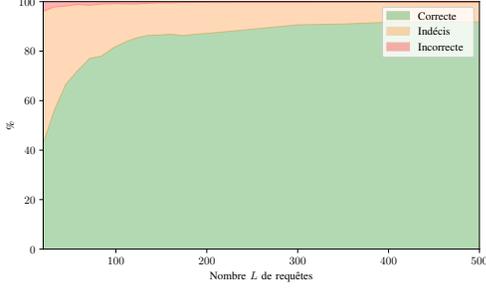


Fig. 4: Distribution de probabilités pour l'identification en fonction du nombre  $L$  de requêtes. Seuil fixé pour avoir un maximum de 5% d'erreurs dans les cas négatifs.

identification ( $\mathcal{H}_i$ ) ou s'abstenir ( $\mathcal{H}_0$ ). En fonction de ce qui se trouve dans la boîte noire, deux réponses correctes sont possibles. Dans le cas négatif où  $b \in \mathcal{F}(m')$  mais  $m' \notin \mathcal{A}$ , la réponse correcte est de s'abstenir. Le seuil  $\tau$  est fixé de manière à ce que la probabilité d'erreur dans un cas négatif soit fixée à 5%. En d'autres termes, Alice minimise son erreur de décision lorsqu'elle doit s'abstenir. À cette fin,  $\mathcal{A}$  est maintenant composé de 30 modèles et les 5 restants sont utilisés pour générer les cas négatifs. Alice calcule les distances entre  $b$  et les 30 modèles vanillas de  $\mathcal{A}$ . Nous répétons ceci 20 fois où les 5 modèles exclus est échantillonné aléatoirement dans  $\mathcal{P}$ . Dans le cas positif où  $b \in \mathcal{F}(m_i)$  et  $m_i \in \mathcal{A}$ , la réponse correcte est de choisir l'hypothèse  $\mathcal{H}_i$ .

Dans la deuxième partie, Alice identifie la variation, sachant qu'elle a fait une identification correcte de la famille  $\mathcal{F}(m)$ . Dans ce cas, Alice doit identifier la variation parmi 6 familles  $\{\mathcal{F}(m, \Psi_j)\}_{j=1:6}$ : random smoothing, trois différents pruning, JPEG, postérisation. Alice calcule donc 6 distances en fonction de leurs délégués et identifie la famille  $i^* = \arg \min_j D_L(b, \mathcal{F}(m, \Psi_j))$ . Aucun seuil n'est nécessaire ici. Pour chaque famille, 20 variantes avec des paramètres aléatoires et conformes à (1) sont créées. Cela conduit à 700 nouveaux modèles testés dans la boîte noire, différents des 1081 modèles considérés jusqu'à présent.

TABLE III: TVP pour  $(D, \mathcal{F}, \mathcal{A} \subsetneq \mathcal{B}, k)$  avec des requêtes aléatoires sélectionnées avec 30/70, TFP = 5%.

Nombre de requêtes		$L = 20$	$L = 50$	$L = 100$	$L = 500$
$\mathcal{F}(m)$	top-1	<b>55.0</b>	<b>83.4</b>	<b>91.1</b>	<b>98.7</b>
	top-3	47.9	81.4	90.3	97.9
	top-5	39.3	78.2	88.2	97.2
$\mathcal{F}(m, \Psi)$	top-1	<b>32.0</b>	84.7	96.9	99.8
	top-3	31.2	<b>91.3</b>	<b>97.8</b>	<b>100</b>
	top-5	29.4	84.7	96.9	<b>100</b>

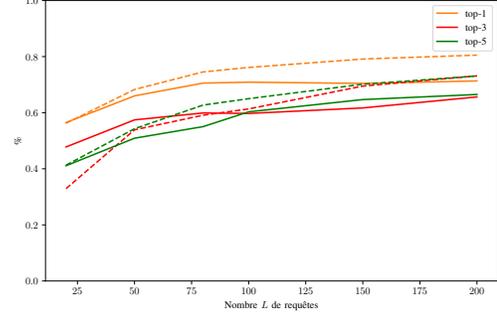


Fig. 5: Taux d'identification correcte pour  $\mathcal{F}(m, \Psi)$  en fonction du nombre de requêtes. Un (en clair) ou deux (en pointillés) délégués par famille.

TABLE IV: Taux d'identification correcte avec des requêtes aléatoires sélectionnées avec Entropie.

Nombre de requêtes		50	100	500
$\mathcal{F}(m)$ délégué = {proche}	top-1	<b>66.7</b>	<b>81.3</b>	<b>91.8</b>
	top-3	22.6	37.6	79.1
	top-5	21.1	48.0	82.0
$\mathcal{F}(m, \Psi)$ delegate = {médian}	top-1	66.0	70.9	73.7
	top-3	57.4	59.7	71.5
	top-5	50.9	60.3	69.8
$\mathcal{F}(m, \Psi)$ delegate = {proche, médian}	top-1	<b>68.3</b>	<b>76.1</b>	<b>82.5</b>
	top-3	53.9	61.4	80.6
	top-5	54.3	65.0	80.5

b) Identifier  $\mathcal{F}(m)$ : Alice identifie presque sûrement la famille  $\mathcal{F}(m)$  de la boîte noire comme le montre la figure 4 et le tableau IV. Elle atteint son taux de réussite maximal à environ 300 requêtes. Au-delà de cette quantité, il n'y a pas d'identification incorrecte mais il reste 10% d'abstention. Ceci est dû au seuillage qui empêche Alice d'être mal classée dans le cas négatif. Si aucun seuillage n'est effectué, le taux de réussite atteint 92,8% en 100 requêtes et 98,2% à 500.

Le nombre de requêtes est plus élevé que pour la détection. A niveau de performance équivalent, 4 fois plus de requêtes sont nécessaires pour l'identification que pour la détection. Néanmoins, l'identification par détection séquentielle nécessiterait en moyenne 3 000 requêtes.

c) Identifier  $\mathcal{F}(m, \Psi)$ : Avec un seul délégué, le tableau IV et la figure 5 montrent une identification difficile. Les variantes éloignées du modèle vanilla sont correctement identifiées. Les variations s'éloignent linéairement de  $m$  sur la figure 1. La principale difficulté provient des variations modifiant légèrement le modèle. Ces variantes sont proches de  $m$ , qui est le centre du cluster  $\mathcal{F}(m)$ . Il est donc difficile de les distinguer. Le composé (13) avec les délégués proche et médian donne une augmentation de 12 points.

d) *Observation du top-k*: Les meilleurs résultats sont obtenus pour  $k = 1$  dans le tableau IV sur chaque tâche, comme pour la détection. Pour la famille  $\mathcal{F}(m)$ , l’information obtenue par top- $k$  a besoin de trop de requêtes pour rattraper le top-1. Pour la famille  $\mathcal{F}(m, \Psi)$ , la différence est plus faible. En effet, le top- $k$  avec  $k \leq 3$  donne des résultats légèrement meilleurs à partir de  $\approx 1.000$  requêtes et plus.

#### IV. BENCHMARK

##### A. Travaux précédents

Depuis les travaux de IP-Guard [2], tous les travaux sur les empreintes traitent d’exemples adverses. Ils commencent avec une petite collection d’entrées (sauf [12] à partir de bruit) et appliquent une attaque précise en boîte blanche comme CW. Il falsifie les exemples adverses pour être proches des frontières de décision, qui sont les signatures d’un modèle.

Deux tendances sont liées à deux applications. La première concerne l’intégrité du modèle. Dans ce scénario, Alice s’assure que Bob a placé son modèle dans la boîte noire sans aucune altération. L’objectif est de trouver une *empreinte fragile* tel que toute modification du modèle vanilla change l’empreinte et devient détectable. Le papier [8] crée des exemples sensibles qui ne sont adverses que pour le modèle vanilla.

La deuxième application est l’*empreinte robuste* telle que présentée jusqu’à présent dans ce papier. Les adeptes de IP-Guard [2] forgent des exemples adverses qui sont plus robustes car ils restent adverses pour toute variation du modèle. L’article [3] utilise les perturbations adverses universelles du modèle vanilla. Le papier [13] introduit le concept d’exemples conférables, soit des exemples adverses qui ne se transfèrent qu’aux variations du modèle ciblé. AFA [5] active le dropout comme substitut bon marché des variantes lors de la création d’exemples adverses. TAFA [4] étend cette idée à d’autres primitives d’apprentissage automatique comme la régression.

Dans cet article, nous pensons que l’utilisation d’images non modifiées est suffisante. Nous avons résolu le problème d’empreinte sans avoir besoin de s’appuyer sur une technique modifiant les images pour les rapprocher des frontières. En effet, il est assez simple de créer des exemples adverses, mais les doter de spécificités supplémentaires (fragiles ou robustes aux variations) est complexe. Il se trouve que tous les articles considèrent de petites dimensions d’entrée, comme MNIST ou CIFAR (images de 32 pixels); aucun d’entre eux n’utilise ImageNet (224 pixels), à l’exception de IP-Guard [2]. De plus, aucun article ne considère que les entrées peuvent être modifiées par une défense (afin d’éliminer une perturbation adverse avant d’être classées) ou détectées comme adverses [14].

##### B. Benchmark

IP-Guard [2] est le seul travail qui s’est montré efficace sur des entrées de grande taille comme celles d’ImageNet. Il s’appuie sur plusieurs attaques en boîte blanche pour créer des exemples adverses. Les meilleurs résultats démontrés dans l’article sont obtenus avec l’attaque CW [15]. Nous utilisons plutôt BP [16] qui présente des performances similaires

TABLE V: Comparaison des taux de vrais positifs pour la tâche (D,  $\mathcal{F}(m)$ , 1). TFP fixé à 5%.

Empreinte utilisée	Paramètre	L requêtes	
		100	200
IP-Guard [2]	BP [16] & 50 iter.	66.9	72.7
FBI	Aléatoire	79.4	91.5
	30/70	91.1	97.4
	Entropie	<b>95.2</b>	<b>97.6</b>

tout en étant beaucoup plus rapide (seulement 50 itérations). L’implémentation de BP provient de GitHub<sup>2</sup>.

Le tableau V compare les performances avec 100 et 200 requêtes et les observations top-1. Toute sélection des entrées bat IP-Guard [2]. Certaines variations sont plus faciles à détecter (‘precision’, ‘pruning’) où les deux méthodes sont à égalité. Au contraire, le random smoothing, une variation jamais considérée dans la littérature, est plus difficile. IP-Guard [2] est basée sur l’élaboration d’exemples adverses proches de la frontière qui sont fortement ‘‘écrasés’’ par le random smoothing. Le fait de ne pas s’appuyer sur des exemples adverses semble être un net avantage ici. Notre méthode offre une plus grande stabilité dans les résultats: aucune variation ne descend le TVP en dessous de 85%.

#### V. CONCLUSION

Des empreintes précises et efficaces pour les modèles précieux sont importantes. Cet article démontre qu’une telle demande peut être satisfaite en utilisant des entrées authentiques et non modifiées, non seulement dans la tâche classique de détection, mais aussi dans la nouvelle tâche d’identification qui a été introduite. Cela implique qu’un modèle en boîte blanche n’est plus nécessaire pour calculer les empreintes.

Nous en tirons les leçons suivantes:

La tâche de détection est résoluble en une centaines d’entrées mais le schéma n’est pas itératif et la sélection est moins cruciale. De manière surprenante, l’observation de sorties plus riches n’apporte pas de gain dans cette configuration.

La tâche d’identification est plus complexe que la détection, mais seul un faible nombre de requêtes supplémentaires est nécessaire. Notre identification est beaucoup plus efficace que la recherche séquentielle naïve. Une des limites de notre

travail est qu’il ne peut pas traiter les classifieurs dont la précision est presque parfaite. Cela se produirait pour des classifications trop faciles, où la valeur des modèles est plus faible et où l’empreinte est un enjeu moins critique. Nous nous attendons néanmoins à ce que les futurs modèles et applications soient des tâches complexes, où atteindre de bons niveaux de précision restera un défi.

#### REFERENCES

- [1] S. Wang and C.-H. Chang, ‘‘Fingerprinting deep neural networks - a deepfool approach,’’ in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.

<sup>2</sup>Boundary Projection’s GitHub : <https://github.com/hanwei0912/walking-on-the-edge-fast-low-distortion-adversarial-examples>

- [2] X. Cao, J. Jia, and N. Z. Gong, "Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. Association for Computing Machinery, 2021.
- [3] Z. Peng, S. Li, G. Chen, C. Zhang, H. Zhu, and M. Xue, "Fingerprinting deep neural networks globally via universal adversarial perturbations," 2022.
- [4] X. Pan, M. Zhang, Y. Lu, and M. Yang, "Tafa: A task-agnostic fingerprinting algorithm for neural networks," in *European Symposium on Research in Computer Security*. 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," *Computer Communications*, vol. 150, pp. 488–497, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S014036641931686X>
- [6] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.
- [7] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.
- [8] Z. He, T. Zhang, and R. Lee, "Sensitive-sample fingerprinting of deep neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [9] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 135–147.
- [10] R. Tamir and N. Merhav, "The mmi decoder is asymptotically optimal for the typical random code and for the expurgated code," 2020. [Online]. Available: <https://arxiv.org/abs/2007.12225>
- [11] E. Abbe and L. Zheng, "Linear universal decoding for compound channels," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 5999–6013, 2010.
- [12] S. Wang, P. Zhao, X. Wang, S. Chin, T. Wahl, Y. Fei, Q. A. Chen, and X. Lin, "Intrinsic examples: Robust fingerprinting of deep neural networks," in *British Machine Vision Conference (BMVC)*, 2021.
- [13] N. Lukas, Y. Zhang, and F. Kerschbaum, "Deep neural network fingerprinting by conferrable adversarial examples," in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=VqzVhqxkjH1>
- [14] A. Kherchouche, S. A. Fezza, W. Hamidouche, and O. Déforges, "Detection of adversarial examples in deep neural networks with natural scene statistics," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.
- [15] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 39–57.
- [16] B. Bonnet, T. Furon, and P. Bas, "Generating Adversarial Images in Quantized Domains," *IEEE Transactions on Information Forensics and Security*, 2022. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-03467692>