Taxonomie des attaques sur les méthodes d'apprentissage automatique

Tom Djaaleb*, Mohamed-Lamine Messai*

*Univ Lyon, Univ Lyon 2, ERIC
{tom.djaaleb, mohamed-lamine.messai}@univ-lyon2.fr

Résumé. L'apprentissage automatique gagne de plus en plus de terrains d'applications. Différentes méthodes existent et qui permettent la construction de modèles à des fins d'aide à la décision. Néanmoins, les modèles d'apprentissage automatique sont vulnérables et exposés à différents types d'attaques de sécurité durant le processus d'apprentissage des modèles et après leur déploiement. Par conséquent, ces menaces doivent être, dans un premier temps, identifiées, définies et classées afin que, dans un deuxième temps, proposer des mesures de défense pour faire face à ces menaces. Dans cet article, Nous nous sommes intéressés à l'étude des menaces pouvant toucher un processus d'apprentissage automatique. Nous présentons une classification des menaces autour de l'objectif, la connaissance et la capacité de l'attaquant. Ensuite, nous montrons quelques exemples d'attaques sur des applications utilisant l'apprentissage automatique.

1 Introduction

L'apprentissage automatique ou machine (*Machine Learning (ML)*) est un sous-domaine de l'intelligence artificielle. Le ML inclut un ensemble d'algorithmes permettant de créer automatiquement des modèles à partir de données exemples. Ces données, utilisées dans la phase d'entraînement sont appelées les jeux de données d'apprentissage. Traditionnellement, un programme informatique effectue une tâche en suivant des instructions précises, et donc systématiquement de la même façon. Par contre, un système d'apprentissage automatique ne suit pas d'instruction, mais apprend à partir de l'expérience. Par conséquent, ses performances s'améliorent au fil de son entraînement sur le jeu de données. La finalité du modèle de ML est d'imiter un comportement semblable au cerveau humain en termes de prise de décision afin d'automatiser des tâches. Ces dernières années, grâce à des capacités de calcul beaucoup plus puissantes et des ensembles de données beaucoup plus importants pour l'entraînement des modèles, les technologies d'apprentissage automatique, en particulier les réseaux neuronaux artificiels et les architectures d'apprentissage profond, ont fait des progrès considérables. L'apprentissage automatique a de nombreux domaines d'application comme la détection des spams et des logiciels malveillants en cybersécurité, la classification d'images et la reconnaissance d'objets pour le contrôle des véhicules autonomes, le diagnostic médical, la reconnaissance vocale, ... etc.

Avec le succès impressionnant de l'application du ML dans des domaines de plus en plus nombreux, des faiblesses de sécurité sont présentes dans les étapes des algorithmes d'appren-

tissage ce qui peut rendre les modèles générés inutilisables. En raison de ces faiblesses, un système d'apprentissage automatique est vulnérable à divers types d'exploitations adverses qui peuvent compromettre l'ensemble du système. En fait, un processus d'apprentissage automatique typique, qui consiste en la collecte de données, l'extraction de caractéristiques, l'entraînement du modèle, la prédiction et le réentraînement du modèle, est vulnérable aux attaques malveillantes à chaque étape. Nous nous intéressons dans ce travail à l'étude des menaces et à l'identification et la classification des différents types d'attaques sur l'apprentissage automatique.

Le reste de l'article est organisé comme suit : la Section 2 explique brièvement le processus d'apprentissage automatique en donnant les sources de vulnérabilités et présente les travaux antérieurs sur la sécurité de l'apprentissage automatique. Ensuite, la Section 3 présente une taxonomie des menaces en présentant notre classification axée sur les dépendances entre les objectifs de l'attaquant, sa connaissance sur le système ML et les attaques qu'il peut exécuter. Dans la Section 4, nous présentons quelques attaques sur des exemples d'applications. La section 5 conclut l'article en donnant quelques perspectives.

2 Apprentissage automatique

La Figure 1, schéma utilisé par McGraw et al. (2020) en amont de leur travaux sur la sécurité du ML, représente toutes les étapes du processus d'apprentisage automatique. Les cellules rectangulaires bleues 2, 4, 5, 8 sont des processus ("des actions") alors que les cellules ovales jaunes 1, 3, 6, 7 et 9 représentent des objets ou des ensembles d'objets. Pour comprendre d'où proviennent les vulnérabilités de l'apprentissage automatique, il est nécessaire de définir formellement son fonctionnement. Le livre "*An Introduction to Statistical Learning*" de James et al. (2021) propose une définition claire et précise du problème. Soit $X = (X_1, ..., X_p)$ un ensemble de variables prédictives et Y l'ensemble des réponses associées. Il existe une fonction f qui établit la relation entre X et Y :

$$Y = f(X) + \epsilon \tag{1}$$

Le terme ϵ représente l'erreur irréductible et est indépendant de X. Cette relation n'est pas toujours connue, l'apprentissage automatique cherche donc à l'estimer pour plusieurs raisons :

— La **prédiction** : généralement, X est connu et accessible mais la réponse Y ne l'est pas. On peut donc utiliser l'estimation $\widehat{f}$ pour prédire $\widehat{Y}$, l'estimation de la réponse.

— L'**inférence** : l'estimation de f permet parfois de mieux comprendre la relation entre Y et les variables explicatives.

L'estimation de f se fait en appliquant des méthodes d'apprentissage statistique sur les données déjà à notre disposition. Lorsque l'apprentissage est supervisé, chaque observation i est constituée de j prédicteurs x_{ij} et de la réponse associée y_i. Si l'apprentissage est non supervisé, seul l'ensemble des X est disponible, les réponses associées aux observations sont inconnues.

Les deux principales raisons de la vulnérabilité des systèmes d'apprentissage automatique, outre les problèmes de sécurité classique, sont les suivantes :

— $\widehat{f}$ est **dépendante** de X^{train} l'ensemble des données d'entrainement : une modification de l'ensemble modifie également la fonction.

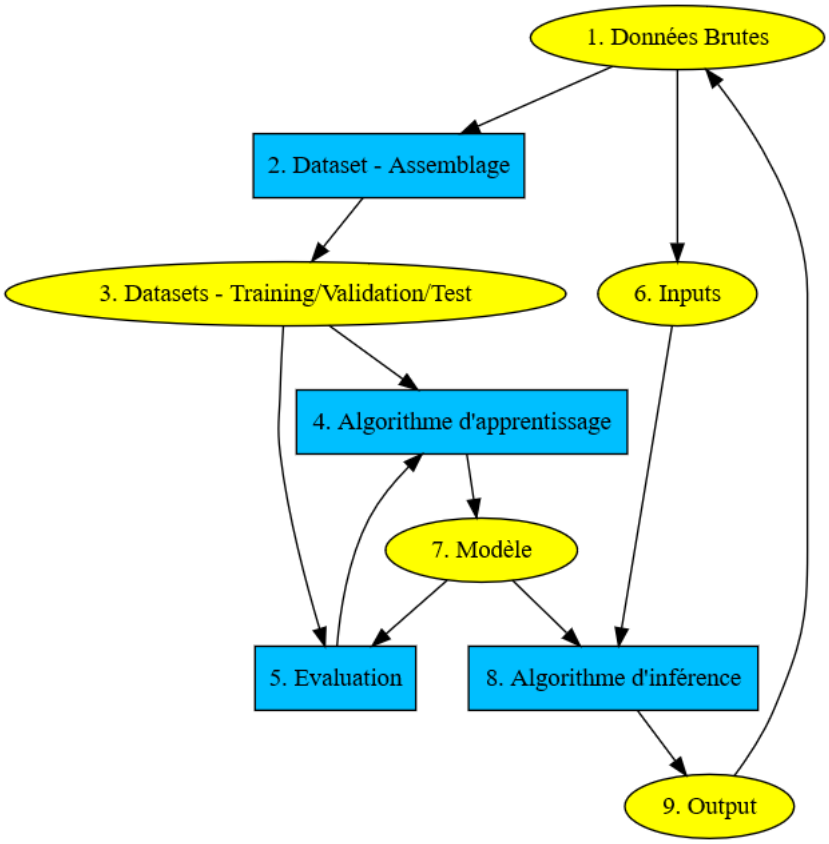

FIG. 1 – Framework de l'apprentissage automatique.

— $\widehat{f}$ est une **estimation** de la relation réelle entre le variables et la réponse : il existe des paires (X'_i, y'_i) telles que

$$\widehat{f}(X'_i) \neq y'_i \tag{2}$$

Plusieurs travaux antérieurs présentent les menaces et les attaques sur l'apprentissage automatique. Barreno et al. (2006) étaient les premiers à s'intérésser aux vulnérabilités des méthodes d'apprentissage automatique. Ils ont proposé une première classification des attaques sur le ML en 2006. Leur classification a été reprise et étendue par Huang et al. (2011) et par Biggio et al. (2013b). Plus récemment, le travail Biggio et al. (2013b) est repris et complété par Xiao et al. (2015) et Muñoz-González et Lupu (2019). La Table 1 présente un résumé de chaque travail antérieur ainsi que l'année de publication.

Dans ce travail, nous enrichissons la classification de Biggio et al. (2013b) en ajoutant la notion de dépendance entre les caractéristiques d'une attaque. Cela permet de mieux comprendre les possibilités de l'attaquant dans la construction de son attaque et ainsi déterminer la faisabilité de celle-ci.

Publication	Résumé
Barreno et al. (2006)	Les auteurs proposent un premier modèle des attaques et discutent ensuite des défenses possibles à explorer. Ils présentent une attaque par empoisonnement simple.
Huang et al. (2011)	Le modèle de Barreno et al. (2006) est repris. Étude de cas sur deux systèmes : *SpamBayes* et *Anomalous Traffic Detection*. Introduction de la confidentialité des données dans le ML.
Biggio et al. (2013b)	Généralisation du modèle de Huang et al. (2011). Introduction des connaissances de l'attaquant. Étude de cas sur un modèle de clustering.
Xiao et al. (2015)	Reprise du modèle de Biggio et al. (2013b). Application à la sélection de variables et expérimentation d'une attaque par empoisonnement.
Muñoz-González et Lupu (2019)	Extension du modèle de Biggio et al. (2013b). Optimisation des attaques par empoisonnement. Présentation de propriétés sur les attaques par empoisonnement et par évasion. Proposition de pistes de défenses.
Tabassi et al. (2019)	Proposition d'un nouveau modèle. Inclusion des défenses et des conséquences.

TAB. 1 – Publications antérieures.

3 Taxonomie des menaces

Dans cette section, nous présentons le modèle de l'attaquant sur les systèmes d'apprentissage automatique. La stratégie de l'attaquant est représentée autour des trois points clés : son **Objectif**, ses **Connaissances** et ses **Capacités**.

La Figure 2 représente le modèle de l'attaquant. La capacité de l'attaquant représente la phase attaquée (Phase d'entrainement ou lorsque le modèle est actif), ses connaissances sont modélisées en cinq points qui sont précisés dans la section 3.2 et son objectif est défini par la règle de sécurité enfreinte ainsi que si l'attaque est ciblée ou non.

Bien que ces trois pilliers définissent une attaque sur un modèle d'apprentissage automatique, ils ne sont pas indépendants. En effet, les connaissances que l'attaquant possède sur le modèle sont essentielles pour que celui-ci puisse déterminer l'objectif de son attaque, ainsi que son champ d'action. La capacité influe également sur l'objectif de l'attaquant, notamment sur le type de sécurité enfreint, celui-ci pourra être limité s'il n'a pas accès à la phase d'entraînement. Nous détaillons par la suite chaque point.

3.1 Objectif de l'attaquant

Xiao et al. (2015) caractérisent l'objectif de l'attaquant en deux catégories ; la spécificité et la règle de sécurité enfreinte. La Figure 3 illustre les deux catégories.

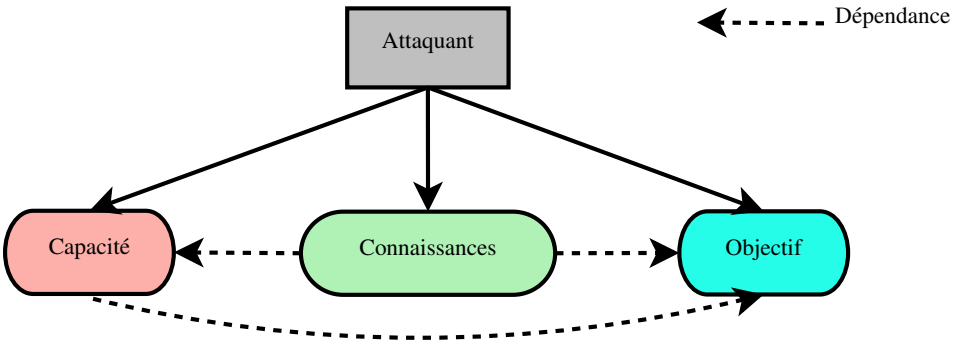

FIG. 2 – Modèle de l'attaquant.

3.1.1 Spécificité

Barreno et al. (2006) puis Huang et al. (2011) introduisent la spécificité comme une des trois caractéristiques principales de l'attaquant. Elle sépare les attaques ciblées et les attaques dites "non discriminante" (*Indiscriminate Attack*). Dans le cadre d'une attaque ciblée, l'attaquant vise un type d'échantillon particulier, par exemple une classe précise. L'attaquant aura pour objectif de maximiser l'erreur pour cette classe sans prendre en compte les autres. Une attaque non discriminante cherche à dégrader les performances du modèle en maximisant le taux d'erreur global du modèle.

3.1.2 Sécurité

La norme ISO/CEI 27001 publiée en 2005 et révisée en 2013 définit la sécurité des systèmes d'informations en trois points fondamentaux : Confidentialité, Disponibilité, Intégrité. Xiao et al. (2015) ainsi que Muñoz-González et Lupu (2019) intègrent ces trois règles dans leur modèle des attaques sur les systèmes ML.

Confidentialité On parle de "*Privacy Attack*" ou "*Confidentiality Attack*" lorsque l'attaquant a pour objectif d'obtenir des informations confidentielles sur le modèle ou bien sur les données d'entraînement (qui peuvent être des données personnelles d'utilisateurs réels).

Disponibilité Les attaques visant à compromettre la disponibilité du modèle cherchent à le rendre inutilisable. Cela se produit généralement en modifiant le modèle pour que celui-ci prédise un trop grand nombre de "faux positifs".

Intégrité Les "*Integrity Attack*" n'ont pas pour objectif de rendre inutilisable le modèle mais que celui-ci produise des erreurs pour fausser les utilisateurs dans leurs prises de décisions.

Tabassi et al. (2019) dans leur taxonomie ne considèrent pas que l'infraction d'une de ces trois règles est une caractérisique de l'attaque mais plutôt une conséquence de l'attaque. L'attaquant ayant généralement choisi quelle règle il souhaite enfreindre, nous préférons définir le

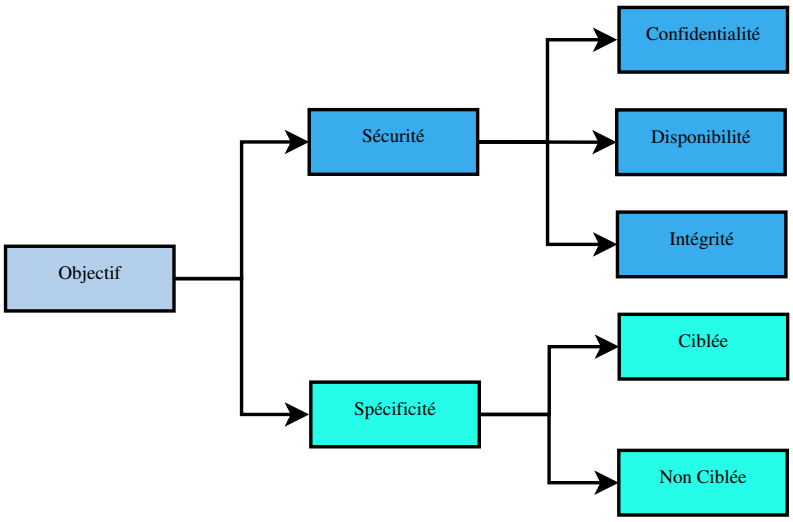

FIG. 3 – Objectif de l'attaquant.

type d'infraction comme un objectif de l'attaquant plutôt que comme une conséquence de ses actions. Muñoz-González et Lupu (2019) proposent également une troisième catégorie pour définir l'objectif de l'attaquant, la spécificité de l'erreur. Cette caractéristique n'étant propre qu'à un nombre restreint de modèles, nous ne la prendrons pas en considération.

3.2 Connaissances de l'attaquant

Les informations détenues par l'attaquant sur le modèle jouent un rôle essentiel dans l'élaboration de sa stratégie. Muñoz-González et Lupu (2019) définissent ces connaissances en cinq composantes :

— Le jeu de données d'entraînement $\mathcal{D}_{tr}$
— L'ensemble des variables explicatives $\mathcal{X}$
— L'algorithme d'apprentissage $\mathcal{M}$
— La fonction objective de l'algorithme $\mathcal{L}$
— Les paramètres de l'algorithme w

L'attaquant peut connaître tout ou une partie de ces cinq paramètres, cependant, Barreno et al. (2006) assument que l'algorithme $\mathcal{M}$ est connu de tous. Dans la continuité de cette idée, Huang et al. (2011) citent le principe de Kerckhoffs en expliquant que la sécurité d'un système ne doit pas reposer sur des attentes irréalistes de confidentialité. L'état de l'art propose deux visions pour étudier les connaissances de l'attaquant :

— Connaissances limitées (*limited knowledge*) vs connaissances parfaites (*perfect knowledge*)
— Boite noire (*black box*) vs boite grise (*grey box*) vs boite blanche (*white box*)

La première vision assume que l'attaquant connaît toujours au moins une des cinq informations présentées ci-dessus (connaissances limitées) et qu'il connaît parfaitement le modèle s'il

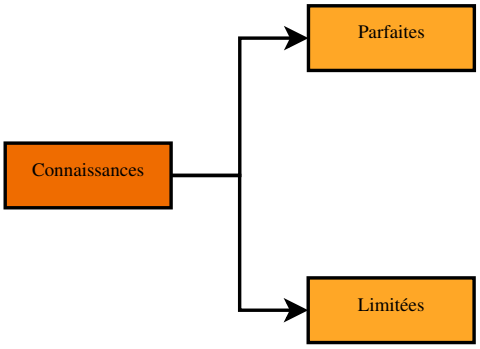

FIG. 4 – Connaissances de l'attaquant.

détient les cinq informations. La deuxième vision est une extension de la première dans laquelle on accepte d'étudier le cas où l'adversaire n'a aucune information sur le modèle (*Black Box*), ce qui est rarement le cas dans une situation réelle, c'est pourquoi dans notre taxonomie, nous décidons de classer les connaissances en deux catégories seulement (Figure 4). Dans cette deuxième vision, *Grey Box* signifie que l'attaquant a des connaissances limitées, et *White Box*, celui-ci a des connaissances parfaites. Les attaques avec connaissances limitées sont les plus fréquentes, la confidentialité des données étant aujourd'hui un enjeu majeur, $\mathcal{D}_{tr}$ est très souvent inconnu. Les informations possédées par l'attaquant sont généralement les paramètres $\mathcal{M}$, $\mathcal{X}$ et $\mathcal{L}$. Dans cette configuration, l'attaquant peut chercher un échantillon $\hat{\mathcal{D}}_{tr}$ statistiquement proche de $\mathcal{D}_{tr}$ pour obtenir une estimation $\hat{w}$ des paramètres du modèle. Tabassi et al. (2019) expliquent que si ce modèle de substitution est suffisament ressemblant au modèle original, les connaissances de l'attaquant deviennent "parfaites". Il est rare que l'attaquant ait une parfaite connaissance du modèle, mais ce type d'attaque est étudié dans la littérature car dans cette configuration, l'attaquant peut attaquer le modèle de façon très performante (celui-ci peut tester son attaque en amont avant de la déployer). Implémenter une défense contre les attaques *White Box* renforce considérablement la résistance du modèle puisque celles-ci sont d'ordinaire, optimisées. Cependant, il existe de rare cas où trouver une attaque optimale est difficile dû à la complexité de certains algorithmes de décision, même avec une connaissance parfaite du modèle.

3.3 Capacité de l'attaquant

La capacité de l'adversaire définit comment et dans quelle mesure l'attaquant peut contrôler le processus d'apprentissage (Biggio et al. (2013b)). On retrouve également le terme d'"influence" pour définir la capacité de l'attaquant (Barreno et al. (2006) Huang et al. (2011)). Lors de la mise en place d'un modèle d'apprentissage automatique, deux phases sont vulnérables aux attaques malveillantes : la phase d'entraînement et la phase d'inférence (Figure 5). Une attaque pendant la phase d'apprentissage est dite "causale" (*Causative Attack*), la caractéristique principale de ce type d'attaque est qu'elles modifient la structure même de l'algorithme pour que celui-ci produise des erreurs de classifications lors de la mise en production. Les attaques "exploratoires" (*Exploratory Attack*) surviennent lors de la phase d'inférence, lorsque le modèle

est mis en production. Ce type d'attaque ne modifie pas le modèle mais cherche plutôt à le leurrer ou à le rendre inutilisable.

3.3.1 Attaque Causale

Les attaques causales interviennent pendant la phase d'entraînement du modèle, on appelle aussi ce type d'attaque "empoisonnement". Les attaques par empoisonnement sont découpées en trois catégories selon Tabassi et al. (2019) :

— Injection de données
— Manipulation des données
— Corruption du modèle

L'injection de données revient à ajouter des échantillons malveillants dans l'ensemble des données d'entraînement pour dégrader les performances de l'algorithme (Saadi et al. (2022)). Dans le cadre d'une attaque ciblée, l'injection permet à l'attaquant d'introduire des *backdoor* pour pouvoir les utiliser lors de la mise en production et d'éviter la détection. La manipulation des données est une extension de l'injection puisqu'elle permet également à l'attaquant de pouvoir modifier/supprimer les échantillons d'entraînement. Cela peut être une modification des données en entrée (X_i) ou bien des labels (y_i). La corruption du modèle (*Logic Corruption*) est l'attaque la plus difficile à mettre en place mais la plus "puissante". L'attaquant cible le modèle d'apprentissage lui-même, il modifie ainsi le processus d'apprentissage. Ce type d'attaque peut survenir à l'entraînement initial ou alors pendant les phases de réentrainement si le système le permet. Il arrive que dans certains cas, les données d'entraînements soient volées avec des attaques informatiques plus classiques. Cela ne modifie pas le modèle mais viole la confidentialité des données, l'attaquant pourra utiliser les informations récoltées pour créer un modèle de substitution et réaliser une attaque exploratoire.

3.3.2 Attaque Exploratoire

Lors d'une attaque exploratoire, l'attaquant n'a pas accès au modèle (il peut tout de même connaître des informations sur ce dernier). Plusieurs possibilités s'offrent à l'adversaire, celui-ci peut dans un premier temps déstabiliser le modèle en envoyant des données en entrée qui produiront des prédictions "faussement positives" (Shi et al. (2018)). Dans le cadre d'un IDS (*Intrusion Detection System*) par exemple, le système alertera sans cesse d'une attaque alors que la situation est normale, ce qui rendra la modèle inutilisable. L'attaquant peut également essayer de traverser le système en trouvant une configuration de variables en entrée qui renvoient une prédiction "faussement négative". Lorsque le modèle est connu, cela peut se faire en analysant le modèle et donc en une seule étape, mais dans le cas où l'adversaire a des connaissances limitées sur le système, il peut trouver cette configuration de manière itérative en ajustant les paramètres à chaque étape. Pour ces types d'attaques, on parle généralement d'attaque par "évasion". Les attaques exploratoires posent également des problèmes de confidentialité. L'attaquant peut, avec un jeu de données en entrées et les prédictions du modèle associées, déduire des informations sur le modèle comme sa nature (paramétrique ou non) ou bien les paramètres choisis par le constructeur (par exemple la fonction objective associée). Ces informations peuvent être utilisées pour créer un modèle de substitution et donc avoir une meilleure connaissance du modèle original. Ce nouveau modèle sera utilisé pour trouver les

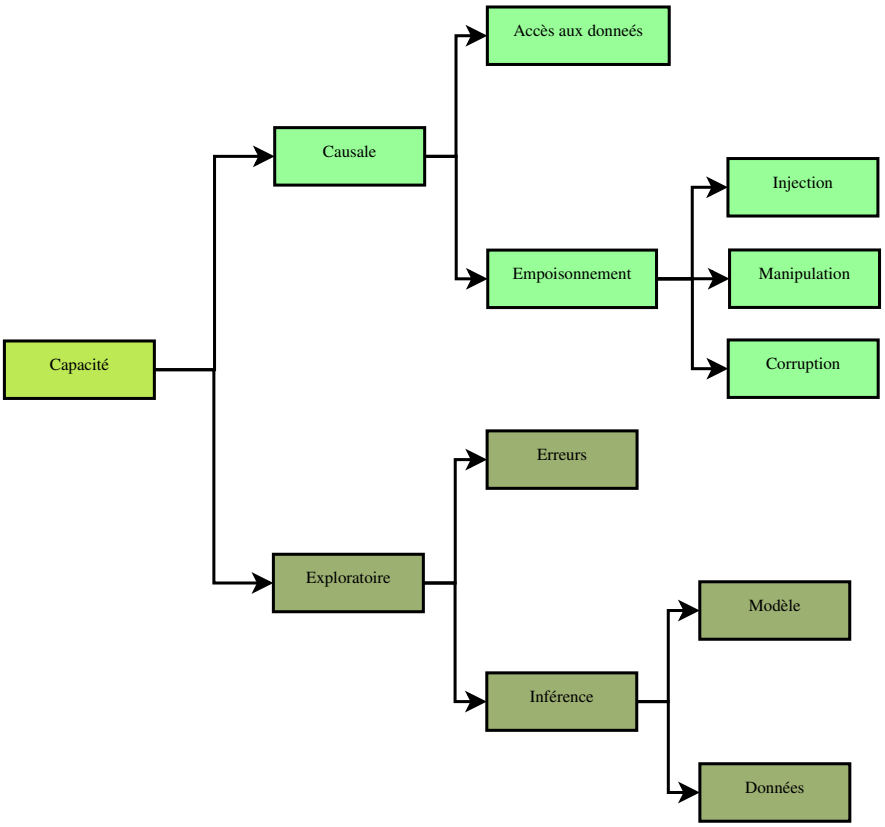

FIG. 5 – Capacité de l'attaquant.

entrées optimales qui permettent de créer des erreurs de classifications. Le modèle peut également détenir des informations sur le jeu de données d'apprentissage et des attaques peuvent être construites pour récupérer le jeu de données d'entrainement (Shokri et al. (2017)).

Les trois caractéristiques présentées précédemment sont dépendantes. Plus particulièrement, les connaissances influent sur l'objectif et la capacité de l'attaquant, de même que la capacité de l'attaquant influe sur son objectif.

Dans le cas où $\mathcal{D}_{tr}$ est connu, l'attaquant n'aura pas besoin de réaliser une attaque exploratoire pour retrouver le jeu de données d'entraînement (Inférence), on peut donc considérer que ce type d'attaque n'est pas une menace pour notre modèle. En revanche, si l'attaquant connaît $\mathcal{D}_{tr}$, deux possibilités sont retenues : les données sont publiques, il n'y a pas de problème de confidentialité ; les données sont privées, il y a une violation de la confidentialité des données. Dans le deuxième cas, en plus de la sécurité enfreinte par l'attaque en question, il faudra aussi considérer qu'il y a eu des failles dans la sécurité des données. Comment l'attaquant a-t-il pu avoir accès aux données ? Méthode classique ou inférence par attaque exploratoire ? Lorsque l'attaquant ne connais ni $\mathcal{D}_{tr}$ ni $\mathcal{X}$, cela devient très compliqué pour lui de

générer des exemples contradictoires : premièrement si le format des données empoisonnées n'est pas le bon, celles-ci pourront tout simplement être rejetées par l'algorithme ; deuxièmement, comment déterminer la pertinence des exemples ? seront-ils assez biaisés pour baisser les performances du modèle. On peut donc se demander si une attaque par empoisonnement est vraiment à craindre lorsque ces informations manquent pour l'attaquant.

La Figure 5 illustre la classification des différentes attaques et la Table 2 présente un résumé du modèle de l'attaquant.

Objectif		
Spécificité		
	Ciblée	L'attaquant ne se préocuppe que d'un échantillon de données précis.
	Non Discriminante	Maximise l'erreur globale du modèle.
Sécurité		
	Confidentialité	Obtention d'informations secrètes comme des données personnelles.
	Disponibilité	Tente de rendre inutilisable le modèle en dégradant drastiquement ses performances.
	Intégrité	Cherche à tromper le modèle dans sans remettre en cause son état de marche.
Connaissances		
Limitées		L'attaquant a des connaissances restreintes parmi $\mathcal{D}_{tr}$, $\mathcal{X}$, $\mathcal{M}$, $\mathcal{L}$, w.
Parfaites		L'attaquant a toutes les informations concernant le modèle.
Capacité		
Causale		
	Empoisonnement	Injecte ou modifie des données pour perturber l'entrainement du modèle.
	Accès aux données	Vole les données d'entraînement.
Exploratoire		
	Évasion	Envoie des données en entrée qui provoquent des erreurs de classifications.
	Inférence	Déduit les paramètres du modèle ou les données d'entraînement en analysant les sorties du modèle.

TAB. 2 – Résumé du modèle de l'attaquant.

4 Quelques exemples d'attaques

Dans cette section, nous présentons trois exemples d'attaques bien connues dans la littérature. Ces exemples permettent d'illustrer la diversité des menaces qui plannent sur l'appren-

tissage automatique. Il existe bien d'autres attaques appliquées à différentes applications ou différents modèles.

4.1 Attaque par empoisonnement

Goodfellow et al. (2015) proposent une méthode rapide pour générer des exemples contradictoires : *Fast Gradient Sign Method* (FGSM). Leurs travaux traitent de la classification d'images. Cette méthode consiste a ajouter une quantité linéaire de bruit inperceptible à l'image et à faire en sorte que le modèle la classe de manière incorrecte. L'exemple contradictoire est généré à partir d'une image des données d'entraînement, à laquelle on ajoute une constante ϵ multipliée par le signe du gradient. Plus ϵ est grand, plus le modèle sera perturbée, mais la perturbation sera également plus visible. La formule pour générer les exemples contradictoires est la suivante :

$$\eta = \epsilon \times sign(\nabla_x J(\theta, x, y)) \tag{3}$$

Avec η la perturbation finale ajoutée à l'image de départ, ϵ le montant de la perturbation, J la fonction de coût, θ les paramètres du modèles, x les données en entrées et y la réponse associée. Une fois les exemples générés, ils sont introduits dans l'ensemble des données d'entraînement et vont perturbés la construction du modèle. Le modèle sera biaisé et ses performances seront dégradées.

Bien que cette méthode ne soit pas la plus efficace, c'est l'une des plus rapide, elle est donc toujours très utilisée pour tester la robustesse des modèles face aux exemples contradictoires. Des méthodes plus efficaces ont été proposées par la suite comme l'attaque C & W du nom de ses auteurs par Carlini et Wagner (2017).

4.2 Attaque exploratoire : Erreur de classification

Biggio et al. (2013a), en plus de proposer un modèle de l'attaquant, publie également un article qui traite des attaques exploratoires visant à produire des erreurs de classifications. Les auteurs construisent une attaque basée sur la descente de gradient pour déterminer x^*, l'echantillon optimal qui "échappe" au modèle (mauvaise classification). Leurs tests sont effectués sur deux systèmes, le premier est un algorithme de reconnaissance d'image et le deuxième, un algorithme ayant pour but de détecter les malwares dans les documents PDF. Leurs expérimentations montrent que les méthodes d'apprentissage populaires comme les SVM ou les réseaux de neurones sont sensibles à ce genre d'attaque, même lorsque les connaissances de l'attaquant sont limitées.

4.3 Attaque exploratoire : Inférence

Shokri et al. (2017) construisent une attaque permettant de déterminer si une observation a été utilisée ou non pour construire un modèle. Cette attaque exploite le fait que les modèles d'apprentissage automatique se comportent souvent différement lorsqu'ils ont en entrée des données sur lesquels ils ont été entraînés comparé à des données qu'ils rencontrent pour la première fois. L'attaque se base uniquement sur les sorties du modèle attaqué. Le modèle d'attaque est une collection de modèles, un pour chaque classe de sortie du modèle cible. Cela augmente la précision de l'attaque car le modèle cible produit différentes distributions sur ses

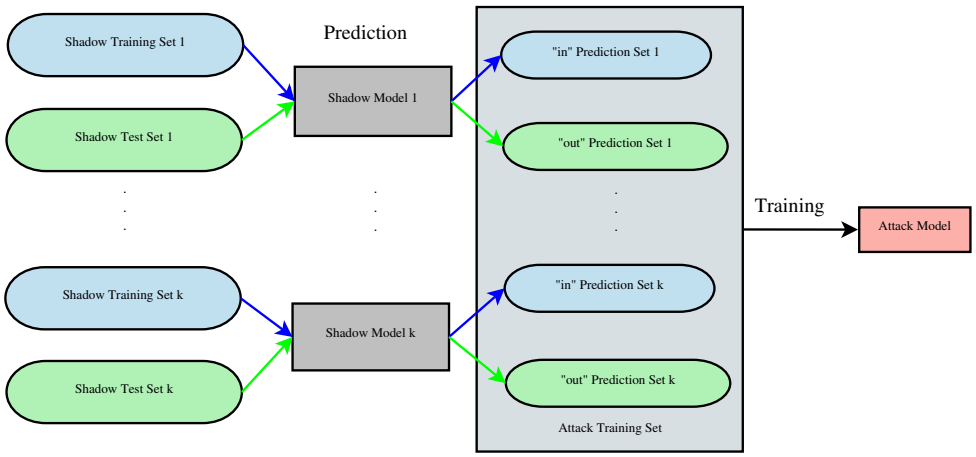

FIG. 6 – Création du modèle d'attaque par inférence.

classes de sortie en fonction de la classe réelle de l'entrée. k modèles "fantômes" f^i_{shadow} (un pour chaque classe) sont construits et destinés à se comporter de manière similaire au modèle cible. Chaque modèle fantôme i est entraîné sur un jeu de données $D^{train}_{shadow^i}$ de distribution similaire aux données d'entraînement du modèle ciblé. Les sorties de ses modèles sont "in" si l'observation fait partie du jeu de données d'entrainement et "out" si ce n'est pas le cas. Plus il y a de modèles fantômes, plus l'attaque sera performante. Pour finaliser l'attaque, les sorties de chaque modèle fantôme sont associées à leurs données en entrée pour créer le jeu de données d'entrainement du modèle final. C'est ce modèle qui sera utilisé pour attaquer le modèle ciblé et déterminer les observations utilisées pour son entraînement. La construction de cette attaque est schématisée dans la Figure 6.

5 Conclusion

L'intelligence artificielle occupe de plus en plus de place dans nos applications de tous les jours. Les modèles d'apprentissage automatique, utilisés dans ces applications, sont vulnérables et sont exposés à différents types d'attaques. Avant d'utiliser massivement l'apprentissage automatique dans les différents domaines notamment le domaine de la cyber-sécurité, il est très important d'étudier la sécurité du processus d'apprentissage automatique. Dans cet article, nous avons présenté les vulnérabilités et une taxonomie des attaques qui peuvent affecter le fonctionnement d'un système ML. Comme perspectives de ce travail, nous envisageons d'étudier l'ensemble de mesures proposées pour faire face aux différents types d'attaques présentés. Les techniques de défense ont pour objectif d'éliminer ou de diminuer l'impact des attaques sur les modèles tout en assurant une précision acceptable.

N. B. : Ce travail a été effectué dans le cadre d'un stage de recherche financé par l'équipe SID du laboratoire ERIC.

Références

Barreno, M., B. Nelson, R. Sears, A. D. Joseph, et J. D. Tygar (2006). Can machine learning be secure ? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*.

Biggio, B., I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, et F. Roli (2013a). Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*.

Biggio, B., I. Pillai, S. R. Bulò, D. Ariu, M. Pelillo, et F. Roli (2013b). Is data clustering in adversarial settings secure ? In *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*.

Carlini, N. et D. Wagner (2017). Adversarial examples are not easily detected : Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*.

Goodfellow, I. J., J. Shlens, et C. Szegedy (2015). Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations*.

Huang, L., A. D. Joseph, B. Nelson, B. I. Rubinstein, et J. D. Tygar (2011). Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*.

James, G., D. Witten, T. Hastie, et R. Tibshirani (2021). *An Introduction to Statistical Learning, 2nd edition*. Springer Texts in statistics.

McGraw, G., R. Bonett, V. Shepardson, et H. Figueroa (2020). The top 10 risks of machine learning security. *Computer Vol. 53*.

Muñoz-González, L. et E. C. Lupu (2019). The security of machine learning systems. In *AI in Cybersecurity*, pp. 47–79. Springer International Publishing.

Saadi, F. Z. K., A. E. Kiouche, K. Amrouche, H. Seba, et M.-L. Messai (2022). Détection d'anomalies dans les flux de graphes et attaques d'empoisonnement. *Revue des Nouvelles Technologies de l'Information Extraction et Gestion des Connaissances, RNTI-E-38*, 273–280.

Shi, Y., Y. Sagduyu, K. Davaslioglu, et J. Li (2018). Generative adversarial networks for black-box api attacks with limited training data. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*.

Shokri, R., M. Stronati, C. Song, et V. Shmatikov (2017). Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*.

Tabassi, E., J. K. Burns, M. Hadjimichael, A. D. Molina-Markham, et J. T. Sexton (2019). A taxonomy and terminology of adversarial machine learning. Technical report, NIST.

Xiao, H., B. Biggio, G. Brown, G. Fumera, C. Eckert, et F. Roli (2015). Is feature selection secure against training data poisoning ? In *Proceedings of the 32nd International Conference on Machine Learning*.

Taxonomie des attaques sur les méthodes d'apprentissage automatique

Summary

Machine learning is gaining more and more application fields. Different methods exist that allow model constructions for decision support purposes. Nevertheless, machine learning models are vulnerable and exposed to different types of security attacks during the model learning process and after their deployment. Therefore, these threats must be first identified, defined and classified in order to propose defensive measures in the aim to have robust models. In this paper, we study various threats that can affect a machine learning process. We present a classification of threats divided into three parts, the objective, the knowledge and the capability of the attacker. Then, we show some examples of attacks on applications using machine learning.