RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ A. MIRA-BEJAIA

جامعة بجاية
Tasdawit n Bgayet
Université de Béjaïa

FACULTÉ DES SCIENCES EXACTES

DÉPARTEMENT D'INFORMATIQUE

MÈMOIRE

EN VUE DE L'OBTENTION DU DIPLÔME DE

MASTER RECHERCHE

Domaine : Mathématiques et Informatique **Filière :** Informatique

Spécialité : Réseaux et Sécurité

Spécialité : Systèmes d'Information Avancés

Présenté par

M. DJEFEL Juba

Mlle. MEBROUK Thafath

Thème

Application des méthodes de Machine Learning pour la détection des Advanced Persistent Threat

Soutenu le 26 Juin 2024 **Devant le jury composé de :**

Nom et Prénom	Grade		
Mme. BOUALLOUCHE Louiza	**Professeur**	**Université de Béjaia**	**Présidente**
M. MOKTEFI Mohand	**MCB**	**Université de Béjaia**	**Raporteur**
Mme. OUYAHIA Samira	**MCB**	**Université de Béjaia**	**Examinatrice**

Année Universitaire : 2023/2024

⁕ *Remerciements* ⁕

Avant tout, nous remercions le Bon ≪**Dieu**≫ , le Tout-Puissant, de nous avoir accordé la force, le courage et la patience dont nous avions besoin pour mener à bien ce travail.

Nous souhaitons exprimer notre profonde gratitude à notre encadrant, **Dr MOKTEFI Mohand**, pour sa patience, son expertise et ses conseils avisés tout au long de ce travail. Sa guidance éclairée et son soutien constant ont été essentiels à l'aboutissement de ce mémoire.

Nous remercions Madame **BOUALLOUCHE Louiza** pour l'honneur qu'elle nous a fait en acceptant de présider le jury. Nos remerciements s'adressent également à **Dr OUYAHIA Samira**, membre du jury, pour l'intérêt qu'elle a manifesté envers notre travail en acceptant de l'examiner et de l'enrichir par ses précieuses suggestions.

Nous exprimons notre reconnaissance à nos famille pour leur soutien moral inconditionnel et leur présence réconfortante lors des moments difficiles.

Nous remercions aussi l'ensemble des enseignants et des membres de l'administration de l'université qui ont participé a notre formation et nous ont permis d'acquérir les connaissances nécessaires pour la réalisation de ce modeste travail.

✳ *Dédicaces* ✳

Je dédie ce mémoire .....

À mes chers parents

Je profite de cette occasion pour exprimer ma profonde gratitude envers vous. Votre soutien continu et votre amour infini ont été ma source de force et d'inspiration tout au long de ce parcours académique. Vous êtes le socle sur lequel je me suis construit et cette réussite est aussi la vôtre.

À ma sœur et mes frères bien-aimés

Cette dédicace est un témoignage de l'amour et du lien familial qui nous unit. Votre soutien moral et votre affection ont été des essentiels dans la réussite de ce projet. Chaque moment partagé avec vous a renforcé ma détermination . Merci d'être toujours présents pour moi.

À ma binôme Thafath

Je tiens à exprimer ma reconnaissance pour ta collaboration et ton engagement tout au long de ce travail. Notre collaboration a été une source d'inspiration et de motivation. Tes idées, ton soutien et ton travail acharné ont contribué de manière significative à la réussite de ce projet.

Je tiens également à exprimer ma gratitude envers toutes les personnes qui, par leurs conseils, leur soutien et leur expertise, ont contribué à enrichir ce travail. Leur impact ne peut être sous-estimé, et je leur suis reconnaissant pour leur précieuse contribution.

DJEFEL Juba

✳ *Dédicaces* ✳

Je dédie ce mémoire .....

À ma très chère mère

Quoi que je fasse ou que je dise je ne saurai comment te remercie comme il se doit, ton affection me couvre, ta bienveillance me guide et ta présence à mes côtés a toujours été ma source de force.

À mon très cher père

Tu as toujours été à mes côtés pour me soutenir et m'encourager et affronter les différents obstacles.

À mes frères et sœurs bien-aimées

Je tiens à exprimer ma plus profonde gratitude à mes frères et sœurs ainsi qu'a leurs enfants. Votre soutien , votre amour et votre présence bienveillante m'ont permis de persévérer dans les moments difficiles. À chacun d'entre vous, je dis Merci pour votre patience et votre écoute .

À mon binôme Juba

Ta détermination et ton esprit d'équipe ont été des piliers dans l'accomplissement de ce mémoire. Merci pour ton soutien constant et tes idées brillantes. Ça était un privilège de travailler à tes côtés.

MEBROUK Thafath

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABRÉVIATIONS

ACC	Accuracy
ACL	Access Control List
ACP	Analyse en Composantes Principales
AES	Advanced Encryption Standard
API	Application Programming Interface
APT	Advanced Persistent Threat
BDD	Base De Données
CC	Commande et Contrôle
CDT	Composition based Decision Tree
CPU	Central Processing Unit
DDOS	Distributed Denial of Service
DES	Data Encryption Standard
DNS	Domain Name System
DNN	Deep Neural Network
DOH	DNS Over HTTPS
DOS	Denial of Service
DT	Decision Tree
ECC	Elliptic Curve Cryptography
FNR	False Negative Rate
FPR	False Positive Rate
FW	Firewall

GBDT	Gradient Boosting Decision Tree
GPU	Graphics Processing Unit
HIDS	Host-based Intrusion Detection System
HTTPS	Hyper Text Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IP	Internet Protocol
KNN	K-Nearest Neighbors
LR	Logistic Regression
ML	Machine Learning
MLAPT	Machine Learning Advanced Persistent Threat
MLP	Multi-Layer Perceptron
NIDS	Network-based Intrusion Detection System
NSL-KDD	National Science Laboratory-Knowledge Discovery in Databases
OSINT	Open Source INTelligence
PC	Pearson Correlation
PRE	Precision
R2L	Remote-to-Local
RBF	Radial Basis Function
RC4	Rivest Cipher 4
RSA	Rivest, Shamir, Adleman
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SVM	Support Vector Machine
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TPR	True Positive Rate
TXT	Text Record
U2R	User to Root
UDP	User Datagram Protocol
USB	Universal Serial Bus
VPN	Virtual Private Network
XGBOOST	eXtreme Gradient Boosting

Introduction générale

La sécurité informatique représente un défi crucial pour les entités, qu'elles soient des organisations, des gouvernements ou des individus. Face à la multiplication de menaces cybernétiques sophistiquées, le repérage et la prévention des intrusions s'avèrent être des défis majeurs. Parmi ces menaces, les Advanced Persistent Threat (APT) se distinguent en combinant des techniques complexes et discrètes pour pénétrer et maintenir leur présence dans les systèmes informatiques sur de longues périodes.

Les APT posent une menace sérieuse en raison de leur origine souvent liée à des groupes hautement qualifiés bénéficiant de ressources financières et humaines considérables. Ces attaques ciblent spécifiquement des secteurs critiques tels que la finance, l'énergie et la défense, avec pour objectif de dérober des données sensibles ou de perturber des infrastructures vitales,leurs répercussions peuvent être catastrophiques sur les plans économique, politique et social.

Devant cette menace croissante, les dispositifs de protection classiques tels que les pare-feux, les antivirus ou les systèmes de détection d'intrusions atteignent leurs limites. Il devient essentiel de concevoir des stratégies plus avancées et flexibles pour repérer efficacement les APT.

C'est dans ce contexte que les techniques d'apprentissage automatique offrent des perspectives prometteuses pour renforcer la détection des APT. En analysant de grandes quantités de données, ces méthodes permettent d'identifier des schémas complexes et d'anticiper les comportements malveillants des attaquants.

L'objectif principal de cette étude est d'explorer de nouvelles approches basées sur l'apprentissage automatique et de concevoir un système de détection performant, capable d'identifier les APT de manière précise à partir de l'analyse des données réseau.

Ce mémoire est structuré comme suit :

Dans le **premier chapitre**, nous allons donner un aperçu des bases de la sécurité informatique et des attaques réseau, puis étudier le processus d'évolution des APT, leurs particularités en analysant certains groupes connus et les méthodes qu'ils emploient pour leurs attaques. Nous conclurons en examinant quelques exemples de méthodes de défense traditionnelles et leurs restrictions.

Dans le **deuxième chapitre**, nous présenterons une revue approfondie de l'utilisation du Machine Learning pour aborder les Menaces Persistantes Avancées. Après avoir expliqué les principes fondamentaux de ce sujet, nous examinerons les traveaux les plus récents dans ce domaine. Une catégorisation par approches est suggérée, permettant de mettre en avant les diverses méthodes utilisées et les résultats obtenus. Enfin, une analyse constructive permettra d'identifier les avantages, les inconvénients et les pistes d'amélioration des solutions actuelles.

Le **troisième chapitre** examinera en détail tout le processus de conception, de mise en œuvre et d'évaluation de notre solution.Nous exposerons la proposition architecturale globale, nous présenterons le jeu de données utilisé, les outils et environnements employés. Ensuite, nous détaillerons les diverses étapes importantes telles que le traitement préalable des données, la création du modèle et son entraînement. Enfin, nous évaluerons les résultats en les confrontant aux méthodes déjà existantes.

Nous terminerons ce mémoire par une **conclusion** et des **perspectives**, proposant ainsi des pistes pour des recherches futures.

Chapitre 1

Généralités sur la sécurité et les menaces persistantes avancées

1.1 Introduction

La multiplication des cybermenaces, portées par des attaques de plus en plus sophistiquées, représente un risque critique pour les organisations, avec un impact potentiel sur leurs infrastructures, leurs données sensibles et leur pérennité. Parmi les plus redoutables figurent les attaques dites **APT**, qui combinent des techniques avancées pour contourner les défenses traditionnelles et rester persistantes dans le temps.

L'objectif de ce chapitre est d'apporter un éclairage approfondi sur la problématique des APT. Nous commencerons par consolider les fondamentaux en sécurité informatique et en attaques réseau, avant d'examiner le cycle de vie des APT, leurs caractéristiques spécifiques en passant en revue quelques groupes connus et les techniques qu'ils adoptent pour mener leurs attaques. Nous terminerons avec l'illustration de quelques moyens de défense traditionnels et de leurs limites.

1.2 Sécurité informatique

1.2.1 Définition

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridiques et humains mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles[11].

1.2.2 Objectifs de la sécurité

La sécurité informatique vise à protéger les systèmes d'information contre diverses menaces, en assurant plusieurs objectifs de sécurité [29].

Confidentialité

Il s'agit de limiter l'accès aux informations sensibles aux personnes autorisées seulement.

Intégrité

Garantie que les informations n'ont pas été modifiées ni détruites sans autorisation.

Disponibilité

Vise à garantir que les utilisateurs autorisés puissent réellement accéder aux informations et systèmes selon leurs besoins et dans les délais prévus.

Authentification

Vérifie l'identité d'un utilisateur, d'un système ou d'une entité, garantissant que seules les parties légitimes ont accès aux ressources.

Contrôle d'accès

La sécurité des systèmes d'information repose en grande partie sur le contrôle d'accès. Ce principe consiste à limiter l'accès aux ressources uniquement aux utilisateurs possédant les autorisations nécessaires.

Non-répudiation

Apporte la preuve des actions et transactions réalisées par les entités d'un système d'information permettant ainsi de prouver sans ambiguïté qu'une personne était aux commandes à un instant T pour réaliser une action donnée.

La figure 1.1 présente les objectifs de la sécurité qui constituent la base d'une protection efficace des systèmes d'information.

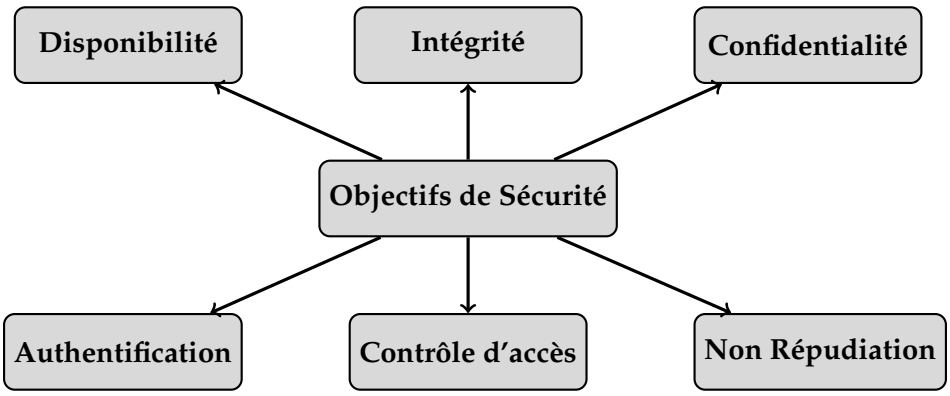

FIGURE 1.1 – Objectifs de la sécurité.

1.3 Attaque informatique

1.3.1 Définition

Une attaque informatique est toute tentative d'obtenir un accès non autorisé à un ordinateur, un système informatique ou un réseau informatique dans le but de causer des dommages.

Les attaques informatiques visent à désactiver, perturber, détruire ou contrôler les systèmes informatiques, ou à modifier, bloquer, supprimer, manipuler ou voler les données contenues dans ces systèmes.

1.3.2 Catégories des attaques informatiques

Comprendre la nature d'une attaque est fondamental pour analyser cet impact potentiel et choisir les défenses appropriées. Passons en revue les catégories d'une attaque.

Elle est subdivisée en deux catégories : l'attaque basée sur le comportement de l'attaquant et l'attaque basée sur la position de l'attaquant [20].

Attaque basée sur le comportement de l'attaquant

Ce type comprend deux catégories d'attaques qui sont les suivantes :

Attaque Active : ce type d'attaque modifie les messages, accède aux équipements d'un réseau ou perturber son fonctionnement. Contrairement aux attaques passives, celles-ci ont des conséquences détectables en raison des dommages qu'elles engendrent.

Attaque Passive : fait référence à l'écoute et la surveillance de la transmission sur le canal sans modifier les données du réseau. Les attaquants peuvent intercepter et capturer des données ou analyser le trafic réseau pour rechercher des informations sensibles telles que des mots de passe qui peuvent

être utilisés dans d'autres types d'attaques. Les attaques passives sont souvent indétectables, mais peuvent être évitées en chiffrant les transmissions entre les nœuds du réseau.

Basée sur la position de l'attaquant

Comprend deux types d'attaques qui sont les suivantes [20] :

Attaque interne : se produit lorsqu'une personne ayant un accès autorisé au système ou au réseau utilise délibérément cette autorisation pour compromettre la sécurité, pouvant résulter en la divulgation, la modification ou la destruction de données sensibles.

Attaque externe : se produit lorsqu'un individu ou une entité tente d'exploiter des vulnérabilités depuis l'extérieur d'un système ou d'un réseau pour accéder illégalement à des informations, perturber les opérations ou compromettre la sécurité.

1.4 Attaque réseaux

1.4.1 Définition

Les attaques réseaux impliquent des actions malveillantes visant à perturber les informations et services des réseaux informatiques en envoyant des flux de données compromettant l'intégrité, confidentialité ou disponibilité des systèmes. Elles varient de courriels ennuyeux à des intrusions dans des données sensibles et infrastructures critiques.

1.4.2 Catégories des attaques réseaux

Logiciels malveillants

Un logiciel malveillant est un programme ou un fichier conçu pour endommager un système informatique ou voler des données sensibles. Certains sont juste ennuyeux, d'autres peuvent causer des violations graves. Il existe plusieurs logiciels malveillants connus. Parmi eux, nous trouvons :

1. **Virus :** est un programme informatique inséré dans une application ou un système d'exploitation sans le consentement de l'utilisateur, pouvant causer des dommages et voler des informations [5].
2. **Ver informatique :** se propage entre les systèmes sans fichier hôte, causant des dommages comme la surutilisation des ressources et l'arrêt du système, contrairement aux virus qui doivent infecter un fichier [5].

3. **Cheval de Troie :** un programme nuisible que les utilisateurs installent en toute connaissance de cause, en le prenant pour un programme légitime. Il utilise des techniques de manipulation pour pénétrer dans l'appareil de ses victimes [5].
4. **Rootkit :** est un programme informatique qui permet à un pirate de contrôler à distance l'ordinateur d'une victime avec des privilèges d'administrateur. Les rootkits sont transmis via le hameçonnage, les pièces jointes malveillantes, les téléchargements infectés et peuvent être utilisés pour cacher d'autres logiciels malveillants comme les *keyloggers*.
5. **Ransomware :** est un logiciel qui utilise le cryptage pour bloquer l'accès aux données d'une victime jusqu'au paiement d'une rançon. Payer la rançon ne garantit pas la restitution des données par les attaquants.

Ingénerie sociale

Une pratique manipulatrice utilisée en cybersécurité, qui consiste à utiliser des stratégies psychologiques pour obtenir des données confidentielles ou encourager des actions spécifiques en exploitant la confiance, la crédulité ou d'autres aspects humains afin de contourner les niveaux de sécurité. Les attaquants peuvent recourir à diverses méthodes comme la manipulation émotionnelle, la simulation d'autorités de confiance ou la mise en place de scénarios trompeurs pour amener les individus à révéler des informations sensibles [48].

Spamming

Aussi connu sous le nom de pourriel, ce sont des messages non sollicités et indésirables. Envoyés en grand nombre à de nombreux destinataires, ils visent principalement la promotion commerciale, faisant la publicité de produits, services ou sites web. Cependant, les courriels de spam peuvent également inclure des arnaques, des chaînes de lettres et des pièces jointes malveillantes [16].

Phishing

Consiste à utiliser des messages frauduleux pour inciter les utilisateurs à divulguer des informations sensibles. Ces messages cherchent souvent à se faire passer pour des messages légitimes en imitant les caractéristiques visuelles de sources familières ou non menaçantes pour la victime [41].

Analyse de ports

L'analyse de port est une méthode fréquemment utilisée pour identifier les failles des systèmes réseau. Avant de s'introduire dans un système, l'attaquant collecte des informations sur l'hôte ou le réseau ciblé en effectuant des scans de port. Les réponses obtenues fournissent à l'attaquant des

données précieuses, telles que les adresses IP actives, les services disponibles et les types de protocoles en usage. Grâce à ces informations, les attaquants peuvent lancer des attaques réseau hautement destructrices, comme les attaques par déni de service distribué et la propagation de vers. Ainsi, la détection précoce des scans de port est essentielle pour éviter des dommages graves aux systèmes réseau[45].

Attaque par déni de service

Une attaque par déni de service **(DoS)** se produit lorsqu'un attaquant malveillant sature le système avec un trafic frauduleux et amplifié, épuisant ainsi les ressources réseau et perturbant le temps de CPU et/ou la bande passante des utilisateurs légitimes [40].

Nous distinguons habituellement deux types de déni de service :

1. **Déni de service par saturation :** consiste à submerger une machine de requête, afin qu'elle ne soit plus capable de répondre aux requêtes réelles.
2. **Déni de service par exploitation de vulnérabilités :** consiste à exploiter une faille du système distant afin de le rendre inutilisable.

Attaque par déni de service distribué

Une attaque par déni de service distribué (**DDoS**) utilise de multiples machines compromises appelées zombies qui attaquent en même temps une cible, causant ainsi un déni de service pour les utilisateurs. L'attaquant crée un botnet en choisissant les machines aléatoirement ou de manière topologique. Contrairement à une attaque DoS facile à contrer en bloquant simplement l'adresse IP, l'attaque DDoS est plus complexe en raison de la diversité des adresses sources [40].

L'attaque ***Smurf*** est une forme d'attaque par déni de service distribué, qui tire partie de la diffusion d'adresses IP pour surcharger les réseaux. Elle implique l'envoi en masse de paquets ***Internet Controle Message Protocole (ICMP)*** Echo Request (ping) à l'adresse de diffusion d'un réseau, avec l'IP de la cible comme adresse source. La réponse de toutes les machines du réseau noie la victime sous un flot de trafic, causant un déni de service.

1.5 Advanced Persistent Threat

1.5.1 Définition

Attaque sophistiquée menée par des acteurs malveillants qualifiés, bénéficie de ressources financières substantielles provenant d'une organisation ou d'un gouvernement, visant à infiltrer les systèmes de manière discrète pour maintenir un accès continu [37].

Le terme a été introduit pour la première fois dans le secteur militaire. Cette attaque se caractérise par trois termes, selon l'appellation[37] :

1. **Advanced :** fait référence à l'ensemble des ressources financières et moyens significatifs qui permettent aux attaquants de mener des attaques complexes. Ils déploient plusieurs vecteurs d'attaque en exploitant les vulnérabilités zero-day, les courriels d'hameçonnage et les injections ***Structured Query Language (SQL)*** pour maximiser leurs chances d'intrusion dans le système cible et échapper à la détection.
2. **Persistant** : les attaquants font preuve de rigueur, de détermination et de persévérance pour mener leurs attaques de manière efficace. Après s'être infiltrés, ils cherchent à rester discrets dans le réseau ciblé en combinant plusieurs techniques d'attaque. La durée d'une attaque APT est généralement de plusieurs mois, voire de plusieurs années [44].
3. **Threat** : la menace dans le cadre des APT se manifeste par la perte d'informations sensibles et confidentielles, ou à travers la mise hors service de plusieurs infrastructures des entreprises. Par conséquent, celles-ci doivent disposer de moyens avancés pour sécuriser leurs réseaux et infrastructures contre toute menace.

Le tableau 1.1 ci-dessous présente une comparaison des caractéristiques distinctives entre une attaque traditionnelle et une attaque ciblée de type APT [3].

Caractéristique	**Attaque Traditionnelle**	**Attaque APT**
Attaquant	une seule personne	groupe organisé et compétent
Cible	des individus	organisations, entreprises commerciales et gouvernements politiques
Objectif	gain financier, apprentissage	bénéfices financiers importants, divulgation d'informations sensibles
Techniques employées	connues de tous	furtives et hautement qualifiées
Ressources	peu de ressources	ressources importantes
Impact	généralement à court terme et peu conséquent	pertes financières , réputation des entreprises
Coût de mise en œuvre	faible	très élevé

TABLE 1.1 – Comparaison entre une attaque traditionnelle et une attaque APT [3].

1.5.2 Caractéristiques des APT

Contrairement aux menaces conventionnelles, les APT se démarquent par cinq caractéristiques distinctives mises en évidence par Chen et al. [9].

Cibles Spécifiques

Les menaces persistantes avancées(MPA) visent spécifiquement les gouvernements ou organisations avec des ressources intellectuelles stratégiques, notamment dans les secteurs financiers, politiques, éducatifs, énergétiques et de la santé. Cette stratégie implique la recherche délibérée d'informations sensibles dans des domaines clés, avec une sélection méthodique des cibles pour maximiser l'acquisition d'informations critiques et la perturbation des secteurs visés.

Objectifs clairs

Dans le domaine numérique en évolution constante, les APT se démarquent par leur nature subtile, visant particulièrement des entités de grande taille dans des secteurs critiques et poursuivant divers objectifs divisibles en deux catégories, d'une part les hackers qui cherchent à extorquer de l'argent, saboter des activités économiques et nuire à la réputation sur le marché mondial dans un but financier. D'autre part, des assaillants qui opèrent dans un but politique, cherchant à obtenir des informations sensibles, à semer la perturbation politique lors d'événements cruciaux et à affaiblir la confiance au sein des gouvernements.

Techniques Furtives et Évasives

Les APT opèrent de manière discrète afin de rester indétectables en se dissimulant dans le trafic réseau. Les attaques ***zero-day*** sont utilisées pour exploiter des vulnérabilités inconnues, tandis que le chiffrement est une autre stratégie visant à prolonger la discrétion de leur présence dans le réseau cible. Ces attaques évitent les mécanismes de détection classiques en se dissimulant au sein du flux d'informations, ce qui rend leur identification plus complexe.

Stratégies à long terme

Les APTévoluant dans la durée, restent indétectables dans le système cible pendant plusieurs mois, voire des années. Les attaquants ont ainsi la possibilité d'ajuster leurs plans, minimisant les risques de découverte et renforçant leur capacité à atteindre leurs objectifs stratégiques au fil du temps grâce à cette durée prolongée.

Tentatives répétées

Les acteurs APT montrent de la patience, ils persistent dans leurs attaques contre leurs cibles en ajustant constamment leurs tentatives lorsqu'elles échouent. Ils peuvent maintenir une pression constante sur leurs cibles tout en recherchant des failles grâce à cette approche. La continuité de l'adaptabilité face aux obstacles renforce la complexité et l'efficacité des attaques.

1.5.3 Cycle de vie d'une APT

Le cycle de vie de cette menace durable est un processus complexe,il se divise en plusieurs phases distinctes qui sont les suivantes [3] :

Reconnaissance

Chaque attaque exige une étape de préparation pendant laquelle l'attaquant tente d'obtenir des renseignements sur sa cible. Cela inclut l'identification des failles du système, la description de l'architecture du réseau. La collecte minutieuse de données permet aux assaillants de planifier leurs futures actions de manière ciblée, augmentant ainsi les chances de succès de l'attaque.

Compromission initiale

Une fois le système de cartographie est terminé, l'attaquant passe à la phase critique de repérage des failles. L'objectif est d'identifier avec soin une faiblesse permettant l'accès, en exploitant les vulnérabilités logicielles. Cela marque la transition de la planification à l'action, le début concret de l'infiltration dans le système visé.

Contrôle et commandement

Une fois identifié l'entrée, le hacker accède dans la machine cible pour initier l'intrusion. Après avoir sécurisé l'accès initial, il effectue une recherche exhaustive d'informations, incluant la montée en niveau d'accès pour atteindre des parties clés du système. Cette progression est essentielle à l'attaque, offrant à l'attaquant un pouvoir accru et une capacité d'explorer plus en profondeur le réseau visé.

Mouvement latéral

Les attaquants, déjà présents dans le réseau, prennent une position latérale. L'objectif premier est de parcourir le réseau afin de localiser les données sensibles, puis d'en tirer parti. En élargissant leur territoire au sein du système, les criminels maximisent leur potentiel d'acquisition d'informations.

Exfiltration

Est un ensemble d'actions destinées à obtenir et à transmettre des données cruciales à l'attaquant. Au cours de cette procédure, les informations stratégiques collectées sont transférées de l'ordinateur cible vers un emplacement externe. L'exfiltration se produit lorsque les criminels atteignent leurs objectifs en obtenant les informations souhaitées, ce qui constitue une menace pour la confidentialité et l'intégrité de la victime.

Post-exfiltration

Cette étape critique implique des activités destinées à effacer les traces de l'attaque et à assurer une sortie discrète du réseau. Ces actions incluent le non-respect des barrières de sécurité destinées à empêcher la détection. De plus, les attaquants tentent d'effacer toutes les empreintes numériques restantes et de supprimer tout reste potentiel de leur présence dans le système. La post-exfiltration est la dernière étape du processus au cours de laquelle les criminels tentent de renforcer leur position en quittant le réseau sans laisser de preuves significatives.

La figure 1.2 schématise les étapes clés qui rythment le déroulement d'une cette attaque élaborée contre des cibles stratégiques.

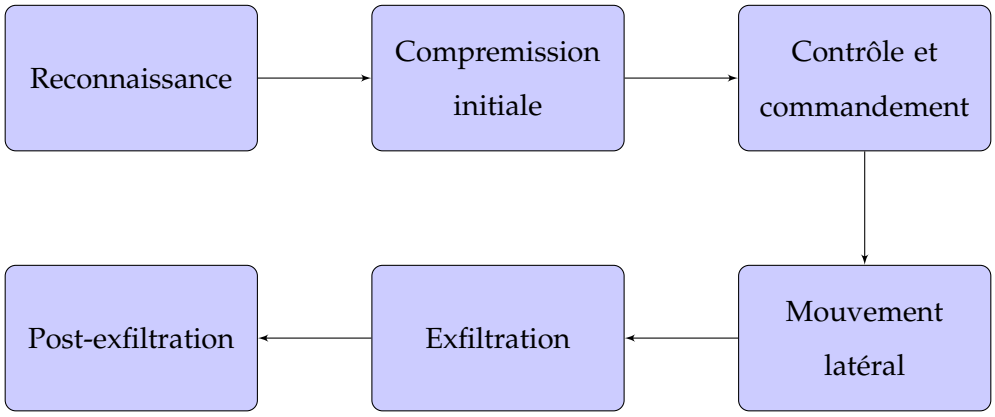

FIGURE 1.2 – Cycle de vie d'une attaque APT.

1.5.4 Groupes APT célèbres

De nombreux groupes ont laissé leur empreinte dans le domaine des APT, marquant ainsi l'histoire de la cybersécurité. Explorer les activités et les attributs de ces groupes notoires offre un aperçu essentiel pour appréhender la sophistication des menaces persistantes avancées, ci-dessous, nous vous présentons quelques groupes célèbres qui ont provoqué d'énormes dommages à l'encontre d'organisations et de gouvernements.

Titan Rain

Nom attribué à une série d'attaques informatiques sophistiquées qui ont été détectées dans les années 2000. Ces attaques étaient principalement dirigées contre des agences gouvernementales américaines et des entreprises de défense. Le terme ***Titan Rain*** a été utilisé pour décrire une série d'activités malveillantes perpétrées par des acteurs cybercriminels, et il est souvent associé à des activités de cyberespionnage. Les attaques *Titan Rain* étaient considérées comme des attaques avancées et persistantes.

TA542

En 2014, un groupe de cybercriminels russes a été identifié comme étant responsable du botnet ***Emotet***, initialement conçu comme un cheval de Troie bancaire. En 2017, le groupe **TA542** a retiré la fonctionnalité de cheval de Troie bancaire d'*Emotet* et a modifié le botnet pour qu'il distribue d'autres types de logiciels malveillants. Les trois premières versions d'*Emotet* ciblaient les clients bancaires afin de réaliser des transferts frauduleux automatiques depuis des comptes compromis.

Stuxnet

Il s'agit d'un ver informatique découvert en 2010 qui a eu un impact sur les installations nucléaires iraniennes. Les politiques et stratégies cybernétiques des États ont subi des changements radicaux à la suite de cet incident. Il utilise quatre vulnérabilités zero-day pour infecter les réseaux informatiques à l'aide de clés Universel Serial Bus (USB). Même s'il était conçu pour l'espionnage industriel, son véritable objectif était de saboter des centrifugeuses dans l'usine nucléaire de *Natanz* en Iran [28].

APT28

Formé de criminels informatiques très compétents. Ce groupe est actif depuis 2004, mais ce n'est qu'en 2014 que ses activités ont commencé à être détaillées. Soupçonné d'avoir des liens avec les services de renseignement russes. APT28 a mené de nombreuses cyberattaques dans le monde entier, ciblant des entités gouvernementales, militaires, des médias et d'autres acteurs stratégiques en Europe, en Asie et en Amérique du Nord.

1.5.5 Moyens d'attaque par stage

Les hackers les plus compétents utilisent une puissante combinaison de méthodes pour pirater des organisations stratégiques.

Nous constatons la sophistication des techniques utilisées à chaque étape : recherche d'information, exploitation de failles inconnues, installation de logiciels malveillants, mouvements dans les réseaux, vol de données sensibles et accès durable au système. Cette synthèse nous aide à comprendre comment ces hackers combinent différentes approches complexes pour prendre le contrôle des systèmes d'information durablement.

Le tableau1.2 résume les méthodes combinées par les attaquants APT lors de leur cyberattaque sur le long terme [3].

Stage	**Techniques Utilisées**
Reconnaissance	Ingénierie sociale, OSINT, Metasploit
Compremission Initiale	Exploitation de faille, faille zero-day , Logiciel malveillant
Contrôle et commandement	Cheval de troie, rootkit, chiffrement , Spear phishing
Mouvement Latéral	Éscalade de privilège, Collecte de données
Exfiltration	Compression, chiffrement
Post-Exfiltration	Cheval de troie, rootkit, scripts avancés

TABLE 1.2 – Techniques utilisées par les groupes APT selon les différentes étapes [3].

1.5.6 Conséquences des APT

Au-delà des dommages techniques directs, les cyberattaques entraînent des répercussions profondes au niveau économique, politique et social comme le montrent les divers exemples concrets suivants [15] [3] :

Conséquences économiques

- Pertes financières directes liées au vol, à l'extorsion ou aux interruptions d'activité 10 milliards de dollars de pertes dues à NotPetya en 2017.
- Coûts élevés de remédiation et de renforcement des défenses après une attaque.
- Impacts sur la stabilité des marchés financiers lors d'attaques contre des institutions financières , 1 milliard de dollars de pertes cumulées dues à Carbanak de 2013 à 2015 .
- Coûts importants liés aux cyberattaques pour les entreprises : 97 des réseaux d'entreprises présentent des signes d'activités suspectes et 82 des entreprises dans le monde ont été victimes de cyberattaques en 2019.

Conséquences politiques

- Tensions géopolitiques accrues entre nations suspectées de mener des opérations offensives crises diplomatiques après des cyberattaques russes présumées lors d'élections aux USA et en France en 2016-2017.
- Risque de déstabilisation d'infrastructures critiques essentielles au fonctionnement des États
- Mise en cause de la responsabilité des États en cas de faille avérée dans leur cybersécurité.
- Renforcement de la surveillance et du contrôle d'Internet, au détriment potentiel des libertés individuelles durcissement des politiques après les révélations de Snowden.

Conséquences sociales

— Perturbation des services publics vitaux suite à des cyberattaques sur les infrastructures critiques 230 000 Ukrainiens privés d'électricité en 2015.
— Perte de confiance dans les institutions et les entreprises après des fuites massives de données personnelles.
— Atteinte à la vie privée avec 5,6 millions de données personnelles piratées ou perdues chaque année.
— Sentiment d'insécurité et crainte généralisée face à la sophistication grandissante des cyberattaques.

1.6 Moyens de défense contre une APT

Combattre efficacement les APT n'a pas de solution universelle. Brewer [7] suggère d'aborder la défense spécifiquement pour chaque phase opérationnelle des APT afin de relever ce défi. Les outils traditionnels de sécurité rencontrent en effet des difficultés à contrer l'ensemble de cette menace persistante avancée. Néanmoins, ils sont utilisés comme première ligne de défense qui peut être astucieusement exploitée pour contrer certaines phases particulières des APT.

1.6.1 Firewall

Définition

Un pare-feu ou Firewall (FW) est un mécanisme de sécurité informatique qui supervise et gère les échanges de données entre un réseau privé et Internet, agissant comme un filtre pour bloquer les dangers et laisser passer les données légitimes, dans le but de protéger les systèmes contre les intrusions et les attaques.

Principe de fonctionnement

Un système de pare-feu comporte des règles prédéfinies pour autoriser, bloquer ou rejeter les connexions sans avertir l'émetteur. De manière spécifique, l'appareil analyse les données entrant et sortant via une interface dédiée au réseau à protéger et une autre pour le réseau externe, souvent l'internet. Le filtrage entrant bloque les connexions suspectes de l'extérieur, même si un tiers malveillant tente d'accéder à une porte dérobée via un cheval de Troie. Le filtrage sortant bloque automatiquement les envois non autorisés, comme des informations confidentielles volées par un cheval de Troie.

Les pare-feu peuvent être classés en fonction de leur méthode de filtrage. Il existe trois types principaux, les pare-feu sans mémoire (Stateless), les pare-feu avec mémoire (Stateful) et les pare-feu applicatifs.

Firewall sans mémoire : les pare-feu sans mémoire effectuent un filtrage simple des paquets, en autorisant ou refusant leur passage entre réseaux en se basant sur l'entête IP du paquet. Cette méthode nécessite la configuration de règles de filtrage, généralement appelées *Access Control List (ACL)*. Cependant, le filtrage simple présente des limites, exige une configuration approfondie, ralentit la bande passante et est vulnérable à des attaques telles que IP Spoofing, déformation de paquets, et certaines formes d'attaques DoS [6].

Firewall avec mémoire : le filtrage de paquet avec état conserve la trace des sessions et connexions dans des tables d'états internes au pare-feu. Les décisions sont prises en fonction des états de connexions, permettant de réagir à des situations protocolaires anormales. Ce type de filtrage offre une protection contre certaines attaques DoS en contrôlant les connexions Internet et en autorisant uniquement celles à la demande. Cependant, une fois l'accès à un service autorisé, aucun contrôle n'est exercé sur le flux concernant ce service [6] .

Firewall applicatif : le pare-feu applicatif fonctionne au niveau de la couche application du modèle Transmission Control Protocol (TCP)/IP. Il filtre les flux non seulement en fonction des entêtes IP, mais aussi en analysant les données contenues dans les paquets. Ce filtrage est réalisé par un programme mandataire qui contrôle l'accès à chaque application en fonction de l'utilisateur et de ses activités. L'utilisateur se connecte d'abord au pare-feu mandataire, qui relaie ensuite le flux vers le serveur demandé. Ce type de pare-feu est efficace car il analyse le contenu des paquets, permettant au filtre de décider de laisser passer ou non en fonction de règles applicables au contenu. Par exemple, l'authentification préalable sur un pare-feu mandataire peut être nécessaire pour accéder à Internet via certains réseaux Wireless Fidelity d'entreprises [6].

Le tableau 1.3 ci-dessous présente un exemple de trois règles spécifiques de filtrage du pare-feu, chacune détaillant les conditions sous lesquelles le trafic réseau est autorisé ou bloqué.

Règle	**Action**	**Protocole**	**Adresse source**	**Port source**	**Adresse destination**	**Port destination**
1	Autoriser	TCP	192.168.1.1	80	160.23.54.134	443
2	Bloquer	UDP	192.168.1.1	*	*	80
3	Autoriser	ICMP	*	25	172.160.43.26	*

TABLE 1.3 – Processus de filtrage du Pare-feu.

— **Règle 1 :** autorise le trafic TCP sortant depuis l'adresse IP 192.168.1.1 sur le port 80 HTTP vers l'adresse IP de destination 160.23.54.134 sur le port 443 HTTPS sortant.

— **Règle 2 :** Bloque tout trafic User Datagram Protocol (UDP) de 192.168.1.1, excepté celui vers le port 80, permettant le trafic HTTP sortant tout en bloquant le reste du trafic UDP.

— **Règle 3 :** Autorise le trafic ICMP depuis n'importe quelle source sur le port source 25 Simple Mail Transfer Protocol (SMTP), vers 172.160.43.26, permettant les vérifications de connectivité ICMP sortantes.

Limitations

— Les pare-feux peuvent être visés pour être submergés et désactivés, ne garantissant pas une protection totale contre les attaques qui échappent à leur portée.
— L'installation d'un antivirus sur les machines est essentielle, car les FW ne protègent pas contre les fichiers infectés par des virus.
— Bien qu'ils puissent bloquer la propagation de virus et les accès de programmes malveillants, les FW ne sont généralement pas capables de les éliminer complètement.

1.6.2 Système de détection d'intrusion

Définition

Un système de détection d'intrusion (IDS) est un logiciel conçu pour monitorer le trafic ou les activités du réseau afin d'identifier les actions malveillantes ou non autorisées, dans le but de repérer et signaler les activités suspectes ou potentiellement dangereuses qui pourraient indiquer une faille de sécurité, une intrusion ou une cyberattaque [34].

Types de systémes de detection d'intrusion

Les IDS peuvent se classer selon trois catégories majeures :

IDS réseau : le système de détection d'intrusion *Network Intrusion Detection System (NIDS)* basé sur le réseau collecte et analyse les données collectées directement à partir du réseau . Il capture et inspecte généralement le type et le contenu des paquets ou des flux transitant par un réseau pour identifier d'éventuels modèles d'attaque.

IDS hôte : un système de détection d'intrusion *Host-based Intrusion Detection System (HIDS)* est déployé sur un serveur spécifique pour surveiller les activités, tels que les fichiers, les journaux système, sur cet hôte afin de détecter des signes d'intrusion. Ils se concentrent sur les activités au niveau de l'hôte [25].

IDS hybride : un IDS hybride améliore la détection en combinant les avantages des NIDS et des HIDS pour une vision plus complète de la sécurité du système.

Principe de fonctionnement

Les IDS analysent les attaques connues et en observant les comportements habituels, ils surveillent en permanence les activités du réseau. Les IDS basés sur les signatures détectent des schémas spécifiques à ceux se trouvant dans leurs bases de données de signatures et déclenchent des alertes en cas de correspondance, tandis que ceux basés sur les comportements repèrent les écarts significatifs , toute déviation du comportement habituel déclenche des alertes fournissant des informations cruciales sur l'incident potentiel, prévenant ainsi les administrateurs de sécurité pour y remédier au problème dans les plus brefs délais.

Limitations

— Les IDS génèrent parfois des alertes pour des erreurs se produisent lorsque le système confond des actions légitimes avec une intrusion.
— Des faux négatifs surviennent lorsqu'un IDS ne parvient pas à détecter une intrusion, interprétant cette activité malveillante comme étant légitime.
— Le chiffrement croissant du trafic rend l'analyse plus difficile pour les IDS qui reposent sur l'inspection de paquets.
— Les IDS basés sur des règles statiques sont moins efficaces face aux menaces évolutives et aux changements d'environnement. Ils nécessitent des mises à jour fréquentes.

1.6.3 Réseau privé virtuel

Définition

Un Virtual Private Network (VPN), également connu sous le nom de réseau privé virtuel, est un service qui établit une connexion sécurisée et chiffrée entre un appareil et un serveur distant pour assurer la confidentialité des données en transit sur Internet. Son utilisation permet de masquer l'adresse IP de l'utilisateur, de crypter les informations échangées et ainsi de se protéger contre la surveillance et les cybermenaces. La figure 1.3 explique le princpe d'encapsulation des systèmes VPN.

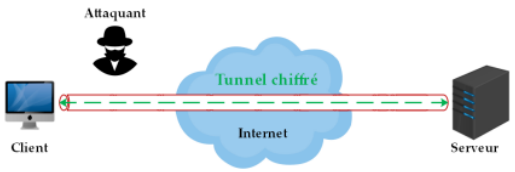

FIGURE 1.3 – Principe d'encapsulation des systèmes VPN.

Principe de fonctionnement d'un VPN

Un réseau privé virtuel, sécurise les transmissions en ligne en chiffrant les données, en les faisant passer par un canal protégé et en masquant l'adresse IP de l'utilisateur, assurant ainsi la confidentialité et la sécurité durant la navigation web , son principe de fonctionnement est détaillé comme suit :

Chiffrement des données : quand un utilisateur active le VPN, toutes les données échangées entre son appareil et le serveur distant sont cryptées. Cela implique que si quelqu'un intercepte ces données, elles resteront incompréhensibles sans la clé de décryptage adéquate.

Tunneling : les données chiffrées sont encapsulées dans un "tunnel" sécurisé qui les protège pendant leur transfert sur Internet. Cela empêche les tiers non autorisés d'accéder aux informations sensibles.

Masquage de l'adresse IP : le VPN change l'adresse IP de l'utilisateur pour cacher sa véritable IP, ce qui rend difficile le suivi de son emplacement et identité, ajoutant ainsi une protection de la vie privée.

Connexion au serveur distant : lorsqu'une connexion sécurisée est établie, toutes les requêtes Internet de l'utilisateur passent par le serveur distant, permettant à l'utilisateur de naviguer sur Internet en utilisant l'adresse IP du serveur et garantissant ainsi un certain degré d'anonymat.

Limitations des VPN

- La performance des VPN dépend du fournisseur de services et les utilisateurs n'ont pas un contrôle total sur tous les paramètres.
- Les normes des VPN ne sont pas toujours suivies et leur fonctionnement varie en fonction du matériel employé.
- Un VPN introduit une étape supplémentaire en faisant passer la connexion de l'utilisateur par un serveur éloigné, pouvant affecter la vitesse et la stabilité de la connexion.

1.6.4 Antivirus

L'antivirus garantit la sécurité de la machine en luttant contre les logiciels dangereux. Il comporte des fonctions Malwarebytes pour protéger contre divers types de logiciels malveillants comme les chevaux de Troie et les vers. Son rôle principal est de repérer et supprimer tout code malveillant contenu dans un logiciel infecté. Mais il ne peut pas empêcher les intrusions réseau de personnes

utilisant des logiciels légitimes. Pour une sécurité maximale, il est essentiel de limiter l'accès non autorisé aux ressources.

1.6.5 Chiffrement des données

C'est la méthode qui permet de transformer des données en un code secret, pour les protéger des personnes non autorisées. Il existe plusieurs types de chiffrement :

le chiffrement symétrique : utilise la même clé pour chiffrer et déchiffrer les données. Il est rapide et efficace, mais il nécessite de partager la clé secrètement entre les parties. Parmi les prococole de chiffrement symétrique nous trouvons : AES, DES, RC4.

le chiffrement asymétrique : utilise deux clés différentes une publique et une privée, pour chiffrer et déchiffrer les données. Il est plus lent et complexe, mais il évite le problème de la distribution des clés. Parmi les prococole de chiffrement asymétrique nous trouvons : RSA, ECC, ElGamal.

1.6.6 Formation et sensibilisation des employés

Les cybercriminels ciblent les employés qui sont la première ligne de défense contre les attaques. Il est essentiel de les instruire sur les bonnes pratiques de sécurité informatique, telles que la création de mots de passe robustes, l'identification des tentatives de phishing et la sauvegarde des données.

1.7 Conclusion

Dans ce premier chapitre, nous avons initialement introduit les concepts fondamentaux de la sécurité en examinant diverses attaques réseau. Par la suite, nous nous sommes concentrés sur les menaces persistantes avancées qui font l'objet de notre étude, en discutant de leurs cycles de vie et de leurs caractéristiques. Nous avons ensuite précisé les techniques employées par les groupes APT pour mener à bien ces attaques sophistiquées. Enfin, nous avons présenté quelques mécanismes de défense traditionnels, en soulignant leurs limites dans la détection de ces menaces avancées.

Le chapitre suivant présentera en détail les techniques d'apprentissage automatique et offrira un aperçu complet de leur utilisation pour détecter les menaces persistantes avancées.

Chapitre 2

État de l'art sur l'application du Machine Learning pour la détection des menaces persistantes avancées

2.1 Introduction

La détection efficace des menaces persistantes avancées reste un défi majeur dans le domaine de la cybersécurité. Les APT sont des attaques sophistiquées et furtives menées par des acteurs malveillants disposant de ressources importantes. Leur objectif est de compromettre les systèmes de manière durable afin d'accéder à des données sensibles ou de perturber les opérations. Face à ces menaces évoluées, les approches traditionnelles de détection d'intrusion montrent leurs limites. C'est dans ce contexte que les techniques d'apprentissage automatique offrent des perspectives prometteuses pour renforcer la détection des APT.

Ce chapitre dresse un état de l'art détaillé sur l'application du Machine Learning à la problématique des APT. Après avoir présenté les concepts de base de ce dernier, nous analyserons les travaux récents les plus pertinents dans ce domaine. Une classification par approches est proposée, mettant en évidence les différentes méthodologies adoptées ainsi que les résultats obtenus. Enfin, une discussion critique permettra d'identifier les forces, les limites et les perspectives d'amélioration des solutions existantes.

2.2 Apprentissage Automatique

2.2.1 Définition

L'apprentissage automatique ou Machine Learning (ML) est un sous-ensemble de l'intelligence artificielle (IA), qui permet aux machines de s'adapter à des situations inconnues et de prendre des décisions de manière autonome en apprenant des données d'entrée, sans nécessiter de programmation explicite [32].

2.2.2 Approches de l'apprentissage automatique

En général, il existe trois grandes approches en apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

Apprentissage supervisé

L'apprentissage supervisé s'inspire des expériences précédentes pour produire des sorties de données. Le système s'entraîne sur un ensemble de données étiquetées, avec les informations qu'il est censé déterminer. Les données peuvent même être déjà classifiées de la manière dont le système est supposé le faire. L'apprentissage supervisé comprend des algorithmes de régression et des algorithmes de classification binaire ou multi-classes [2].

Apprentissage non supervisé

L'apprentissage non supervisé utilise une approche plus indépendante dans laquelle un ordinateur apprend à identifier des processus et des schémas complexes sans aucun guidage humain constant et rigoureux. C'est l'algorithme lui-même qui classe et analyse les données pour aboutir aux résultats corrects. Les algorithmes d'apprentissage non supervisé sont nombreux ; parmi eux, nous citons le K-means, l'analyse en composantes principales (PCA) et les règles d'association [2].

Apprentissage par renforcement

L'apprentissage par renforcement est une méthode qui permet d'entraîner des modèles d'IA de manière spécifique. Un agent ou un algorithme apprend des stratégies de manière autonome pour ainsi faire des choix de manière correcte sur la base des informations que l'agent pourrait percevoir [2]. La figure 2.1 présente les différentes approches de l'apprentissage automatique.

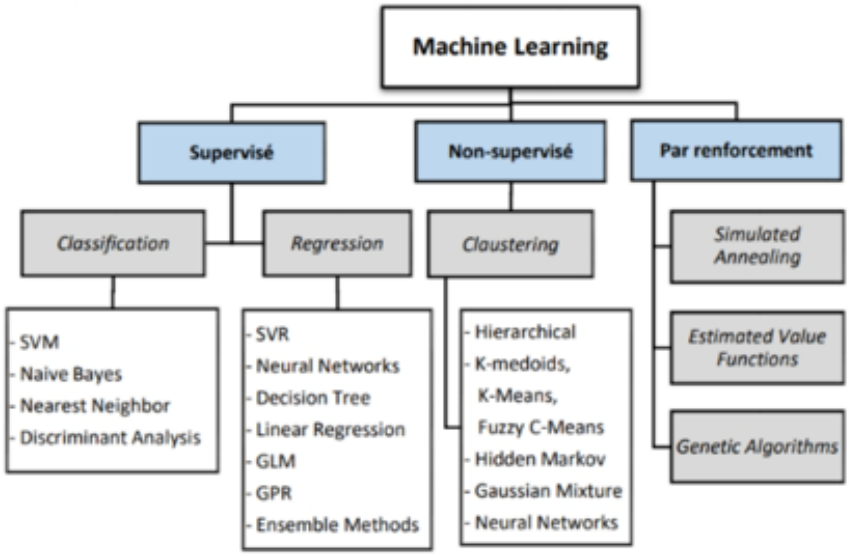

FIGURE 2.1 – Différentes approches de l'apprentissage automatique [17].

2.2.3 Notions liées au Machine Learning

Préparation des données

La préparation des données consiste à transformer les données brutes en informations utiles et exploitables pour les utilisateurs. Ci-dessous nous présentons les différentes phases de la préparation des données :

Collecte des données : est un processus de regroupement de toutes les données nécessaires pour résoudre un problème de Machine Learning. Les données sont recueillies à partir de différentes sources telles que des bases de données, des interface de programmation d'application et des enquêtes, etc. Ces données doivent être pertinentes pour la réussite du projet.

Exploration des données : après la collecte des données, il est nécessaire de les explorer pour analyser leur structure, leurs tendances, les valeurs manquantes, les corrélations, etc. Cela passe généralement par la visualisation des données (histogrammes, boîtes, graphiques, etc.) et des analyses statistiques descriptives pour obtenir des premières informations.

Nettoyage des données : cette phase est essentielle car elle comprend le traitement des données pour supprimer les valeurs aberrantes, les doublons, les valeurs manquantes. La qualité des données utilisées pour l'analyse est assurée par le nettoyage des données, ce qui aide à obtenir des résultats précis et significatifs dans les analyses suivantes.

Transformation des données : une fois que les données ont été collectées, explorées et nettoyées, arrive la phase de transformation des données. Il existe plusieurs méthodes telles que la normalisation, l'encodage des variables catégorielles et le remplacement des données manquantes pour préparer les données pour la phase d'entrainement .

Division des données

L'ensemble d'entrainement : est utilisé pour former notre modèle. C'est là que notre algorithme apprend les schémas et les relations entre les caractéristiques et les résultats attendus.

L'ensemble de test : est utilisé pour évaluer les performances du modèle. Il permet de vérifier comment le modèle se comporte sur des données qu'il n'a jamais vues auparavant.

L'ensemble de validation : est utilisé pour ajuster les hyperparamètres du modèle. Les hyperparamètres sont des paramètres qui ne sont pas appris par le modèle lui-même, mais qui influencent ses performances.

Une répartition courante est de diviser les données en environ 60% pour l'ensemble d'entraînement, 20% pour l'ensemble de test et 20% pour l'ensemble de validation. Cette répartition équilibrée permet d'avoir suffisamment de données pour l'entraînement, l'évaluation et l'ajustement des hyperparamètres.

Entrainement du modèle

La phase d'entraînement consiste à ajuster les paramètres du modèle en utilisant l'ensemble d'entraînement pour minimisant ainsi l'erreur entre les prédictions du modèle et les résultats réels. Cette optimisation utilise généralement un algorithme comme la descente de gradient, afin de permettre au modèle de généraliser ses connaissances pour de nouvelles données et obtenir des prédictions précises sur des exemples inconnus.

Évaluation du modèle

La phase d'évaluation consiste a tester la performance d'un modèle sur de nouvelles données avec un ensemble de test. Pour cela des critères comme la précision (PRE), le rappel et le F1-score sont utilisés en classification, ou l'erreur quadratique moyenne dans le cas régression, varient selon le problème. Un modèle de qualité doit être capable de généraliser et de prédire avec précision sur des données qu'il n'a pas vu auparavant.

Sous-apprentissage / sur-apprentissage

Le sous-apprentissage (underfitting) survient quand le modèle ne capte pas les motifs sous-jacents des données d'entraînement, résultant en une faible performance tant sur les données d'entraînement que sur les données de test.

Le sur-apprentissage (overfitting) survient lorsque le modèle s'adapte excessivement aux données d'entraînement, mémorisant les exemples spécifiques au lieu d'apprendre des schémas généraux. Cette situation se traduit par de bons résultats avec les données d'entraînement mais des performances décevantes avec les données de test ou nouvelles. Le modèle peut être trop complexe et spécialisé pour les données, entraînant une capacité de généralisation réduite [47]. La figure 2.2 illustre représentation d'un underfitting et d'un overfitting.

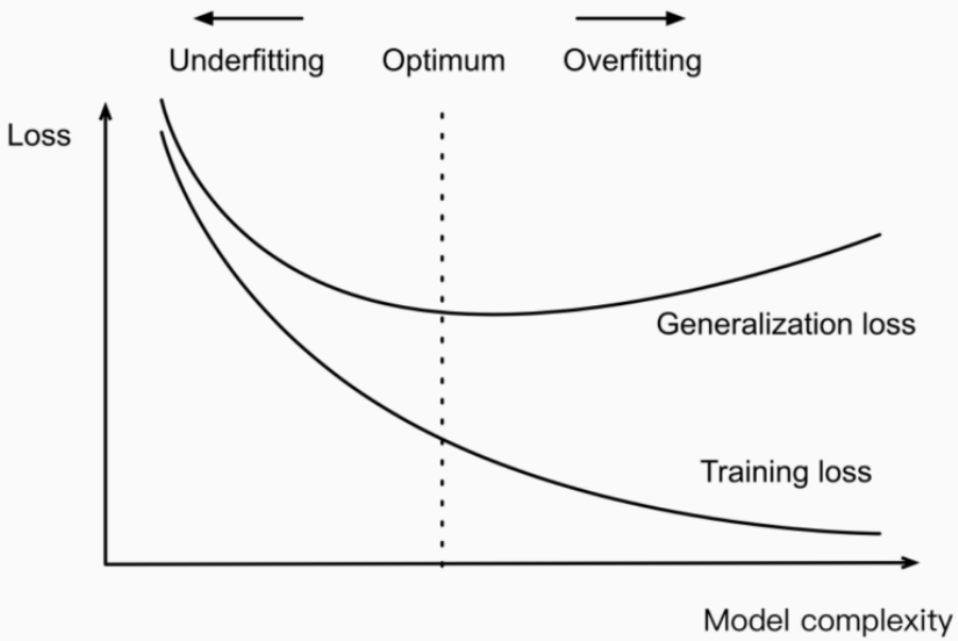

FIGURE 2.2 – Représentation d'un underfitting et d'un overfitting [4].

2.2.4 Algorithmes appliqués pour la détection des APT

Arbre de décision

Un arbre de décision (DT) est un modèle d'apprentissage machine utilisé à des fins de classification et de régression,représenté grâce à un ensemble de choix hiérarchiques basés sur les caractéristiques, organisés sous forme d'un arbre constitués d'un ensemble de nœuds intermédiaires et des feuilles . Un critère de séparation, consiste souvent en une condition portant sur un ou plusieurs attributs des données d'apprentissage est utilisé lors de la prise de décision. L'indice de Gini et l'entropie sont utilisés pour sélectionner l'attribut de séparation et la condition précise sur l'attribut. La figure 2.3 monte un exemple du principe de fonctionnement d'un arbre de décision.

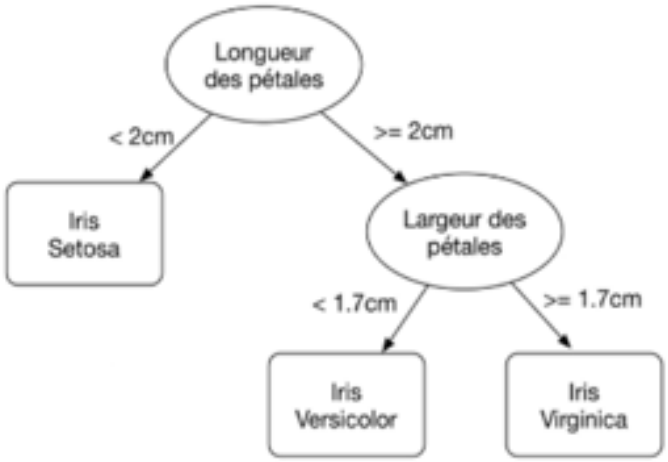

FIGURE 2.3 – Exemple d'arbre de décision [33].

De nombreux algorithmes avancés, comme Random Forest et eXtreme Gradient Boosting sont constitués de plusieurs arbres de décision. La figure 2.4 présente l'évolution de l'arbre de décision.

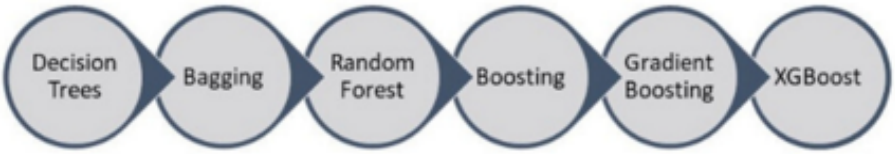

FIGURE 2.4 – Évolution de l'arbre de décision [31].

Random Forest

Random Forest (RF) est un algorithme d'apprentissage automatique qui combine les prédictions de plusieurs arbres de décision pour améliorer la précision prédictive globale. Chaque arbre est formé sur un sous-ensemble aléatoire des données d'apprentissage, ce qui permet de réduire la variance et d'améliorer la robustesse de l'algorithme. La prédiction finale est obtenue en faisant la moyenne des prédictions pour la régression, ou en prenant le vote majoritaire pour la classification [18]. La figure 2.5 offre une illustration du mécanisme d'un Random Forest.

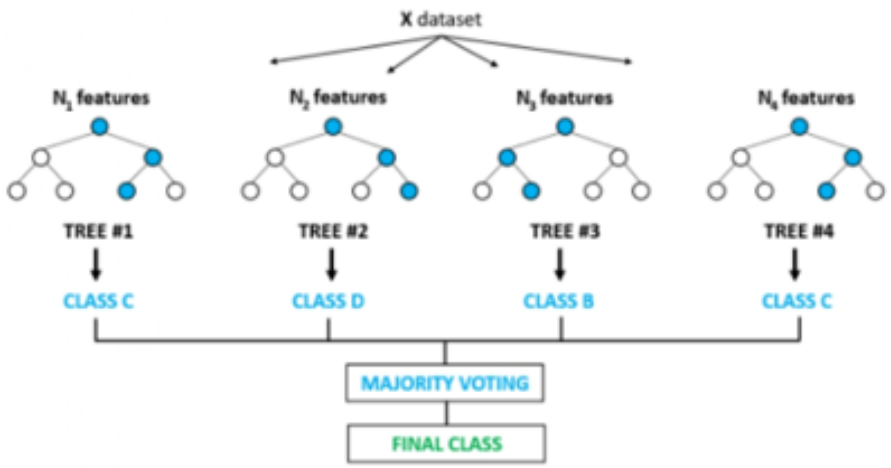

FIGURE 2.5 – Fonctionnement de Random Forest[8].

Machines à vecteurs de support

Machines à vecteurs de support (SVM) sont des algorithmes de classification utilisés pour établir une frontière de décision optimale géneralement sous forme d'une droite dans le cas linéaire pour diviser l'espace à d dimensions en deux classes soit attaque ou normal . La frontière optimale est celle qui maximise la distance aux données des deux classes favorisant une meilleure généralisation aux nouvelles données.

Régression logistique

La régression logistique (LR) , est un algorithme d'apprentissage automatique qui sert à représenter la relation entre une variable dépendante binaire et un ensemble de variables indépendantes. cet algorithme permet de déterminer la probabilité qu'un événement se produise en fonction des valeurs des variables explicatives. La régression logistique emploie une fonction logistique pour transformer la somme pondérée des caractéristiques en une probabilité qui varie de 0 à 1. Ainsi faciliter la représentation de la non-linéarité des liens entre les variables explicatives et la variable cible binaire.

Réseaux de neurone

Un réseau de neurone artificiel est formé par des unités interconnectées, appelées neurones, organisées en couches, inspiré par le fonctionnement du neurone biologique, et utilisé dans l'apprentissage automatique. Un réseau de neurones typique se compose de trois types de couches : une couche d'entrée recevant les données initiales, une ou plusieurs couches intermédiaires effectuant des calculs complexes et une couche de sortie produisant la sortie du réseau.

Perceptron multicouches

Perceptron multicouches (MLP) consiste en un ensemble de neurones artificiels organisés en couches, où les données se déplacent uniquement de la couche d'entrée vers la couche de sortie. De nombreuses couches de traitement lui permettent d'établir des connexions non linéaires entre l'entrée et la sortie.

Gradient Boosting

Le Gradient Boosting est une technique d'apprentissage automatique en ensemble qui combine les prédictions de plusieurs modèles pour améliorer la précision prédictive globale. Il est particulièrement utile pour les problèmes de régression et de classification. Le terme "gradient" dans le Gradient Boosting fait référence à la méthode d'utilisation du gradient de la fonction de perte pour minimiser les erreurs pendant l'entraînement. La partie "boosting" du nom implique que l'algorithme combine des modèles prédictifs faibles pour former un apprenant fort [14].

eXtreme Gradient Boosting

L'algorithme Extreme Gradient Boosting (XGBoost) est une méthode d'apprentissage automatique supervisé appartenant à la famille des techniques d'ensemble. Il se distingue par sa capacité à construire un modèle prédictif de haute précision en combinant de manière itérative des modèles de prédiction faibles, généralement des arbres de décision [36]. La figure 2.6 représente un exemple du fonctionnement d'un XGBoost.

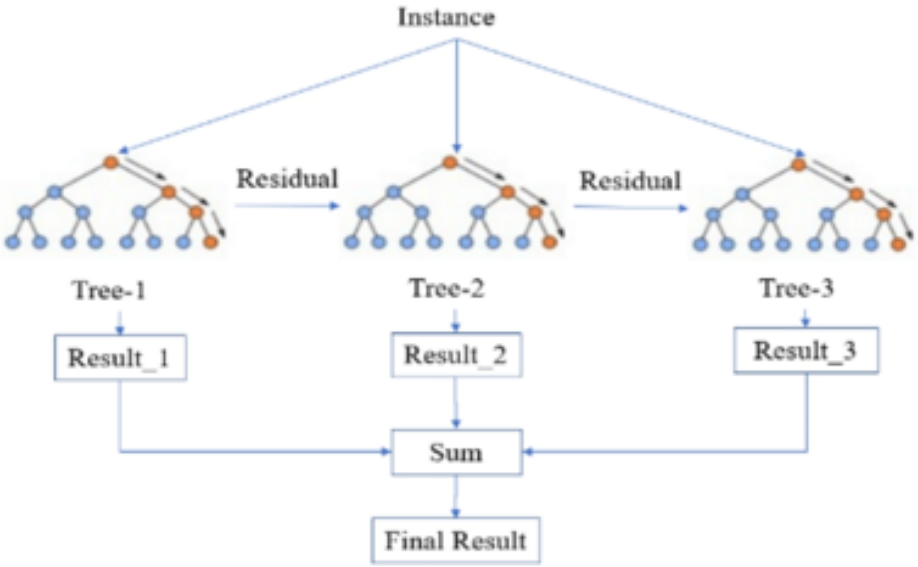

FIGURE 2.6 – Structure simplifiée de XGBoost[43].

2.2.5 Avantages de l'application du Machine Learning dans la sécurité

— Le Machine Learning repère rapidement toute activité anormale en analysant le comportement normal des utilisateurs, des systèmes et des réseaux. Les modèles peuvent détecter des schémas inhabituels tels que des accès non autorisés, des tentatives de phishing ou des comportements de logiciels malveillants.

— Les algorithmes de Machine Learning peuvent détecter précocement les nouvelles menaces en analysant de larges ensembles de données pour repérer des tendances, des signatures et des signes de compromission, facilitant ainsi l'identification de nouvelles variantes d'attaques.

— Les modèles de Machine Learning peuvent être constamment mis à jour avec de nouvelles données pour rester pertinents face aux évolutions des menaces et des tentatives illégales des attaquants.

— Les algorithmes de Machine Learning proposent une automatisation de la réponse aux incidents, grâce a des recommandations pour neutraliser ou atténuer les effets des menaces, accélérant ainsi la détection, l'analyse et la réaction, et réduisant le temps nécessaire pour contrer une attaque.

2.3 Synthèse des travaux connexes sur l'application du Machine Learning pour la détection des APT

La figure 2.7 présente une taxonomie des principales méthodes de détection des APT utilisant le Machine Learning, classées en trois catégories distinctes : les approches basées sur la corrélation des alertes, celles axées sur la détection des noms de domaine APT, et enfin les techniques reposant sur l'analyse du trafic réseau. Cette classification permet de comprendre les différentes stratégies adoptées par les chercheurs pour aborder ce problème complexe et met en évidence l'évolution des techniques de détection au fil du temps.

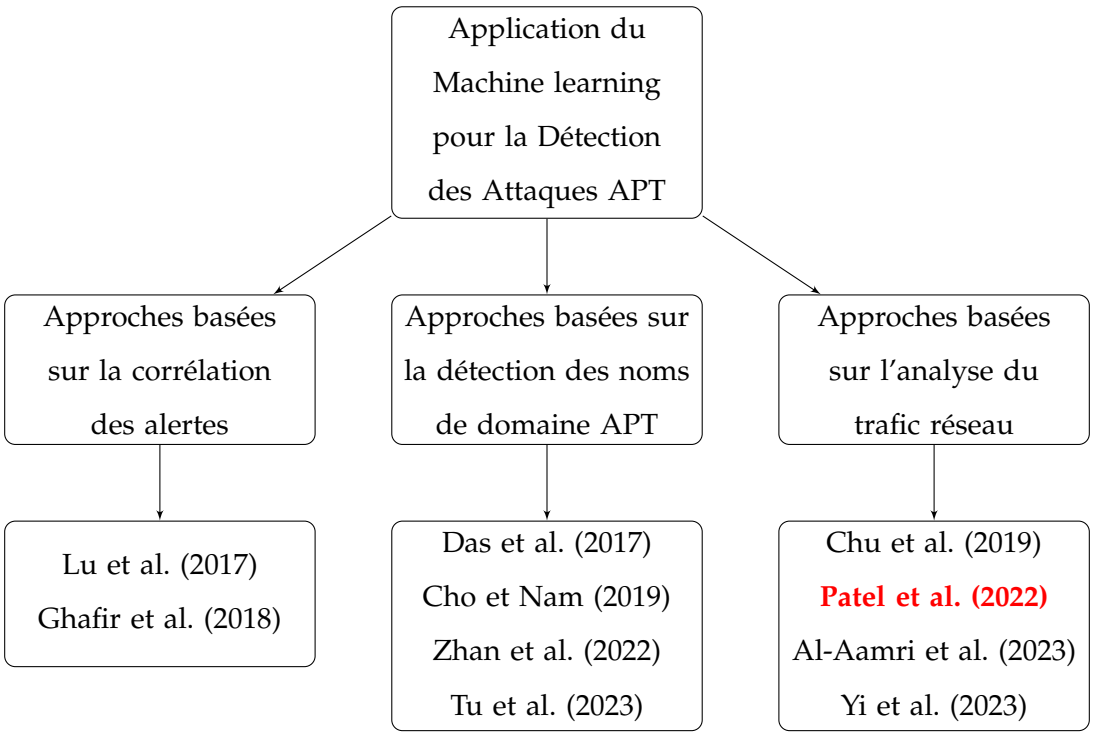

FIGURE 2.7 – Taxonomie des approches de détection des attaques APT basées sur le ML.

2.3.1 Approches basées sur la corrélation des alertes

Travail de Lu et al. [30]

Dans [30], une méthode de détection de trafic malveillant de type APT est proposée. L'architecture proposée comporte quatre composants principaux : la capture de trafic, le filtrage de trafic, l'extraction de caractéristiques et la détection d'anomalies.

Le trafic réseau à analyser est d'abord capturé au niveau de la passerelle du réseau en utilisant un outil de capture de paquets, tel que ***Wireshark***, pour reconstruire les flux TCP/UDP entrants et sortants. Ensuite, les données brutes sont envoyées à un module de filtrage qui élimine la plupart des données normales en se basant sur des caractéristiques comme la taille des paquets et la durée des échanges. Les attributs extraits incluent les adresses IP, les ports et le nombre de paquets de chaque flux. Les flux restants après filtrage sont analysés par le module d'extraction de caractéristiques pour étudier la corrélation entre la taille des paquets et la durée des flux, en distinguant les flux réguliers des flux nuisibles. Les flux conformes à la relation linéaire sont considérés comme non malveillants. Les flux restants sont comparés à des modèles de flux normaux, permettant d'identifier les flux non malveillants. Les flux anormaux sont conservés après cette double analyse.

La méthode ATCTDS utilise l'algorithme Gradient Boosting Decision Tree (GBDT) pour classifier les flux après le filtrage des APT, obtenant un taux de faux positifs (FPR) de 3,0% et un taux de faux négatifs (FNR) de 2,95%. Ces faibles taux d'erreurs permettent une détection précise des APT dans les communications réseaux. La figure 2.8 montre l'architecture du modéle proposé .

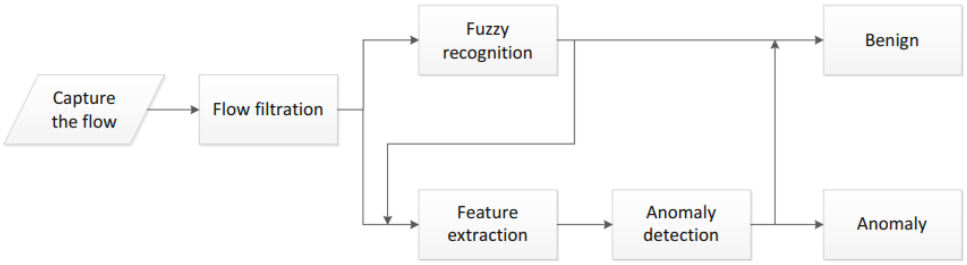

FIGURE 2.8 – L'architecture du modéle proposé par Lu et al.

Travail de Ghafir et al. [19]

Ghafir et al. ont mis en place un système performant Machine Learning Advanced Persistent Threat (MLAPT) pour repérer et anticiper rapidement les attaques APT, facilitant la protection des systèmes.

Le modèle MLAPT comporte trois étapes principales. En premier lieu, la phase de détection des menaces où huit modules ont été développés pour détecter différentes techniques utilisées dans les étapes du cycle de vie d'une attaque APT. Vient par la suite la phase de corrélation d'alertes impliquant trois étapes qui sont : le filtrage des alertes qui identifie et filtre les alertes redondantes ou répétées ; le regroupement des alertes qui utilise un algorithme de regroupement basé sur trois règles qui sont : type d'alerte, horodatage, hôte infecté et l'indexation des corrélations avec un algorithme qui calcule un indice de corrélation pour chaque cluster en fonction des attributs des alertes. Selon la valeur de l'indice, le cluster peut représenter un scénario d'attaque APT complet, partiel ou non corrélé, dans le but de concevoir un schéma de corrélation pour associer les alertes des différents modules de détection, afin d'identifier celles liées à un même scénario d'attaque APT.

La dernière étape du modèle proposé par les chercheurs est la mise en œuvre d'un modèle de classification basé sur l'apprentissage automatique pour prévoir les attaques en se basant sur l'historique du réseau. Plusieurs algorithmes sont appliqués, à savoir : DT, KNN (K-Nearest Neighbors), SVM et la méthode d'ensemble.

Le SVM linéaire a donné de meilleurs résultats pour prédire si des alertes précoces allaient se développer en une attaque APT complète avec un taux d'Accuracy (ACC) de 84,8%, un taux de vrais positifs (TPR) de 81,8% et FPR de 4,5%. La figure 2.9 offre une représentation de l'architecture suggérée.

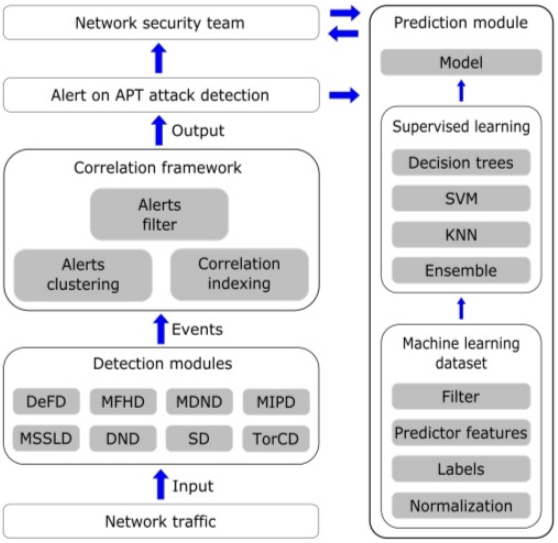

FIGURE 2.9 – Architecture de l'IDS proposé par Ghafir et al.

2.3.2 Classification des travaux basés sur la détection des APT basés sur la corrélation d'alertes

Le Tableau 2.1 a pour objectif de synthétiser et analyser les articles étudiés dans la section précédente. Il mettra en évidence les méthodologies et résultats présentés par les auteurs, permettant ainsi une comparaison approfondie des approches basées sur la corrélation des alertes.

Auteurs	Année	DataSet	Modèle	Résultats (ACC, FPR, TPR , FNR)
Lu et al. [30]	2017	trafic normal provenant des serveurs universitaires , trafic APT malveillant généré à partir de la BDD des logiciels malveillants Mila Parkour fournie par Contagio.	GBDT	FPR= 3.0% FNR= 2.95%
Ghafir et al. [19]	2018	trafic d'attaques APT simulées sur un réseau universitaire	MLAPT	ACC= 84.8% TPR= 81.8% FPR= 4.5%

TABLE 2.1 – Tableau comparatif des traveaux basés sur la corrélation des alertes.

2.3.3 Approches basées sur la détection des noms de domaines APT

Travail de Das et al. [13]

L'article présente deux méthodes différentes pour détecter l'exfiltration de données et les tunnels Controle et Commande(CC) sur Domain Name System(DNS).

Dans la phase d'exfiltration de données, l'approche développée par les auteurs comporte deux étapes essentielles : le marquage de chaque sous-domaine de requête DNS en fonction de caractéristiques telles que la longueur, l'apparition sur une liste blanche de domaines, etc. Ensuite, les sous-domaines marqués sont regroupés par adresse IP source et domaine de niveau 2, puis fusionnés. Une série de caractéristiques est obtenue à partir de cette chaîne combinée, incluant l'entropie, les proportions de caractères alphanumériques, la taille, le nombre de sous-domaines uniques. Un modèle de régression logistique régularisée est développé en utilisant des caractéristiques provenant de chaînes concaténées, obtenant une ACC de 99,93% et un F1-score de 96%.

Dans la phase de détection des tunnels CC sur les enregistrements DNS de type texte , une approche non supervisée a été élaborée pour contourner le manque de données d'entraînement labellisées pour les tunnels. Cette méthode consiste à extraire les réponses Text Record (TXT) qui partagent des caractéristiques similaires mais spécifiques, telles que les quantités de majuscules, de minuscules, de chiffres, de caractères spéciaux et d'entropie. Ensuite, un algorithme de regroupement K-means regroupe les réponses TXT en fonction de ces caractéristiques. Les clusters repérés manuellement comme contenant potentiellement du contenu encodé sont considérés comme des tunnels CC possibles. Avec l'injection de tunnels malveillants, le modèle obtient un TPR de 91,68% avec seulement 0,40% de FNR.

Travail de Cho et Nam [10]

La référence [10] propose un modèle pour repérer les attaques APT basées sur des noms de domaines inconnus qui est composé de deux parties : la détection de domaines suspects et la surveillance d'accès.

Le modèle intègre un centre de stockage des données, un module de détection APT et un composant d'alerte émettant des avertissements. Une base de données est utilisée pour enregistrer les signatures et les résultats de surveillance. L'algorithme se base sur des signatures et un algorithme Random Forest pour détecter les domaines inconnus. Les 25 attributs les plus significatifs sont sélectionnés pour la classification, l'outil open source Spark-Suricata est utilisé pour la génération de règles et la détection des signes d'attaques APT.

Un total de 10 000 domaines normaux et 30 000 domaines malveillants ont été rassemblés pour entraîner une forêt aléatoire. Le jeu de test se compose de 43 fichiers PCAP d'attaques APT et de 90 fichiers de code malveillant provenant d'attaques APT réelles, avec une répartition de 70% pour

l'entraînement et 30% pour les tests. Les données de test ont été réparties en quatre ensembles avec des proportions différentes à chaque fois.

Les résultats expérimentaux obtenus indiquent un TPR de 98,9%, un FPR de 15,4%, un FNR de 1,1% et un taux de vrais négatifs (TNR) de 84,6% et une précision de 96,1%. L'entraînement a pris 4,175 secondes et la prédiction 0,117 seconde.

Travail de Zhan et al. [49]

L'étude aborde la problématique de la détection de l'exfiltration de données basée sur DNS over HTTPS (DoH). L'approche proposée repose sur l'utilisation d'informations de poignée de main Transport Layer Security (TLS) et d'une analyse du trafic pour détecter le tunneling DoH. Pour ce faire, les chercheurs ont utilisé des algorithmes de classification tels que RF, DT, et LR pour entraîner le modèle de détection. Les caractéristiques utilisées pour l'entraînement des modèles comprennent des informations liées à la longueur des enregistrements TLS et des caractéristiques temporelles. Les métriques d'évaluation incluent la précision (PRE), le rappel et le F1-score pour les différents modèles et configurations.

Les résultats numériques des modèles RF, DT et LR, le modèle RF affiche des valeurs de précision, de rappel et de score F1 toutes évaluées à 99,9%, de même, le modèle DT présente des performances similaires avec des valeurs également élevées à 99,9%. Pour ce qui est de LR, bien que ses scores soient légèrement inférieurs, ils demeurent néanmoins très solides, avec une précision de 98,7%, un rappel de 96,8% et un F1-score de 97,8%.

Travail de Tu et al. [42]

L'article propose une nouvelle méthode pour détecter les tunnels DNS, en se concentrant sur l'efficacité et la précision, deux aspects souvent limités dans les approches existantes.

L'approche repose sur l'utilisation des paires de requête-réponse DNS comme unité de base, combinant l'extraction de caractéristiques d'encodage à l'aide d'un encodeur à mécanisme d'attention multi-têtes avec des caractéristiques statistiques sur le trafic DNS. Ces caractéristiques sont ensuite fusionnées pour entraîner un classifieur XGBoost. Les résultats obtenus sur les datasets de base et de test de généralisation dépassent ceux des méthodes de référence, démontrant une amélioration significative de la précision et du F1-score. Plus précisément, sur le dataset de base, l'Accuracy atteint 99,57%, avec un F1-score 99,68%. Sur le dataset de test de généralisation, la précision est de 98,77%, également avec un F1-score de 98,82%. Ces résultats sont supérieurs à ceux des méthodes de référence, avec des gains de 1,44% et 10,02% sur le F1-score pour les datasets de base et de généralisation respectivement. La figure 2.10 présente l'approche proposé par les auteurs.

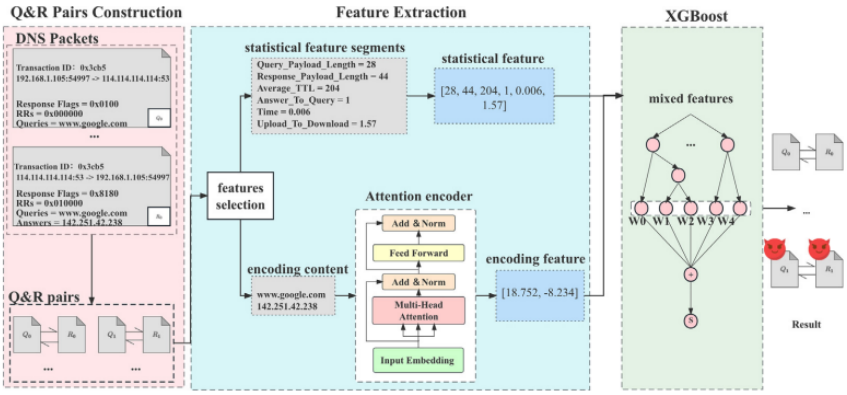

FIGURE 2.10 – L'architecture du modèle proposé par Tu et al.

2.3.4 Classification des travaux basés sur la détection des noms de domaines APT

Le tableau 2.2 a pour objectif de synthétiser et analyser les articles étudiés dans la section précédente. Il mettra en évidence les méthodologies et résultats présentés par les auteurs, permettant ainsi une comparaison approfondie des approches basées sur la détection des noms de domaines APT .

Auteurs	Année	DataSet	Modèle	Résultats(ACC, PRE, Recall, F1-score, TPR, FPR)
Das et al. [13]	2017	Données collectées dans une compagnie nommée (Company1)	LR	ACC= 99,93%, FPR= 0,189%, FNR= 5,5%
Cho et Nam [10]	2019	trafic d'attaques APT simulées et de noms de domaines les plus connus sur Internet	RF	ACC= 96,1%, TPR= 98,9%, FPR= 15,4%
Zhan et al. [49]	2022	Les données malveillantes ont été générées en simulant des attaques d'exfiltration de données, tandis que les données bénignes proviennent de la visite de sites web populaires.	DT	ACC= 99,9%, PRE= 99,9%, Recall= 99,9% , F1-score= 99,9%
			RF	ACC= 99,9%, PRE= 99,9%, Recall= 99,9% , F1-score= 99,9%
			LR	ACC= 97,8%, PRE= 98,7%, Recall= 96,8% , F1-score= 97,8%
Tu et al. [42]	2023	Les données malveillantes proviennent de simulations d'attaques, les données bénignes sont collectées en capturant du trafic réel d'entreprise.	XGBoost	PRE= 98,77%, Recall= 98,87%, F1-score= 98,82%

TABLE 2.2 – Tableau comparatif des traveaux basés sur la détection des noms de domaines APT.

2.3.5 Approches basées sur l'analyse du trafic réseau

Travail de Chu et al. [12]

Les auteurs cherchent à concevoir un dispositif pour détecter les attaques Advanced Persistent Threat et ainsi limiter leur impact en mettant en place des alertes précoces.

Wen-Lin et al. emploient le dataSet NSL-KDD avec des enregistrements décrivant le trafic réseau comme normal ou comme des attaques. Ils utilisent l'outil WEKA pour encoder les attributs texte en numériques, résultant en 122 dimensions par enregistrement. Ensuite, après avoir prétraité les données, ils utilisent l'ACP pour réduire les dimensions de 122 à 94, tout en conservant 90% de la variance initiale, améliorant ainsi la vitesse de calcul sans réduire significativement les informations.

Quatre algorithmes ont été expérimentés : SVM, naïve bayes, J48, MLP avec des réglages spécifiques pour SVM (noyau RBF, C=1.0, gamma=0.0), paramètres par défaut pour naïve bayes, et divers paramètres testés pour J48 et MLP. Chaque méthode est entraînée sur un sous-ensemble A de 5000 exemples, puis évaluée sur trois autres sous-ensembles B,C et D de taille similaire.

Les SVM avec un noyau RBF et le MLP sont les meilleurs en termes de taux d'exactitude, atteignant respectivement jusqu'à 97,22% et 97,82%. Malgré cela, le temps de calcul du SVM diminue grâce à l'ACP, ce qui en fait le choix des auteurs pour le système de détection des attaques APT. Ce modèle couvre toutes les phases d'une attaque APT en combinant des informations de multiples protocoles et sources réseau. Ainsi, même si une phase échappe à la détection, il est très difficile pour l'attaque de passer inaperçue dans sa globalité.

Travail de Patel et al. [38]

Patel et al. ont examiné les performances de la détection d'intrusion en utilisant SVM et DNN sur le jeu de données NSL-KDD de DARPA celui-ci contient des données normales et quatre types d'attaques distincts, à savoir Probe, DoS, U2R(User to Root) et R2L (Remote to Local). Les données ont été prétraitées et une réduction de la dimension des données a été réalisée , cela visait à économiser du temps et de l'espace pour la classification en supprimant les caractéristiques peu corrélées et en évitant le sur-ajustement, ce qui a permis d'améliorer les performances.

Un processus d'entraînement 75% et de validation 25% a été effectué en utilisant les méthodes SVM et DNN sur le jeu de données prétraité. Les auteurs ont examiné l'algorithme SVM en intégrant la bibliothèque LIBSVM dans l'outil Meka, un logiciel intégré offrant différents modèles SVM, et le modèle DNN a utilisé l'approche de transfert d'apprentissage.

Les chercheurs ont utilisé des métriques comme l'Accuracy, la précision et le Racall pour évaluer chaque type d'attaque seule. Ainsi, les résultats de leurs évaluations sont présentés dans la figure 2.11.

Attack Name	Accuracy		Precision		Recall	
	SVM Model	DNN Model	SVM Model	DNN Model	SVM Model	DNN Model
Dos	0.9875	**0.9878**	0.9825	**0.9836**	**0.9845**	0.9812
Probe	0.9765	**0.9768**	0.9612	**0.9665**	0.9921	**0.9960**
R2L	0.9118	**0.9121**	**0.8916**	0.8664	**0.9512**	0.9148
U2R	0.8909	**0.8912**	**0.9043**	0.8892	0.8914	**0.8945**
normal	0.9760	**0.9763**	0.9912	**0.9967**	0.9832	**0.9871**

FIGURE 2.11 – Résultats obtenus du modéle proposé par Patel et al.

Travail d'Al-Aamri et al. [1]

AL-Aamri et al. ont proposé une approche utilisant l'apprentissage automatique pour détecter les menaces persistantes avancées en distinguant les modèles de trafic normaux et anormaux. Le système, implémenté via Microsoft Azure, détecte les comportements malveillants, même avec des signatures uniques d'attaques inconnues.

Le jeu de données utilisé contient 57 caractéristiques extraites de fichiers de paquets réseau enregistrés à différents intervalles. Après un prétraitement comprenant des étapes telles que le nettoyage, la transformation, la réduction, l'intégration, la discrétisation, la division, la sélection des caractéristiques et l'augmentation des données, les informations ont été converties au format CSV à l'aide de Cloud MS Azure. De plus, le jeu de données a été personnalisé par date afin de réduire la quantité de journaux générés.

Un nouvel algorithme appelé Composition based Decision Tree (CDT) a été testé par rapport à d'autres (PRISM, JRip, OneR). Les résultats montrent que le modèle CDT surpasse les autres algorithmes dans la détection des attaques, avec une précision moyenne de 94,3% , un rappel de 96% et un F1-score de 94,3%.

Travail de Yi et al. [46]

Dans cette étude, une méthode basée sur l'apprentissage automatique pour détecter précocement les menaces persistantes avancées, renforçant ainsi la sécurité en anticipant leurs activités dès le début de leur cycle de vie.

La procédure se concentre sur trois étapes d'une attaque APT. Elle débute par l'extraction des caractéristiques des APT à partir d'un dataset nommé DAPT 2020, utilisant 57 caractéristiques initiales. Des techniques telles que la corrélation de Pearson (PC) et la méthode PCA sont employées pour réduire la dimension des caractéristiques spécifiques à chaque étape. Ensuite, des algorithmes de ML sont appliqués pour repérer les activités malveillantes des APTs. Un arbre de décision avec entropie est utilisé pour optimiser entre les classes, et le renforcement du gradient avec une profondeur maximale de 10 et 50 combine plusieurs modèles simples pour créer un modèle prédictif puissant. Enfin, une méthode de clustering agglomératif est employée pour créer des empreintes digitales des

MPA. Initialement regroupées en 3,6 ou 8 clusters, elles sont étudiées pour identifier les distinctions et les stratégies d'attaques. Les résultats obtenus démontrent que le processus d'extraction de sous-chemins d'attaque avec 6 clusters est fortement corrélé.

L'évaluation repose sur des métriques telles que l'ACC, le F1 score, le rappel et FPR. Chaque phase est évaluée séparément. Lors de la phase d'exfiltration, le modèle proposé par Yi et al. atteint une précision de 81,82% pour le DT avec entropie et 90,91% pour Gradient Boosting. Les détails de l'architecture du modèle sont explicités dans la figure 2.12.

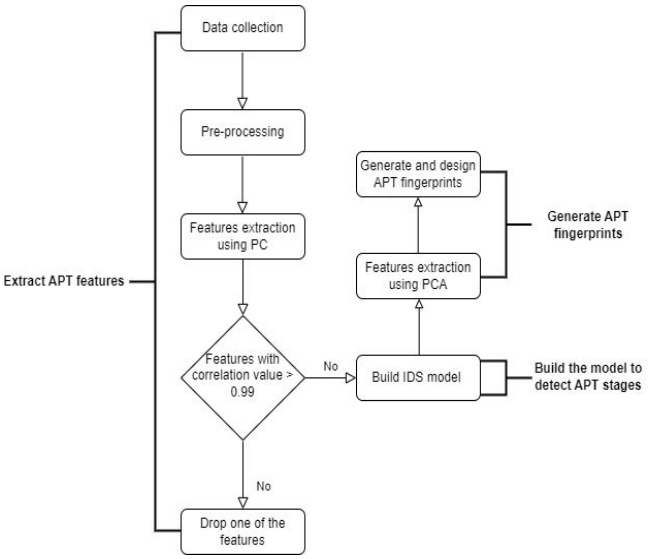

FIGURE 2.12 – L'architecture du modéle proposé par Yi et al.

2.3.6 Classification des travaux basés sur l'analyse du trafic réseau

Le Tableau 2.3 a pour objectif de synthétiser et analyser les articles étudiés dans la section précédente. Il mettra en évidence les méthodologies et résultats présentés par les auteurs, permettant ainsi une comparaison approfondie des approches basées sur sur l'analyse du trafic réseau.

Auteurs	Année	DataSet	Modèle	Résultats (ACC, PRE, F1 score , Recall, FPR)
Chu et al. [12]	2019	NSL-KDD	MLP	ACC=97.82 %
Patel et al. [38]	2022	NSL-KDD	Linear SVM	ACC= 94.85% , PRE= 94.61% , Recall= 96.04%
			DNN	ACC= 94.88% , PRE= 94.04% , Recall= 95.47%
Al-Aamri et al. [1]	2023	non spécifié	CDT	PRE= 94,3% , F1-score= 94,3% , Recall= 96 %
Yi et al. [46]	2023	DAPT 2020	DT	ACC= 93.91% , FPR= 14.11% , Recall= 93.16% , F1-score= 95.99%
			Gradient Boosting avec pronfendeur= 10	ACC= 96.25% , FPR= 2.12% , Recall= 98.19% , F1-score= 97.09%
			Gradient Boosting avec pronfendeur= 50	ACC= 96.16% , FPR= 1.62% , Recall= 98.50% , F1-score= 97.04%

TABLE 2.3 – Tableau comparatif des traveaux basés sur l'analyse du trafic réseau.

2.4 Discussion des travaux existants

Après une analyse approfondie, nous avons identifié trois principales approches dans le domaine de la détection des menaces persistantes avancées utilisant les techniques de Machine Learning : la corrélation d'alertes, la détection des noms de domaines APT et la détection basée sur l'analyse du trafic réseau .

D'une part, l'approche basée sur la corrélation d'alertes offre un potentiel considérable pour retracer le déroulement complet d'une attaque APT. En effet, ces attaques sophistiquées se déploient

généralement en plusieurs étapes distinctes qu'il est crucial de pouvoir relier. Cependant, des défis tels que la grande quantité d'alertes à traiter et la difficulté à filtrer le bruit restent à relever pour ces approches.

D'autre part, la détection des noms de domaines APT en analysant notamment le trafic DNS semble très prometteuse. De nombreux travaux utilisant des techniques comme les forêts aléatoires, le gradient boosting ou encore la régression logistique ont donné lieu à des résultats satisfaisants en termes de précision et de rappel. Cependant, la majorité de ces approches reposent sur des datasets avec des données simulées ou anciennes, ce qui soulève des interrogations quant à leur capacité de généralisation face à de nouvelles variantes d'attaques.

Par ailleurs, les approches basées sur l'analyse du trafic réseau, utilisant diverses caractéristiques des flux, ont également démontré des performances intéressantes. Cependant, une limite majeure réside dans la difficulté à extraire des caractéristiques pertinentes du trafic réseau. En effet, de nombreuses techniques se concentrent sur des attributs statistiques bas niveau comme les tailles de paquets ou les durées de flux. Or, ces caractéristiques ne permettent pas toujours de capturer les spécificités comportementales complexes des menaces APT évoluées.

Par conséquent, il est justifié que cette dernière approche basée sur l'analyse du trafic réseau vaut la peine d'être explorée en détail et de poursuivre des recherches dans ce domaine pour explorer pleinement son potentiel.

2.5 Conclusion

Dans ce chapitre, nous avons examiné plusieurs travaux avancés dans le domaine de la détection des menaces persistantes avancées utilisant les techniques de Machine Learning. Nous avons constaté que les approches basées sur la corrélation d'alertes offrent des perspectives intéressantes pour retracer le déroulement complet d'une attaque APT, tandis que la détection par analyse des noms de domaine semble prometteuse et a donné lieu à des résultats satisfaisants en termes de précision et de rappel . Cependant, l'approche par analyse du trafic réseau apparaît comme particulièrement pertinente à approfondir. Malgré des performances encourageantes, cette approche souffre de limites liées à l'extraction de caractéristiques pertinantes à partir des flux réseaux.

Notre contribution se concentrera sur la conception d'un système de détection des APT, en se focalisant sur une nouvelle méthode de sélection des caractéristiques les plus significatives pour améliorer davantage les performances du modèle conçu.

Chapitre 3

Contribution, résultats et discussion

3.1 Introduction

Dans ce chapitre, nous présentons en détail notre contribution visant à relever les défis de la détection des menaces persistantes avancées en exploitant l'analyse approfondie du trafic réseau. Comme identifié dans l'état de l'art, cette approche offre un potentiel prometteur, mais souffre encore de limites, notamment liées à l'extraction de caractéristiques discriminantes à partir des flux réseau.

Notre méthodologie se concentre sur le développement d'un modèle d'apprentissage machine capable de capturer efficacement les anomalies présentes dans les APT à partir d'une analyse fine des flux réseau.

Ce chapitre détaillera l'ensemble du processus de conception, d'implémentation et d'évaluation de notre solution. Nous présenterons l'architecture générale proposée, une description du jeu de données exploité ainsi que les outils et environnements utilisés. Nous expliquerons ensuite en profondeur les différentes étapes clés telles que le prétraitement des données, la construction de notre modèles et son entraînement. Enfin, nous analyserons les performances obtenues en les comparant aux approches existantes.

3.2 Motivations

La principale motivation réside dans la volonté de relever les défis majeurs dans la détection des menaces persistantes avancées par l'analyse du trafic réseau. Bien que prometteuse, cette approche souffre de limites importantes à améliorer. Un défi clé concerne l'extraction de caractéristiques discriminantes à partir des flux réseau, souvent limitée à des attributs statistiques simples. Ces caractéristiques bas niveau ne capturent pas toujours pleinement les schémas complexes des attaques APT.

Notre objectif est d'explorer une nouvelle méthodologie pour identifier efficacement ces attributs révélateurs d'activités malveillantes, en exploitant les avancées des méthodes ensemblistes, notamment les forêts aléatoires.

3.3 Approche proposée

Notre proposition consiste à exploiter le jeu de données NSL-KDD afin de développer un modèle d'apprentissage automatique pour la détection des APT. Pour cela, nous suivrons une méthodologie rigoureuse en plusieurs étapes.

Tout d'abord, nous procéderons à un prétraitement approfondi des données brutes contenues dans l'ensemble d'entraînement KDDTrain+ et l'ensemble de test KDDTest+. Cela implique de gérer les différents types de variables (binaires, catégorielles, numériques) présentées dans les jeux de données à l'aide des différentes techniques de prétraitement des données.

Le jeu de données NSL-KDD contient des informations sur différents types d'attaques classées de manière détaillée. Afin de simplifier le problème de classification, nous avons créé un dictionnaire qui permettra de regrouper ces différents types d'attaques en cinq classes : Normal, DoS, Probe, R2L et U2R.

Ensuite, une sélection des caractéristiques les plus pertinentes à partir des données prétraitées a été effectuée. Nous avons utilisé une méthode de sckit-learn appelée SelectKBest basée sur l'analyse de la variance afin d'identifier les variables les plus discriminantes et de réduire la dimensionnalité du problème.

Enfin, une étape de suréchantillonnage des classes minoritaires est effectuée a travers la méthode SMOTE (Synthetic Minority Over-sampling Technique) afin d'améliorer l'apprentissage du modèle sur ces enregistrements les plus rares. Cette approche vise à équilibrer les différentes classes, facilitant ainsi la généralisation du classifieur.

Le modèle de classification choisi est un Random Forest , une méthode d'apprentissage supervisé reconnue pour sa robustesse et sa capacité à gérer des jeux de données complexes. Comme notre jeu de données contient deux ensembles distincts, l'entraînement du modèle RF s'effectue sur l'ensemble d'apprentissage dans le but d'identifier les cinq classes d'attaques contenues dans le jeu de données.

Pour évaluer les performances de notre modèle, des métriques classiques sont calculées sur les données de test, à savoir l'Accuracy, la précision et le rappel. Ces indicateurs permettent de quantifier la capacité de notre modèle à prédire correctement les différents types d'attaques. La figure 3.1 montre l'architecture de notre approche proposée.

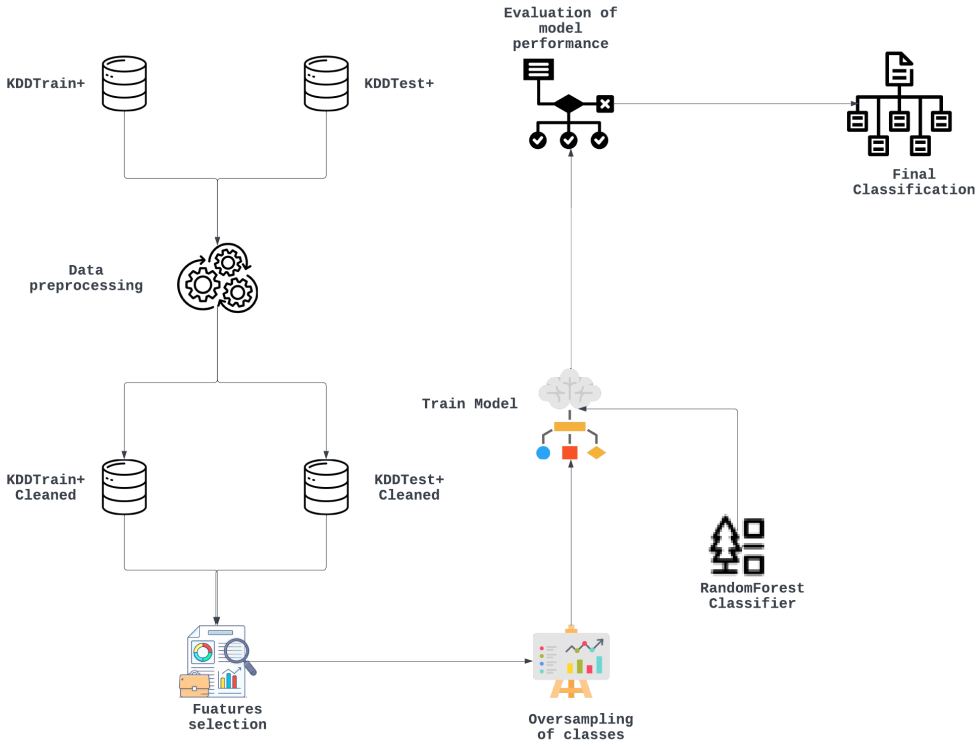

FIGURE 3.1 – Architecture globale de l'approche proposée.

3.4 Description du jeu de données

Le choix du dataSet est une étape importante pour la création de modèle d'apprentissage automatique capable de détecter les menaces persistantes avancées . Dans le tableau 3.1 ci-dessous nous allons présenter quelques ensembles de données conçu pour la détection des cyberattaques [35].

Dataset	**Type d'attaque**	**Etiqueté**	**Année**
DARPA	Attaques réseaux	Oui	1998
KDD Cup	Attaques réseaux	Oui	1999
Mawi	Attaques réseaux	Oui	2006
NSL-KDD	Attaques réseaux	Oui	2009
UNB-15	Attaques réseaux	Oui	2015
CICIDS2017	Attaques réseaux	Oui	2017

TABLE 3.1 – DataSets publiés sur les cyberattaques avec année[35].

Concernant le DataSet utilisé pour la réalisation de notre approche, il s'agit du DataSet NSL-KDD (NSL Knowledge Discovery and Data Mining), conçu en 2009 pour capturer les différents aspects des attaques en temps réel. NSL-KDD est développé comme une amélioration du dataSet KDD Cup 1999, il vise à résoudre les problèmes de son prédécesseur, à savoir la redondance des données dans

l'ensemble d'entraînement, ainsi que le nombre de données dans les ensembles d'entraînement et de test est suffisamment élevé pour permettre des expériences complètes sans nécessité de sélection aléatoire d'un petit échantillon.

Le dataset que nous avons utilisé se compose d'un ensemble d'apprentissage total (KDDTrain+), contenant 125 973 échantillons, et d'un ensemble de test total (KDDTest+), contenant 22 544 échantillons.

La composition de NSL-KDD comprend des enregistrements étiquetés comme trafic normal ou comme attaque, celle-ci est regroupée en quatre catégories principales : les attaques par déni de service (DoS), les attaques par sondage dites Probe, les attaques (R2L) et les attaques (U2R).

Le tableau 3.2 ci-dessous présente plus de détails sur la composition du DataSet NSL-KDD.

Dataset	Nombre d'enregistrements				
	Normal	DoS	Probe	U2R	R2L
KDDTrain+.txt	67343	45927	11656	52	995
KDDTest+.txt	9711	7636	2423	200	2574

TABLE 3.2 – NSL-KDD DataSet [21].

3.5 Outils de développement

3.5.1 Matériel

Les principales caractéristiques de la machine utilisée pour implémenter et tester notre approche sont représentées dans le Tableau 3.3 :

Matériel	**Caractéristiques**
HP EliteBook 840 G5	Processeur : Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz 2.11 GHz Mémoire vive : 16,00 Go Disque dur : 512 Go SSD Systéme d'exploitation : Windows 10 Famille

TABLE 3.3 – Caractéristiques de la machine.

3.5.2 Langage, Logiciels et bibliothèques

Nous avons utilisé Google Colaboratory , et les différentes bibliothèques que nous allons présenté ci-dessous :

Python : est un langage de programmation interprété, multi-paradigme et multiplateformes, open source, avec une bibliothèque standard étendue et des fichiers sources modifiables. Il offre de nombreuses bibliothèques additionnelles pour divers domaines, une grande portabilité sur différents systèmes d'exploitation, des structures de données efficaces et une approche simple mais robuste de la programmation orientée objet.

FIGURE 3.2 – Python [39].

Google Colaboratory : Google Colab est un environnement cloud gratuit pour exécuter des notebooks Python dans un navigateur, avec accès à des ressources de calcul gratuites, notamment des processeurs graphiques (GPU) via l'infrastructure Google.

FIGURE 3.3 – Google Colaboratory.

Numpy : abréviation de "Numerical Python", est une bibliothèque open source en Python pour manipuler des tableaux à n dimensions. Elle allie flexibilité de Python et vitesse du code C compilé, rendant la programmation facile et productive.

FIGURE 3.4 – Numpy.

Pandas : le terme "Pandas" combine "Panel Data" et "Python Data Analysis", bibliothèque Python pour la science des données offre structures robustes, expressives et adaptatives pour manipulation aisée de tableaux numériques.

FIGURE 3.5 – Pandas.

Matplotlib : est une bibliothèque open source qui facilite la création de divers types de graphiques pour représenter des données de manière statique ou animée. Souvent employée en science des données et en ingénierie, elle aide à la compréhension des données grâce à ses différents graphiques.

FIGURE 3.6 – Matplotlib.

Scikit-Learn : est une bibliothèque open source écrite en Python, Elle propose une panoplie d'outils performants pour l'apprentissage automatique, couvrant des tâches telles que la classification, la régression, le regroupement et la réduction de la dimensionnalité.

FIGURE 3.7 – Scikit-Learn.

3.6 Implémentation

3.6.1 Préparation des données

Le jeu de données NSL-KDD comporte trois types d'attributs : nominaux, binaires et numériques. Les attributs 2, 3 et 4 sont nominaux, tandis que les attributs 7, 12, 14, 15, 21 et 22 sont binaires et les autres attributs sont de type numérique.

Le tableau 3.4 ci-dessous montre en détail la liste des attributs du jeu de données, ainsi que leurs types et une description de chacun.

N°	Nom de l'attribut	Type	Description
1	Duration	Numérique	La durée de connexion
2	Protocol_type	Nominal	Protocole utilisé dans la connexion (tcp, udp, icmp)
3	Service	Nominal	Service réseau de destination, (http, telnet, ftp_data, etc.)
4	Flag	Nominal	Statut de la connexion – Normal ou Erreur (SF, REJ, S0, S1, etc.)
5	Src_bytes	Numérique	Nombre d'octets de données transférés de la source à la destination (491, etc.)
6	Dst_bytes	Numérique	Nombre d'octets de données transférés de la destination à la source (0, etc.)
7	Land	Binaire	Si l'adresse IP de source et destination et le numéro de port sont les mêmes alors, land=1 sinon land=0
8	Wrong_fragment	Numérique	Nombre total de fragments erronés dans cette connexion
9	Urgent	Numérique	Nombre de paquets urgents
10	Hot	Numérique	Nombre d'indicateurs "Hot"
11	Num_failed_logins	Numérique	Nombre de tentatives de connexion échouées
12	Logged_in	Binaire	Si connecté avec succès alors logged_in=1 sinon logged_in=0
13	Num_compromised	Numérique	Nombre de conditions compromises
14	Root_shell	Binaire	1 si le root shell est obtenu, 0 autrement
15	Su_attempted	Binaire	1 si la commande "su root" a été tentée ou utilisée, sinon 0
16	Num_root	Numérique	Nombre d'accès " root " ou nombre d'opérations effectuées comme racine dans la connexion
17	Num_file_creations	Numérique	Nombre d'opérations de création de fichiers
18	Num_shells	Numérique	Nombre d'invites du shell
19	Num_access_files	Numérique	Nombre d'opérations sur les fichiers de contrôle d'accès
20	Num_outbound_cmds	Numérique	Nombre de commandes sortantes dans une session FTP
21	Is_host_login	Binaire	1 si la connexion appartient à la liste du ≪ hot ≫ (root ou admin) ; sinon 0
22	Is_guest_login	Binaire	1 si le login est un login ≪ guest ≫ ; sinon 0
23	Count	Numérique	Nombre de connexions vers le même hôte de destination que la connexion en cours dans les deux dernières secondes
24	Srv_count	Numérique	Nombre de connexions vers le même service (N° Port) que la connexion en cours dans les deux dernières secondes
25	Serror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag s0, s1, s2 ou s3, parmi les connexions agrégées dans count
26	Srv_serror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag s0, s1, s2 ou s3, parmi les connexions agrégées dans srv_count
27	Rerror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag REJ, parmi les connexions agrégées dans count
28	Srv_rerror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag REJ, parmi les connexions agrégées dans srv_count
29	Same_srv_rate	Numérique	Le pourcentage de connexions qui sont au même service, parmi les connexions agrégées dans count
30	Diff_srv_rate	Numérique	Le pourcentage de connexions qui sont aux différents services, parmi les connexions agrégées dans count
31	Srv_diff_host_rate	Numérique	Le pourcentage de connexions qui sont à différentes machines de destination, parmi les connexions agrégées dans srv_count
32	Dst_host_count	Numérique	Nombre de connexions ayant la même adresse IP de l'hôte de destination
33	Dst_host_srv_count	Numérique	Nombre de connexions ayant le même numéro de port
34	Dst_host_same_srv_rate	Numérique	Le pourcentage de connexions qui sont au même service, parmi les connexions agrégées dans dst_host_count
35	Dst_host_diff_srv_rate	Numérique	Le pourcentage de connexions qui sont aux différents services, parmi les connexions agrégées dans dst_host_count
36	Dst_host_same_src_port_rate	Numérique	Le pourcentage de connexions qui sont au même port de source, parmi les connexions agrégées dans dst_host_srv_count
37	Dst_host_srv_diff_host_rate	Numérique	Le pourcentage de connexions qui sont à différentes machines de destination, parmi les connexions agrégées dans dst_host_srv_count
38	Dst_host_serror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag s0, s1, s2 ou s3, parmi les connexions agrégées dans dst_host_count
39	Dst_host_srv_serror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag s0, s1, s2 ou s3, parmi les connexions agrégées dans dst_host_srv_count
40	Dst_host_rerror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag REJ, parmi les connexions agrégées dans dst_host_count
41	Dst_host_srv_rerror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag REJ, parmi les connexions agrégées dans dst_host_srv_count
42	Label	Nominal	Label de classification

TABLE 3.4 – Les détails des attributs de NSL-KDD utilisés [27].

Chargement des données

Dans la figure 3.8, nous importons les données depuis des fichiers *.txt* , nous avons fait usage des fichiers ci-dessous :

— **Le fichier "KDDTrain+" :** constitue notre ensemble d'entrainement.

— **Le fichier "KDDTest+" :** constitue notre ensemble de test.

```
df_train = pd.read_csv('/content/KDDTrain+.txt' , names= header_names)
df_test = pd.read_csv('/content/KDDTest+.txt' , names= header_names)
```

FIGURE 3.8 – Chargement des données

Numérisation des données

Comme mentionné précédemment, la base de données NSL-KDD comporte 3 attributs avec des données nominales : "protocol_type", "service" et "flag".

Les trois attributs mentionnés ont été transformés en données numériques en utilisant la méthode de conversion alphabétique simple pour numériser les attributs de type nominal, par exemple, l'attribut "protocole_type" a trois valeurs catégorielles distinctes : "tcp", "icmp" et "udp". Les valeurs sont d'abord triées par ordre alphabétique avant de recevoir un numéro pour chaque catégorie de valeurs distincte . De méme , pour l'attribut "service" qui contient 70 services , et l'attribut "flag" contient 10 flags. La figure 3.9 présente le processus de numérisation effectué.

```
from sklearn.preprocessing import LabelEncoder

variables_catégorielles = df_train.select_dtypes(include=['object'])

def encoder(df_encoded, colonnes):
    for colonne in colonnes:
        encoder = {value: i+1 for i, value in enumerate(sorted(df_encoded[colonne].unique()))}
        df_encoded[colonne] = df_encoded[colonne].map(encoder)
    return df_encoded

df_train = encoder(df_train, variables_catégorielles)

df_test = encoder(df_test, variables_catégorielles)
```

FIGURE 3.9 – Numérisation des variables catégorielles

Les trois tableaux 3.5, 3.6, 3.7 ci-dessous montrent les valeurs numériques attribuées à chaque catégorie dans les attributs "protocole_type", "flag" et "service" de la base de données NSL-KDD après leur transformation en données numériques.

Numéro	protocole_type
1	icmp
2	tcp
3	udp

TABLE 3.5 – Liste des valeurs uniques de l'attribut protocole_type.

Numéro	flag
1	OTH
2	REJ
3	RSTR
4	RSTO
5	RSTOS0
6	S0
7	S1
8	S2
9	S3
10	SF
11	SH

TABLE 3.6 – Liste des valeurs uniques de l'attribut flag.

Numéro	service	Numéro	service
1	aol	36	netbios_dgm
2	auth	37	netbios_ns
3	bgp	38	netbios_ssn
4	courier	39	netstat
5	csnet_ns	40	nnsp
6	ctf	41	nntp
7	daytime	42	ntp_u
8	discard	43	other
9	domain	44	pm_dump
10	domain_u	45	pop_2
11	echo	46	pop_3
12	eco_i	47	printer
13	ecr_i	48	private
14	efs	49	red_i
15	exec	50	remote_job
16	finger	51	rje
17	ftp	52	shell
18	ftp_data	53	smtp
19	gopher	54	sql_net
20	harvest	55	ssh
21	hostnames	56	sunrpc
22	http	57	supdup
23	http_2784	58	systat
24	http_443	59	telnet
25	http_8001	60	tim_i
26	imap4	61	time
27	IRC	62	tftp_u
28	iso_tsap	63	urh_i
29	klogin	64	urp_i
30	kshell	65	uucp
31	ldap	66	uucp_path
32	link	67	vmnet
33	login	68	whois
34	mtp	69	X11
35	name	70	Z39_50

TABLE 3.7 – Liste des valeurs uniques de l'attribut service.

Normalisation des données

Normaliser les données implique de réajuster les valeurs d'un ensemble de données pour les rendre comparables et analysables, en les ramenant généralement dans une plage de 0 à 1, afin d'éviter les biais dus à l'échelle des données. Pour ce faire , nous avons choisi d'utiliser la fonction **MinMaxScaler()** de la bibliothèque sklearn.preprocessing définie par la formule suivante 3.1 :

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{3.1}$$

.

où :

X est le vecteur de données que nous souhaitons normaliser.

X_{min} est la valeur minimale dans X.

X_{max} est la valeur maximale dans X.

et X_{norm} est le vecteur de données normalisé résultant.

Comme le montre la figure 3.10, nous allons procéder à la normalisation des données pour les ensembles d'apprentissage et de test.

```
from sklearn.preprocessing import MinMaxScaler
colonnes_numeriques = df_train.select_dtypes(include='number').drop('label', axis=1).columns
scaler = MinMaxScaler(feature_range=(0, 1))
df_train[colonnes_numeriques] = scaler.fit_transform(df_train[colonnes_numeriques])
df_test[colonnes_numeriques] = scaler.transform(df_test[colonnes_numeriques])
```

FIGURE 3.10 – Normalisation des données avec MinMaxScaler().

Mappage des classes

Le tableau 3.8 organise les différentes catégories d'attaques, en spécifiant le nombre d'attaques dans chaque catégorie et en listant les attaques spécifiques associées à chacune d'elles.

Catégorie	Nombre d'attaques	Liste d'attaques
DoS	11	apache2, back, land, mailbomb, neptune, pod, processtable, smurf, teardrop, udpstorm, worm
Probe	6	ipsweep, mscan, nmap, portsweep, saint, satan
U2R	7	buffer_overflow, loadmdoule, perl, ps, rootkit, sqlattack, xterm
R2L	15	imap, multihop, warezclient, phf, sendmail, snmpgetattack, snmpguess, spy, xclock, warezmaster, xsnoop, ftp_write, guess_passwd, http_tunnel, named

TABLE 3.8 – Liste des attaques par catégories du DataSet NSL-KDD.

Nous avons établi un processus de mappage à l'aide d'un dictionnaire Python nommé `attack_mapping`, où chaque attaque est associée à sa catégorie respective. Ce mappage a été implémenté dans une fonction `map_attack` , qui, lorsqu'elle reçoit le type d'attaque en entrée, renvoie sa catégorie selon le dictionnaire de mappage. La figure 3.11 présente ce processus de mappage qui a été appliqué aux données d'entraînement et de test, étiquetant ainsi chaque attaque avec sa catégorie appropriée, sauf si elle était déjà de type "normal".

```
def map_attack(label):
    if label == "normal":
        return "normal"
    else:
        return attack_mapping.get(label, label)

df_train['label'] = df_train["label"].apply(map_attack)
df_test['label'] = df_test["label"].apply(map_attack)
```

FIGURE 3.11 – Fonction de mappage des éthiquettes.

3.6.2 Sélection des caractéristiques

Nous avons utilisé la méthode **SelectKBest()** de la bibliothèque `scikit-learn` en Python pour choisir les caractéristiques les plus pertinentes pour l'apprentissage.

La figure 3.12 détaille cette méthode qui vise à sélectionner les k meilleures caractéristiques d'un ensemble de données en utilisant le test statistique `f_classif` de l'analyse de variance (ANOVA).

Ce test évalue la variabilité des caractéristiques en calculant les scores F et les valeurs p, comparant les variances inter-classes et intra-classes pour déterminer si les moyennes des valeurs de la caractéristique varient significativement entre les différentes classes de la cible.

```
from sklearn.feature_selection import SelectKBest, f_classif
k_values = range(1, X_train.shape[1] + 1)

f_scores = []
for k in k_values:
    selector = SelectKBest(f_classif, k=k)
    selector.fit(X_train, y_train)
    f_scores.append(selector.scores_)

mean_f_scores = np.mean(f_scores, axis=1)
best_k = k_values[np.argmax(mean_f_scores)]

selector = SelectKBest(f_classif, k=best_k)
selector.fit(X_train, y_train)
X_train_selected = selector.transform(X_train)
X_test_selected = selector.transform(X_test)
selected_features_indices = selector.get_support(indices=True)
selected_features = X_train.columns[selected_features_indices]

print("Nombre de caractéristiques sélectionnées:", best_k)
print("Caractéristiques sélectionnées:" , selected_features)
```

FIGURE 3.12 – Sélection des attributs les plus importants

Grâce à **SelectKBest()** identifie les caractéristiques les plus discriminantes, et dans notre cas, elle a sélectionné 12 caractéristiques : 'flag', 'wrong_fragment', 'logged_in', 'count', 'serror_rate', 'srv_serror_rate', 'same_srv_rate', 'dst_host_srv_count', 'dst_host_same_srv_rate','dst_host_same_src_port_rate', 'dst_host_serror_rate' et 'dst_host_srv_serror_rate'.

3.6.3 Suréchantillonage des classes

Le tableau 3.9 ci-dessous représente les pourcentages de différents types de trafic dans deux ensembles de données, KDDTrain+.txt et KDDTest+.txt.

Dataset	Classe				
	Normal	DoS	Probe	U2R	R2L
KDDTrain+.txt	53.45 (%)	36.46(%)	9.26(%)	0.04(%)	0.79(%)
KDDTest+.txt	43.07(%)	33.87(%)	10.76 (%)	0.89(%)	11.41(%)

TABLE 3.9 – Répartition par pourcentage des classes de NSL-KDD [21].

Nous avons appliqué la technique **SMOTE** pour remédier au problème de déséquilibre de classes constatés dans le tableau 3.9 notamment dans les ensembles de données U2R et R2L.

Cette technique résout ce problème en générant de manière synthétique de nouveaux exemples de la classe minoritaire, en se basant sur des techniques d'interpolation entre les exemples existants en utilisant la stratégie de suréchantillonnage automatique (`sampling_strategy='auto'`) .

Les données d'entraînement initialement sélectionnées (`X_train_selected`) et les étiquettes associées (`y_train`) sont ensuite remplacées par les données resamplées (`X_train_resampled`) et les nouvelles étiquettes (`y_train_resampled`) pour l'entraînement de notre modèle.

3.6.4 Construction du modèle

Nous avons opté pour un modèle de classification `RandomForestClassifier` . D'abord, le modèle est entraîné sur les données d'entraînement resamplées (`X_train_resampled` et `y_train_resampled`) qui ont été prétraitées avec la méthode SMOTE pour gérer le déséquilibre de classes. Ensuite, le modèle entraîné est utilisé pour faire des prédictions sur l'ensemble de test (`X_test_selected`), générant ainsi des prédictions de classe stockées dans `y_pred`.

Le choix du modèle `RandomForestClassifier` est pertinent pour plusieurs raisons puisqu'il s'agit d'un modèle d'ensemble puissant qui combine les prédictions de multiples arbres de décision, réduisant ainsi le sur-apprentissage et améliorant la généralisation. La figure 3.13 illustre l'entraînement de notre modéle.

```
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train_resampled, y_train_resampled)
y_pred = rf_model.predict(X_test_selected)
```

FIGURE 3.13 – Entrainement du RandomForestClassifier.

3.6.5 Évaluation du modèle

Pour évaluer les performances de notre classifieur , nous avons utilisé les métriques de performance suivantes :

Accuracy

appelée également l'exactitude, est une mesure qui représente le pourcentage total de prédictions correctes réalisées par le modèle parmi toutes les prédictions effectuées, qu'elles concernent des attaques détectées correctement ou non. Elle est donnée par la formule 3.2 suivante :

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

Précision

Une mesure qui évalue évalue le pourcentage d'attaques correctement identifiées parmi toutes les alertes générées par le modèle. Elle est donnée par la formule 3.3 suivante :

$$\text{Précision} = \frac{TP}{TP + FP} \tag{3.3}$$

Rappel

Appelé également "Recall" en anglais , mesure la proportion d'attaques détectées correctement parmi toutes les attaques réellement présentes dans les données. Il est donné par la formule 3.4 suivante :

$$\text{Rappel} = \frac{TP}{TP + FN} \tag{3.4}$$

F1-score

Définit comme étant la moyenne harmonique de la précision et du rappel , il offre une mesure équilibrée de la capacité d'un modèle à identifier correctement à la fois les instances positives et négatives. Elle est donnée par la formule 3.5 suivante :

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.5}$$

Matrice de confusion

Un tableau qui sert à évaluer l'efficacité d'un modèle de classification sur un jeu de données test avec des valeurs réelles connues. Ce tableau montre les prévisions du modèle par rapport aux données réelles, avec quatre cellules clés : False Negative (FN), False Positive (FP), True Positive (TP) et True Negative (TN) [24].

FN : Le modèle prédit à tort une classe négative (pas d'attaque) alors que la vraie classe est positive (il y a une attaque). Cela signifie que le modèle a manqué de détecter une véritable attaque.

FP : Le modèle prédit à tort une classe positive (attaque détectée) alors que la vraie classe est négative (pas d'attaque). Cela indique que le modèle a généré une alerte incorrecte pour une activité qui n'était pas une attaque.

TP : Le modèle prédit correctement une classe positive (attaque détectée) et la vraie classe est également positive (il y a bien une attaque). Cela signifie que le modèle a correctement identifié une attaque.

TN : Le modèle prédit correctement une classe négative (pas d'attaque) et la vraie classe est également négative (il n'y a pas d'attaque). Cela montre que le modèle a correctement identifié une situation où il n'y avait pas d'attaque.

La figure 3.14 illustre la représentation tabulaire d'une matrice de confusion.

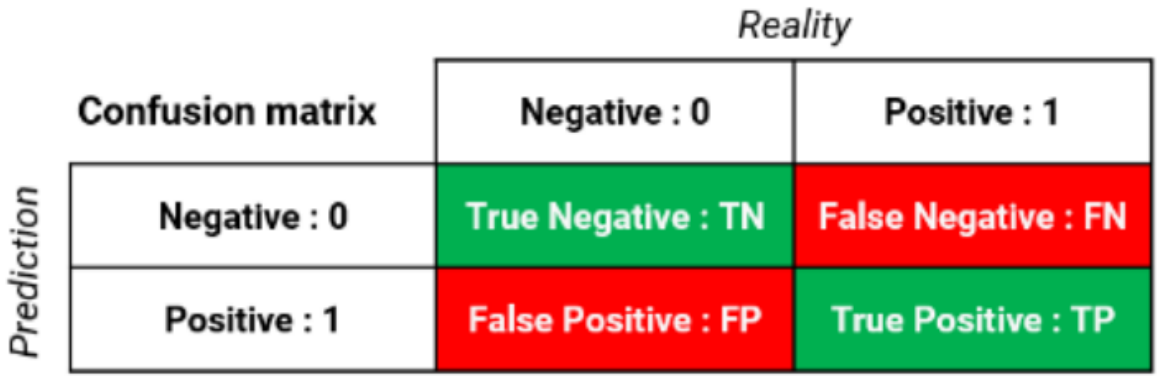

FIGURE 3.14 – Matrice de confusion [26].

Dans le cadre de notre analyse, nous avons employé une matrice de confusion afin d'examiner de près les performances de notre modèle Random Forest en ce qui concerne la détection des diverses attaques APT. En effet, la figure 3.15 illustre cette matrice de confusion générée par notre modèle, ce qui nous a permis d'effectuer une analyse approfondie et de déduire les conclusions suivantes :

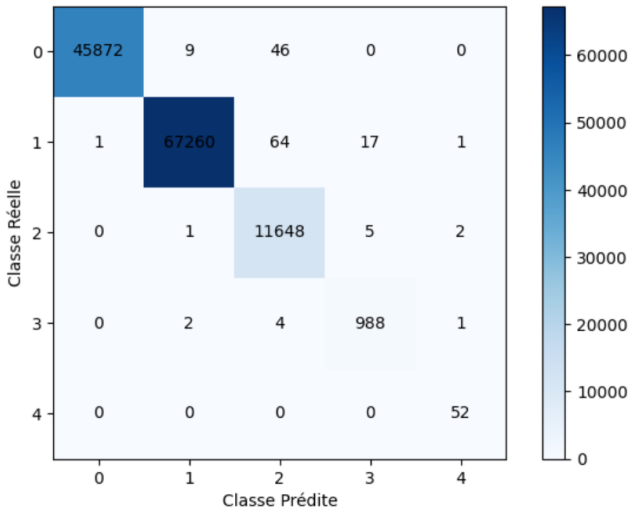

FIGURE 3.15 – Matrice de confusion de notre modéle.

Pour la classe DoS, nous avons observé un nombre élevé de TP à 45 872, ce qui montre que notre modèle a bien détecté les attaques de type DoS. De plus, le nombre élevé de TN à 80 045 indique une capacité à reconnaître correctement le trafic non-DoS. Cependant, un FP de 1 signifie que le modèle a une précision élevée et émet très peu d'alertes erronées. Le nombre de FN est de 55 ;

le modèle a échoué à détecter 55 attaques DoS réelles, les classant incorrectement comme d'autres classes. Bien que ce nombre soit relativement bas par rapport au total, une marge d'amélioration doit être envisagée pour la détection des attaques DoS.

Concernant la classe Normal, nous avons obtenu un nombre élevé de TP à 67260, indiquant une détection précise du trafic normal. Les TN à 58618 montrent également une capacité à reconnaître le reste du trafic comme n'appartenant pas à cette classe. Cependant, quelques FP à 12 et FN à 83 nécessitent des ajustements pour améliorer la spécificité et la sensibilité de la détection pour cette classe.

Pour la classe Probe, le nombre de TP à 11648 montre que notre modèle a bien identifié cette classe dans les données. Les TN à 114203 montrent également une capacité à reconnaître le reste du trafic comme n'appartenant pas à cette classe. Cependant, quelques FP à 114 et FN à 8 nécessitent des ajustements pour améliorer la précision de la détection.

La classe R2L présente un nombre de TP à 988, indiquant une détection correcte de cette classe spécifique. Les TN à 124956 montrent également une capacité à reconnaître le reste du trafic comme n'appartenant pas à cette classe. Cependant, quelques FP à 22 et FN à 7 nécessitent également des améliorations pour une détection plus précise.

Enfin, pour la classe U2R, nous avons obtenu un nombre de TP à 52, indiquant une capacité de notre modèle à détecter cette classe spécifique. Les TN à 125917 montrent également une capacité à reconnaître le reste du trafic comme n'appartenant pas à cette classe. De plus, un FP de 4 et l'absence de FN pour cette classe est un signe positif de la performance de notre modèle.

Rapport de classification

La figure 3.16 présente le rapport de classification qui démontre d'excellentes performances pour les classes DoS et Normal, avec une précision, un rappel et un score F1 de 1.00, indiquant une classification parfaite pour ces classes majoritaires. La classe Probe affiche également des performances satisfesantes avec une précision de 0.99 et un rappel de 1.00. Les classes R2L et U2R présentent des scores légèrement inférieurs mais toujours très bons, avec des scores F1 de 0.99 et 0.96 respectivement.

	precision	recall	f1-score	support
1	1.00	1.00	1.00	45927
2	1.00	1.00	1.00	67343
3	0.99	1.00	0.99	11656
4	0.98	0.99	0.99	995
5	0.93	1.00	0.96	52
accuracy			1.00	125973
macro avg	0.98	1.00	0.99	125973
weighted avg	1.00	1.00	1.00	125973

FIGURE 3.16 – Rapport de classification généré par notre modèle.

3.7 Étude comparative

Le tableau 3.10 suivant présente une comparaison des performances de nos modèles par rapport aux études antérieures.

Modèle	Accuracy%	Precision%	Recall%
SVM	94.85%	94.61%	96.04%
DNN	94.88%	94.04%	95.47%
RF	**99.87 %**	**97.93%**	**99.79%**

TABLE 3.10 – Comparaison des résultats.

En termes d'accuracy , notre modèle atteint un score de 99,87%, dépassant largement les 94,85% de SVM et 94,88% de DNN. Pour la précision, nous obtenons 97,93%, contre 94,61% pour SVM et 94,04% pour DNN, démontrant une meilleure capacité à éviter les faux positifs. Notre rappel de 99,79% est également supérieur à celui de SVM 96,04% et DNN 95,47%, ce qui signifie que notre modèle détecte plus d'attaques réelles.

Ces résultats mettent en évidence la performance de notre approche basée sur les forêts aléatoires, capable de mieux capturer la complexité des flux réseau malveillants .

La figure 3.17 ci-dessous contient la représentation graphique des métriques de performances exposées ci-dessus :

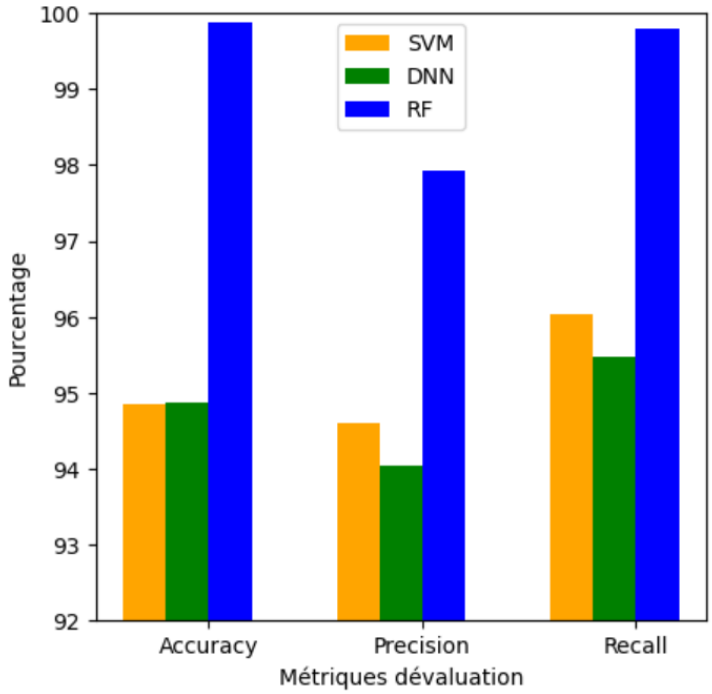

FIGURE 3.17 – Performances des modéles exitants et de notre modéle.

3.8 Discussion des résultats

Les résultats obtenus démontrent clairement l'efficacité de notre approche basée sur les forêts aléatoires pour la détection des menaces persistantes avancées à partir de l'analyse du trafic réseau.

Notre modèle identifie avec précision les différents types d'attaques présents dans le jeu de données NSL-KDD. De plus, il se distingue par des temps de réponse et de détection remarquablement rapides, inférieurs à **20 millisecondes** et **12 millisecondes** respectivement, ce qui est crucial dans la détection proactive des menaces. Cette rapidité permet non seulement une réaction immédiate aux incidents détectés, mais aussi une minimisation des dommages potentiels en réduisant le temps d'exposition aux attaques. Cependant, certaines limitations subsistent. Bien que notre modèle ait atteint des performances remarquables sur les données de test, il est important de souligner que celles-ci proviennent d'un environnement simulé.

Dans des conditions réelles, le trafic réseau est susceptible d'être plus complexe et dynamique, ce qui pourrait impacter les capacités de détection. De plus, les attaques APT évoluent constamment, adoptant de nouvelles techniques pour contourner les systèmes de sécurité. Il sera essentiel de mettre régulièrement à jour notre modèle avec de nouveaux jeux de données reflétant les menaces émergentes.

Il convient également de noter que notre approche ne couvre que les trois premières phases des APT (reconnaissance, la compremission initiale, le contrôle et commandement, mais ne traite pas les phases ultérieures telles que le mouvement latéral et l'exfiltration et la post-exfiltration.

3.9 Conclusion

Ce chapitre a présenté notre contribution visant à relever les défis de la détection des menaces persistantes avancées par l'analyse approfondie du trafic réseau. Nous avons conçu et mis en œuvre un modèle d'apprentissage automatique basé sur les forêts aléatoires, capable de capturer efficacement les anomalies présentes dans les APT à partir d'une analyse fine des flux réseau. Les résultats obtenus se sont avérés très prometteurs, dépassant les performances des approches existantes sur le jeu de données NSL-KDD.

Conclusion générale

Dans le cadre de ce mémoire, nous avons exploré l'application des méthodes de Machine Learning pour améliorer la détection des menaces persistantes avancées. Après avoir présenté les concepts fondamentaux de la sécurité, des Advanced Persistent Threat et du Machine Learning, nous avons mis l'accent sur son utilisation pour la détection des attaques APT.

L'état de l'art a été réalisé pour examiner les travaux connexes dans le domaine de l'application des méthodes de ML pour améliorer la détection des APT. Nous avons identifié trois approches principales : la détection basée sur la corrélation d'alertes, la détection basée sur les noms de domaines APT et la détection basée sur l'analyse du trafic réseau. Force est de constater que cette dernière possède une limitation majeure qui réside dans la difficulté d'extraire des caractéristiques pertinentes du trafic réseau.

Notre approche est axée sur l'élaboration d'un modèle basé sur les forêts aléatoires pour une détection meilleure des APT mettant en évidence l'importance de la phase de sélection des caractéristiques dans l'analyse du trafic réseau. En utilisant des attributs plus significatifs, notre modèle a pu mieux capturer les comportements malveillants typiques des APT et a démontré une capacité supérieure à détecter les anomalies associées à ces attaques. Les phases couvertes par notre approche incluent la reconnaissance, la compromission initiale et le commandement et contrôle.

Les tests expérimentaux ont été menés sur le jeu de données NSL-KDD. Les résultats obtenus en termes d'accuracy, de précision et de rappel s'avèrent très prometteurs, et des FPR et FNP assez faibles, dépassant ainsi les approches existantes (SVM et DNN). Étant donné la sophistication des APT, il est important de souligner que ces données ne reflètent pas toujours la complexité des environnements réels.

Les perspectives futures de cette recherche visent à étendre notre modèle. Premièrement, nous devons élargir notre approche pour qu'elle couvre l'ensemble du cycle de vie des APT, incluant les phases de mouvement latéral, d'exfiltration et de post-exfiltration des données. Deuxièmement, l'application de modèles de deep learning plus sophistiqués pourrait encore améliorer les performances de détection. Une combinaison des dispositifs de sécurité avec notre modèle de détection pourrait être une solution pour diminuer l'impact majeur de ces attaques. Enfin, tester notre modèle sur des

jeux de données plus récents et variés, tels que le dataset DAPT 2020, permettra d'évaluer sa robustesse et sa généralisation face aux APT émergentes.

Bibliographie

[1] AL-AAMRI, A. S., ABDULGHAFOR, R., TURAEV, S., AL-SHAIKHLI, I., ZEKI, A., AND TALIB, S. Machine learning for apt detection. *Sustainability 15*, 18 (2023), 13820.

[2] ALBERT, P. *Le Machine Learning avec Python : De la Théorie à la Pratique*. Éditions TechPress, Paris, France, 2022.

[3] ALSHAMRANI, A., MYNENI, S., CHOWDHARY, A., AND HUANG, D. A survey on advanced persistent threats : Techniques, solutions, challenges, and research opportunities. *IEEE Communications Surveys amp ; Tutorials 21*, 2 (2019), 1851–1877.

[4] ANALYTICS YOGI. Overfitting & underfitting in machine learning. https://vitalflux.com/overfitting-underfitting-concepts-interview-questions/, Consulté le : 04/04/2024.

[5] BLOCH, L., AND WOLFHUGEL, C. *Sécurité informatique Principes et méthode à l'usage des DSI, RSSI et administrateurs* . Editions Eyrolles, Paris, France, 2009.

[6] BOUKHARROU, R. Sécurité des réseaux. Support de cours, 2020. Disponible à : https://elearning.univ-constantine2.dz/elearning/enrol/index.php?id=2503.

[7] BREWER, R. Advanced persistent threats : minimising the damage. *Network Security 2014*, 4 (2014), 5–9.

[8] CHAUHAN, A. Random forest classifier and its hyperparameters, 2021.

[9] CHEN, P., DESMET, L., AND HUYGENS, C. *A Study on Advanced Persistent Threats*. Springer Berlin Heidelberg, 2014, pp. 63–72.

[10] CHO, D. X., AND NAM, H. H. A method of monitoring and detecting apt attacks based on unknown domains. *Procedia Computer Science 150* (2019), 316–323.

[11] CHOUARFIA, A. Introduction à la sécurité des réseaux et des systèmes d'information. Support de cours, 2010. Disponible à : https://www.exoco-lmd.com/securite/introduction-a-la-securite-des-reseaux-et-des-systemes-dinformation/?action=dlattach;attach=7788.

[12] CHU, W.-L., LIN, C.-J., AND CHANG, K.-N. Detection and classification of advanced persistent threats and attacks using the support vector machine. *Applied Sciences 9*, 21 (2019), 4579.

[13] DAS, A., SHEN, M.-Y., SHASHANKA, M., AND WANG, J. Detection of exfiltration and tunneling over dns. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (2017), IEEE, pp. 735–741.

[14] DATA SCIENCE TEAM. Gradient boosting – ce que vous devez savoir. https://datascience.eu/fr/apprentissage-automatique/gradient-boosting-ce-que-vous-devez-savoir, Consulté le : 27/03/2024.

[15] DJEBARI, N. Sécurité des systèmes d'information. Support de cours, 2023. Disponible à : https://elearning.univ-bejaia.dz/course/view.php?id=15997.

[16] DOSHI, J., PARMAR, K., SANGHAVI, R., AND SHEKOKAR, N. A comprehensive dual-layer architecture for phishing and spam email detection. *Computers amp ; Security 133* (2023), 103378.

[17] DUC, T. L., LEIVA, R. G., CASARI, P., AND ÖSTBERG, P.-O. Machine learning methods for reliable resource provisioning in edge-cloud computing : A survey. *ACM Computing Surveys 52*, 5 (2019), 1–39.

[18] FLORIAN ZYPRIAN. Guide du développeur random forest : 5 façons de l'implémenter en python. https://konfuzio.com/fr/random-forest, Consulté le : 29/03/2024.

[19] GHAFIR, I., HAMMOUDEH, M., PRENOSIL, V., HAN, L., HEGARTY, R., RABIE, K., AND APARICIO-NAVARRO, F. J. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems 89* (2018), 349–359.

[20] HAMZA, L. Sécurité des réseaux. Support de cours, 2022. Disponible à : https://elearning.univ-bejaia.dz/course/view.php?id=15032.

[21] HONG, R.-F., HORNG, S.-C., AND LIN, S.-S. Machine learning in cyber security analytics using nsl-kdd dataset. In *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)* (2021), IEEE.

[22] IMADEDDINE, M. D. A. M. O. Analyse et prédiction de la consommation d'électricité pour l'internet des comportements, 2022.

[23] JONES, R., AND SMITH, A. *Data Scientist et langage R*. Data Science Press, Paris, France, 2023.

[24] KHEHRA, B. S., AND PHARWAHA, A. P. S. Classification of clustered microcalcifications using mlffbp-ann and svm. *Egyptian Informatics Journal 17*, 1 (Mar. 2016), 11–20.

[25] KHRAISAT, A., GONDAL, I., VAMPLEW, P., AND KAMRUZZAMAN, J. Survey of intrusion detection systems : techniques, datasets and challenges. *Cybersecurity 2*, 1 (2019).

[26] KOBIA. Matrice de confusion, la comprendre et l'utiliser. https://kobia.fr/classification-metrics-matrice-de-confusion/, Consulté le 16/04/2024.

[27] KUNHARE, N., AND TIWARI, R. Study of the attributes using four class labels on kdd99 and nsl-kdd datasets with machine learning techniques. In *2018 8th International Conference on Communication Systems and Network Technologies (CSNT)* (2018), IEEE.

[28] LANGNER, R. Stuxnet : Dissecting a cyberwarfare weapon. *IEEE Security amp ; Privacy Magazine 9*, 3 (2011), 49–51.

[29] LE, V. H., DEN HARTOG, J., AND ZANNONE, N. Security and privacy for innovative automotive applications : A survey. *Computer Communications 132* (Nov. 2018), 17–41.

[30] LU, J., CHEN, K., ZHUO, Z., AND ZHANG, X. A temporal correlation and traffic analysis approach for apt attacks detection. *Cluster Computing 22*, S3 (Oct. 2017), 7347–7358.

[31] MAIMON, O. Z., AND ROKACH, L. *Data Mining with Decision Trees : Theory and Applications*, vol. 81. World Scientific, 2014.

[32] MATEYAUNGA, I. Predictive maintenance using machine learning. Master's thesis, Université de Tlemcen, Faculté de technologie, 2020.

[33] MAZIERES, A. *Cartographie de l'apprentissage artificiel et de ses algorithmes*. PhD thesis, Université Paris Diderot, 2016.

[34] MILENKOSKI, A., VIEIRA, M., KOUNEV, S., AVRITZER, A., AND PAYNE, B. D. Evaluating computer intrusion detection systems : A survey of common practices. *ACM Computing Surveys 48*, 1 (2015), 1–41.

[35] MYNENI, S., CHOWDHARY, A., SABUR, A., SENGUPTA, S., AGRAWAL, G., HUANG, D., AND KANG, M. *DAPT 2020 - Constructing a Benchmark Dataset for Advanced Persistent Threats*. Springer International Publishing, 2020, pp. 138–163.

[36] NVIDIA. Xgboost. https://www.nvidia.com/en-us/glossary/xgboost/, Consulté le : 27/03/2024.

[37] OF STANDARDS, N. I., AND (NIST), T. Managing information security risk : Organization, mission, and information system view. Tech. rep., National Institute of Standards and Technology (NIST), 2011.

[38] PATEL, N. D., MEHTRE, B. M., AND WANKAR, R. Detection of intrusions using support vector machines and deep neural networks. In *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)* (2022), IEEE.

[39] PYTHON SOFTWARE FOUNDATION. Python.org. https://www.python.org/. Consulté le : 21/03/2024.

[40] SASI, T., LASHKARI, A. H., LU, R., XIONG, P., AND IQBAL, S. A comprehensive survey on iot attacks : Taxonomy, detection mechanisms and challenges. *Journal of Information and Intelligence* (2023).

[41] STURMAN, D., BELL, E. A., AUTON, J. C., BREAKEY, G. R., AND WIGGINS, M. W. The roles of phishing knowledge, cue utilization, and decision styles in phishing email detection. *Applied Ergonomics 119* (2024), 104309.

[42] TU, Y., LIU, S., AND SUN, Q. Dns tunnelling detection by fusing encoding feature and behavioral feature. *Computers amp ; Security 132* (2023), 103357.

[43] WANG, Y., PAN, Z., ZHENG, J., QIAN, L., AND LI, M. A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science 364*, 8 (2019).

[44] WRESSNEGGER, C., SCHWENK, G., ARP, D., AND RIECK, K. A close look on n -grams in intrusion detection : anomaly detection vs. classification. In *Proceedings of the 2013 ACM workshop on Artificial intelligence and security* (2013), CCS'13, ACM.

[45] WU, H., SHAO, Z., YANG, F., CHENG, G., HU, X., REN, J., AND WANG, W. Pd-cps : A practical scheme for detecting covert port scans in high-speed networks. *Computer Networks 231* (2023), 109825.

[46] YI, S. Y., SINGH, M. M., SODHY, G. C., AND JABAR, T. Fingerprinting generation for advanced persistent threats (apt) detection using machine learning techniques. In *2023 13th International Conference on Information Technology in Asia (CITA)* (2023), IEEE.

[47] YING, X. An overview of overfitting and its solutions. *Journal of Physics : Conference Series 1168* (2019), 022022.

[48] ZAMBRANO, P., TORRES, J., TELLO-OQUENDO, L., YÁNEZ, , AND VELÁSQUEZ, L. On the modeling of cyber-attacks associated with social engineering : A parental control prototype. *Journal of Information Security and Applications 75* (2023), 103501.

[49] ZHAN, M., LI, Y., YU, G., LI, B., AND WANG, W. Detecting dns over https based data exfiltration. *Computer Networks 209* (2022), 108919.

RÉSUMÉ

Ce mémoire explore l'application des techniques de Machine Learning pour détecter les menaces persistantes avancées dans le domaine de la sécurité informatique. Les APT, en tant que cyberattaques sophistiquées, présentent des défis majeurs pour les méthodes de détection classiques. L'objectif principal est de concevoir un modèle d'apprentissage automatique détectant les comportements suspects en se basant sur les données du trafic réseau. Notre méthodologie implique la collecte de données, un prétraitement des données, et le choix des caractéristiques appropriées pour l'apprentissage, qui est crucial dans notre approche. Nous avons sélectionné 12 caractéristiques sur 41, effectué un suréchantillonnage des classes pour éviter le sur-apprentissage, puis entraîné un algorithme Random Forest et évalué ses performances. Les résultats montrent que notre modèle basé sur les forêts aléatoires surpasse les autres techniques existantes dans la littérature avec une accuracy de 99,87%, une précision de 97,93%, et un rappel de 99,79%, démontrant une capacité plus que satisfaisante à capturer la complexité des flux réseau malveillants. Ces chiffres mettent en évidence l'efficacité et la robustesse de notre approche face aux défis posés par les APT.

Mots clés : Machine Learning , Menaces Persistantes Avancées , comportements suspects , forêts aléatoires .

ABSTRACT

This thesis explores the application of Machine Learning techniques to detect Advanced Persistent Threats (APTs) in the field of computer security. APTs, as sophisticated cyberattacks, present major challenges for traditional detection methods. The main objective is to create a machine learning model that detects suspicious behaviors based on network traffic data. Our methodology involves data collection, data preprocessing, and the selection of appropriate features for learning, which is crucial in our approach. We selected 12 features out of 41, performed class oversampling to avoid overfitting, then trained a Random Forest algorithm and evaluated its performance. The results show that our Random Forest-based model outperforms other existing techniques in the literature with an accuracy of 99.87%, a precision of 97.93%, and a recall of 99.79%, demonstrating more than satisfactory ability to capture the complexity of malicious network flows. These figures highlight the effectiveness and robustness of our approach in facing the challenges posed by APTs.

Keywords : Machine Learning, Advanced Persistent Threats, suspicious behaviors, random forests.