REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université SAAD DAHLAB de BLIDA 1
Faculté des sciences
Département d'informatique
Domaine : informatique
Filière : informatique

SPÉCIALITÉ : SYSTÈMES INFORMATIQUES ET RÉSEAUX

MÉMOIRE DE MASTER

Une application de prévention contre les malwares basé sur l'apprentissage automatique KNN

Présenter par : *Chouali Ramdane*

Promoteur : *Mme. Yasmine Ghebghoub*

Devant le jury *: Mme. Oukid Saliha et Mr. Chikhi Nassim*

Année Universitaire : 2024-2025

REMERCIMENT

TOUT D'ABORD, JE VEUX DIRE "ALHAMDOULILAH" QUI M'A PERMIS D'ACCOMPLIR MON TRAVAIL ET M'A ACCORDÉ LA FORCE ET LA PATIENCE POUR ÊTRE PRÉSENT ICI AUJOURD'HUI.

JE REMERCIE MA FAMILLE, EN PARTICULIER MES PARENTS, POUR LEUR SOUTIEN CONSTANT, LEUR ENCOURAGEMENT ET TOUT CE QU'ILS M'APPORTENT AFIN QUE JE PUISSE MIEUX COMPRENDRE ET SUIVRE MON TRAVAIL.

A LA FIN, JE REMERCIE MA PROMOTRICE MME Y. GHEBHGHOUB DE M'AVOIR PROPOSÉ CE THÈME, POUR SES PRÉCIEUX CONSEILS, AINSI QUE POUR SON SUIVI ATTENTIF DE MON TRAVAIL, ÉTAPE PAR ÉTAPE. MERCI BEAUCOUP MADAME.

RESUMÉ

La détection des malwares par l'apprentissage automatique KNN (en français K-Plus-Proche) est une technique qui permet de classer les comportements réseau en comparant chaque activité à des modèles connus, afin d'identifier les menaces.

Dans notre solution, nous développons une application de prévention des malwares en utilisant l'algorithme K-NN et le jeu de données NSL-KDD. Après un prétraitement des données, le modèle apprend à distinguer les connexions normales des connexions malveillantes. Cette approche vise à offrir une détection simple, rapide et fiable des attaques connues.

Mots-clés : Sécurité, Malwares, Apprentissage automatique, K-NN, NSL-KDD, Détection des malwares, connexions malveillantes, connexions normales

ملخص

تعد تقنية الكشف عن البرمجيات الخبيثة باستخدام خوارزمية الجار الأقرب KNNمن تقنيات التعلم الآلي التي تهدف إلى تصنيف سلوكيات الشبكة من خلال مقارنة كل نشاط بنماذج معروفة، وذلك بهدف التعرف على التهديدات.

في حلنا المقترح، قمنا بتطوير تطبيق للوقاية من البرمجيات الخبيثة باستخدام خوارزمية KNN ومجموعة البيانات NSL-KDD. بعد إجراء معالجة أولية للبيانات، يتعلم النموذج التمييز بين الاتصالات العادية والاتصالات الخبيثة. تهدف هذه المقاربة إلى توفير وسيلة بسيطة وسريعة وفعّالة لاكتشاف الهجمات المعروفة.

الكلمات المفتاحية: الأمن، البرمجيات الخبيثة، التعلم الآلي، الجار الأقرب KNN، NSL-KDD, كشف البرمجيات الخبيثة، الاتصالات الخبيثة، الاتصالات العادية.

ABSTRACT

Malware detection using the K-Nearest Neighbor (K-NN) algorithm is a machine learning technique that classifies network behaviors by comparing each activity to known patterns, in order to identify potential threats.

In our proposed solution, we develop a malware prevention application using the K-NN algorithm and the NSL-KDD dataset. After preprocessing the data, the model learns to distinguish between normal and malicious connections. This approach aims to provide a simple, fast, and reliable method for detecting known attacks.

Keywords: Security, Malware, Machine Learning, K-NN, NSL-KDD, Malware Detection, Malicious Connections, Normal Connections.

TABLE DE MATIERE

Liste des figures

Liste des Tableaux

Introduction générale

À l'ère du numérique, les systèmes informatiques sont devenus des piliers essentiels de notre quotidien, que ce soit dans les secteurs économiques, industriels, sociaux ou gouvernementaux. Cependant, cette transformation s'accompagne d'un risque croissant : la vulnérabilité des systèmes face aux cybermenaces, en particulier les logiciels malveillants (malwares). Ces derniers ne cessent d'évoluer en complexité, mettant à rude épreuve les solutions de sécurité classiques.

Face à cette problématique majeure,l'apprentissage automatique (machine learning) s'imposent comme des approche prometteuse pour renforcer la cybersécurité. En exploitant des algorithmes capables d'apprendre à partir de données, il devient possible de détecter, classifier et anticiper les menaces de manière plus efficace, plus rapide et plus autonome.

Dans ce contexte, ce mémoire s'intéresse particulièrement à d'abord à réaliser une simulation d'une attaque sur un réseau virtuel sous forme d'un pentsting afin de vous montrer le danger existant sur les réseaux et ensuite nous appliquons l'algorithme K-Nearest Neighbors (KNN) à la détection des malwares. KNN est un algorithme supervisé simple mais puissant, basé sur la proximité entre les données. Il a montré des performances notables dans le domaine de la sécurité informatique, avec des taux de précision atteignant jusqu'à 95 % dans la classification de trafic réseau malveillant selon certaines études récentes.

Ce travail a pour objectif de concevoir et de développer une application pratique basée sur KNN, capable d'analyser les comportements suspects au sein d'un système informatique. L'approche proposée consiste à intégrer l'apprentissage KNN dans un outil de sécurité, afin d'offrir une solution intelligente, adaptative et facilement implémentable contre les attaques de type malware.

Ainsi, ce mémoire s'articule autour de quatre grands chapitres complémentaires.

Dans le premier chapitre, nous abordons les notions fondamentales de la sécurité informatique, en mettant l'accent sur les concepts clés tels que les vulnérabilités, les tests d'intrusion (pentests), ainsi que les systèmes de prévention et de défense contre les malwares. Ce cadre théorique permet de mieux comprendre les enjeux actuels auxquels sont confrontés les systèmes d'information.

Le deuxième chapitre est consacré à l'apprentissage automatique, en présentant ses différents types (supervisé, non supervisé, semi-supervisé) et en se focalisant plus précisément sur l'algorithme K-Nearest Neighbors (KNN). Nous discutons également des travaux connexes relatifs à la détection de malwares à l'aide du KNN, en analysant leurs résultats et en soulignant l'importance de cet algorithme dans le renforcement de la sécurité des systèmes informatiques.

Le troisième chapitre présente notre solution pratique : une application conçue autour de l'algorithme KNN. Nous y décrivons son mode de fonctionnement, accompagné de schémas explicatifs, et expliquons comment l'apprentissage automatique y est intégré pour détecter les activités malveillantes.

Enfin, le quatrième chapitre est dédié à la simulation de notre solution dans un environnement de développement réel. Nous détaillons les outils logiciels utilisés, le processus de scan et d'exploitation des vulnérabilités, ainsi que l'intégration du code de classification KNN utilisant le jeu de données NSL-KDD. Ce chapitre se conclut par une présentation et une analyse des résultats de classification obtenus, mettant en lumière la performance de notre application dans la détection des malwares.

CHAPITRE 1 : NOTIONS DE BASES

1.1 Introduction

À l'ère numérique actuelle, la sécurité de l'information est de la plus haute importance, car les systèmes d'information sont aujourd'hui de plus en plus ouverts sur Internet. Cette ouverture, pose néanmoins un problème majeur : il en découle un nombre croissant d'attaques. La mise en place d'une politique de sécurité autour de ces systèmes est donc primordiale.

Dans ce chapitre, nous introduisons les principales notions de base de sécurité informatique, y compris sa définition, ses objectifs, les vulnérabilités et des exemples des systèmes de préventions, ainsi que les mécanismes permettant d'améliorer la sécurité.

1.2 Sécurité informatique

1.2.1 Définition

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridiques et humains pour protéger l'intégrité et la confidentialité des informations stockées dans un système informatique.

1.2.2 Les enjeux de la sécurité informatique en entreprise

Aucun système informatique ne peut atteindre l'inviolabilité. Toutefois, les techniques mobilisées par la sécurité informatique sont en mesure de protéger les systèmes à un degré « suffisant » pour éloigner les menaces. Le concept de sécurité informatique s'est démocratisé en même temps que l'émergence des ordinateurs personnels. Les bonnes pratiques comme l'installation d'un antivirus avec une base de données constamment mise à jour, la mise en place de pare-feu ou le cryptage des informations sont aujourd'hui largement adoptées par les particuliers.

Naturellement, à mesure que les entreprises ont amorcé le virage numérique, la sécurité informatique s'est imposée comme une condition à leur pérennité, puis à leur compétitivité. Les conséquences des attaques liées à la cybercriminalité sont toujours plus graves : espionnage industriel, escroquerie financière, vol de données, demandes de rançon, etc. Si les grandes groupes peuvent disposer des moyens de déployer d'énormes dispositifs de sécurité informatique pour limiter les risques, les TPE/PME doivent sécuriser leurs ressources matérielles et logicielles ainsi que leurs données à moindre coût. L'élaboration d'une stratégie de sécurité informatique efficace se résume en 5 étapes essentielles.

a. Etablir une charte de sécurité informatique

La rédaction d'une charte de sécurité informatique permet à l'entreprise de sensibiliser l'ensemble de ses collaborateurs aux bonnes pratiques de l'informatique dans un cadre professionnel, et de les responsabiliser davantage quant à l'usage des outils mis à leur disposition.

b. Contrôler les accès internet de l'entreprise

Bien qu'internet soit, de loin, le premier moyen d'intrusion et d'attaque malveillante, peu d'entreprises peuvent s'en passer. Pour minimiser les risques, il est préférable d'utiliser des pare-feux qui permettent de filtrer les virus et de bloquer les contenus de sites douteux en se basant sur des listes régulièrement mises à jour.

c. Sauvegarder ses données

Il est très important pour une entreprise de pouvoir récupérer ses données en cas de défaillance du système d'information mis en place. Cette capacité de restauration peut être mise en œuvre en réalisant une copie de sauvegarde, en suivant la règle du 3-2-1 (3 sauvegardes, sur 2 supports différents dont 1 ailleurs) ou via des logiciels spécifiques. Elle permet à l'entreprise de continuer son activité et de se protéger contre les attaques des logiciels de type « ransomware » notamment.

d. Contrôler et limiter l'accès a certaines ressources

De plus en plus de salariés ont recours à des solutions de stockage en ligne pour sauvegarder certaines données. Cette pratique pose un problème de sécurité à l'entreprise qui risque de voir ses données parfois plus sensibles divulguées au grand public. La solution consiste à recourir à un système de partage et de sauvegarde de données en interne, contrôlé en permanence par un administrateur.

e. Entretenir et mettre a jour le parc informatique

Pour sécuriser le parc informatique d'une entreprise, il est important d'établir un inventaire du matériel présent et de le sécuriser en commençant par uniformiser les systèmes d'exploitation et les logiciels de protection (anti-virus, pare-feu…). De même, garder les logiciels et systèmes d'exploitation à jour permet de corriger d'éventuelles failles de sécurité.

1.2.3 Les principes de la securité informatique

Dans un contexte de digitalisation, les entreprises sont de plus en plus exposées aux attaques informatiques. Il est donc capital de disposer d'un système de sécurité informatique fiable en mesure de remplir 5 objectifs principaux.

Figure 1. Les principes de la sécurité informatique [1]

a. Confidentialité

La confidentialité consiste à s'assurer que les informations diffusées ne sont accessibles qu'aux personnes qui disposent d'un droit d'accès. Elle constitue un axe important de la sécurité des systèmes d'information.

b. Intégrité

La sécurité informatique doit permettre à l'entreprise de garder ses données intactes, et veiller à ce qu'elles ne subissent aucun dommage ou destruction volontaire ou accidentelle.

c. Disponibilité

La disponibilité est l'une des pierres angulaires de la sécurité informatique. Elle désigne la capacité des collaborateurs à utiliser l'ensemble des données qui leur sont nécessaires pour l'accomplissement d'une tâche de manière sécurisée.

d. Non-Répudiation

La non-répudiation suppose la possibilité de vérifier l'identité de l'expéditeur et du destinataire d'un message transmis dans un cadre professionnel. Cette vérification se fait généralement grâce à la technologie du certificat numérique qui permet de prouver l'identité d'un individu.

e. Authentification

L'authentification consiste à s'assurer de l'identité d'un utilisateur puis de déterminer sa légitimité pour accéder à certaines ressources de l'entreprise (données, logiciels, etc.). Une entreprise qui respecte ces 5 principes de sécurité informatique limite de manière considérable les risques liés aux attaques et tentatives d'intrusion. Pour déployer les meilleures pratiques, il est primordial de disposer des compétences nécessaires dans un environnement de plus en plus turbulent, où les manifestations de la cybercriminalité sont toujours plus innovantes. Les entreprises ont fortement besoin de profils qualifiés dans le domaine. [2]

1.3 Vulnérabilité informatique

1.3.1 Définition

Une vulnérabilité informatique est un défaut de sécurité. Elle se situe dans un système d'information, une application, un logiciel, ou même au cœur d'un composant matériel. Ces failles proviennent également des utilisateurs et de leur façon de se servir de leurs outils informatiques. La **faiblesse de la sécurité** d'une entreprise possède donc des origines multipless.

Toutes les vulnérabilités ne sont pas la porte ouverte des pirates informatiques. La majorité d'entre elles sont rendues publiques, et corrigées. C'est le cas notamment des mises à jour des logiciels, des appareils et des systèmes d'exploitation. La grande majorité des failles détectées ne sont pas intéressantes pour les **blacks Hat**. Il faut qu'elles disposent d'un **intérêt lucratif (ransomware)** ou qu'elles donnent la possibilité de voler des données pour les revendre ou effectuer un chantage financier.

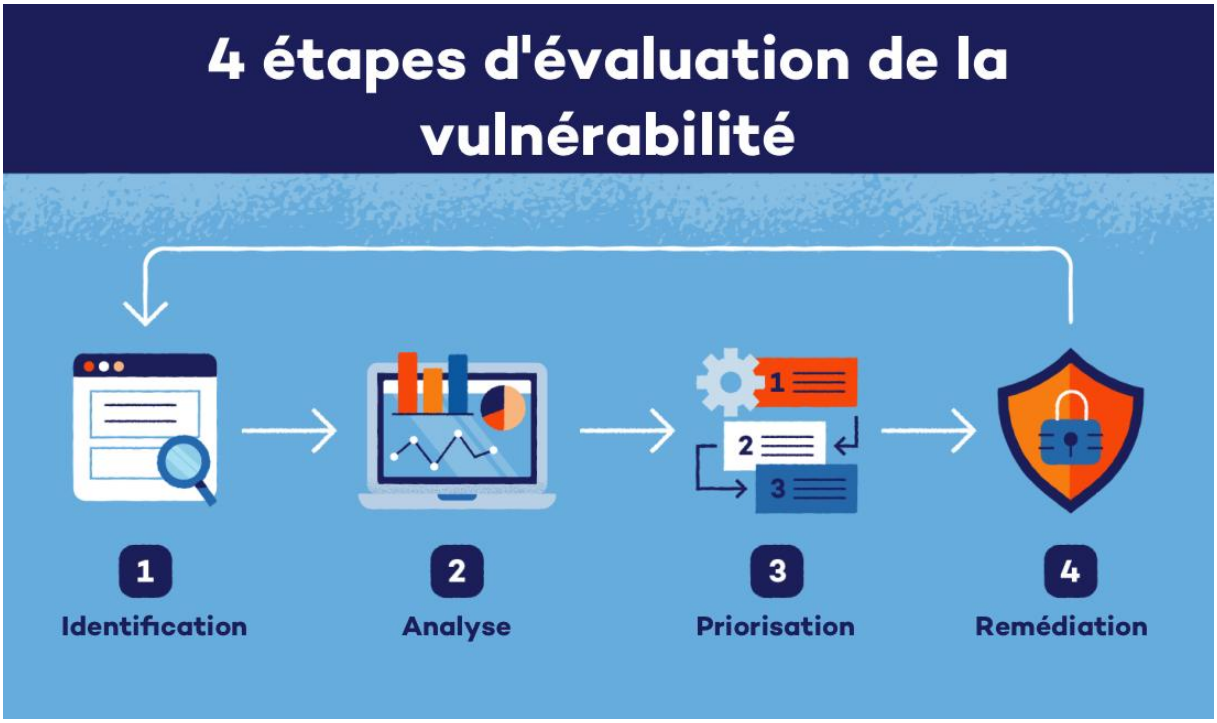

Figure 2. Les étapes d'évaluation de la vulnérabilité [3]

1.3.2 Les vulnérabilités informatiques majeures

Les vulnérabilités informatiques majeures sont à prendre au sérieux. Ce sont elles que les hackers exploitent. Les équipes de cybersécurité se concentrent sur la correction de ces défauts détectés.

a. Broken authentification ou la mauvaise gestion des mots de passe

Les failles de sécurité les plus répandues concernent les mots de passe des utilisateurs. Les identifiants sont souvent trop faibles et faciles à récupérer par des logiciels malveillants. Il n'est pas rare non plus que les personnes saisissent leur mot de passe sur un site web frauduleux, ou qu'ils partagent leurs codes d'accès. Aujourd'hui, la connexion à double identifiant (mot de passe et code de sécurité) permet de mieux protéger les données personnelles.

b. L'exposition de données mal protégées sur le cloud ou les réseaux

Sur les réseaux d'entreprises, les données sont exposées. Une mauvaise gestion de leur accès est le point de départ d'intrusion des cybercriminels. Ce point de vulnérabilité concerne aussi bien les données en ligne que vos ressources d'hébergement. Votre infrastructure peut, physiquement, être accessible. Il suffit également qu'un employé laisse une session ouverte pour qu'un intrus accède aux informations en son absence.

c. Des logiciels obsolètes et sans correctifs : un risque assuré

Pour des questions de budget ou par simple mauvaise gestion, d'anciens logiciels sont encore utilisés. Les mises à jour de sécurité ne sont pas réalisées et vous travaillez avec des applications obsolètes. Ce sont des points d'entrée pour les hackers. Vos outils de protection peuvent ne plus être compatibles pour détecter les défauts de sécurité. L'obsolescence des logiciels, mais aussi du matériel, génère de nombreuses menaces d'intrusion.

d. Messagerie d'entreprise et ingénierie sociale : les vulnérabilités humaines

L'ingénierie sociale rassemble toutes les techniques des pirates reposant sur la méconnaissance et la naïveté des utilisateurs. L'une des tactiques les plus courantes est d'envoyer de faux emails contenant des liens frauduleux. En cliquant dessus, les employés font entrer dans le système des malwares, dont les ransomwares cryptant toutes les données.

e. Les risques des lignes de code sensibles mises en production

Les développeurs écrivent des commentaires au sein de leur code sur les plateformes en préproduction. Ces lignes peuvent être un échange entre membre d'une équipe travaillant sur le même projet, ou pire, les codes d'accès à une base de données. L'utilisateur pense à le retirer avant la mise en production, mais l'erreur humaine étant, les lignes de codes sont publiées. Il suffit alors, pour le pirate, d'accéder au code source de la page et d'obtenir des informations compromettantes.

1.3.3 Les causes de la vulnérabilité informatique

Les causes des vulnérabilités informatiques sont multiples. Elles sont dues notamment à :

- Une mauvaise configuration des sécurités.
- Des mises à jour de sécurité non faites portant sur des défauts connus des pirates.
- De l'absence d'outils de supervision.
- De l'absence de correction des failles détectées.
- Une méconnaissance, tant de la part des utilisateurs que des administrateurs, des cybermenaces existantes.
- De l'utilisation d'appareils personnels pour accéder à l'infrastructure de l'entreprise.

1.3.4 Les impacts d'une vulnérabilité informatique en cybersécurité

Un défaut de sécurité peut avoir différentes conséquences sur les données et services des entreprises. Tout dépend de la faille exploitée et du niveau d'intrusion réalisé. Un simple dysfonctionnement peut rapidement être solutionné. D'autres ont, en revanche, plus d'impacts négatifs.

a. Le cryptage des données contre une demande de rançon

Le ransomware est une technique très répandue de la part des hackers. Après s'être introduit dans le réseau, le malware crypte les fichiers, les rendant totalement inaccessibles. Une rançon vous est demandée en échange d'une clé de chiffrement. Cette dernière vous donne, en théorie, l'accès à vos données.

Le risque ici est de perdre l'ensemble de vos fichiers si vous ne disposez pas d'une sauvegarde récente. La perte financière, si vous décidez de payer la rançon, peut être difficile à assumer pour une petite organisation.

b. L'arrêt complet des services d'une entreprise

Selon la nature de l'attaque, l'exploitation des vulnérabilités informatiques peut même mettre en péril l'activité d'une entreprise, voire la sécurité physique des personnes. C'est notamment le cas lors des attaques des établissements médicaux. L'activité d'une entreprise est impactée, générant une perte financière. L'inaccessibilité des services en ligne, comme pour les plateformes marchandes, suscite également un désintérêt de la part des clients. L'entreprise concernée perd aussi en réputation. [4]

1.4 Le Pentest (Test d'intrusion)

1.4.1 Définition

Le Pentest, ou test d'intrusion en français, est une méthode qui consiste à simuler une attaque informatique afin d'évaluer la sécurité d'un actif numérique, qu'il s'agisse d'applications, de sites web, d'objets connectés, de systèmes d'information complets, etc. En d'autres termes, lors d'un test d'intrusion, une organisation engage des professionnels en sécurité, appelés pentesteurs ou hackers éthiques (white hats), pour qu'ils agissent comme s'ils étaient des hackers malveillants (black hats) afin de mettre à l'épreuve la sécurité de leurs actifs.

Les différentes formes de pentest : **Black Box**, **White Box** et **Grey Box**

Les tests d'intrusion peuvent être classés en trois catégories principales en fonction de la méthodologie employée par le pentester et de son niveau de connaissance.

a. **Pentest BlackBox :** Dans ce scénario, l'auditeur simule une attaque en adoptant le rôle d'un hacker, dans des conditions similaires à une véritable intrusion. Il dispose de peu ou d'aucune information sur la cible. Cette approche permet de déterminer de manière fiable les vulnérabilités de la sécurité de l'entreprise. Les pirates informatiques ont généralement un accès limité aux informations sur le système d'information (SI) qu'ils tentent de compromettre, ce qui signifie qu'ils doivent consacrer du temps à l'exploration, laissant ainsi aux entreprises ciblées la possibilité de réagir, si elles en ont les moyens.
b. **Pentest WhiteBox :** Contrairement au pentest BlackBox, l'auditeur travaille en étroite collaboration avec le département des systèmes d'information (DSI) de son client. Il a accès à toutes les informations concernant la configuration du SI. Le pentest WhiteBox ressemble davantage à un audit informatique officiel, mais il offre la possibilité d'identifier plus en profondeur les vulnérabilités en accédant à toutes les couches du SI.
c. **Pentest GreyBox :** De plus en plus courant, le pentest GreyBox représente une méthodologie intermédiaire qui combine les avantages du BlackBox et du WhiteBox. Le pentester effectue ses tests avec un ensemble limité d'informations. Par exemple, il peut intégrer l'entreprise en tant que salarié d'un service sensible et avoir un compte utilisateur. À mesure qu'il avance dans l'attaque, il obtient progressivement de nouvelles informations. Le GreyBox se révèle être une stratégie de test d'intrusion optimale, car elle permet de simuler divers types d'attaques, y compris celles provenant de l'intérieur de l'entreprise. Le pentester peut élaborer des scénarios d'attaques émanant de membres de l'entreprise, d'anciens employés, voire de prestataires externes, en fonction des droits qui lui sont attribués.

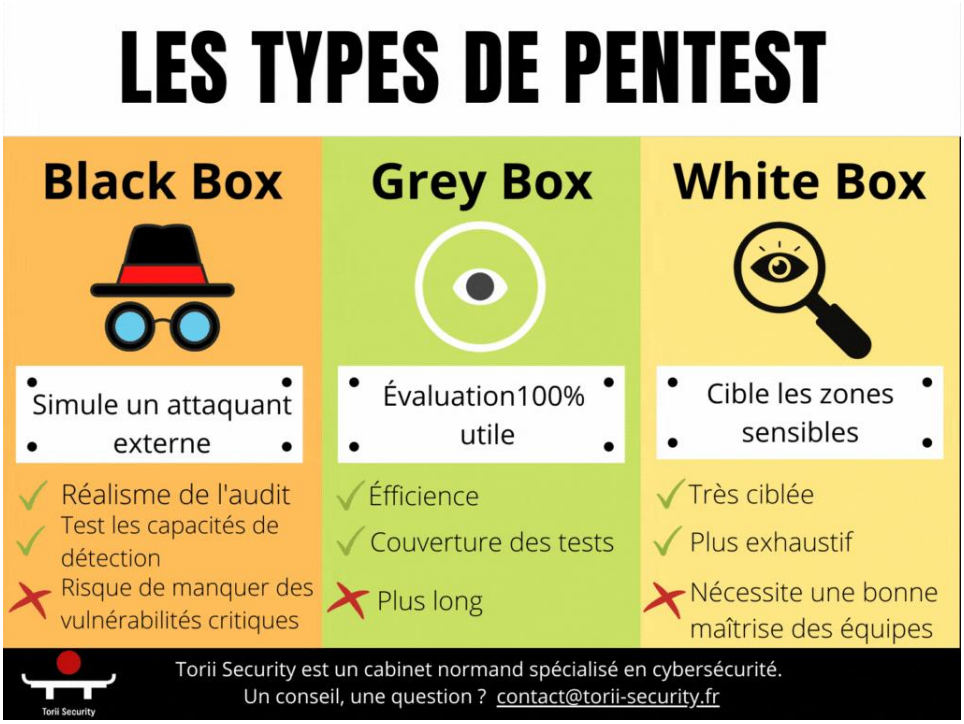

Figure 3. Types de Pentest [5]

1.4.2 Les objectifs du test d'intrusion

Un test d'intrusion implique l'évaluation de la sécurité d'un système informatique en lançant des attaques dans le but de repérer les vulnérabilités du système et de recommander des mesures de sécurité.

Le test d'intrusion inclue la recherche de failles logiques qui échappent aux outils automatiques, ainsi qu'une phase manuelle d'exploitation des vulnérabilités détectées. Il s'agit d'une méthode d'audit de sécurité complète et éprouvée, permettant d'évaluer l'impact réel de divers types de failles. Ce test peut se décliner en trois approches : en BlackBox, en WhiteBox ou en GreyBox.

À la suite d'un test d'intrusion, un rapport d'audit de sécurité est remis. Ce rapport répertorie les vulnérabilités détectées, classées en fonction de leur gravité, et propose des recommandations techniques pour les corriger. En complément de ce rapport, une synthèse non technique peut également être fournie pour une présentation au comité de direction ou à des partenaires.

1.4.3 Les étapes clés d'un pentest réussi

Le processus est segmenté en quatre phases distinctes :

a. **Planification et reconnaissance :** l'objectif initial de cette première étape est de définir la portée et les objectifs du test, y compris les systèmes à évaluer et les méthodes de test à mettre en œuvre. De plus, une collecte d'informations est réalisée, comprenant des éléments tels que les noms de réseau et de domaine, les serveurs de messagerie, les fuites de données, les données provenant de sources publiques (OSINT, etc.). Cette collecte permet d'acquérir une compréhension plus approfondie du fonctionnement de la cible ainsi que de ses vulnérabilités potentielles.

b. **Analyse :** la phase de scan se divise en deux composantes distinctes :

- **Analyse statique :** interprétation des données collectées lors de la phase 1. Cette analyse concerne les services, les ports, les adresses, les domaines, les comportements, les employés, etc. Elle se déroule sans aucun impact direct sur la cible.
- **Analyse dynamique :** cette partie consiste à obtenir des informations actives à l'aide d'outils d'analyse. Ces outils peuvent être issus des leaders du domaine, de sources open-source, de développements internes en R&D, et peuvent également englober des techniques manuelles.

c. **Obtention d'accès :** cette troisième phase recourt à diverses méthodes d'attaque, qu'elles soient orientées vers les applications, les systèmes, ou le réseau, telles que les « cross-site scripting, » les « SQL injection, » les « Buffer Overflow, » les « Heap Overflow, » etc. L'objectif est d'exploiter les vulnérabilités identifiées sur la cible. En vue d'obtenir des privilèges supplémentaires, des mouvements latéraux peuvent être effectués. L'objectif ultime est d'atteindre la cible définie par le client.

d. **Maintien de l'accès :** cette quatrième et dernière étape a pour but de déterminer si les vulnérabilités identifiées peuvent être exploitées pour établir une présence durable dans le système visé, offrant ainsi suffisamment de temps à un attaquant malveillant pour accéder en profondeur. L'objectif ici est de simuler les tactiques des menaces persistantes avancées, qui demeurent discrètement dans un système pendant des mois pour exfiltrer les données les plus sensibles de l'organisation. [6]

1.5 Malware

1.5.1 Définition

Le terme « malware » est un mot-valise dérivé de « malicious software » (logiciel malveillant) et décrit une attaque numérique conçue pour infiltrer des ordinateurs individuels ou de vastes réseaux de systèmes. Les logiciels malveillants peuvent être créés pour endommager les

systèmes, obtenir un accès non autorisé aux données ou verrouiller un réseau entier. Ils sont également souvent utilisés pour voler des données à des fins lucratives, comme arme dans le cadre d'attaques soutenues par des États, comme forme de protestation numérique par des hacktivistes ou pour demander une rançon à des entreprises.

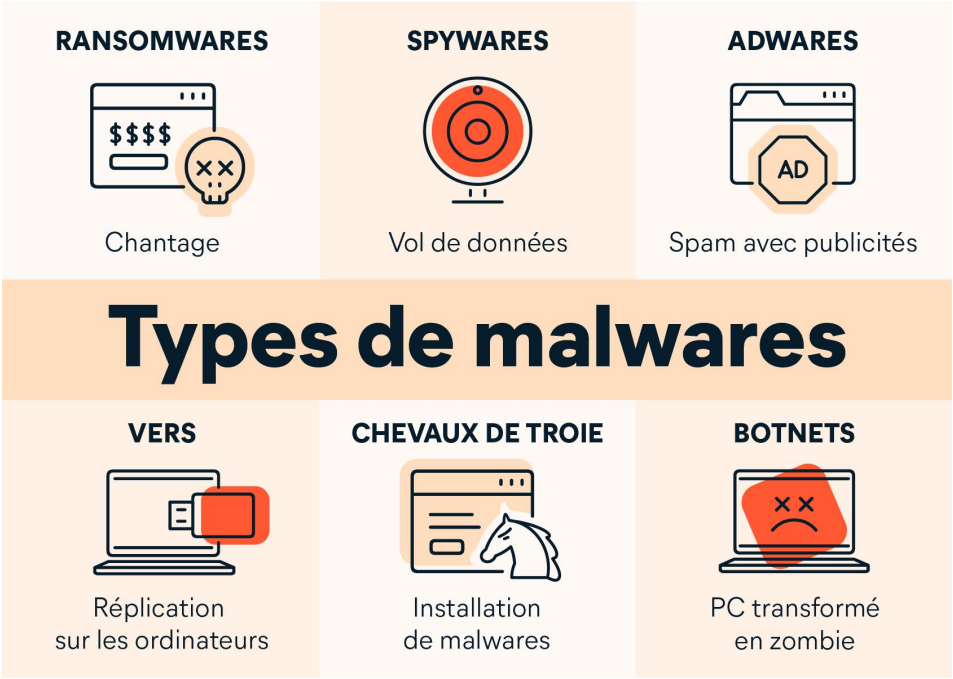

Figure 4. Types de malwares [7]

Le terme malware est un terme générique qui désigne une variété de logiciels malveillants. Les types de logiciels malveillants les plus courants sont les suivants :

- **Ransomware** – Menace bien connue, le ransomware empêche l'accès à un système informatique particulier jusqu'à ce que la victime verse de l'argent.
- **Vers** – Copient automatiquement leur code malveillant d'un système à l'autre. Les vers n'ont pas besoin d'être attachés à une application logicielle pour s'infiltrer dans un ordinateur ou un réseau.
- **Trojans** – Un type de virus qui incite les utilisateurs à l'ouvrir et à l'exécuter en se faisant passer pour des fichiers ou des URL inoffensifs.
- **Logiciel espion** – Recueille des informations sur l'utilisateur et les envoie à un acteur de la menace qui prévoit de lui nuire ou de l'exposer.
- **Logiciel publicitaire** – Affiche automatiquement des publicités (souvent intrusives) à l'intention d'un utilisateur lorsqu'il est en ligne.

1.5.2 Système de prévention contre les malwares

Une solution anti-malware est un logiciel ou un service essentiel qui protège les systèmes informatiques contre les logiciels malveillants. Pour ce faire, ces programmes détectent les éventuelles menaces de logiciels malveillants, bloquent les menaces avant qu'elles n'accèdent au système et éliminent les menaces existantes afin qu'elles ne causent pas d'autres dommages au système. L'anti-malware est également un outil nécessaire pour protéger vos réseaux et vos données contre les logiciels malveillants et les attaques **MaaS** . L'anti-malware est l'outil de prédilection d'un MSP ou d'un professionnel de l'informatique lorsqu'il doit localiser et supprimer le malware de l'ordinateur d'un client.

1.5.3 L'évolution des stratégies de lutte contre les logiciels malveillants

Le logiciel anti-malware original utilisait une base de données de signatures simples pour détecter les signes de programmes malveillants connus. Lorsque l'outil anti-malware analyse un ordinateur, il recherche ces signes. Les logiciels malveillants détectés sont alors mis en quarantaine ou supprimés. Les équipes informatiques ont également utilisé une forme de solutions anti-malware basées sur des signatures. Cependant, les cybercriminels peuvent éviter cette approche en modifiant quelque chose dans le code, de sorte que la signature ne soit plus reconnue.

Les professionnels de la cybersécurité avaient donc besoin d'une nouvelle méthode de détection des logiciels malveillants et se sont tournés vers l'heuristique. La détection heuristique est conçue pour rechercher des modèles de comportement suspect plutôt que des signatures spécifiques. Lorsqu'un logiciel tente de « mal se comporter » en faisant quelque chose qu'un logiciel légitime ne ferait pas normalement, il est signalé comme étant un logiciel malveillant.

Aujourd'hui, la plupart des logiciels anti-malware modernes utilisent une combinaison puissante de détection de signatures, d'analyse heuristique et d'une certaine forme d'apprentissage automatique. Une approche ML (parfois appelée intelligence artificielle) ne se contente pas d'analyser ce que fait le logiciel, mais analyse également sa composition. Cette stratégie permet à l'outil d'introduire des heuristiques comportementales dans un modèle de détection afin d'améliorer son propre algorithme grâce à un entraînement continu.

De plus, une autre stratégie a vu le jour, appelée sandboxing, qui consiste à exécuter le programme anti-malware dans un « bac à sable » sûr (une machine virtuelle simulant

l'environnement réel). Le logiciel peut alors observer le comportement du programme afin de déterminer son intention et tout dommage potentiel qu'il pourrait causer. Le sandboxing est en effet une technique anti-malware largement utilisée.

1.5.4 10 meilleures solutions de prévention contre les logiciels malveillants

a. Bitdefender

Bitdefender s'est imposé comme un leader dans le domaine des logiciels anti-malware. Les utilisateurs ont noté que le logiciel était doté d'une interface utilisateur élégante et de fonctions de sécurité robustes. Bitdefender propose une suite complète d'outils de sécurité pour protéger la vie numérique.

Outre sa bonne interface utilisateur et ses fonctions de sécurité, Bitdefender offre également une protection informatique essentielle grâce à la surveillance en temps réel et à l'obstruction des menaces potentielles. La plateforme dispose de fonctions exceptionnelles de détection des menaces qui s'appuient sur l'analyse comportementale pour identifier et atténuer les cyberattaques potentielles, ce qui est idéal pour tous les niveaux d'utilisateurs, des petites entreprises aux grandes sociétés.

Caractéristiques principales :

- Protection en temps réel : Surveille et bloque en permanence les menaces malveillantes.
- Détection avancée des menaces : Utilise l'analyse comportementale pour identifier et neutraliser les menaces émergentes.
- Pare-feu: Protège contre les accès non autorisés au réseau grâce à des paramètres personnalisables.

b. SentinelOne

SentinelOne s'est imposé comme un acteur de premier plan dans le paysage de la cybersécurité, en proposant une plateforme en nuage conçue pour une détection et une réponse rapides aux menaces. Son produit de sécurité informatique, Next-gen Antivirus ou NGAV, suit une approche basée sur l'IA, permettant une protection proactive contre les attaques les plus sophistiquées.

La plateforme de SentinelOne souligne également l'importance de la sécurité des terminaux dans le cadre de la stratégie antivirus et anti-malware. Pour ce faire, il intègre des outils de sécurité dans la gestion des terminaux, en tirant parti des capacités de détection avancée des logiciels malveillants.

Caractéristiques principales :

- Prévention alimentée par l'IA : Utilise l'intelligence artificielle pour prédire et prévenir les menaces.
- Détection et réponse des terminaux (EDR) **:** Fournit des capacités avancées d'investigation et de réponse aux incidents.
- Architecture cloud-native : Permet un déploiement et une évolutivité rapides.

c. CrowdStrike

CrowdStrike s'est imposé comme un leader de la protection des terminaux sur le cloud. L'accent mis par l'entreprise sur la rapidité de réaction et la prévention a suscité une grande attention dans le secteur de la cybersécurité. La plateforme de CrowdStrike s'appuie sur une technologie basée sur le cloud pour offrir une protection en temps réel contre les menaces en constante évolution.

En plus de ses capacités principales de protection des terminaux, CrowdStrike offre une suite de solutions de sécurité complémentaires. Il s'agit notamment de modules de renseignement sur les menaces, de réponse aux incidents et d'hygiène informatique. En proposant une approche globale de la cybersécurité, CrowdStrike permet aux entreprises de gérer et d'atténuer les risques de manière efficace.

Caractéristiques principales :

- Antivirus de nouvelle génération (NGAV) : Combine un antivirus traditionnel avec une détection avancée des menaces.
- Renseignements sur les menaces : Fournit des informations en temps réel sur les menaces émergentes.
- Réponse rapide aux incidents : Permet d'enquêter rapidement sur les attaques et d'y remédier.

d. Kaspersky Anti-Ransomware

Kaspersky est un leader mondialement reconnu dans le domaine de la cybersécurité, qui propose une gamme complète de solutions de sécurité. L'une de ses solutions de sécurité informatique est Kaspersky Anti-Ransomware, qui fournit des outils pour protéger les utilisateurs contre toutes les étapes des attaques de ransomware. La plateforme est également connue pour offrir son service gratuitement avec des fonctionnalités limitées. Ce niveau est utile pour les particuliers ou les entreprises qui souhaitent tester l'approche de la plateforme pour rester à la pointe de l'évolution du paysage des menaces.

Au-delà de ses capacités antimalware de base, Kaspersky propose une gamme de fonctionnalités supplémentaires, notamment la protection contre les menaces liées aux fichiers, la protection contre les menaces liées au courrier électronique, l'effacement des données à partir d'un appareil Windows, etc. L'entreprise se concentre également sur l'amélioration continue de l'expérience des utilisateurs en s'adressant à une base d'utilisateurs diversifiée.

Caractéristiques principales :

- Analyse du comportement : Utilise l'apprentissage automatique avec la protection de la mémoire pour détecter des schémas malveillants inconnus jusqu'alors aux premiers stades de l'exécution.
- Prévention des exploits : Protège les appareils contre les attaques de logiciels malveillants en bloquant en temps réel les tentatives de cyberattaques qui exploitent les vulnérabilités du système.
- Sécurité multicouche : La plateforme fonde sa sécurité sur de multiples techniques de protection qui vont des enregistrements AV classiques à la détection basée sur le comportement.

e. Avast Antivirus

Avast est un fournisseur de sécurité qui propose une gamme de produits, notamment son logiciel antivirus qui protège l'utilisateur contre les logiciels malveillants. Il a établi une forte présence sur le marché avec sa version gratuite de l'antivirus, attirant une large base d'utilisateurs, des particuliers aux grandes entreprises.

La protection antivirus de base d'Avast prend en charge les systèmes d'exploitation les plus répandus tels que Windows, macOS, Android et iOS. Comme nous l'avons mentionné, la plateforme comporte un volet gratuit qui offre une protection essentielle, tandis que les formules premium proposent des fonctionnalités et une assistance améliorée. La plateforme a

également élargi ses fonctionnalités qui incluent une protection totale avec Avast Ultimate, Avast SecureLine VPN, Avast Secure Browser, et plus encore.

Caractéristiques principales :

- Smart scan : Inspection approfondie et continue des appareils pour détecter les vulnérabilités.
- Bouclier Web : Détecte les téléchargements Internet suspects qui constituent des menaces potentielles pour le système.
- Alerte de sécurité Wi-Fi : Avertit les utilisateurs des faiblesses potentielles du réseau Wi-Fi en temps réel.

f. Microsoft Defender XDR

Anciennement connu sous le nom de Microsoft 365 Defender, Microsoft Defender XDR est une suite antivirus développée pour maintenir la posture de sécurité des systèmes gérés. Elle propose différentes plateformes spécialement conçues pour s'intégrer à d'autres solutions et plateformes de sécurité Microsoft, créant ainsi un écosystème de sécurité unifié. Ces plateformes incluent Microsoft Defender for Endpoint, Identity, Office 365, et plus encore.

Microsoft Defender XDR offre une approche rationalisée de la protection contre les logiciels malveillants. Il commercialise son système de perturbation automatique des cyberattaques avancées pour remédier aux menaces potentielles avant même qu'elles n'affectent les systèmes gérés. La plateforme s'intègre également à Copilot pour aider à atténuer les menaces, depuis la détection des cyberattaques jusqu'au signalement des incidents.

Caractéristiques principales :

- Détection des cybermenaces : La plateforme se targue d'une chasse proactive aux cybermenaces avant qu'elles ne se transforment en un problème plus grave.
- Perturbation des attaques : Offre une protection contre les logiciels malveillants en stoppant le mouvement latéral des cyberattaques à la vitesse de la machine.
- Protection de la suite Office : Offre Microsoft Defender pour Office 365 qui permet une protection multicouche contre le phishing par courriel, les ransomwares et le vol d'informations d'identification qui peuvent être exécutés dans des liens, des fichiers et des outils de collaboration.

g. Norton AntiVirus

Norton s'est fait un nom dans le domaine de la cybersécurité grâce à son programme antivirus et antimalware, facilement reconnaissable rien qu'à son logo. Le programme, distribué par Gen Digital (anciennement Symantec), compte plus de 80 millions d'utilisateurs, ce qui en fait l'une des solutions les plus privilégiées pour favoriser la cybersécurité.

La marque a développé Norton AntiVirus, une plateforme complète de lutte contre les virus informatiques et les logiciels malveillants, qui fait partie de l'ensemble de sa suite de sécurité informatique. Norton AntiVirus doit en partie son succès à ses offres complètes, adaptées à tous les utilisateurs, des particuliers aux grandes entreprises. Cela consolide la position de cette plateforme en tant que solution de pointe pour la lutte contre les menaces de cybersécurité.

Caractéristiques principales :

- Protection en temps réel : Norton offre une protection en temps réel contre les menaces en ligne nouvelles et existantes, éradiquant les cyberattaques potentielles avant qu'elles ne s'infiltrent dans un système informatique.
- Smart Firewall : Commercialisée sous le nom de Smart Firewall pour PC et Smart Firewall pour Mac, cette fonction permet de bloquer le trafic non autorisé entre les ordinateurs d'un réseau.
- Protection multicouche : La plateforme offre une approche multicouche pour lutter contre les logiciels malveillants, les ransomwares, les spywares, etc.

h. Webroot

Webroot est un éditeur de logiciels de cybersécurité, spécialisé dans les solutions de lutte contre les menaces de cybersécurité. La plateforme comporte des niveaux spécifiques pour les particuliers et les entreprises, ce qui en fait une solution mieux adaptée à une large base d'utilisateurs. Webroot offre une protection en temps réel contre les logiciels malveillants et les ransomwares et est connu pour son approche basée sur le cloud qui minimise l'impact sur le système tout en offrant une sécurité robuste.

Webroot offre des fonctionnalités qui permettent une protection solide contre les cyberattaques. Il s'agit notamment d'une protection contre les menaces pour Windows et Mac, d'une détection et d'un blocage en temps réel de l'hameçonnage, de capacités de protection de l'identité et d'une expérience de navigation sécurisée grâce à son bouclier Web Threat Shield.

Caractéristiques principales :

- Protection des Chromebooks **:** Comme les développeurs ont en quelque sorte négligé la base d'utilisateurs des Chromebooks, Webroot a développé une protection personnalisée dédiée à la plateforme.
- Analyses rapides comme l'éclair **:** Webroot est connu pour sa capacité d'analyse rapide, qui permet une analyse rationalisée des menaces sans perturbation.
- Légèreté : La plateforme est présentée comme légère et peu encombrante, tout en offrant une protection contre les menaces établies telles que les logiciels malveillants et les ransomwares.

i. Sophos

Sophos est une plateforme qui offre une suite de cybersécurité adaptée aux entreprises. La solution est axée sur la protection des terminaux et les services de sécurité pour se défendre contre les attaques de logiciels malveillants, qu'elles soient nouvelles ou déjà établies. Cela fait de Sophos le choix privilégié des grandes entreprises comme solution de protection de leurs actifs numériques.

Au-delà de ses fonctions principales de plate-forme de cybersécurité, Sophos offre une protection complète contre les ransomwares, conçue pour atténuer les menaces potentielles et alléger le fardeau associé aux cyberattaques préjudiciables. Cela aide les entreprises à établir une posture de sécurité efficace en répondant efficacement aux cyberincidents.

Caractéristiques principales :

- Détection et réponse réseau **:** Sophos agit comme un chien de garde contre les cyber-attaques en surveillant en permanence le trafic réseau pour aider à réduire les risques de sécurité.
- Plate-forme de messagerie **:** Sophos propose également une plate-forme de messagerie autonome qui permet de bloquer les tentatives de phishing et de nombreuses autres attaques de malwares par courrier électronique.
- Alertes : La plateforme dispose d'un système d'alerte qui se déclenche en réponse à des signaux suspects, ce qui permet aux utilisateurs de réagir immédiatement aux menaces potentielles.

j. Malwarebytes

Malwarebytes est une marque de cybersécurité bien connue qui se concentre principalement sur la protection des appareils contre les logiciels malveillants. Comme d'autres plateformes

anti-malware, Malwarebytes utilise des stratégies telles que l'analyse comportementale et le balayage heuristique pour s'assurer que les menaces sont identifiées et éliminées avant qu'elles ne causent des dommages importants au système ou des pertes de données.

Malwarebytes est également reconnu pour son logiciel anti-malware gratuit, destiné aux particuliers. Il offre des fonctionnalités de base qui analysent les appareils et suppriment les menaces de logiciels malveillants existantes. La plateforme offre également un niveau de qualité supérieure appelé ThreatDown, conçu pour les entreprises et les organisations qui ont besoin d'une protection avancée et de renseignements sur les menaces.

Caractéristiques principales :

- Systèmes d'exploitation pris en charge **:** Malwarebytes offre une prise en charge d'un ensemble varié de systèmes d'exploitation les plus populaires, tels que Windows, macOS, ChromeOS, Android et iOS.
- Alimentée par l'IA : La plateforme exploite l'intelligence artificielle pour protéger les appareils contre différents types de menaces telles que les virus, les chevaux de Troie, les logiciels malveillants, les logiciels espions et bien d'autres encore. [8]

1.5.5 Discussion

Système	**Technologie principale**	**Spécificité / Point fort**	**Public ciblé**
Bitdefender	Analyse comportementale + détection en temps réel	Équilibre entre interface utilisateur et sécurité avancée	Utilisateurs individuels & PME
SentinelOne	IA + EDR + cloud-native	Détection et réponse autonome, très adapté aux entreprises	Entreprises de taille moyenne/grande
CrowdStrike	NGAV + Intelligence sur les menaces + Cloud	Rapidité de réponse & renseignement sur les menaces	Grands comptes, entreprises exigeantes
Kaspersky Anti-Ransomware	Apprentissage automatique + sécurité multicouche	Spécialisé anti-ransomware, disponible gratuitement	Particuliers, TPE/PME
Avast Antivirus	Analyse heuristique + fonctions gratuites	Large base d'utilisateurs, bonne couverture multiplateforme	Grand public et PME

Microsoft Defender XDR	Intégration complète Microsoft + détection proactive	Idéal dans l'écosystème Microsoft, réponse automatisée	Entreprises déjà clientes Microsoft
Norton AntiVirus	Multicouche + pare-feu intelligent	Solution complète avec forte reconnaissance de marque	Grand public & professionnels
Webroot	Cloud + légèreté système	Très léger et rapide, adapté aux Chromebooks	Utilisateurs recherchant performance et discrétion
Sophos	Détection réseau + alertes + messagerie sécurisée	Très adapté aux infrastructures d'entreprise	Grandes entreprises
Malwarebytes	IA + analyse heuristique	Facilité d'usage, bon niveau gratuit pour particuliers	Particuliers & entreprises via "ThreatDown"

Tableau 1. Comparaison entre les solutions de prévention contre les logiciels malveillantes

- Pour les entreprises recherchant une protection avancée et intégrée : SentinelOne, CrowdStrike, Sophos.
- Pour un usage personnel ou petites structures avec bonnes performances gratuites : Avast, Malwarebytes, Kaspersky.
- Pour ceux déjà dans l'écosystème Microsoft : Microsoft Defender XDR est fortement recommandé.
- Pour une solution légère et rapide : Webroot.
- Pour une suite antivirus classique mais fiable : Norton ou Bitdefender.

1.6 Conclusion

La sécurité informatique constitue aujourd'hui un enjeu majeur dans un monde de plus en plus numérisé. Les vulnérabilités, qu'elles soient d'origine humaine, logicielle ou matérielle, offrent des portes d'entrée aux cyberattaques, mettant en péril la confidentialité, l'intégrité et la disponibilité des données. Les malwares, sous leurs nombreuses formes (virus, ransomware, spyware, etc.), illustrent bien l'ingéniosité croissante des attaquants.

Face à ces menaces, il est indispensable de mettre en place des stratégies de défense efficaces. Celles-ci reposent sur des outils techniques tels que les antivirus, les pare-feu, les systèmes de détection d'intrusion, mais aussi sur la sensibilisation des utilisateurs et la mise à jour régulière des systèmes. La sécurité ne peut jamais être absolue, mais une approche proactive,

combinant prévention, détection et réaction rapide, permet de réduire considérablement les risques.

En somme, la sécurité informatique est un domaine dynamique qui nécessite une vigilance constante, une adaptation continue et une coopération entre les utilisateurs, les professionnels du numérique et les institutions.

CHAPITRE 2 : ETAT DE L'ART

2.1 Introduction

Ces dernières années, l'apprentissage automatique (Machine Learning) a émergé comme un outil puissant pour résoudre des problèmes complexes dans divers domaines, notamment la cybersécurité. Parmi les algorithmes les plus utilisés figure **K-Nearest Neighbors (KNN)**, une méthode simple mais efficace pour la classification et la détection d'anomalies.

Dans un contexte où les cyberattaques deviennent de plus en plus sophistiquées, les approches traditionnelles de détection basées sur des signatures présentent des limites. L'apprentissage automatique, et notamment KNN, offre une alternative en permettant une analyse dynamique des données pour identifier des comportements malveillants.

Ce chapitre explore le rôle de l'algorithme KNN dans la classification et la détection d'attaques en cybersécurité. Nous aborderons son fonctionnement, ses avantages, ses limites et son application pratique dans la lutte contre les menaces informatiques.

2.2 L'apprentissage automatique

L'apprentissage automatique (**Machine Learning**) est l'un des principaux domaines dans l'intelligence artificielle qui traite des méthodes d'identification et des algorithmes par lesquels un ordinateur peut apprendre, ce domaine est associé à l'intelligence artificielle et plus spécifiquement intelligence computationnelle. L'intelligence computationnelle est une méthode d'analyse de données qui pointe vers la création automatique de modelés analytiques. Autrement dit, permettant à un ordinateur d'élaborer des concepts, d'évaluer, prendre des décisions et prévoir les options futures. Le machine Learning nécessite deux ensembles de données :

- **Ensemble de données pour l'entraînement :** c'est la base de connaissance utilisée pour entrainer, notre l'algorithme d'apprentissage, pendant cette phase, les paramètres du modèle peuvent être réglés (ajustés) en fonction des performances obtenues.
- **Ensemble de données pour le test :** cela est utilisé juste pour évaluer les performances du modèle sur les données non-vues.

La théorie de l'apprentissage utilise des outils mathématiques dérivés de la théorie des probabilités et de la théorie de l'information, Cela vous permet d'évaluer l'optimalité de certaines méthodes par rapport aux autres. [9]

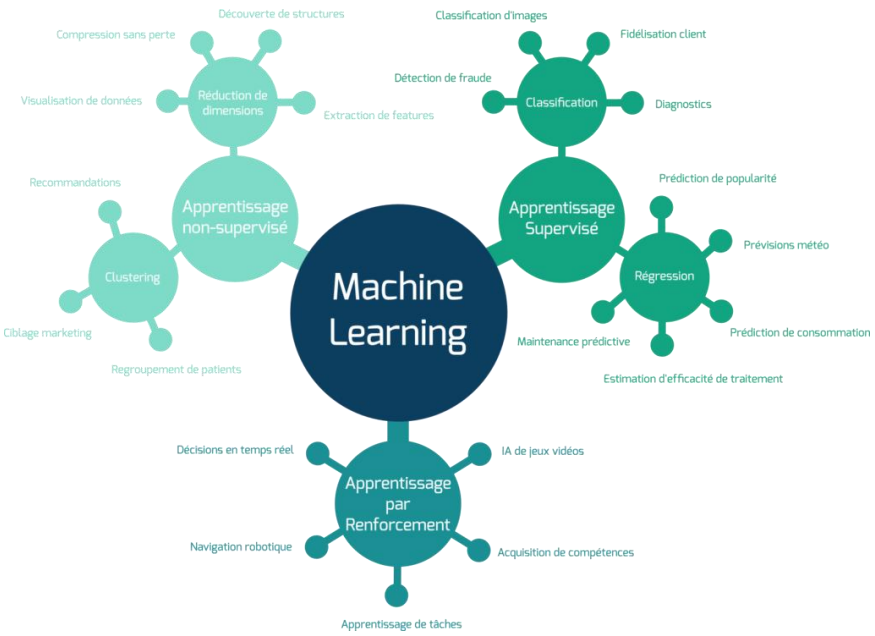

Figure 5. L'apprentissage automatique [10]

2.3 Types d'apprentissage automatique

On peut citer trois types d'algorithme d'apprentissage automatique :

- Apprentissage supervisé.
- Apprentissage non supervisé.
- Apprentissage par renforcement.

2.3.1 Apprentissage supervisé

L'apprentissage supervisé est fait en utilisant une **vérité**, c'est-à-dire qu'on a une connaissance préalable de ce que les valeurs de sortie pour nos échantillons devraient être. Par conséquent, le but de ce type d'apprentissage est d'apprendre une fonction qui, compte tenu d'un échantillon de données et de résultats souhaités, se rapproche le mieux de la relation entre les entrées et les sorties observables dans les données.

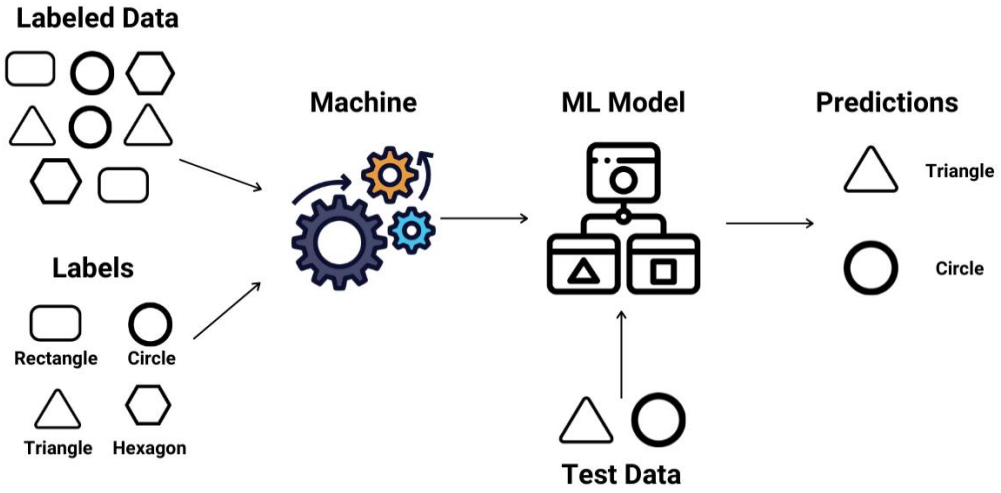

Figure 6. Schéma de fonctionnalité d'apprentissage supervisé

Dans l'apprentissage supervisé, on a deux types d'algorithmes :

- Les algorithmes de **régression**, qui cherchent à prédire une valeur continue, une quantité.
- Les algorithmes de **classification**, qui cherchent à prédire une classe/catégorie.

2.3.2 Apprentissage non supervisé

Les algorithmes d'apprentissage automatique non supervisés sont utilisés lorsque l'information utilisée pour entraîner le modèle n'est ni classifiée ni étiquetée. Le modèle en question étudie ses données d'entrainement dans le but de déduire une fonction pour décrire une structure cachée à partir ces données. À aucun moment le système ne connaît la sortie correcte avec certitude. Au lieu de cela, il tire des inférences des ensembles de données quant à ce que la sortie devrait être.

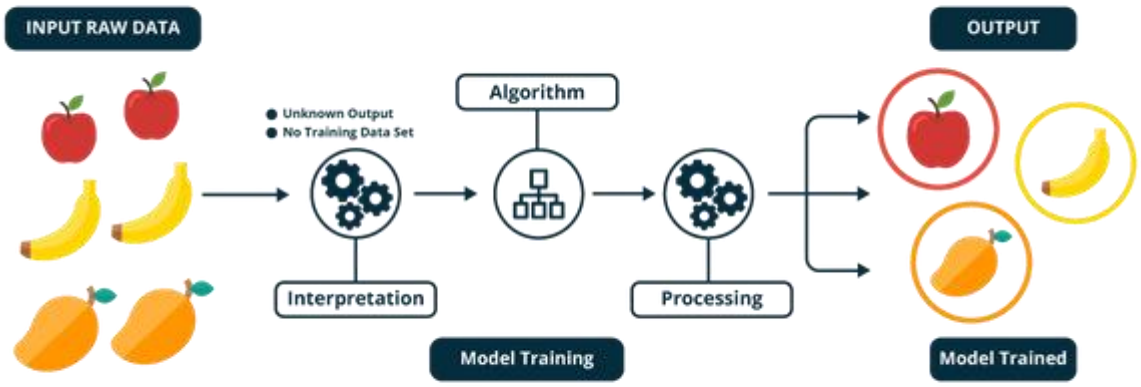

Figure 7. Schéma de fonctionnalité d'apprentissage non-supervisé

Les algorithmes de ce type d'apprentissage peuvent être utilisés pour trois types en problèmes.

- **Association :** un problème où on désire découvrir des règles qui décrivent de grandes portions de ses données. Par exemple, dans un contexte d'une étude de comportement d'achat d'un groupe de clients, les personnes qui achètent tel produit ont également tendance à acheter un autre produit spécifique.
- **Regroupement :** un problème où on veut découvrir les groupements inhérents aux données, comme le regroupement des clients par le comportement d'achat.
- **La réduction de dimension :** on vise à réduire le nombre de variables à prendre en compte dans l'analyse.

2.3.3 Apprentissage par renforcement

L'apprentissage par renforcement est une méthode qui consiste à optimiser de manière itérative un algorithme uniquement à partir des actions qu'il entreprend et de la réponse associée de l'environnement dans lequel il évolue.

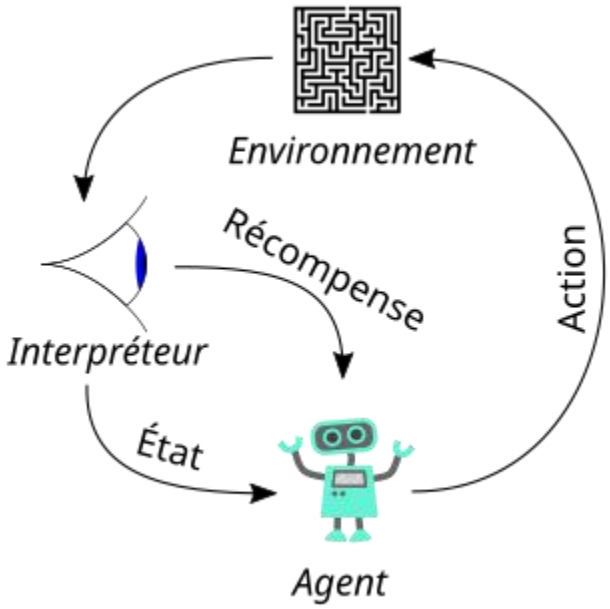

Figure 8. Schéma de fonctionnalité d'apprentissage par renforcement

Cette méthode permet aux machines et aux agents de déterminer automatiquement le comportement idéal dans un contexte spécifique pour maximiser ses performances. Une simple rétroaction de récompense, connue sous le nom de signal de renforcement, est nécessaire pour que l'agent apprenne quelle action est la meilleure.

2.4 Fonctionnement de l'apprentissage supervisé

Dans l'apprentissage supervisé, les données d'entrainement fournies aux machines fonctionnent comme le superviseur qui apprend aux machines à prédire correctement la sortie. Il applique le même concept qu'un élève apprend dans la supervision de l'enseignant.

L'apprentissage supervisé, c'est lorsque l'on a des variables d'entrée x et une variable de sortie y et qu'on utilise un algorithme pour apprendre la fonction de mappage de l'entrée à la sortie.

$$y=f(x)$$

Cela se fait sur plusieurs étapes : pour mieux comprendre, nous prendrons comme exemple un jeu de données où x représente les caractéristiques d'un logement, le phénomène étudié correspondant au marché immobilier, la réponse $f(x)$ correspondrait au prix de ce logement

En pratique, x représente presque toujours plusieurs points de données. Dans notre cas, le prédicteur du prix du logement pourrait prendre non seulement la superficie x_1, mais aussi le

nombre de chambres x_2, le nombre de salles de bains x_3, le nombre d'étages x_4, et ainsi de suite.

La détermination des entrées à utiliser est une partie importante de la conception du modèle, cela se fait généralement à l'aide du Feature Engineering (ou l'ingénierie des caractéristiques en français). Une fois les données sont transformées en formes adaptées à la modélisation et que l'on a conservé que les entrées pertinentes, l'étape suivante consiste à entraîner son modèle.

Cependant, dans notre cas, il est plus facile de supposer qu'une seule valeur d'entrée est utilisée. Supposons que le jeu de donnée soit représenté par le nuage de points suivant (le prix du logement en fonction de la superficie).

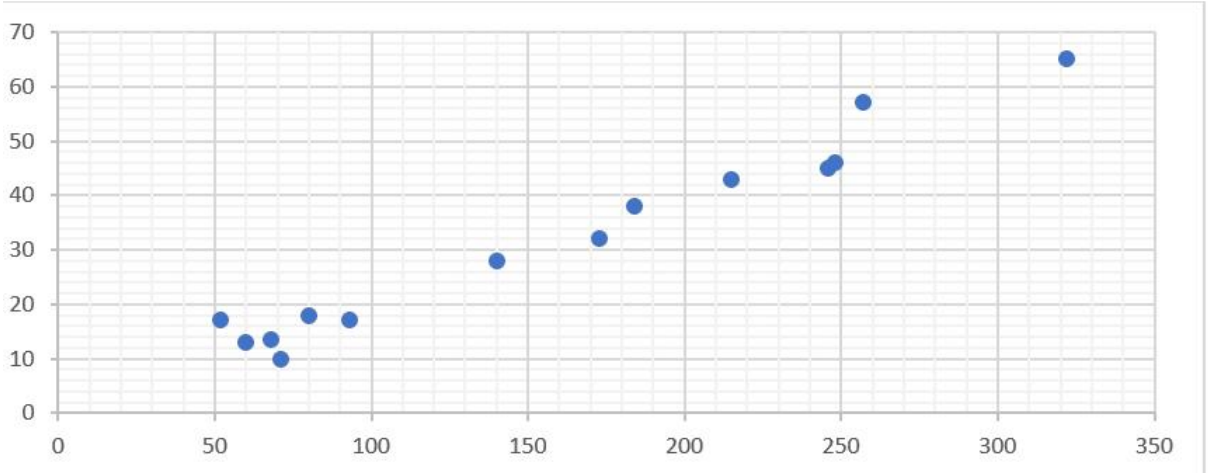

Figure 9. Courbe graphique d'ensemble de jeu de données

Et supposons que la fonction de prédiction a cette forme : $f(x)=ax+b$.

a et b sont des constantes. L'objectif est de trouver les valeurs optimales de a et b pour que le prédicteur **fonctionne le mieux possible**. À chaque fois que le modèle est entraîné avec une observation du jeu de données, on modifie a et b en résolvant l'équation $ax+b=y$ avec y le prix du logement et x est sa superficie. Ce processus est répété encore et encore **jusqu'à ce que le système ait convergé vers les valeurs optimales**.

Au fur et à mesure que le modèle s'entraîne, et les valeurs de a et b changent, on se trouve avec plusieurs droites reliant la superficie du logement à son prix.

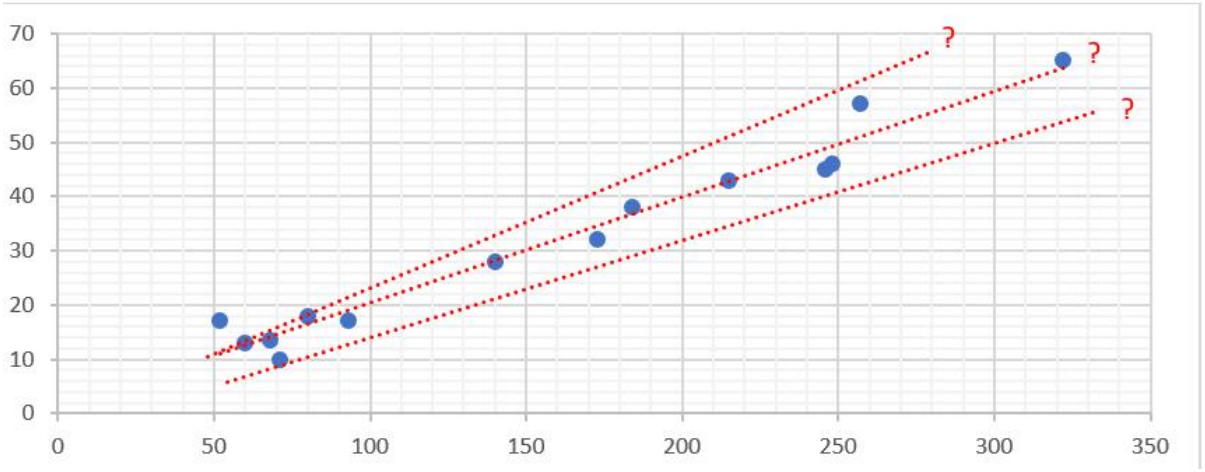

Figure 10. Courbe graphique De teste des prédicteurs

À un certain stade, si on répète le processus plusieurs fois, on constate que ***a*** et ***b*** ne changeront plus et ainsi on voit que le système a convergé. Cela signifie que nous avons trouvé le prédicteur optimal.

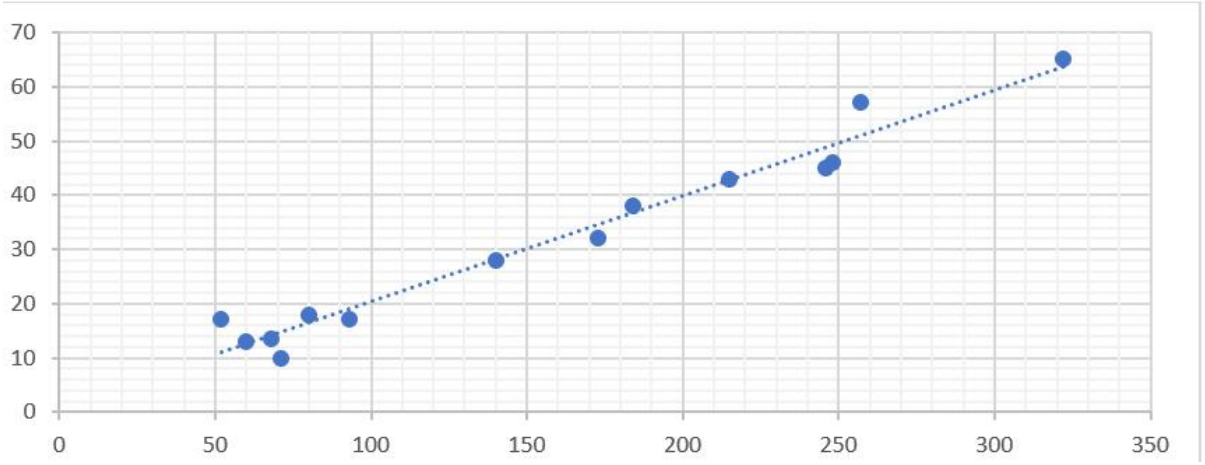

Figure 11. Courbe graphique de choix le prédicteur optimal

La dernière étape du processus de l'apprentissage supervisé consiste à vérifier la capacité de l'algorithme à généraliser et produire de résultats une fois mis en production avec des données qu'il ne connaît pas.

Alors comment mesurer si ce modèle est performant ? En réalité, c'est très simple, la meilleure droite est celle qui minimise les écarts entre la réalité et les prédictions. Dans notre cas, la droite rouge est moins bonne que la droite verte, car elle présente des écarts plus importants que l'autre droite.

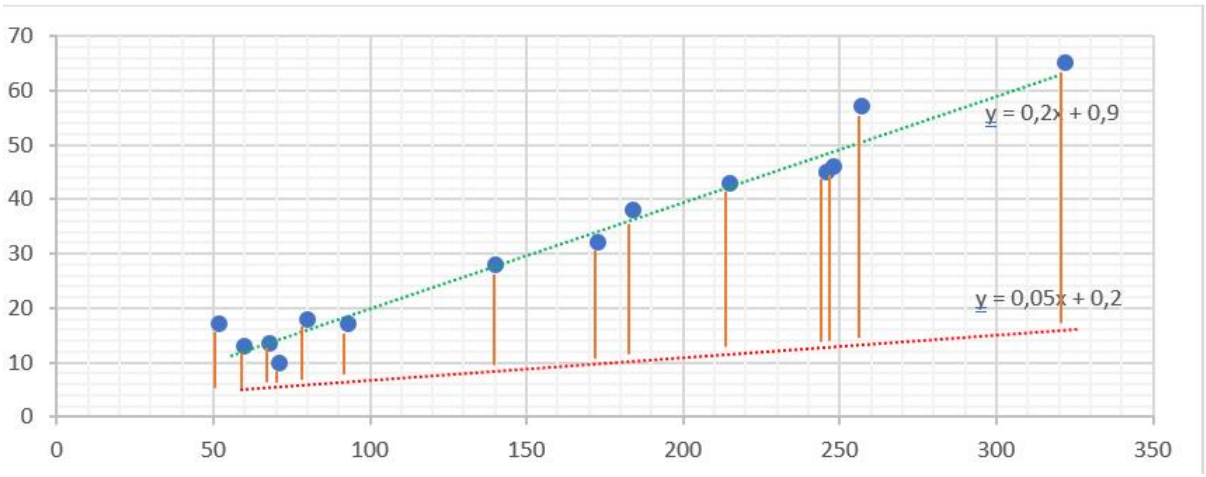

Figure 12. Courbe graphique de choix de meilleure droite

Dans le monde réel, les problèmes sont beaucoup plus complexes et par suite les fonctions de prédiction le sont aussi. Et c'est pourquoi les algorithmes de Machine Learning existent.

- Scores et performance

Maintenant que l'on a déterminé la fonction f comment évaluer la qualité et les performances de son modèle ? Il existe plusieurs métriques permettant de mesurer les performances d'un modèle d'un point de vue quantitatif.

- Cas de la classification

Considérons un problème de classification binaire : les réponses prennent les valeurs 0 ou 1.

L'accuracy est une métrique de score élémentaire qui calcule le **nombre moyen d'observations correctement prédites**. Elle peut être calculée directement par la formule suivante pour **n** prédictions.

$$\textbf{accuracy}=\frac{1}{n}\sum_{i=1}^{n} 1_{\hat{y}_i=y_i}$$

Par exemple, supposons que y= [0,1,1,0,1,0,0,1,1,1] les valeurs réelles et ŷ= [0,0,1,0,1,0,1,1,0,1] les valeurs prédites, alors :

$$\textbf{accuracy}=\frac{1+0+1+1+1+1+0+1+0+1}{10}=70\%$$

Le **F1-score** est une métrique plus fine qui prend en compte la notion de faux positifs et faux négatifs. Elle se base sur le calcul de deux mesures, qui font appel à la **matrice de confusion**.

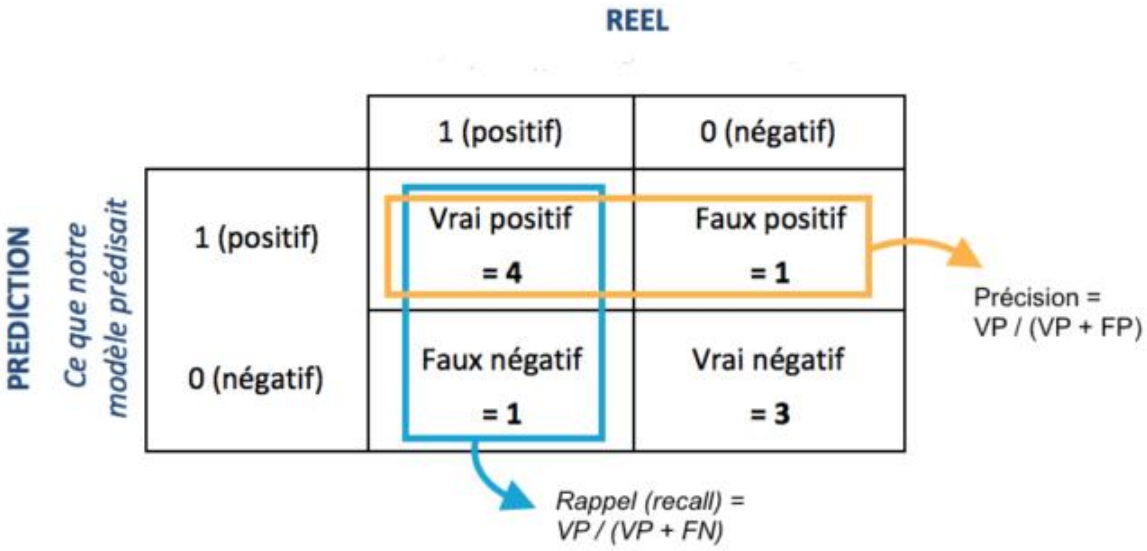

Tableau 2. Matrice de confusion

La **précision**, qui est la proportion d'observations correctement prédites positivement parmi toutes celles qui sont prédites positivement. Quand je prédis positif, est-ce que la bonne réponse était souvent positive ?

$$\textbf{Précision}=\frac{VP}{VP+FP}$$

Le **rappel**, qui est la proportion d'observations correctement prédites positivement parmi toutes celles qui devraient être prédites positivement. Est-ce que je n'ai pas tendance à trop souvent prédire négatif à tort ?

$$\textbf{Recall}=\frac{VP}{VP+FN}$$

Cela aboutit donc à la création du **F1 score** :

$$\textbf{ScoreF1}=2\times\frac{Precision\times Recall}{Precision+Recall}$$

- Cas de la régression

Dans le cadre de la régression, le calcul est différent : les valeurs ne sont plus discrètes (0 ou 1) mais continues (à valeurs dans R). Il faut donc utiliser des métriques adaptées. Le meilleur moyen pour comparer les performances sur un modèle de régression est de calculer les différences entre les réponses théoriques et les réponses prédites.

La **RMSE** (Root-Mean-Squared-Error) calcule cette différence avec une élévation au carré pour ne pas avoir de problèmes de signe. Le meilleur moyen pour comparer les performances sur un modèle de régression est de calculer les différences entre les réponses théoriques et les réponses prédites.

$$\mathbf{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\hat{y_i}\right)^2}$$

La **MAE** (Mean-Absolute-Error) ressemble fortement à la **RMSE**, puisque la valeur absolue est utilisée à la place de l'élévation au carré.

MAE=\frac{1}{n}\sum_{i=1}^n |y_i-\hat{y}_i|

L'avantage de ces deux métriques est qu'elles permettent de quantifier les performances en terme d'écart (absolu ou quadratique), mais ne permettent pas d'obtenir un score en pourcentage. Dans ce dernier cas de figure, on se tournera plutôt vers le coefficient de détermination **R2**:

R^2=1-\frac{\sum_{i=1}^n (y_i-\hat{y}_i)^2}{\sum_{i=1}^n (y_i-\bar{y})^2}

- **Les algorithmes d'apprentissage supervisé**

Pour créer un modèle d'**apprentissage supervisé**, on peut recourir à différents algorithmes , on peut citer en guise d'exemple la régression linéaire et logistique, **l'arbre de choix** avec différentes variables de sortie, le **Naive Bayes**, **Random Forest**, **SVM** et **k-NN**. [11]

2.5 Travaux connexes

Il existe plusieurs recherches qui s'intéresse à l'utilisation de l'apprentissage automatique dans le domaine de la sécurité informatique plus précisément dans la détection des attaques, dans cette section nous allons citer quelques travaux :

i. Les systèmes de détection d'intrusion sont essentiels à la protection contre les cyberattaques. Cet article présente une architecture de système de détection d'intrusion (IDS) qui utilise de nombreux modèles d'apprentissage automatique pour accroître la sensibilité et la spécificité. Des modèles courants, notamment la machine à vecteurs de support linéaire, la SVM quadratique, le K-plus proche voisin (KNN), l'analyse discriminante linéaire, le perceptron multicouche, le LSTM et l'encodeur automatique, sont testés sur l'ensemble de données NSL-KDD. Leurs recherches mesurent

l'exactitude, la précision, le rappel et le score F1 de ces modèles afin d'évaluer leur capacité à identifier les données réseau normales et malveillantes. Ils utilisent la capacité des algorithmes à reconnaître les schémas de cyberattaque mineurs pour construire une défense puissante. Ils étudient plusieurs méthodes pour trouver les meilleurs modèles d'apprentissage automatique pour la détection d'intrusion. Cette étude montre les avantages et les inconvénients de chaque modèle et comment ils contribuent à l'identification des intrusions. Ces recherches aideront à choisir des modèles d'apprentissage automatique IDS pour améliorer la sécurité du réseau et les contre-mesures. La « précision expérimentale » mesure la précision de prédiction du modèle et est nommée d'après « précis ». L'algorithme des K plus proches voisins est le plus précis à 98,55 %. La méthode du perceptron multicouche (97,78 %) et l'algorithme de mémoire à long terme (97,77 %) suivent de près. [12]

ii. Dans cet article [13] , ils présentent un modèle IDS basé sur une méthodologie d'apprentissage profond. Ils prennent que l'apprentissage profond a le potentiel d'être plus performant pour extraire des caractéristiques de données massives, compte tenu de l'ampleur du cybertrafic réel. Par conséquent, ils proposent d'entraîner un modèle IDS basé sur les réseaux de neurones à convolution (CNN), une méthode classique d'apprentissage profond, en utilisant l'ensemble de données NSL-KDD. Ils donnent les performances du modèle par classification multi-classes afin de les comparer aux performances des méthodes traditionnelles d'apprentissage automatique, telles que Random Forest (RF) et Support Vector Machine (SVM), et aux méthodes d'apprentissage profond, telles que Deep Belief Network (DBN) et Long Short Term Memory (LSTM). Les résultats expérimentaux montrent que les performances de ce modèle IDS sont supérieures à celles des modèles basés sur les méthodes traditionnelles d'apprentissage automatique et des nouvelles méthodes d'apprentissage profond en classification multi-classes.

iii. La sécurité de l'information est un enjeu crucial pour toute organisation, car elle doit protéger ses données contre les manipulations inutiles du trafic réseau ou les intrusions. Les systèmes de détection d'intrusion jouent un rôle essentiel dans la protection des données et des informations contre les comportements malveillants, car ils sont capables de détecter les attaques dans divers environnements. Par conséquent, de nombreuses recherches se concentrent sur le développement de nouveaux algorithmes

permettant de traiter les jeux de données différemment. Dans ce travail [14] , Ils proposent une nouvelle méthode PCA-Flou Clustering-KNN, qui combine l'analyse en composantes principales et le clustering flou avec des techniques de sélection de caractéristiques des K plus proches voisins. Cependant, nous effectuons deux classifications de classes principales pour construire notre modèle. Afin de vérifier la robustesse du modèle, nous avons utilisé le célèbre Dataset NSL-KDD pour l'analyse des anomalies. Cet ensemble de données est basé sur des données de référence utilisées pour la détection d'intrusion, KDDCup 1999.

iv. L'objectif de [15] est d'utiliser les algorithmes d'apprentissage automatique dans le but de créer un système de détection d'intrusion dans les environnement IdO (Internet des Objets) car ils voient que La croissance exceptionnelle et l'utilisation d'Internet soulèvent des préoccupations sur la façon de communiquer et de protéger les informations numériques et que dans le monde d'aujourd'hui, les pirates utilisent différents types d'attaques pour obtenir des informations précieuses.

2.6 Systèmes de prévention contre les malwares utilisant l'apprentissage automatique

Dans Nos jours, les solutions traditionnelles de cybersécurité, comme les antivirus basés sur des signatures, ne suffisent plus à contrer les menaces complexes et évolutives telles que les malwares zero-day et les attaques avancées persistantes (APT). Pour répondre à ces défis, plusieurs systèmes de prévention contre les malwares intègrent des techniques d'apprentissage automatique. Voici quelques exemples concrets de ces systèmes ainsi que les algorithmes qu'ils utilisent.

a. Microsoft Defender Advanced Threat Protection (ATP)

Fonctionnement : Ce système analyse des milliards de signaux issus de terminaux Windows à travers le monde afin de détecter des comportements anormaux.
Algorithmes utilisés : Apprentissage supervisé avec des modèles de forêts aléatoires (Random Forest) et de réseaux de neurones profonds (Deep Neural Networks)**.**

Apprentissage non supervisé pour la détection d'anomalies, souvent avec des techniques de clustering (K-means, DBSCAN). [16]

b. CylancePROTECT (BlackBerry)

Fonctionnement : Cylance utilise l'apprentissage automatique pour analyser les caractéristiques statiques d'un fichier (code binaire) sans avoir besoin de l'exécuter.
Algorithmes utilisés : Apprentissage supervisé, principalement à l'aide de réseaux de neurones profonds (DNN).

Utilisation de modèles linéaires (ex. : régression logistique) lors des premières phases d'analyse. [17]

c. CrowdStrike Falcon

Fonctionnement : CrowdStrike combine des données massives en provenance de millions d'endpoints avec des analyses comportementales.
Algorithmes utilisés : Deep Learning**,** notamment avec des réseaux de neurones récurrents (RNN) pour l'analyse des séquences d'événements.

Clustering non supervisé pour identifier des schémas de comportement inédits.

Random Forest pour la classification initiale de fichiers. [18]

d. Sophos Intercept X

Fonctionnement : Ce système utilise le deep learning pour prédire si un fichier est malveillant en se basant uniquement sur son contenu binaire.
Algorithmes utilisés :

Deep Learning, avec des **réseaux convolutifs (CNN)** adaptés à la classification de fichiers.

Régression logistique pour des analyses secondaires de risques. [19]

e. Malwarebytes Nebula

Fonctionnement : Nebula combine les techniques d'analyse comportementale et de classification pour détecter les menaces en temps réel.
Algorithmes utilisés : Forêts aléatoires (Random Forest) pour la détection précoce.

K-means pour le regroupement de comportements similaires (clustering).

Support Vector Machines (SVM) pour la classification des fichiers à risque. [20] [21] [22]

Système	Fonction principale	Algorithmes de ML utilisés	Type d'apprentissage	Particularité
Microsoft Defender ATP	Analyse comportementale à grande échelle sur les endpoints Windows	- Forêts aléatoires (Random Forest) - Réseaux de neurones profonds (DNN) - K-means	Supervisé & Non supervisé	Intégré à l'écosystème Microsoft, analyse massive dans le cloud
CylancePROTECT	Analyse statique des fichiers sans exécution	- Régression logistique - Réseaux de neurones profonds (DNN)	Supervisé	Détection rapide sans besoin de base de signatures
CrowdStrike Falcon	Analyse comportementale avec corrélation cloud	- Réseaux de neurones récurrents (RNN) - Clustering (DBSCAN) - Random Forest	Supervisé & Non supervisé	Fortement basé sur l'analyse cloud, très adapté aux grandes entreprises
Sophos Intercept X	Classification des fichiers binaires à l'aide du deep learning	- Réseaux convolutifs (CNN) - Régression logistique	Supervisé	Spécialisé dans la détection de ransomwares
Malwarebytes Nebula	Surveillance temps réel combinée avec analyse heuristique et comportementale	- Forêts aléatoires - K-means - SVM (Support Vector Machine)	Supervisé & Non supervisé	Interface intuitive et adaptée aux PME

Tableau 3. Comparaison des solutions de préventions contre les malwares qu'ils utilisent l'apprentissage automatique

2.7 Desscussion

Déterminer l'algorithme d'apprentissage à utiliser dans notre cas d'étude et sur le jeu de données est un choix très important ou souvent loin d'être évident. Dans la plupart des cas, l'approche empirique qui consiste à tester et évaluer plusieurs algorithmes d'évaluation pour en choisir le meilleur est la seule façon de garantir que le choix que nous avons effectué va dans le bon sens. En se basant sur le travail de [14] Nous avons décidé de réaliser notre proposition en utilisant l'algorithme KNN.

Algorithme	Accuracy	Precision	Recall	F1
Régression Logistique	87,75%	85,57%	90,81%	88,11%
Analyse discriminante linéaire	87,52%	86,36%	89,11%	87,71%
Méthode des k plus proches voisins	98,86%	98,84%	98,89%	98,86%
Arbres de décision	99,57%	99,56%	99,57%	99,57%
Classification naïve bayésienne	74,92%	66,69%	99,55%	79,87%
Réseaux de neurones	98,50%	98,65%	98,35%	98,50%
Machine à vecteurs de support	90,91%	89,21%	93,09%	91,10%

Tableau 4. Les statistiques de teste des algorithmes d'apprentissage automatique

Dans le tableau 4 la moyenne seulement est montrée sous forme de pourcentage avec arrondi au plus proche en deux chiffres après la virgule. Les trois algorithmes qui ont la précision la plus élevée (98 à 99%) sont "Arbre de décision (CART)", "k voisins les plus proches (KNN)" et "Réseau de neurones (MLP)", nous croyons que la précision de ce dernier pourrait encore être améliorer en personnalisant la configuration du réseau et le taux d'apprentissage

2.8 L'apprentissage automatique par l'algorithme KNN

L'algorithme des K plus proches voisins ou K-Nearest-Neighbors (KNN) est un algorithme de Machine Learning qui appartient à la classe des algorithmes d'apprentissage supervisé simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression. Dans cet article, nous allons revenir sur la définition de cet algorithme, son fonctionnement ainsi qu'une application directe en programmation.

En apprentissage supervisé, un algorithme reçoit un ensemble de données qui est étiqueté avec des valeurs de sorties correspondantes sur lequel il va pouvoir s'entraîner et définir un modèle de prédiction. Cet algorithme pourra par la suite être utilisé sur de nouvelles données afin de prédire leurs valeurs de sorties correspondantes.

Voici une illustration simplifiée :

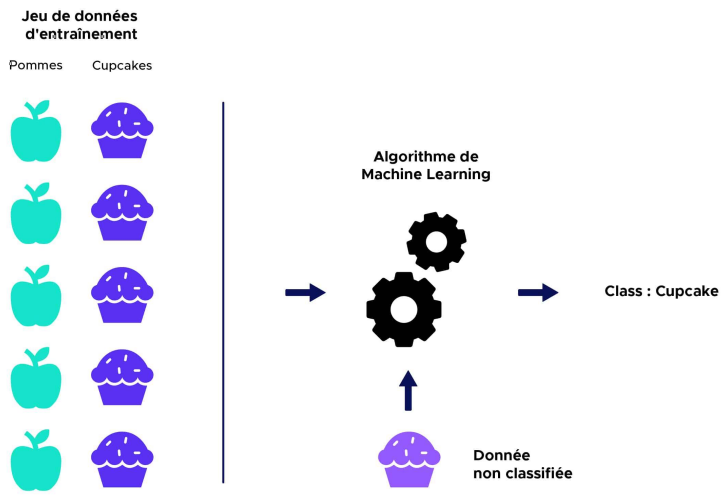

Figure 13. Schéma de fonctionnalité d'un algorithme d'apprentissage automatique

L'intuition derrière **l'algorithme des K** plus proches voisins est l'une des plus simples de tous les algorithmes de Machine Learning supervisé :

Étape 1 : Sélectionnez le nombre K de voisins

Étape 2 : Calculez la distance

$$\sum_{i=1}^{n} |x_i - y_i| \qquad \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Manhattan **Euclidienne**

Du point non classifié aux autres points.

Étape 3 : Prenez les K voisins les plus proches selon la distance calculée.

Étape 4 : Parmi ces K voisins, comptez le nombre de points appartenant à chaque catégorie.

Étape 5 : Attribuez le nouveau point à la catégorie la plus présente parmi ces K voisins.

Étape 6 : Notre modèle est prêt : [23]

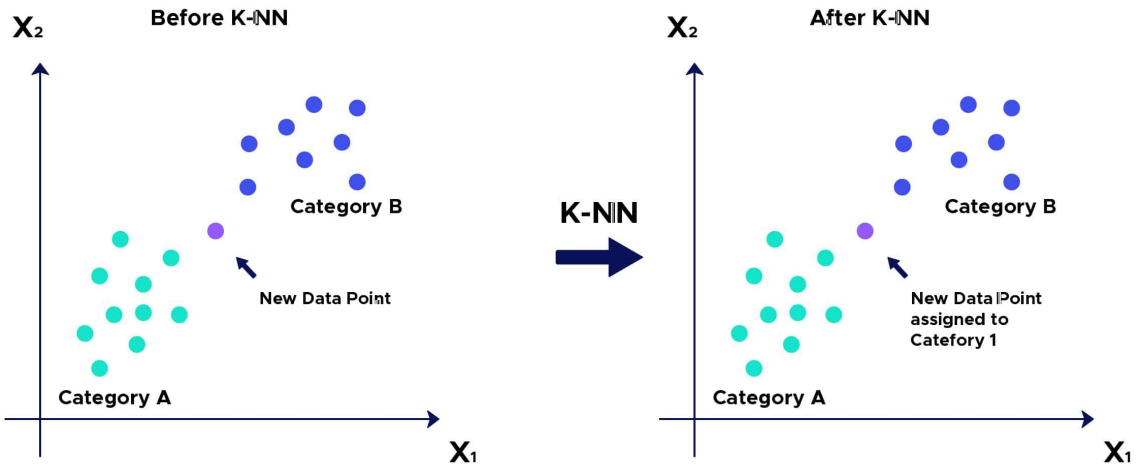

Figure 14. Schéma de fonctionnalité de l'algorithme d'apprentissage KNN

2.9 Conclusion

L'intégration de l'intelligence artificielle et de l'apprentissage automatique dans les systèmes de sécurité a permis une évolution significative dans la détection des malwares [24] [25] [26] . En utilisant des algorithmes avancés comme KNN, cet algorithme joue un rôle clé dans la détection et la classification des attaques informatiques grâce à son approche intuitive et efficace. En comparant les nouvelles données réseau aux exemples historiques étiquetés, il permet d'identifier avec précision diverses menaces comme les DDoS, scans de ports et intrusions. Ses principaux atouts résident dans sa simplicité de mise en œuvre, son adaptabilité aux nouvelles formes d'attaques et l'interprétabilité de ses résultats. Bien qu'il présente certaines limites, notamment en termes de sensibilité au bruit et de performance sur les gros volumes de données, des solutions comme l'optimisation du paramètre K, la normalisation des données ou l'utilisation de structures KD-Tree permettent d'en faire un outil performant. Particulièrement utile pour les systèmes nécessitant des décisions transparentes et rapides, le KNN s'intègre parfaitement dans les IDS/IPS, l'analyse de logs et la sécurité des infrastructures cloud. Pour maximiser son potentiel, il peut être combiné à d'autres algorithmes ou enrichi par des techniques d'apprentissage en ligne. Ainsi, malgré l'émergence de méthodes plus complexes, le KNN reste une solution de choix pour une détection d'attaques à la fois robuste et explicable dans de nombreux scénarios opérationnels.

CHAPITRE 3 : CLASSIFICATION DES MALWRES A PROPOS D ’ALGORITHME D’APPRENTISSAGE AUTOMATIQUE KNN

3.1 Introduction

L’utilisation de l’apprentissage automatique (machine learning) offre une alternative prometteuse pour renforcer les capacités de détection. Parmi les algorithmes de classification les plus simples mais efficaces figure le **K-Nearest Neighbors (KNN)**. KNN est un classifieur supervisé qui attribue une classe à un échantillon inconnu en se basant sur les classes des *k* exemples les plus proches dans l’espace des caractéristiques.

Ce chapitre présente une solution de détection des malwares fondée sur l’algorithme KNN. Nous décrivons la méthodologie adoptée, incluant la phase de prétraitement des données, l’extraction des caractéristiques, la sélection du paramètre *k*, ainsi que l’évaluation des performances du modèle. L'objectif est de démontrer que, malgré sa simplicité, KNN peut constituer un outil robuste et interprétable pour distinguer efficacement les comportements malveillants des comportements légitimes dans un environnement système ou réseau

3.2 Détection et Classification des attaques par l’apprentissage automatique KNN

Dans le cadre de ce travail, nous proposons une solution complète de détection et de classification des attaques réseau basée sur l’apprentissage automatique. La solution repose sur deux axes principaux :

L’**exploitation de vulnérabilités** via un scan de ports et une attaque réalisée par une machine externe (Machine 2), illustrant le scénario réel d’intrusion.

L’**utilisation de l’algorithme KNN** pour la détection automatique et la classification de ces attaques à partir d’un jeu de données caractérisant les comportements malveillants (Machine 1), afin de contrôler dynamiquement l'accès aux ports sensibles.

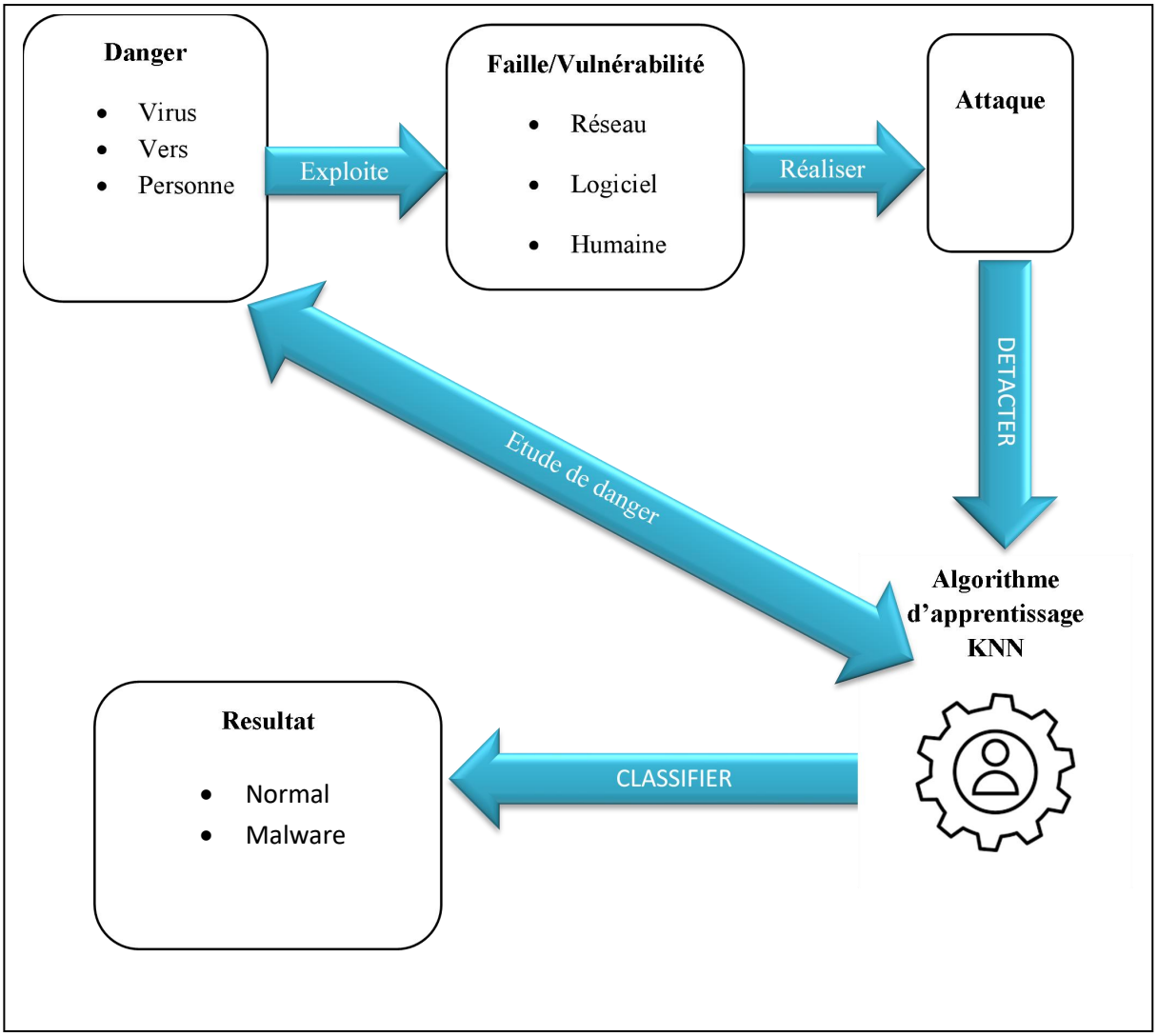

Figure 15. Schéma globale de notre solution

3.2.1 Partie 1 : Simulation d'une attaque réelle (Machine 2)

a. Scan de ports

La machine attaquante utilise l'outil **Nmap** pour réaliser un scan complet du système cible (Machine 1) afin d'identifier les ports ouverts et les services actifs. Cela permet de révéler les points faibles exploitables.

Commande utilisée : nmap -sS -p- <IP_cible>

b. Exploitation de la vulnérabilité

Une fois une vulnérabilité identifiée (ex. : via le port SMB ou HTTP), un exploit est lancé à l'aide de **Metasploit Framework**, dans le but de compromettre le système cible.

Example:

use exploit/windows/smb/ms17_010_eternalblue

set RHOST <IP_cible>

set LHOST <IP_machine_2>

exploit

c. Schéma explicatif

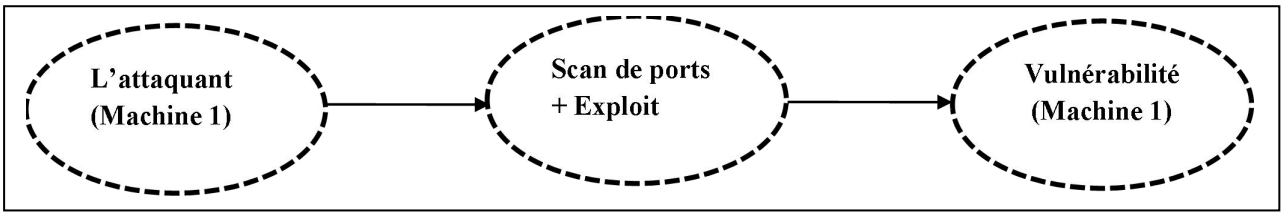

Figure 16. Schéma de partie 1 de notre solution

3.2.2 Partie 2 : Classification par KNN (Machine 1)

a. Constitution du jeu de données

Dans notre solution on a utilisé le **NSL-KDD** data set et ça description comme celui :

Description

a. APPRENTISSAGE DE LA DÉTECTION D'INTRUSIONS

Les logiciels de détection d'intrusions sur les réseaux protègent un réseau informatique contre les utilisateurs non autorisés, y compris éventuellement les utilisateurs internes. La tâche d'apprentissage pour la détection d'intrusion consiste à construire un modèle prédictif (c'est-à-dire un classifieur) capable de faire la distinction entre les connexions « malveillantes », appelées intrusions ou attaques, et les connexions « saines » ou normales.

Le programme d'évaluation de la détection d'intrusion DARPA 1998 a été préparé et géré par le laboratoire MIT Lincoln. L'objectif était d'étudier et d'évaluer les recherches dans le domaine de la détection d'intrusions. Un ensemble de données standard à analyser a été fourni, incluant une grande variété d'intrusions simulées dans un environnement de réseau militaire. Le concours de détection d'intrusions KDD 1999 utilise une version de cet ensemble de données.

Lincoln Labs a mis en place un environnement pour capturer neuf semaines de données brutes TCP (dump TCP) sur un réseau local (LAN) simulant un LAN typique de l'armée de l'air américaine. Ils ont exploité ce réseau comme s'il s'agissait d'un vrai environnement militaire, mais en y injectant de nombreuses attaques.

Les données brutes d'entraînement représentaient environ quatre gigaoctets de données binaires TCP compressées provenant de sept semaines de trafic réseau. Celles-ci ont été transformées en environ cinq millions d'enregistrements de connexions. De même, les deux semaines de données de test ont produit environ deux millions d'enregistrements de connexions.

Une connexion est une séquence de paquets TCP débutant et se terminant à des moments bien définis, durant laquelle les données circulent d'une adresse IP source vers une adresse IP cible selon un protocole bien défini. Chaque connexion est étiquetée comme normale ou comme attaque, avec un type d'attaque précis. Chaque enregistrement de connexion contient environ 100 octets.

b. CATÉGORIES D'ATTAQUES

Les attaques sont regroupées en quatre catégories principales :

- DOS (Denial-of-Service) : déni de service, par exemple syn flood ;
- R2L (Remote to Local) : accès non autorisé à partir d'une machine distante, par exemple deviner un mot de passe ;
- U2R (User to Root) : accès non autorisé aux privilèges administrateur locaux (root), par exemple attaque de type débordement de tampon (buffer overflow) ;
- Probing : exploration du réseau et collecte d'informations, par exemple scan de ports.

Il est important de noter que les données de test ne proviennent pas de la même distribution probabiliste que les données d'entraînement, et qu'elles comprennent des types d'attaques non présents dans les données d'entraînement. Cela rend la tâche plus réaliste. Certains experts pensent que la plupart des attaques inédites sont des variantes d'attaques connues, et que la « signature » de ces attaques peut suffire à détecter leurs variantes. Les ensembles de données contiennent un total de 24 types d'attaques pour l'entraînement, avec 14 types supplémentaires uniquement dans les données de test.

c. FONCTIONNALITÉS DÉRIVÉES (DERIVED FEATURES)

Stolfo et al. Ont défini des fonctionnalités de plus haut niveau (features) pour aider à distinguer les connexions normales des attaques. Ces fonctionnalités sont regroupées en plusieurs catégories.

Les fonctionnalités "same host" (même hôte) examinent uniquement les connexions des deux dernières secondes ayant le même hôte de destination que la connexion actuelle, et calculent des statistiques sur le comportement du protocole, le service utilisé, etc.

Les fonctionnalités "same service" (même service) examinent, de manière similaire, les connexions des deux dernières secondes ayant le même service que la connexion actuelle.

Les fonctionnalités "same host" et "same service" sont regroupées sous le nom de fonctionnalités de trafic basées sur le temps.

Certaines attaques de type probing (exploration) scannent les hôtes ou ports à un intervalle de temps plus long que deux secondes, par exemple une fois par minute. Par conséquent, les enregistrements de connexions ont aussi été triés par hôte de destination, et des fonctionnalités ont été construites en utilisant une fenêtre de 100 connexions vers le même hôte au lieu d'une fenêtre temporelle. Cela donne un ensemble de fonctionnalités de trafic basées sur l'hôte.

Contrairement à la plupart des attaques de type DOS ou probing, il semble qu'il n'y ait pas de motifs séquentiels fréquents dans les enregistrements des attaques R2L et U2R. En effet, les attaques DOS et probing impliquent de nombreuses connexions vers certains hôtes sur une courte période, tandis que les attaques R2L et U2R sont contenues dans les données des paquets eux-mêmes, et ne concernent souvent qu'une seule connexion.

Les algorithmes capables d'analyser automatiquement les données non structurées dans les paquets (comme le contenu des messages) restent un sujet de recherche ouverte. Stolfo et al. ont utilisé leur connaissance du domaine pour ajouter des fonctionnalités capables de détecter un comportement suspect dans ces données, comme le nombre de tentatives de connexion échouées. Ces fonctionnalités sont appelées fonctionnalités de contenu (content features).

Une liste complète des fonctionnalités définies pour les enregistrements de connexions est fournie dans le tableau ci-dessous

Nr	Features	
	Name	Description
1	duration	duration of connection in seconds
2	protocol_type	connection protocol (tcp, udp, icmp)
3	service	dst port mapped to service (e.g. http, ftp, ..)
4	flag	normal or error status flag of connection
5	src_bytes	number of data bytes from src to dst
6	dst_bytes	bytes from dst to src
7	land	1 if connection is from/to the same host/port; else 0
8	wrong_fragment	number of 'wrong' fragments (values 0,1,3)
9	urgent	number of urgent packets
10	hot	number of 'hot' indicators (bro-ids feature)
11	num_failed_logins	number of failed login attempts
12	logged_in	1 if successfully logged in; else 0
13	num_compromised	number of 'compromised' conditions
14	root_shell	1 if root shell is obtained; else 0
15	su_attempted	1 if 'su root' command attempted; else 0
16	num_root	number of 'root' accesses
17	num_file_creations	number of file creation operations
18	num_shells	number of shell prompts
19	num_access_files	number of operations on access control files
20	num_outbound_cmds	number of outbound commands in an ftp session
21	is_hot_login	1 if login belongs to 'hot' list (e.g. root, adm); else 0
22	is_guest_login	1 if login is 'guest' login (e.g. guest, anonymous); else 0
23	count	number of connections to same host as current connection in past two seconds
24	srv_count	number of connections to same service as current connection in past two seconds
25	serror_rate	% of connections that have 'SYN' errors
26	srv_serror_rate	% of connections that have 'SYN' errors
27	rerror_rate	% of connections that have 'REJ' errors
28	srv_rerror_rate	% of connections that have 'REJ' errors
29	same_srv_rate	% of connections to the same service
30	diff_srv_rate	% of connections to different services
31	srv_diff_host_rate	% of connections to different hosts
32	dst_host_count	count of connections having same dst host
33	dst_host_srv_count	count of connections having same dst host and using same service
34	dst_host_same_srv_rate	% of connections having same dst port and using same service
35	dst_host_diff_srv_rate	% of different services on current host
36	dst_host_same_src_port_rate	% of connections to current host having same src port
37	dst_host_srv_diff_host_rate	% of connections to same service coming from diff. hosts
38	dst_host_serror_rate	% of connections to current host that have an S0 error
39	dst_host_srv_serror_rate	% of connections to current host and specified service that have an S0 error
40	dst_host_rerror_rate	% of connections to current host that have an RST error
41	dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST error
42	connection_type	

Tableau 5. KDD Data set features [27]

Ces données sont prétraitées et normalisées avant d'être utilisées pour entraîner un classifieur KNN. [28]

b. Configuration du classifieur KNN

L'algorithme KNN est entraîné en mode supervisé pour classer chaque nouvelle connexion selon sa proximité avec les échantillons du jeu de données.

c. Déroulement :

Lorsqu'une connexion extérieure est détectée sur un port ouvert,

Ses caractéristiques sont extraites en temps réel,

Le classifieur calcule sa distance avec les instances du jeu d'entraînement,

La connexion est classée comme **malware** ou **non malware**, si est-il un malware, il donne son type (dos, vers, …)

Une action est déclenchée : autorisation ou blocage de l'accès.

f. Schéma explicatif

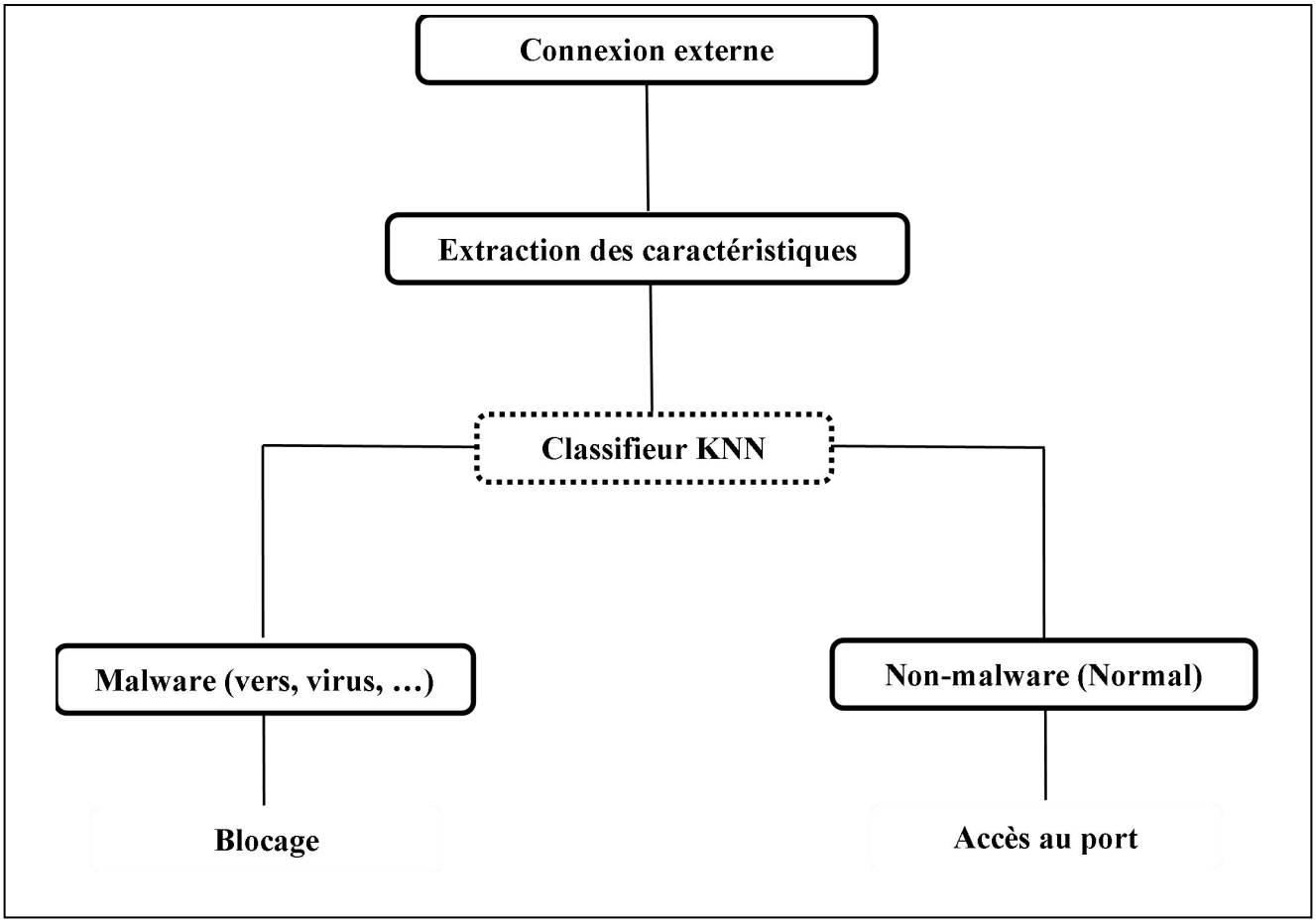

Figure 17. Schéma de partie 2 de notre solution

3.3 Fonctionnement Détaillé de l'Algorithme KNN pour la Détection d'Attaques

L'algorithme **KNN (K-Nearest Neighbors)** classe les attaques en comparant de nouvelles données à des exemples déjà connus [29] [30]. Voici comment cela fonctionne **pas à pas avec des calculs réels**.

a. Préparation des Données

Exemple de Jeu de Données (Attaques Réseau) : [31] [32]

ID	Durée (s)	Paquets	Ports Distincts	Type D'attaque
1	2.1	1500	1	DDoS
2	1800.5	50	3	Normal
3	5.7	800	45	PortScan
4	1.2	2000	1	DDoS
5	0.8	300	30	PortScan

Tableau 6. Jeu de données

Caractéristiques (Features) :

- Durée : Temps de la connexion (secondes)
- Paquets : Nombre de paquets échangés
- Ports_Distincts : Nombre de ports contactés

Classes (Labels) :

Normal, DDoS, PortScan [33] [34].

b. Nouvelle Connexion à Classifier

Supposons une **nouvelle connexion** avec :

Durée = 3.0 s

Paquets = 1200

Ports Distincts = 2

c. Calcul des Distances (K=3)

On utilise la distance Euclidienne) [35] [36] :

distance = $\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2]}$

Calcul pour chaque exemple :

- **Distance avec ID 1 (DDoS)**

$d_1 = \sqrt{[(3.0 - 2.1)^2 + (1200 - 1500)^2 + (2 - 1)^2]} = \sqrt{[0.81 + 90{,}000 + 1]} \approx 300.0027$

- **Distance avec ID 2 (Normal)**

$d_2 = \sqrt{[(3.0 - 1800.5)^2 + (1200 - 50)^2 + (2 - 3)^2]}$

$= \sqrt{[3234300.25 + 1322500 + 1]} \approx 2134.9$

- **Distance avec ID 3 (PortScan)**

$d_3 = \sqrt{[(3.0 - 5.7)^2 + (1200 - 800)^2 + (2 - 45)^2]} = \sqrt{[7.29 + 160{,}000 + 1{,}849]} \approx 402.4$

- **Distance avec ID 4 (DDoS)**

$d_4 = \sqrt{[(3.0 - 1.2)^2 + (1200 - 2000)^2 + (2 - 1)^2]} = \sqrt{[3.24 + 640{,}000 + 1]} \approx 800.002$

- **Distance avec ID 5 (PortScan)**

$d_5 = \sqrt{[(3.0 - 0.8)^2 + (1200 - 300)^2 + (2 - 30)^2]} = \sqrt{[4.84 + 810{,}000 + 784]} \approx 901.4$

d. Sélection des 3 Plus Proches Voisins (K=3)

ID	Distance	Type_Attaque
1	300.0	DDoS
3	402.4	PortScan
4	800.0	DDoS

Tableau 7. Les résultats des calcules

Les 3 plus proches sont :

ID 1 (DDoS)

ID 3 (PortScan)

ID 4 (DDoS)

[29] [30] [35]

e. Vote Majoritaire

DDoS → 2 votes (ID 1 et ID 4)

PortScan → 1 vote (ID 3)

Résultat :

La nouvelle connexion est classée comme **DDoS**.

f. Interprétation

La connexion a une **durée courte** (3s) et un **volume élevé de paquets** (1200), typique d'un **DDoS**.

Elle diffère d'un **PortScan** (qui aurait plus de ports distincts).

Elle est trop différente du trafic **Normal** (distance élevée) [33] [24].

g. Optimisation Possible

Choix du meilleur K

Si on prend **K=5**, on ajoute :

ID 2 (Normal) → Distance = 2,134.9

ID 5 (PortScan) → Distance = 901.4

Nouveau vote :

DDoS (2), PortScan (2), Normal (1)

→ Égalité !

Solution :

Choisir un **K impair** (K=3, 5, 7) pour éviter les égalités.

Utiliser une **pondération par distance** (les voisins les plus proches comptent plus) [35] [36] [25].

h. Applications Réelles

- **Cas 1 : Détection de DDoS**

Caractéristiques : Volume élevé, durée courte.

Exemple :

Paquets = 10,000, Durée = 1s → **DDoS**. [33] [37]

- **Cas 2 : Scan de Ports**

Caractéristiques : Nombre élevé de ports, paquets moyens.

Exemple :

Ports_Distincts = 100, Paquets = 500 → **PortScan**. [33] [34]

- **Cas 3 : Trafic Normal**

Caractéristiques : Faible volume, durée longue.

Exemple :

Paquets = 50, Durée = 1800s → Normal. [31] [32]

3.4 Conclusion

Cette solution montre l'efficacité de combiner une approche offensive (attaque simulée) avec une défense intelligente basée sur le machine-learning. L'algorithme KNN, grâce à sa simplicité et sa capacité à généraliser les comportements observés, constitue un choix pertinent pour discriminer les connexions légitimes des attaques potentielles en temps réel.

CHAPITRE 4 : Simulation de l'application

4.1 Introduction

Dans un monde où les cyberattaques deviennent de plus en plus sophistiquées, de nombreuses entreprises peinent à tester efficacement la sécurité de leurs systèmes d'informations. La sécurité offensive est une approche proactive qui vise à identifier et exploiter les failles des systèmes d'information avant qu'elles ne soient découvertes et exploitées par des attaquants.

Afin de faire un pentest sans endommager notre système, nous devons créer un environnement sécurisé pour simuler des attaques expose les infrastructures à des vulnérabilités non détectées, mettant ainsi en danger les données sensibles et l'intégrité du réseau.

4.2 Mise en Place d'un Lab de Pentest

La création d'un lab de pentest sécurisé permet de pratiquer des tests d'intrusion dans un environnement contrôlé, sans risque pour les systèmes réels.

Pour commencer, il est crucial de suivre plusieurs étapes clés. Tout d'abord, il faut installer une distribution spécialisée comme Kali Linux, qui inclut tous les outils nécessaires aux tests d'intrusion.

Ensuite, la configuration d'un réseau virtuel sécurisé à l'aide de logiciels comme VirtualBox ou VMware permet de simuler des environnements réalistes sans compromettre les systèmes de production.

L'intégration de machines vulnérables telles que Metasploitable ou d'applications comme DVWA est également essentielle pour pratiquer l'identification et l'exploitation des failles. Une fois le lab configuré, vous pouvez effectuer des tests d'intrusion complets afin de détecter les vulnérabilités et de renforcer vos compétences en sécurité dans un cadre sécurisé.

Cela permet d'affiner les techniques de pentesting et d'améliorer la posture de sécurité de l'organisation.

4.2.1 Oracle VirtualBox

VirtualBox est le logiciel de virtualisation gratuit, open source et multiplateforme d'Oracle. Celui-ci permet d'héberger une ou plusieurs machines virtuelles, avec des systèmes d'exploitation différents.

Le logiciel fonctionne sur différents systèmes d'exploitation hôtes à savoir Windows, Linux, MacOs et Solaris et prend en charge une multitude de systèmes d'exploitation invités en tant que machines virtuelles (Windows, Linux, Solaris, Mac, Unix sous différentes versions).

Grâce à ces systèmes d'exploitation invités, vous pouvez par exemple tester des logiciels sur une machine virtuelle (ou plusieurs en simultané) sans prendre le risque d'endommager votre ordinateur (hôte).

La solution propose de nombreuses fonctionnalités telles que :

- L'importation ou l'exportation de vos machines virtuelles vers une solution cloud,
- Le transfert de fichiers d'une machine hôte vers une machine virtuelle (sous Windows uniquement),
- La possibilité de contrôler la machine virtuelle à distance,
- La sécurisation des accès, avec des clés de cryptage de 256 bits.

VirtualBox vous permet d'exécuter un nombre illimité de machines virtuelles, avec pour seules limites l'espace disque et la mémoire de votre ordinateur. Utilisable sur tous supports, le logiciel s'adapte aussi bien aux systèmes embarqués qu'aux ordinateurs, aux datacenters ou encore aux environnements cloud.

Le téléchargement du logiciel est entièrement gratuit et bénéficie d'améliorations continuelles en fonction des retours des utilisateurs, grâce à son format open source. [38]

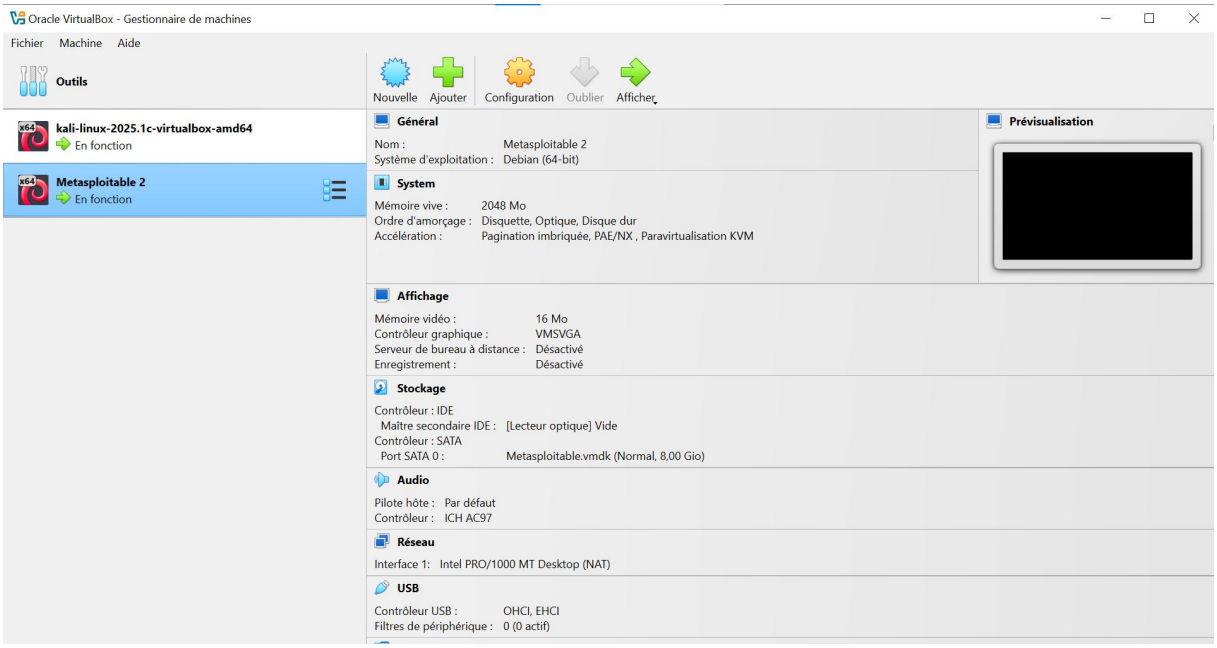

Figure 18. Interface d'Oracle VirtualBox

4.2.2 Kali-linux

Kali Linux est une distribution Linux basée sur **Debian** largement utilisée par les professionnels de la sécurité, les testeurs d'intrusion et les hackers éthiques. C'est un système d'exploitation libre et open-source qui est spécialement conçu pour la forensique numérique, les tests d'intrusion et l'audit de sécurité. Kali Linux est préinstallé avec une large gamme d'outils et d'utilitaires liés à la sécurité, tels que Nmap, Wireshark, Metasploit et Aircrack-ng, pour n'en citer que quelques-uns. La distribution peut être exécutée à partir d'un CD/USB en direct, ce qui permet aux utilisateurs de démarrer le système d'exploitation sans l'installer sur leur disque dur. Il peut également être installé sur un disque dur, permettant une utilisation permanente.

Kali Linux peut être utilisé pour une variété de tâches, telles que les tests d'intrusion sur les réseaux sans fil, la numérisation de vulnérabilités, les tests d'applications web, la récupération de mots de passe et l'ingénierie inverse. La distribution est constamment mise à jour et de nouveaux outils sont régulièrement ajoutés pour suivre les dernières tendances et développements en matière de sécurité. [39]

Nmap : Outil d'analyse du réseau qui permet de découvrir les hôtes, les services et les vulnérabilités potentielles au sein d'un réseau. Il est largement utilisé pour l'exploration des réseaux et l'audit de sécurité.

Metasploit : Metasploit est un outil de test de pénétration très répandu qui permet d'exploiter des vulnérabilités connues et de simuler des attaques réelles afin d'évaluer la sécurité des systèmes informatiques et des réseaux.

Wireshark : Analyseur de protocole réseau qui capture et inspecte les paquets de données en temps réel. Il permet de diagnostiquer les problèmes de réseau, d'analyser le trafic suspect et de détecter les violations de données potentielles ou les activités de logiciels malveillants.

Kali-linux-2025.1c : La version Kali Linux 2025.1c est la version officielle de Kali Linux pour l'année 2025, avec la désignation 1c dans son nom indiquant une mise à jour ou un patch, en plus des modifications principales de la version 2025.1. Cette version inclut des améliorations par rapport à la version 2025.1, mais pas nécessairement une refonte majeure.

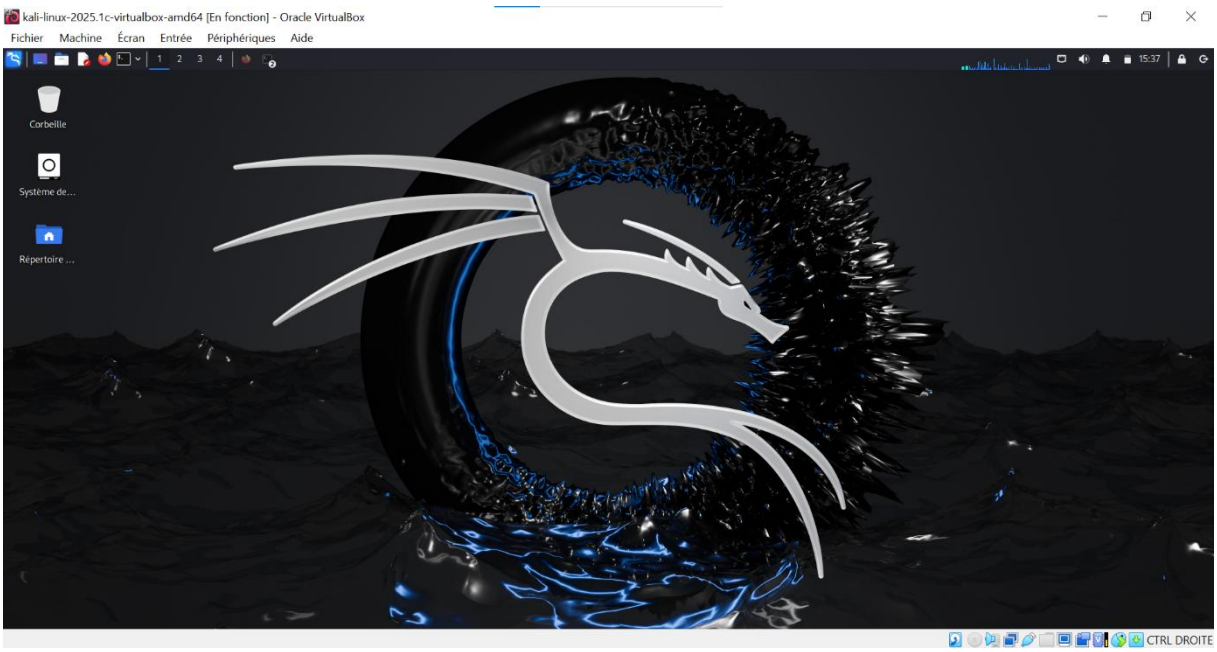

Figure 19. Interface De Kali-linux-2025.1c

4.2.3 Metasploitable-linux-2.0.0

La machine virtuelle Metasploitable est une version intentionnellement vulnérable de Ubuntu Linux conçue pour tester des outils de sécurité et démontrer des vulnérabilités courantes. La version 2 de cette machine virtuelle est livrée avec encore plus de vulnérabilités que l'image originale. Cette machine virtuelle est compatible avec VMWare, VirtualBox et d'autres plateformes de virtualisation courantes.
Par défaut, les interfaces réseau de Metasploitable sont liées aux adaptateurs réseau NAT et réseau privé (Host-only), et l'image ne doit jamais être exposée à un réseau hostile. [40]

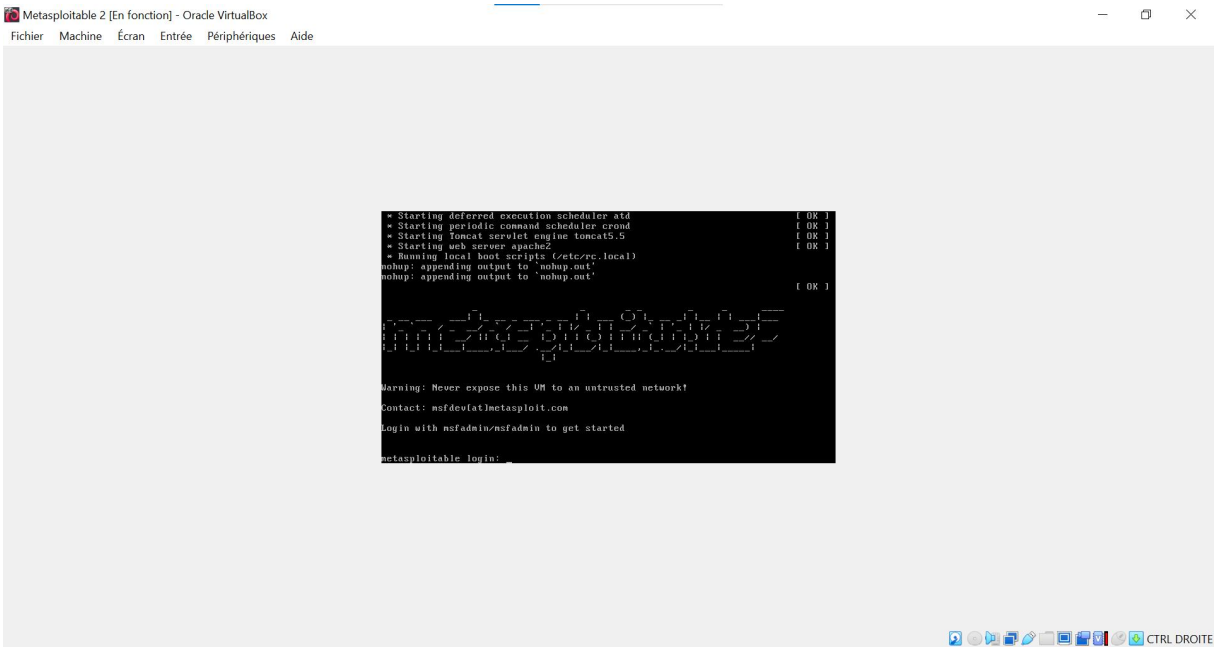

Figure 20. Interface De Metasploitable-linux-2.0.0

4.2.4 Google Colab

Google Colab, abréviation de Google Colaboratory, est une plateforme offerte gratuitement par Google permettant d'écrire et exécuter du code python dans le navigateur. Elle permet en particulier d'exécuter des notebooks Jupyter sans avoir besoin de soucier le matériel ou des logiciels installés sur votre ordinateur. Google Colab est un outil qui facilite également l'accès à des ressources de calcul et aux bibliothèques d'apprentissage automatique usuelles.

Les fichiers Jupyter Notebook, d'extension.ipynb, sont des documents interactifs qui intègrent du code, du texte explicatif et des éléments visuels dans un seul environnement [41]

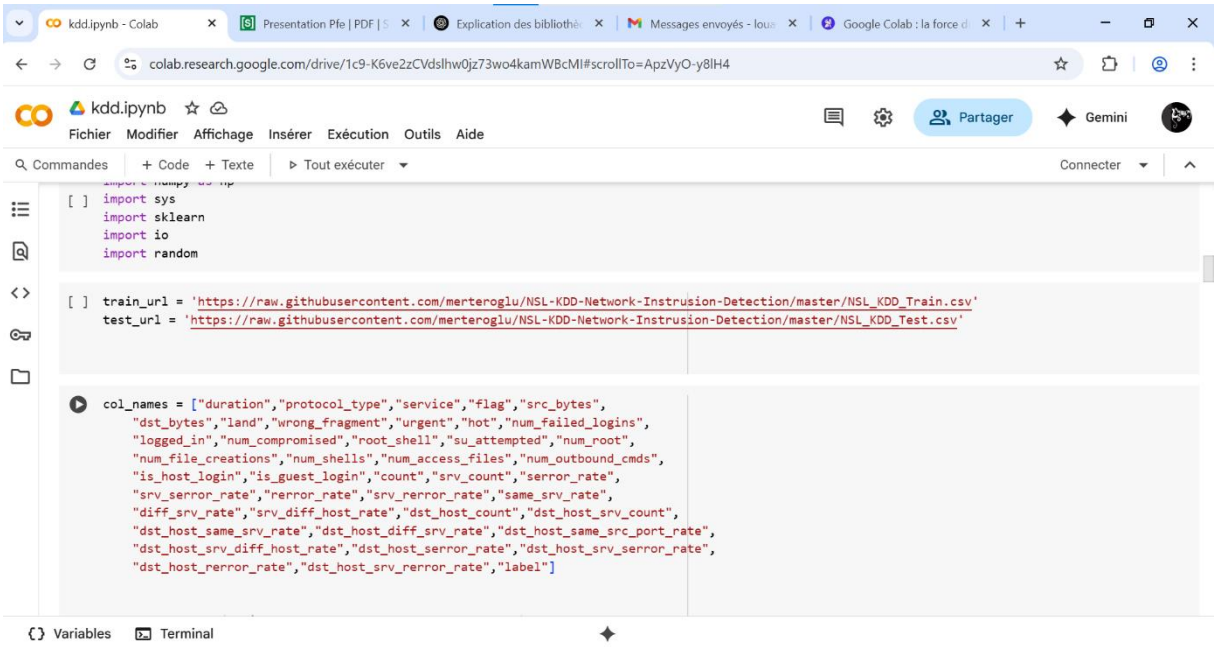

Figure 21. Interface De Google colab

4.3 Configuration de l'@ip

Depuis la metasploitable et le kali on a changé la configuration réseau au **Accès par pont** puis on obtient de l'@IP par la commande ifconfig après login.

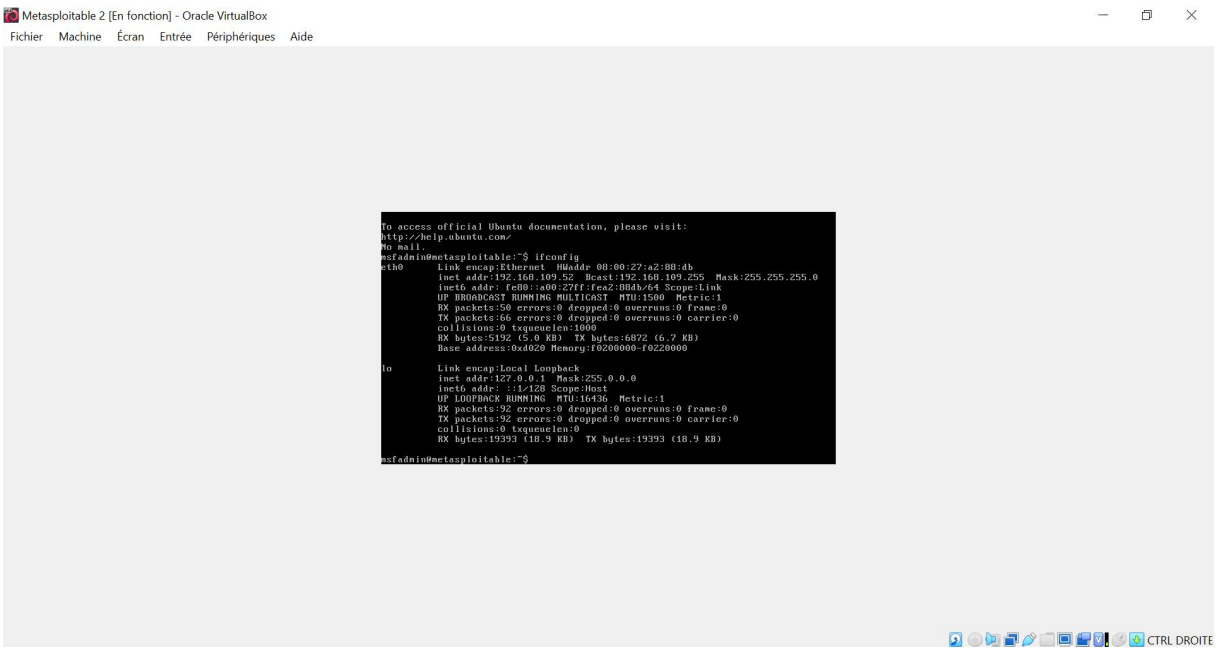

Figure 22. Nouveau @IP

Puis on fait le ping depuis le kali linux vers l'@IP de metasploitable (machine ciblé)par la commande ping <@ip> :

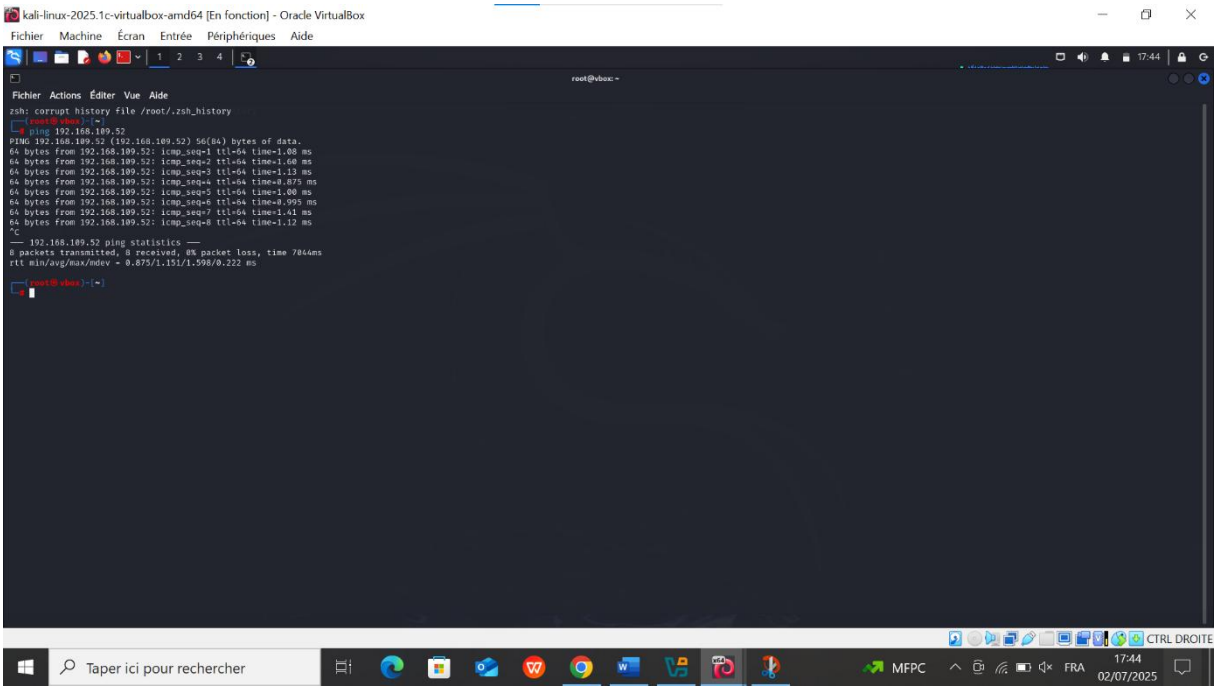

Figure 23. Le ping

4.4La réalisation d'un test de pénetration

4.4.1 Full Scan

Désigne généralement une analyse approfondie d'une cible (souvent un serveur, une machine distante ou un réseau) pour identifier toutes les failles et services accessibles.

On réalise le full scan a travers la commande suivant qui utilise le Nmap :

nmap -sS <@ip> -p -vv -sV -n

- -Ss: syn_scan
- @ip: Adresse IP cible
- -p: Spécifier des ports
- -vv:
- -sV: Version détection
- -n : éviter la résolution DNS

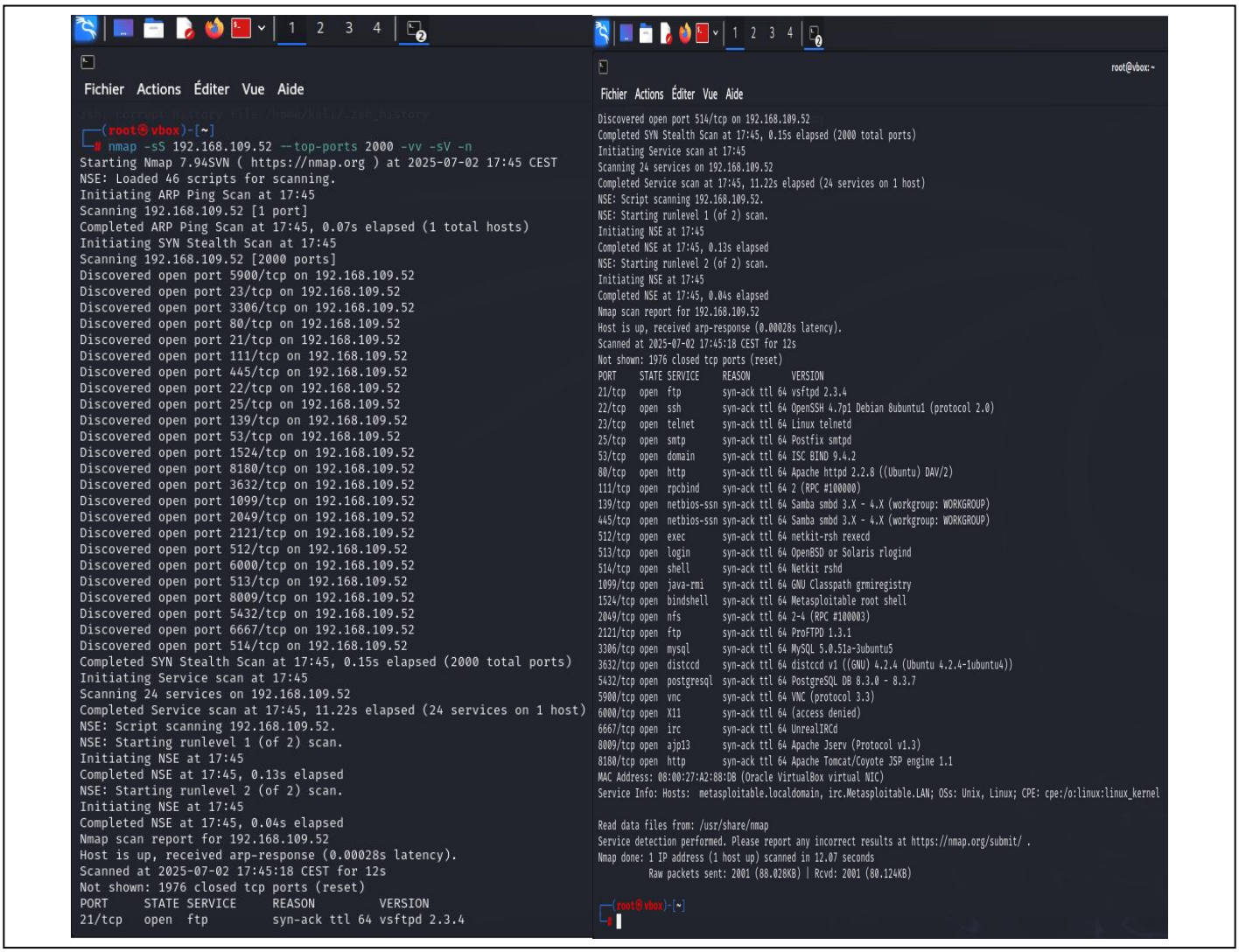

Figure 24. Full scan

4.4.2 Le capture réseau

On lance la capture réseau sur le machine ciblé par la commande :

```
sudo tcpdump -i eth0 -w capture_attack.pcap
```

Après l'exploite de vulnérabilité on arrêter le capture par ctrl+c . Cela va capturer tout le trafic sur l'interface réseau eth0 et l'enregistrer dans un fichier capture_attack.pcap

4.4.3 Définir la version

Identifier précisément les **versions des services** ou des logiciels qui tournent sur les ports ouverts d'une machine cible.

Depuis Kali, on entre dans Metasploit de Kali avec la commande msfconsole, il affiche le nombre des exploits, payloads. Après, on veut voir les exploits disponibles avec la commande show exploits .

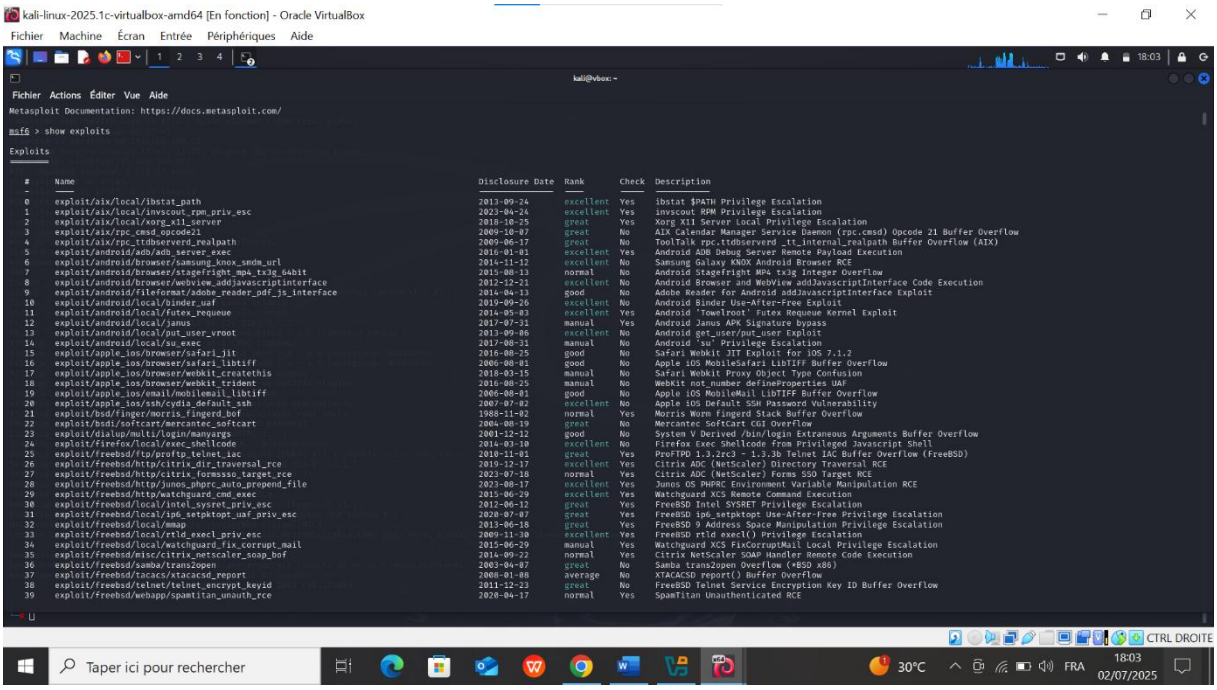

Figure 25. liste des exploits

puis on choisit une version à exploiter en faisant la recherche de l'exploit, ex : search vsftpd 2.3.4. Après, on continue l'exploit pour ouvrir une session vers le service choisi pour réaliser l'attaque

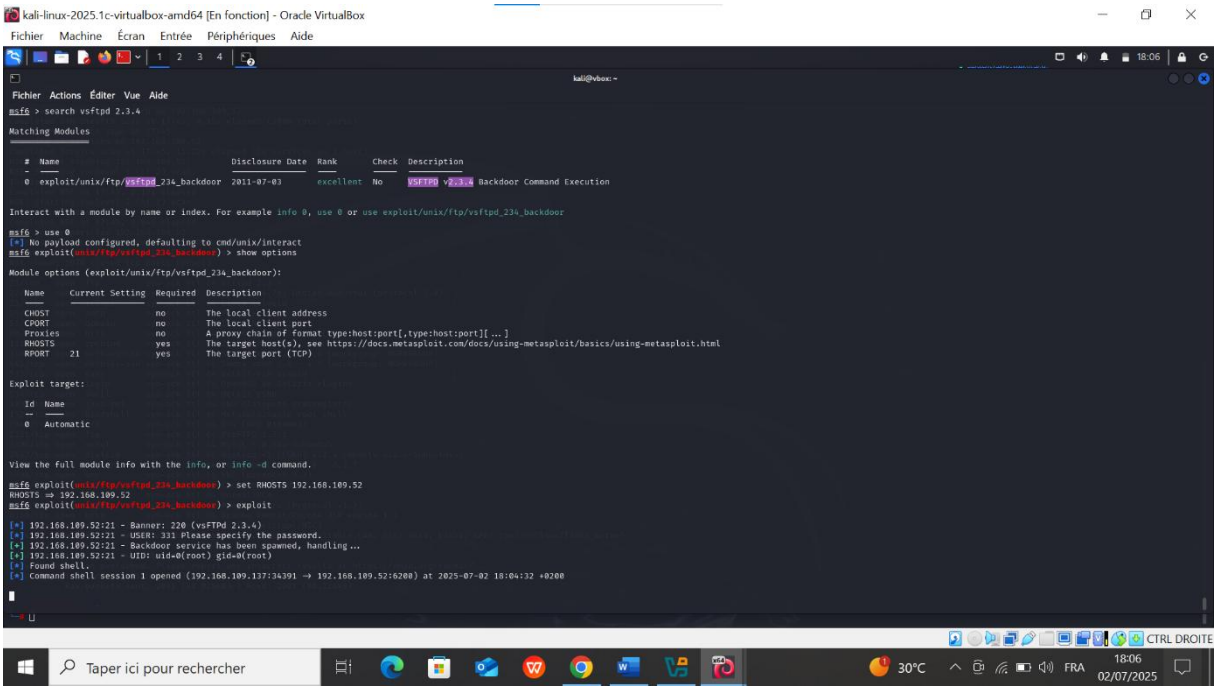

Figure 26. exploit de version choisi

4.4.4 Exploiting DVWA

DVWA (Damn Vulnerable Web Application) est une application web délibérément vulnérable, conçue pour l'entraînement à la sécurité informatique, l'apprentissage des attaques courantes, et le test des outils de sécurité.

Dans notre solution, nous avons créé un fichier shell.php pour exploiter DVWA dans Kali Linux, permettant l'exécution d'une liste de commandes sur le système cible.

Un fichier shell.php est un **script PHP malveillant ou utilitaire** qui permet à un attaquant (ou un testeur en sécurité) d'exécuter des **commandes système** à distance sur un serveur web.

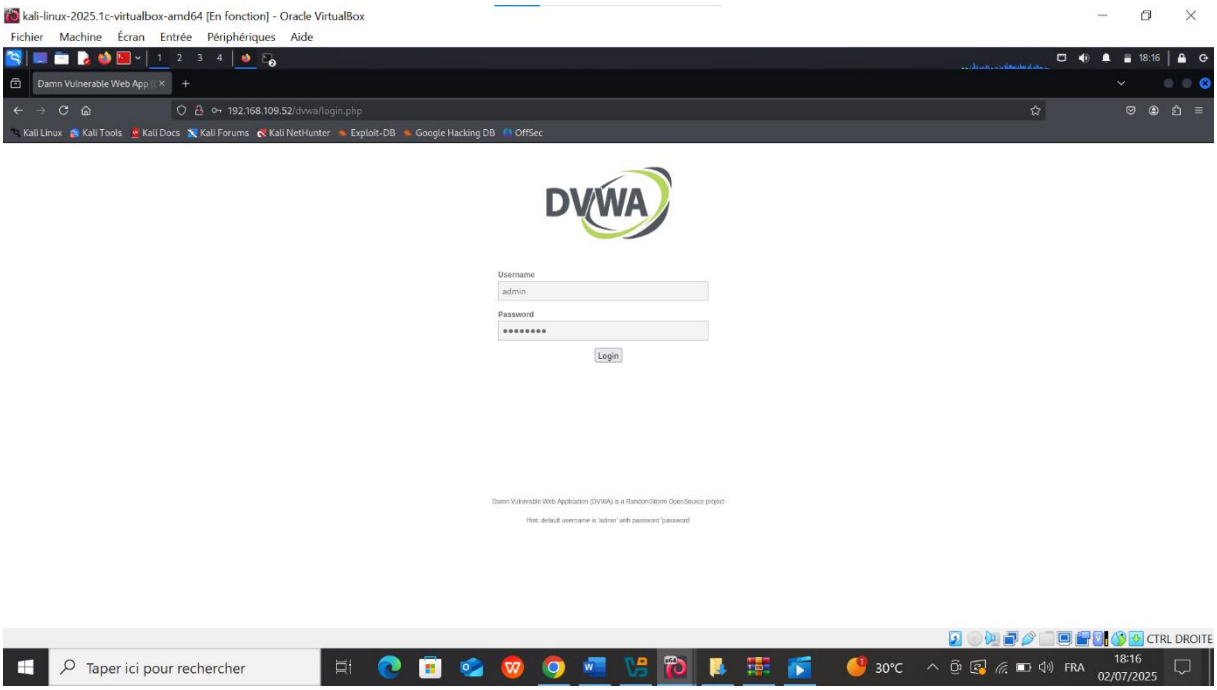

Figure 27. interface de DVWA

Après plusieurs configurations (commandes) sur kali on a réalisé cette attaque DWAV

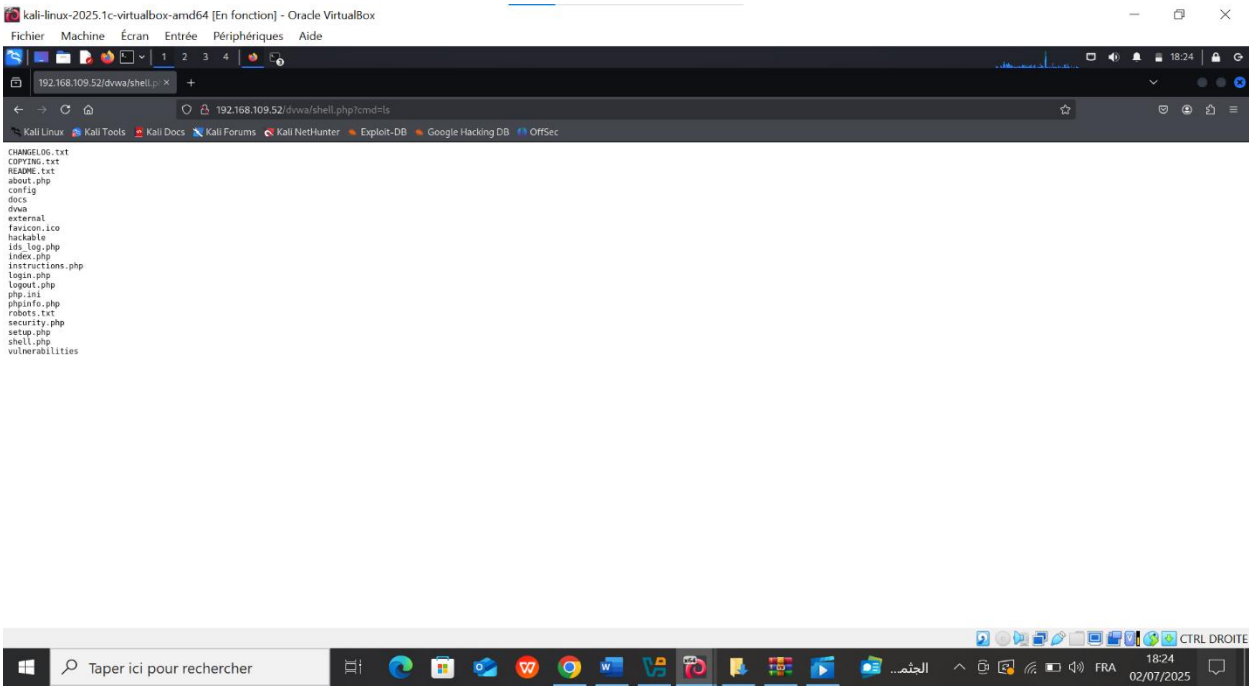

Figure 28. Exploit l'interface DVWA

4.5 KNN pour la classification des attaques

Nous allons tester l'algorithme KNN en utilisant la base test KDD :

4.5.1 Chargement de données

Depuis google colab on recharger le NSL_KDD data set en deux fichier : train-set et test-set par leur lien :

```
import pandas as pd

import numpy as np

import sys

import sklearn

import io

import random

train_url = 'https://raw.githubusercontent.com/merteroglu/NSL-KDD-Network-Instrusion-Detection/master/NSL_KDD_Train.csv'

test_url = 'https://raw.githubusercontent.com/merteroglu/NSL-KDD-Network-Instrusion-Detection/master/NSL_KDD_Test.csv'
```

4.5.2 Apprentissage automatique

Pour appliquer l'algorithme d'apprentissage KNN ou bien autre :

```
from sklearn.neighbors import KNeighborsClassifier

clf_KNN_DoS=KNeighborsClassifier()

clf_KNN_Probe=KNeighborsClassifier()

clf_KNN_R2L=KNeighborsClassifier()

clf_KNN_U2R=KNeighborsClassifier()

clf_KNN_DoS.fit(X_DoS, Y_DoS.astype(int))

clf_KNN_Probe.fit(X_Probe, Y_Probe.astype(int))

clf_KNN_R2L.fit(X_R2L, Y_R2L.astype(int))

clf_KNN_U2R.fit(X_U2R, Y_U2R.astype(int))

#DoS

Y_DoS_pred=clf_KNN_DoS.predict(X_DoS_test)

# Create confusion matrix

print(pd.crosstab(Y_DoS_test, Y_DoS_pred, rownames=['Actual attacks'],
colnames=['Predicted attacks']))

#Probe

Y_Probe_pred=clf_KNN_Probe.predict(X_Probe_test)

# Create confusion matrix

print(pd.crosstab(Y_Probe_test, Y_Probe_pred, rownames=['Actual attacks'],
colnames=['Predicted attacks']))

#R2L

Y_R2L_pred=clf_KNN_R2L.predict(X_R2L_test)

# Create confusion matrix

print(pd.crosstab(Y_R2L_test, Y_R2L_pred, rownames=['Actual attacks'],
colnames=['Predicted attacks']))

#U2R

Y_U2R_pred=clf_KNN_U2R.predict(X_U2R_test)
```

```
# Create confusion matrix

print(pd.crosstab(Y_U2R_test, Y_U2R_pred, rownames=['Actual attacks'], colnames=['Predicted attacks']))

#Accuracy

#DoS Accuracy

from sklearn.model_selection import cross_val_score

from sklearn import metrics

print("\nDoS Accuracy")

accuracy = cross_val_score(clf_KNN_DoS, X_DoS_test, Y_DoS_test, cv=10, scoring='accuracy')

print("Accuracy: %0.5f (+/- %0.5f)" % (accuracy.mean(), accuracy.std() * 2))

precision = cross_val_score(clf_KNN_DoS, X_DoS_test, Y_DoS_test, cv=10, scoring='precision')

print("Precision: %0.5f (+/- %0.5f)" % (precision.mean(), precision.std() * 2))

recall = cross_val_score(clf_KNN_DoS, X_DoS_test, Y_DoS_test, cv=10, scoring='recall')

print("Recall: %0.5f (+/- %0.5f)" % (recall.mean(), recall.std() * 2))

f = cross_val_score(clf_KNN_DoS, X_DoS_test, Y_DoS_test, cv=10, scoring='f1')

print("F-measure: %0.5f (+/- %0.5f)" % (f.mean(), f.std() * 2))

#Probe Accuracy

print("\nProbe Accuracy")

accuracy = cross_val_score(clf_KNN_Probe, X_Probe_test, Y_Probe_test, cv=10, scoring='accuracy')

print("Accuracy: %0.5f (+/- %0.5f)" % (accuracy.mean(), accuracy.std() * 2))

precision = cross_val_score(clf_KNN_Probe, X_Probe_test, Y_Probe_test, cv=10, scoring='precision_macro')

print("Precision: %0.5f (+/- %0.5f)" % (precision.mean(), precision.std() * 2))

recall = cross_val_score(clf_KNN_Probe, X_Probe_test, Y_Probe_test, cv=10, scoring='recall_macro')

print("Recall: %0.5f (+/- %0.5f)" % (recall.mean(), recall.std() * 2))
```

```
f = cross_val_score(clf_KNN_Probe, X_Probe_test, Y_Probe_test, cv=10, scoring='f1_macro')

print("F-measure: %0.5f (+/- %0.5f)" % (f.mean(), f.std() * 2))

#R2L Accuracy

print("\nR2L Accuracy")

accuracy = cross_val_score(clf_KNN_R2L, X_R2L_test, Y_R2L_test, cv=10,
scoring='accuracy')

print("Accuracy: %0.5f (+/- %0.5f)" % (accuracy.mean(), accuracy.std() * 2))

precision = cross_val_score(clf_KNN_R2L, X_R2L_test, Y_R2L_test, cv=10,
scoring='precision_macro')

print("Precision: %0.5f (+/- %0.5f)" % (precision.mean(), precision.std() * 2))

recall = cross_val_score(clf_KNN_R2L, X_R2L_test, Y_R2L_test, cv=10,
scoring='recall_macro')

print("Recall: %0.5f (+/- %0.5f)" % (recall.mean(), recall.std() * 2))

f = cross_val_score(clf_KNN_R2L, X_R2L_test, Y_R2L_test, cv=10, scoring='f1_macro')

print("F-measure: %0.5f (+/- %0.5f)" % (f.mean(), f.std() * 2))

#U2R Accuracy

print("\nU2R Accuracy")

accuracy = cross_val_score(clf_KNN_U2R, X_U2R_test, Y_U2R_test, cv=10,
scoring='accuracy')

print("Accuracy: %0.5f (+/- %0.5f)" % (accuracy.mean(), accuracy.std() * 2))

precision = cross_val_score(clf_KNN_U2R, X_U2R_test, Y_U2R_test, cv=10,
scoring='precision_macro')

print("Precision: %0.5f (+/- %0.5f)" % (precision.mean(), precision.std() * 2))

recall = cross_val_score(clf_KNN_U2R, X_U2R_test, Y_U2R_test, cv=10,
scoring='recall_macro')

print("Recall: %0.5f (+/- %0.5f)" % (recall.mean(), recall.std() * 2))

f = cross_val_score(clf_KNN_U2R, X_U2R_test, Y_U2R_test, cv=10, scoring='f1_macro')

print("F-measure: %0.5f (+/- %0.5f)" % (f.mean(), f.std() * 2))
```

sklearn.metrics.classification_report: résumé textuel de la précision, rappel, score F1 pour chaque classe. Dictionnaire retourné si output_dict est true. Le dictionnaire a la structure suivante:

Les résultats de la classification :

```
Predicted attacks     0     1
Actual attacks
0                  9422   289
1                  1573  5887
Predicted attacks     0     2
Actual attacks
0                  9437   274
2                  1272  1149
Predicted attacks     0  3
Actual attacks
0                  9706  5
3                  2883  2
Predicted attacks     0  4
Actual attacks
0                  9711  0
4                    65  2

DoS Accuracy
Accuracy: 0.99715 (+/- 0.00278)
Precision: 0.99678 (+/- 0.00383)
Recall: 0.99665 (+/- 0.00344)
F-measure: 0.99672 (+/- 0.00320)

Probe Accuracy
Accuracy: 0.99077 (+/- 0.00403)
Precision: 0.98606 (+/- 0.00675)
Recall: 0.98508 (+/- 0.01137)
F-measure: 0.98553 (+/- 0.00645)

R2L Accuracy
Accuracy: 0.96729 (+/- 0.00727)
Precision: 0.95304 (+/- 0.01240)
Recall: 0.95467 (+/- 0.01351)
F-measure: 0.95377 (+/- 0.01030)

U2R Accuracy
Accuracy: 0.99703 (+/- 0.00281)
Precision: 0.93143 (+/- 0.14679)
Recall: 0.85073 (+/- 0.17639)
F-measure: 0.87831 (+/- 0.11390)
```

4.6 Conclusion

Dans ce chapitre, nous avons présenté la mise en œuvre technique de notre solution de détection et de classification des malwares basée sur l'algorithme d'apprentissage supervisé KNN. À travers le traitement du jeu de données NSL-KDD.

Les résultats obtenus montrent que le KNN, bien que basique, peut offrir des performances satisfaisantes en matière de détection d'intrusions, surtout lorsqu'il est correctement entraîné et optimisé. Cette solution ouvre ainsi la voie à des travaux futurs visant à intégrer d'autres algorithmes plus avancés pour renforcer davantage la sécurité des systèmes informatiques.

Conclusion Générale

Dans un contexte où les cybermenaces ne cessent de croître en sophistication et en fréquence, il devient impératif de développer des solutions de sécurité à la fois efficaces, réactives et intelligentes. Ce mémoire a permis de démontrer la pertinence et la puissance de l'intelligence artificielle, en particulier de l'apprentissage automatique, comme levier pour améliorer la détection des logiciels malveillants.

À travers l'étude et la mise en œuvre de l'algorithme K-Nearest Neighbors (KNN), nous avons exploré une approche simple mais robuste pour la classification des comportements malveillants. Notre travail s'est appuyé sur une solide base théorique en cybersécurité et en machine learning, avant de déboucher sur une application concrète, testée à l'aide du jeu de données NSL-KDD. Les résultats obtenus confirment que le KNN peut offrir des performances satisfaisantes, avec des taux de détection élevés et une adaptabilité intéressante face à la nature évolutive des menaces.

Cependant, il convient de souligner que cette approche présente également certaines limites, notamment en termes de sensibilité au choix des paramètres et au traitement préalable des données. Ces aspects ouvrent des perspectives d'amélioration, notamment par l'intégration de techniques hybrides ou l'utilisation d'algorithmes plus avancés comme les forêts aléatoires ou les réseaux de neurones.

En conclusion, ce mémoire met en lumière l'intérêt croissant de l'IA appliquée à la cybersécurité, et propose une première contribution dans cette direction, en illustrant comment un algorithme aussi accessible que KNN peut constituer un socle pour des solutions de sécurité intelligentes, autonomes et évolutives. Ce travail pose ainsi les bases d'approfondissements futurs visant à renforcer davantage la résilience des systèmes informatiques face aux menaces numériques actuelles et émergentes.

Bibliographie

[1] PID, «Sécurité des systèmes d'information,» *L'informatique pour tous,* 20 Février 2018.

[2] GROUPE IGENSIA EDUCATION, LA SÉCURITÉ INFORMATIQUE, Établissement d'enseignement supérieur technique privé, Association à but non lucratif.

[3] Panda Security, «evaluation des vulnerabilité,» 21 11 2022. [En ligne]. Available: https://www.pandasecurity.com/fr/mediacenter/evaluation-vulnerabilites/.

[4] CYBER Mangement School, la vulnérabilité informatique, france.

[5] Torii Security, «Tests d'intrusion : tout ce qu'il faut savoir,» 25 5 2021. [En ligne]. Available: https://torii-security.fr/tests-dintrusion-tout-ce-quil-faut-savoir/.

[6] J. Pierre, «Mailinblack. Pentest : Un outil crucial pour la sécurité informatique,» 28 4 2025. [En ligne]. Available: https://www.mailinblack.com/ressources/glossaire/pentest-un-outil-crucial-pour-la-securite-informatique. [Accès le 10 5 2025].

[7] I. Belcic, «Qu'est-ce qu'un malware et comment s'en protéger ?,» 19 1 2023. [En ligne]. Available: https://www.avast.com/fr-fr/c-malware.

[8] ninjaone, «10 meilleures solutions de protection contre les logiciels malveillants 2025,» 13 10 2024. [En ligne]. Available: https://www.ninjaone.com/fr/blog/les-meilleures-solutions-de-protection-contre-les-logiciels-malveillants/. [Accès le 10 5 2025].

[9] S. Y. e. T. D. Salim, «Université Blida 1,» 15 12 2020. [En ligne]. Available: https://di.univ-blida.dz/jspui/bitstream/123456789/9985/1/Slimane%20Yacine%20et%20Tahar%20Djoudi%20Salim.pdf. [Accès le 27 5 2025].

[10] M. Poissard, Compositeur, *IA COGNITO – EP03 : Le Machine Learning.* [Enregistrement audio]. Neovision. 7 JUIN 2025.

[11] Equipe blent, «L'apprentissage supervisé : définition et exemples,» *Machine Learning,* 12 4 2022.

[12] A. C. &. S. S. Shrivastava, chez *Lecture Notes in Networks and Systems*, International Conference on Computer & Communication Technologies éd., vol. 898, S. —. r. d. l. p. d. l. s. LNNS., Éd., 2023.

[13] Y. Z. Yalei Ding, «Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks,» chez *CSAI '18: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*.

[14] «Improving the intrusion detection system for NSL-KDD dataset based on PCA-fuzzy clustering-KNN,» chez *2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Marrakesh, Maroc, 16–19 octobre 2018.

[15] B. M. Amin, «Détection des intrusions basée sur l'apprentissage automatique dans les systèmes IdO (Internet des Objets),» Juin 2022.

[16] Microsoft Corporation, «Machine learning-based protection in Microsoft Defender Antivirus,» 2020. [En ligne]. Available: https://learn.microsoft.com/en-us/microsoft-365/security/defender-

endpoint/machine-learning?view=o365-worldwide. [Accès le 27 5 2025].

[17] BlackBerry, «BlackBerry (Cyclance). Artificial Intelligence and Predictive Advantage in Cybersecurity.,» 2019. [En ligne]. Available: https://www.blackberry.com/us/en/solutions/artificial-intelligence-cybersecurity. [Accès le 27 5 2025].

[18] CrowdStrike, «How CrowdStrike Uses Machine Learning to Detect Malware.,» 2021. [En ligne]. Available: https://www.crowdstrike.com/epp-101/machine-learning/. [Accès le 27 5 2025].

[19] Sophos, «Deep learning technology in Intercept X,» 2020. [En ligne]. Available: https://www.sophos.com/en-us/products/intercept-x/tech-specs.aspx. [Accès le 27 5 2025].

[20] Malwarebytes, «Machine learning and malware detection,» 2022. [En ligne]. Available: https://www.malwarebytes.com/resources/files/white-papers/mb-machine-learning-white-paper.pdf. [Accès le 27 5 2025].

[21] A. H. S. A. A. Srinivas B. Mukkamala, «Intrusion Detection Using Ensemble of Soft Computing Paradigms,» *Journal of Network and Computer Applications,* vol. 28, n° %12, pp. 167-182, Avril 2005.

[22] Anna L. Buczak et Erhan Guven, «A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,» *IEEE Communications Surveys & Tutorials,* vol. 18, n° %12, pp. 1153-1176, 1 avril 2016.

[23] J. Robert, «KNN : Découvrez cet algorithme de Machine Learning,» 19 11 2020. [En ligne]. Available: https://datascientest.com/knn. [Accès le 27 5 2025].

[24] A. e. V. Varun Chandola, «Anomaly Detection: A Survey,» *ACM Computing Surveys,* vol. 41, n° %13, p. 1–58, 30 Juiellet 2009.

[25] V. P. Robin Sommer, « Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,» chez *IEEE Symposium on Security and Privacy (aussi appelé IEEE S&P ou Oakland Conference)*, Berkeley/Oakland, Californie, États-Unis, 16–19 mai 2010.

[26] Scikit-learn, «Scikit-learn Documentation,» 2023. [En ligne]. Available: https://scikit-learn.org/stable/modules/neighbors.html. [Accès le 27 5 2025].

[27] S. A. Abdur Rehman Sakhawat, «A secure healthcare 5.0 system based on blockchain technology entangled with federated learning technique,» *Computers in Biology and Medicine,* September 2022.

[28] S. J. F. W. L. W. P. A. &. C. P. K. Stolfo, «Cost-based Modeling and Evaluation for Data Mining With Application to Fraud and Intrusion Detection: Results from the JAM Project,» chez *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX 2000)*, 2000.

[29] T. M. C. e. P. E. Hart, «Nearest Neighbor Pattern Classification,» vol. 13, n° %11, pp. 21-27, janvier 1967.

[30] M. S. V. K. Pang-Ning Tan, «Classification: Basic Concepts, Decision Trees, and Model Evaluation,» chez *Introduction to Data Mining (2ᵉ édition)*, pearson, Éd., USA, 2019, p. 860.

[31] E. W. &. A. Mahbod Tavallaee, «A Detailed Analysis of the NSL-KDD CUP 99 Data Set,» Ottawa, Canada, 2009.

[32] UCI Machine Learning Repository, «NSL-KDD Dataset».

[33] C.-H. R. L. Y.-C. L. K.-Y. T. Hung-Jen Liao, «Intrusion Detection System: A Comprehensive Review,» *Journal of Network and Computer Applications,* vol. 36, n° %11, pp. 16-24, 2013.

[34] B. I. e. A. Yadav, « Performance Analysis of NSL-KDD Dataset Using ANN,» *Procedia Computer Science,* p. 92–96, 2015.

[35] J. J. &. F. Salvador García, «Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study,» *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE TPAMI),* vol. 34, n° %13, p. 417–435, 2012.

[36] X. Shang-Zhang & Li, «K-Nearest Neighbors with Mutual Information for Cost- Sensitive Classification,» *Knowledge-based Systems,* vol. 163, p. 276- 285, 2019.

[37] Z. A. O. M. Z. A. N. Wesam L. Al-Yaseen, «Multi-level Hybrid Support Vector Machine and Extreme Learning Machine Based on Modified K-means for Intrusion Detection System,» *Expert Systems with Applications,* vol. 67, p. 296–303, 2017.

[38] BDM, «BDM tools : Oracle VM VirtualBox,» BDM, [En ligne]. Available: https://www.blogdumoderateur.com/tools/oracle-vm-virtualbox/. [Accès le 13 6 2025].

[39] Bilty, «Bilty: definition de kali linux,» [En ligne]. Available: https://bility.fr/definition-kali-linux/. [Accès le 13 6 2025].

[40] RAPID7, «RAPID7: Metasploit.Metasploitable 2 Exploitability Guide,» [En ligne]. Available: https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/. [Accès le 13 6 2025].

[41] A. Jaillet, «Google Colab : la force du cloud pour l'apprentissage automatique,» 8 fevrier 2024. [En ligne]. Available: https://datascientest.com/google-colab-tout-savoir.