République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid
Faculté des Sciences et de la Technologie
Département d'Informatique

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

MEMOIRE

Présenté par

BOUHANNA SAMEH

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Un système de détection d'intrusion pour les objets connecte

Soutenu le : 25 / 06 / 2022

Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Président	Mr. Touahri. Dham aldin	MCB	Chadli Bendjedid El-Tarf
Rapporteur	**Mr. Banmachiche Abdelmadjid**	**MCB**	**Chadli Bendjedid El-Tarf**
Examinateur	Mme.Matallah Majda	MCB	Chadli Bendjedid El-Tarf
CO-encadreur	Mlle Labiod Yasmine	MCB	Chadli Bendjedid El-Tarf

Année Universitaire : 2022/2023

Remerciements

Je tiens à exprimer mes sincères remerciements à Dieu pour m'avoir accordé la force et la persévérance nécessaires pour mener à bien ce travail de mémoire.

Je souhaite également exprimer ma profonde gratitude envers ma mère, dont le soutien constant, l'amour inconditionnel et les encouragements ont été essentiels tout au long de ce parcours. Sa présence et son soutien indéfectibles ont été une source d'inspiration et de motivation pour moi.

Je tiens également à remercier mon encadrant, ***M. Benmachiche****, pour sa guidance, son expertise et ses conseils avisés tout au long de la réalisation de ce travail.*

Un grand merci également à mon encadrante, ***Mlle Labiod,*** *pour son aide précieuse, ses suggestions pertinentes et son soutien constant. Sa contribution a été d'une grande valeur et a enrichi mon travail.*

Enfin, je tiens à exprimer ma reconnaissance envers toutes les personnes qui m'ont soutenu, que ce soit par leurs encouragements, leurs conseils ou leur aide pratique. Leur présence et leur soutien ont été d'une importance capitale dans la réalisation de ce travail.

Merci à tous ceux qui ont contribué à ma réussite et qui ont été présents à mes côtés tout au long de ce parcours

Dédicace

Je dédie ce modeste travail à :

À mon père décédé, dont la présence me manque tant, notamment lors de ma remise des diplômes. Je sais que tu aurais été fier de mes réalisations. Cette thèse est dédiée en ton honneur, en témoignage de l'amour et de l'affection que tu m'as toujours témoignés.

À ma chère mère, ma plus grande source de soutien et de motivation. Ta présence constante dans ma vie a été un pilier de force et de guidage. Je te suis infiniment reconnaissante pour tous les sacrifices que tu as consentis et pour ta tendresse sans faille.

À mes sœurs : Hind, Aya, Zina et Hadil, mes compagnes de vie et de parcours. Votre présence, vos encouragements et vos conseils ont été d'une importance capitale tout au long de ce parcours.

À mes amies Chaima. L, Chaima .F, Asma, Rym, Ferdaous, qui ont partagé avec moi de nombreux moments de joie, de rire et de soutien. Votre présence et votre amitié précieuse ont allégé les difficultés de ce parcours.

Que ces dédicaces expriment ma reconnaissance envers ceux qui ont joué un rôle essentiel dans ma vie et qui ont contribué à ma réussite. Leur soutien et leur amour ont une valeur inestimable.

Table des matières

Liste des figures

Liste des tableaux

Liste des acronymes

IoT	Internet of Things
IDS	Intrusion Detection System,
ML	Machine Learning
SVM	Support Vector Machine
k-NN	k-Nearest Neighbors
RF	Random Forest
IPS	Intrusion Prevention System
NIDS	Network Intrusion Detection System
NIPS	Network Intrusion Prevention System
HIDS	Host-based Intrusion Detection System
DDos	Distributed denial-of-service
DoS	Denial of Service

Introduction Générale

L'Internet des objets (IoT) a connu une expansion massive ces dernières années, permettant la connectivité de nombreux objets du quotidien au réseau Internet. Cependant, cette interconnexion généralisée a également engendré des problèmes de sécurité importants, notamment en ce qui concerne la détection d'intrusion. Dans ce mémoire, nous abordons la problématique de la détection d'intrusion dans l'Internet des objets et proposons une approche novatrice pour y faire face.

1. Contexte du projet et problématique

Dans un monde de plus en plus interconnecté, où les objets du quotidien sont de plus en plus connectés à Internet, la sécurité de l'Internet des objets est devenue une préoccupation majeure. Les attaquants peuvent exploiter les vulnérabilités des objets connectés pour accéder aux réseaux, voler des données sensibles ou perturber les services. Les approches actuelles de détection d'intrusion pour l'IoT ne sont pas totalement satisfaisantes, car elles ne parviennent pas à fournir une protection adéquate contre les attaques sophistiquées et émergentes.

2. Motivations

La sécurité des systèmes IoT est essentielle pour garantir la confidentialité, l'intégrité et la disponibilité des données et des services. Cependant, les approches traditionnelles de détection d'intrusion sont souvent insuffisantes pour faire face aux spécificités de l'IoT. Nous sommes motivés par la nécessité de développer un système de détection d'intrusion spécifiquement conçu pour l'Internet des objets, capable de détecter les attaques émergentes et d'assurer la sécurité des objets connectés.

3. Objectifs

L'objectif principal de ce mémoire est de proposer une approche innovante pour la détection d'intrusion dans l'Internet des objets. Nous visons à développer un système robuste et fiable capable de détecter les activités malveillantes au sein des réseaux d'objets connectés. Pour atteindre cet objectif, nous allons explorer l'utilisation de techniques avancées telles que l'apprentissage automatique et l'analyse des comportements pour détecter les intrusions de manière proactive et précise.

4. Contenu du mémoire

Ce mémoire est structuré en plusieurs chapitres, abordant différents aspects de la détection d'intrusion dans l'Internet des objets.

- **Chapitre 1 :** La sécurité de l'Internet des objets

Ce chapitre introduit les concepts de base de la sécurité de l'IoT, en mettant en évidence les défis et enjeux associés à la protection des objets connectés contre les attaques. De plus, nous mentionnerons le système de détection d'intrusion et ses classifications.

- **Chapitre 2 :** Apprentissage automatique pour la détection d'intrusion dans l'IoT

Ce chapitre examine l'utilisation de techniques d'apprentissage automatique, les algorithmes de classification, pour la détection d'intrusion dans l'IoT.

- **Chapitre 3:** Approche proposée

Dans ce chapitre, nous présenterons notre approche novatrice pour la détection d'intrusion dans l'Internet des objets. Nous détaillerons les approches et les techniques que nous proposons d'utiliser.

1. La sécurité pour internet des objets

1.1. Introduction

L'Internet des objets IoT connecte tous les objets à Internet pour échanger des informations. Il permet un accès universel et une interaction sans intervention humaine grâce à des capteurs intelligents. L'IoT utilise des schémas d'adressage uniques pour créer de nouvelles applications et services dans divers domaines tels que la santé, l'environnement et la domotique [1]. Cependant, la sécurité est un défi majeur car les dispositifs IoT sont accessibles via un réseau non fiable, les exposant à des attaques malveillantes [1].

Dans ce chapitre, nous traitons de la sécurité dans l'Internet des objets (IoT). Nous commençons par présenter la définition de l'Internet des objets, puis nous abordons les défis et enjeux associés à ce domaine. Ensuite, nous explorons les concepts fondamentaux liés à la sécurité dans l'IoT, et examinons les principes des attaques visant les objets connectés. Enfin, nous étudions les mécanismes de sécurité utilisés dans les environnements IoT.

1.2. Internet des objets

1.2.1.Définition

Le terme "Internet des objets" se réfère à un réseau étendu composé d'objets physiques qui sont connectés à Internet et identifiés de la même manière que nos appareils traditionnels tels que les ordinateurs, les tablettes et les Smartphones. Dans la littérature, plusieurs définitions de l'Internet des objets ont été proposées, ce qui rend l'existence d'une définition standard difficile. Récemment perçu comme une révolution technologique, l'Internet des objets peut être simplement défini comme l'extension de l'Internet actuel à tous les objets capables de communiquer directement ou indirectement avec des équipements électroniques connectés à Internet, selon une définition donnée par [2]. Selon une autre définition proposée par [3], l'Internet des objets est une infrastructure de réseau mondiale dynamique qui se configure automatiquement en utilisant des normes et des protocoles de communication interopérables. Dans ce réseau, les objets physiques et virtuels possèdent des identités, des caractéristiques physiques, des personnalités virtuelles et des interfaces intelligentes, et ils sont intégrés de manière transparente au réseau d'information. C'est un monde où les objets interconnectés interagissent avec les humains et entre eux (M2M) [2].

1.2.2.Architecture internet des objets

Voici les principales couches d'architecture de l'IoT

A. La couche perception

La couche physique de l'IoT collecte les données du monde réel à l'aide de capteurs et autres dispositifs, assurant la communication entre les appareils. Des appareils tels qu'Arduino, ZigBee, codes-barres, RFID et divers capteurs sont utilisés. Chaque appareil doit avoir une étiquette unique pour une connexion fiable, tandis que la couche réseau transporte les informations collectées vers le centre de traitement. [4]

B. La Couche réseau

La couche réseau assure le transport des informations collectées des objets physiques via des capteurs vers la couche d'application. Elle permet la connectivité des objets intelligents, des périphériques réseau et des réseaux. Cependant, elle est vulnérable aux attaques et présente des défis en termes de sécurité, d'intégrité et d'authentification des informations transportées sur le réseau. [5]

C. Couche de traitement de données

La couche de traitement, ou couche middleware, collecte et traite les informations de la couche de transport. Elle élimine les données redondantes, extrait les informations utiles et résout le problème du Big Data dans l'IoT. Cependant, cette couche peut être vulnérable à des attaques qui affectent les performances du système. [6]

D. Couche Applicative

L'Internet des objets (IoT) regroupe des services intelligents utilisant divers mécanismes de gestion des données provenant de différents objets. Les applications de l'IoT sont vastes, incluant les maisons intelligentes, le transport intelligent, et plus encore.

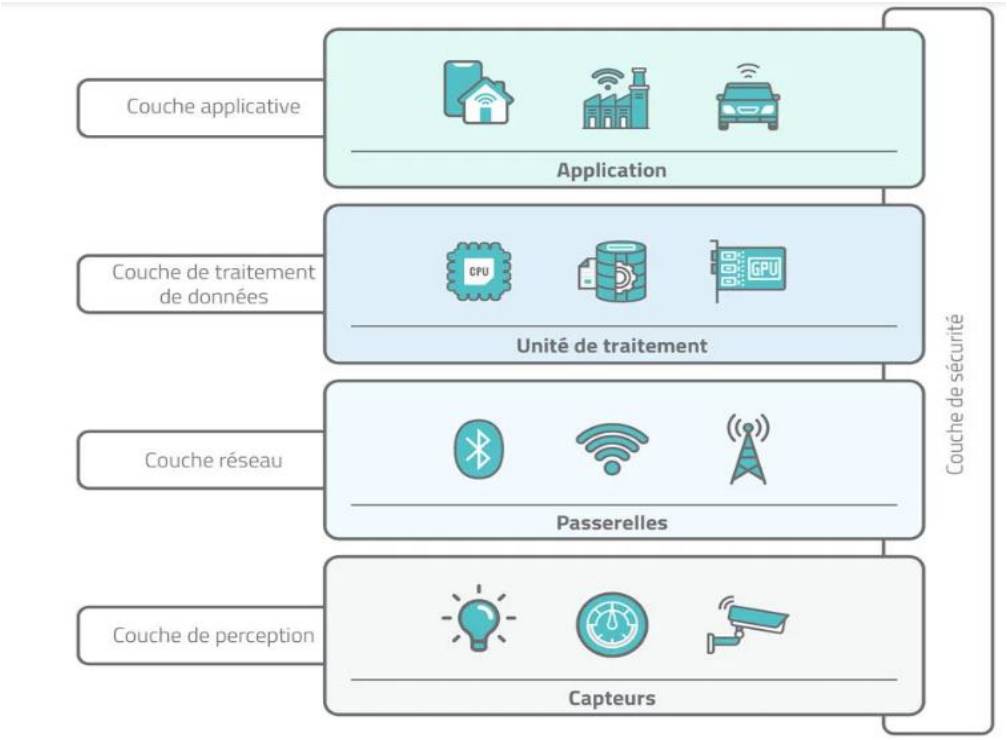

Figure 1.1. Architecture IoT à quatre couches [7]

1.2.3.Défis des internet des objets

L'hétérogénéité des dispositifs

L'IoT rassemble des dispositifs variés avec des normes de communication différentes et des contraintes spécifiques. La gestion efficace des données, du calcul, du stockage et de la sécurité est essentielle pour assurer l'interopérabilité et le bon fonctionnement de l'IoT. [8]

Sécurité

La sécurité dans l'Internet des Objets (IoT) est un défi majeur en raison des limitations des dispositifs contraints, tels que la puissance de traitement et la mémoire, qui les rendent vulnérables aux attaques. De plus, ces contraintes limitent les options de gestion, rendant nécessaire un soutien externe pour renforcer leur sécurité. Ces dispositifs deviennent donc des cibles faciles pour les attaques telles que les attaques par déni de service distribué (DDoS). [9]

Interopérabilité

L'interopérabilité est un défi majeur dans l'IoT, mais elle offre un avantage économique en augmentant la valeur du marché, car les utilisateurs recherchent des produits et services compatibles. C'est essentiel pour un fonctionnement harmonieux de l'IoT. [10]

1.3. La sécurité des internet des objets

1.3.1. Principaux concepts de base en sécurité dans l'internet des objets

- **La confidentialité des données informatiques**

L'ensemble des mécanismes qui garantissent la confidentialité des communications de données entre un émetteur et un destinataire repose sur la cryptographie ou le chiffrement des données. En effet, la cryptographie est la seule solution fiable pour assurer la confidentialité des données [11].

- **L'intégrité des données**

Le message ne doit pas être altère en transit ; il doit être reçu au nœud récepteur tel qu'il a été envoyé au nœud émetteur. L'intégrité garantit que le message n'a pas été modifié par des personnes non autorisées pendant la transmission [12]

- **La disponibilité des données informatiques**

La disponibilité d'un service est assurée en protégeant contre les arrêts intentionnels et non intentionnels, en utilisant des mécanismes de protection tels que la prévention des attaques de déni de service, et en dupliquant le service sur plusieurs serveurs pour maintenir la continuité en cas de défaillance. [13]

- **La non-répudiation**

La non-répudiation garantit que l'expéditeur et le destinataire ne peuvent pas nier Avoir envoyé et reçu le message respectivement [14].

- **L'authentification**

Le contrôle d'accès vise à restreindre l'accès aux ressources aux seules personnes autorisées en garantissant leur identité et en permettant uniquement aux utilisateurs légitimes d'accéder aux ressources. [15].

1.3.2. Principales attaques dans les objets connectés

Principales formes d'attaques courantes liées à l'Internet des objets IoT

Les cybercriminels ont la possibilité de cibler le matériel ou le logiciel de n'importe quel composant au sein d'un système d'IoT. Parmi les formes les plus courantes d'attaques par l'IoT, on trouve :

Les Botnets : Les appareils de l'IoT peuvent être compromis par des cybercriminels et utilisés en tant que "bots zombies" en grand nombre. Les attaquants installent un logiciel malveillant sur ces appareils afin d'exploiter leur puissance de traitement collective, permettant ainsi de mener des

attaques DDoS de grande envergure, d'espionner les utilisateurs, de voler des informations, et bien d'autres actions.

Les attaques par déni de service distribué (DDoS) : Dans ce type d'attaque, un grand nombre de systèmes sont utilisés de manière malveillante pour attaquer une cible unique, entraînant une saturation de sa capacité et rendant un site web, une application ou un service indisponible. Les réseaux d'appareils de l'IoT offrent de multiples possibilités aux pirates informatiques pour mener ces attaques.

Les ransomwares : Les ransomwares sont des types de logiciels malveillants qui peuvent bloquer des appareils ou rendre des fichiers inaccessibles jusqu'à ce que la victime paye une rançon.

Les attaques de l'homme du milieu : Les attaquants exploitent les réseaux et les protocoles non sécurisés pour s'insérer "au milieu" ou entre les canaux de communication, leur permettant ainsi d'intercepter ou de transmettre subrepticement des messages entre deux parties qui pensent communiquer en toute sécurité.

1.3.3.Les mécanismes de sécurité pour l'internet des objets

La sécurité est un aspect crucial de l'Internet des objets (IoT) en raison de la nature des données sensibles traitées et de la diversité des objets connectés. Voici quelques mécanismes de sécurité couramment utilisés pour protéger l'IoT :

Authentification et contrôle d'accès

- Les dispositifs de l'IoT doivent être authentifiés avant de pouvoir accéder au réseau ou aux services. Cela peut inclure des protocoles d'authentification tels que les certificats numériques, les clés d'accès, les jetons d'authentification, etc.
- Des mécanismes de contrôle d'accès sont mis en place pour limiter les privilèges et les autorisations des dispositifs, afin de s'assurer qu'ils n'accèdent qu'aux ressources appropriées

Chiffrement des données

- Le chiffrement est utilisé pour sécuriser les données en transit et au repos. Les protocoles de chiffrement tels que SSL/TLS sont utilisés pour sécuriser les communications entre les objets connectés, les passerelles et les systèmes de gestion.
- Le chiffrement des données stockées dans les dispositifs et les serveurs garantit que seules les personnes autorisées peuvent y accéder.

Intégrité des données

- Des mécanismes de vérification de l'intégrité des données sont utilisés pour détecter les altérations ou les manipulations des données pendant la transmission ou le stockage. Des techniques telles que les codes de hachage, les signatures numériques et les fonctions de somme de contrôle peuvent être utilisées pour garantir l'intégrité des données.

Mises à jour et correctifs de sécurité

- Les fabricants et les développeurs d'objets connectés doivent fournir des mises à jour régulières du firmware et des correctifs de sécurité pour corriger les vulnérabilités connues.

- Les utilisateurs doivent s'assurer que leurs objets connectés sont régulièrement mis à jour avec les dernières versions du logiciel pour bénéficier des correctifs de sécurité.

Surveillance et détection des menaces

- Des mécanismes de surveillance en temps réel sont utilisés pour détecter les activités suspectes des objets connectés. Les systèmes de détection des intrusions et d'analyse des journaux (SIEM) sont employés pour collecter et analyser les données de sécurité, permettant d'identifier les potentielles violations de sécurité.

Sensibilisation à la sécurité

- Les utilisateurs d'objets connectés doivent être sensibilisés aux bonnes pratiques de sécurité, telles que l'utilisation de mots de passe forts, la protection des informations d'identification, l'identification des tentatives de phishing,

Sécurité du réseau

Les systèmes de détection/prévention d'intrusion (IDS/IPS) sont des composants de sécurité réseau qui analysent les échanges entre les systèmes pour détecter et prévenir les intrusions. Ils capturent les échanges, identifient les attaques et empêchent les tentatives d'intrusion ou émettent des alertes. Les IDS/IPS se déclinent en deux types principaux : les IDS/IPS réseaux (NIDS/NIPS) qui surveillent les échanges sur le réseau à l'aide de sondes spécifiques, et les IDS/IPS hôtes (HIDS/HIPS) qui surveillent la sécurité au niveau des hôtes grâce à des sondes implantées. [16]

1.4. Système de détection d'intrusion dans les objets connecte

1.4.1.Définition

La détection d'intrusion consiste à surveiller et analyser les événements sur un système informatique pour repérer les activités malveillantes ou non autorisées qui enfreignent la politique de sécurité. Les systèmes de détection d'intrusion automatisent ce processus en utilisant des composants matériels et logiciels pour détecter les menaces imminentes. Ils agissent comme une deuxième ligne de défense similaire à un pare-feu, en identifiant les activités malveillantes qui échappent aux pare-feu traditionnels. Ces systèmes sont essentiels pour augmenter le niveau de protection et prévenir les dommages en détectant les intrusions avant qu'elles ne causent des problèmes significatifs. [17]

1.4.2.Classification des systèmes de détection d'intrusion

Nous allons présenter ici la technologie de détection d'intrusion d'une manière taxonomique. Il y a plusieurs types d'IDS disponibles aujourd'hui, caractérisé par différente approche de la surveillance et de l'analyse.

- L'emplacement d'IDS
  - Modes de détection
- Les types de réponse

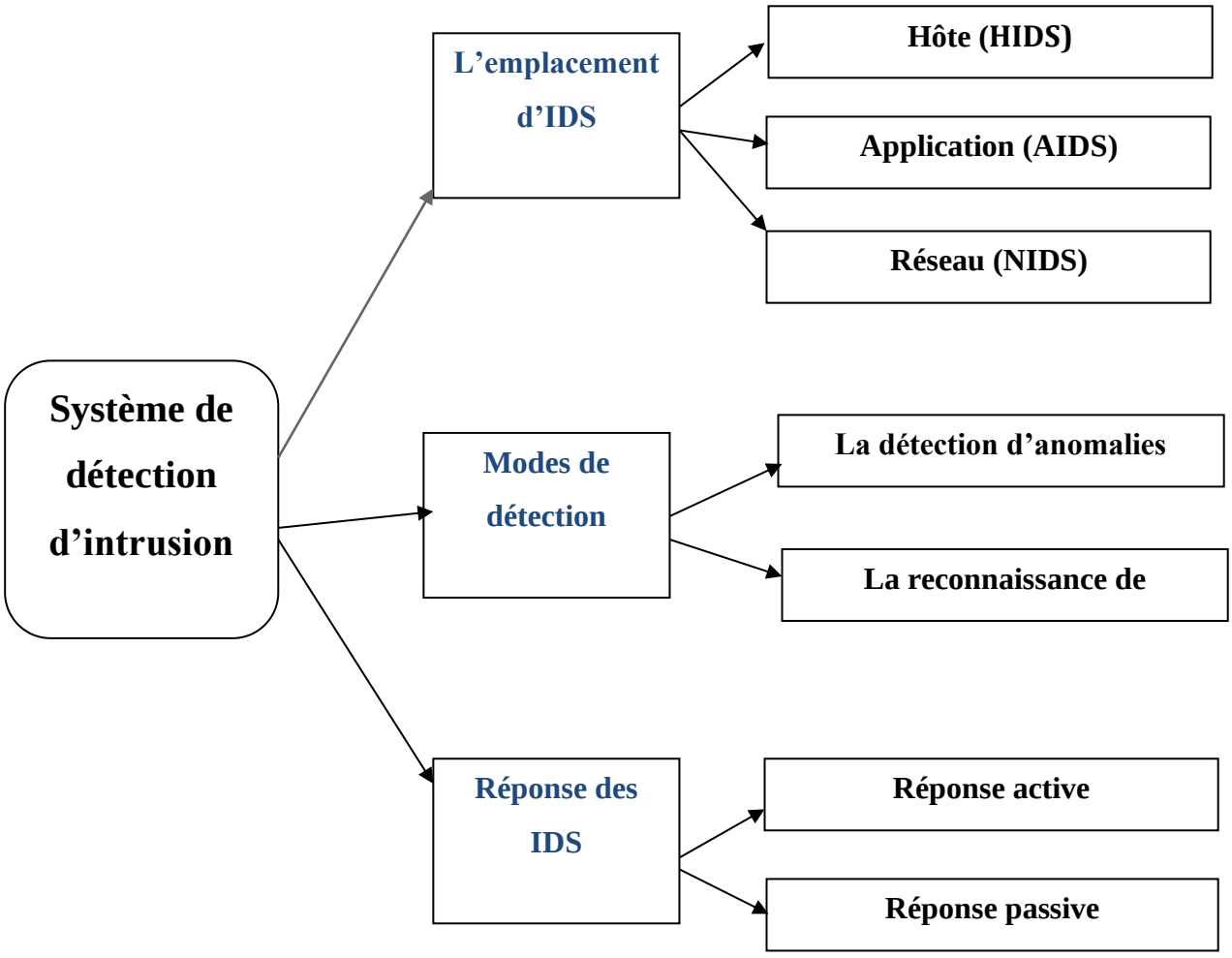

Figure 2.1. Classification ID[18]

1.4.2.1. L'emplacement d'IDS

A. La détection d'intrusion basée sur L'hôte

Les systèmes de détection d'intrusion basés sur l'hôte (HIDS) analysent les informations spécifiques à un hôte pour détecter les attaques, offrant fiabilité et précision. Ils accèdent aux fichiers et processus ciblés, utilisant les traces d'audit pour obtenir des informations sur l'activité de la machine. Les HIDS réagissent rapidement, détectent les attaques "Cheval de Troie" et repèrent les attaques cryptées, mais peuvent être vulnérables aux attaques DoS, générant des fichiers de logs volumineux et affectant les performances. Ils sont déployés sur des machines sensibles pour renforcer la sécurité. [18]

B. La détection d'intrusion basée sur application

Les IDS basés sur les applications (AIDS) contrôlent l'interaction utilisateur-programme, détectent les comportements suspects et empêchent l'exécution de certaines commandes. Cependant, ils offrent une sécurité moindre contre les attaques "Cheval de Troie" et les fichiers de log sont vulnérables. Les AIDS sont utiles pour surveiller les applications sensibles, mais nécessitent généralement un HIDS pour renforcer la sécurité et surveiller l'utilisation du CPU afin de préserver les performances.

C. La détection d'intrusion basée sur Réseau (NIDS)

Les systèmes de détection d'intrusion basés sur le réseau (NIDS) analysent les paquets réseau pour détecter les attaques en utilisant des signatures connues ou en repérant les anomalies du protocole. Ils déclenchent des alertes et ferment les connexions lorsqu'ils détectent des activités suspectes. Des capteurs sont déployés aux points stratégiques du réseau, générant des alertes qui sont analysées dans une console sécurisée.

1.2.2.2. Mode de détection

Nous notons deux modes de détection qui sont :

- La détection d'anomalies.
- La reconnaissance de signature.

Il faut noter que la reconnaissance de signature est le mode de fonctionnement le plus implémenté par les IDS du marché. Cependant, les nouveaux produits tendent à combiner les deux méthodes pour affiner la détection d'intrusion.

- **La détection d’anomalies**

L'approche de détection d'anomalies vise à prédire le comportement en utilisant une base de données de comportements normaux, détectant les comportements significativement différents. Elle peut détecter des attaques connues et inconnues, mais présente un taux élevé de fausses alarmes. Elle nécessite un historique des événements et peut fournir des informations pour définir des signatures dans les systèmes basés sur les signatures.

- **La reconnaissance de signature**

L'approche de détection basée sur les signatures recherche des empreintes d'attaques connues, nécessitant des mises à jour régulières de la base de signatures. Elle est efficace pour détecter les attaques connues, mais ne détecte pas les attaques inconnues et peut être contournée par des attaquants. Les avantages incluent la prise en compte des comportements réels des attaquants et un faible taux de fausses alarmes, mais elle nécessite une manipulation prudente des signatures. [19]

1.2.2.3. Type de réponse

- **Réponse active**

La réponse active dans les systèmes de détection d'intrusion implique des mesures immédiates pour stopper une attaque. Les techniques courantes sont la reconfiguration du pare-feu pour bloquer le trafic malveillant et l'interruption d'une connexion TCP pour déconnecter l'attaquant. Cependant, il faut être prudent pour éviter de déconnecter des utilisateurs légitimes et il est nécessaire d'analyser les fichiers d'alertes pour confirmer les attaques. Les capacités de réponse active dépendent du pare-feu utilisé.

- **Réponse passive**

La réponse passive d'un IDS se traduit par l'enregistrement des intrusions détectées dans des journaux pour une analyse ultérieure. Certains IDS peuvent enregistrer des connexions malveillantes complètes. Cela permet de remédier aux vulnérabilités et de prévenir les attaques similaires à l'avenir. Cependant, cette approche ne prévient pas directement les attaques en cours, mais vise à améliorer les mesures de sécurité pour l'avenir en recueillant des informations précieuses.

1.3. Conclusion

Dans ce chapitre, nous avons débuté en abordant la sécurité dans l'Internet des objets (IdO) et en discutant des principales attaques qui le visent. Nous avons ensuite présenté les mécanismes de sécurité, ainsi qu'une définition des systèmes de détection d'intrusions (IDS). Ensuite, nous avons présenté une classification des IDS, en discutant de leur emplacement, notamment les IDS basés sur l'hôte (HIDS), les IDS basés sur l'application (AB-IDS) et les IDS basés sur le réseau (N-IDS).

Nous avons ensuite examiné les deux approches principales utilisées par les IDS : la détection d'anomalies et la reconnaissance de signatures, en détaillant leurs méthodes respectives, avantages et inconvénients. Enfin, nous avons abordé les deux types de réponses offertes par les systèmes IDS suite à la détection d'une intrusion : la réponse active et la réponse passive.

En conclusion, les IDS jouent un rôle essentiel en tant que deuxième ligne de défense pour garantir la sécurité opérationnelle, surtout dans un environnement où les systèmes informatiques sont de plus en plus interconnectés et ouverts. Le chapitre suivant se concentrera sur l'apprentissage automatique.

2. Apprentissage automatique

2.1. Introduction

L'ampleur des données IoT nécessite de nouveaux mécanismes pour les exploiter efficacement. L'apprentissage automatique (Machine Learning) est considéré comme essentiel pour fournir une intelligence intégrée aux réseaux IoT. [20]

Dans ce chapitre, nous commençons par expliquer les principes fondamentaux de l'apprentissage automatique, en détaillant ses trois principales catégories : supervisée, non supervisée et semi-supervisée. Ensuite, nous abordons la détection d'anomalies, en décrivant les différents types d'anomalies ainsi que le concept essentiel de la détection. Enfin, nous présentons les différents algorithmes d'apprentissage automatique qui ont été mis en œuvre dans le cadre de notre recherche.

2.2. Définition

L'apprentissage automatique (Machine Learning) est un domaine de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances. Cette approche résout des problèmes complexes et trouve des applications dans divers domaines tels que la médecine, la finance et les transports. Les algorithmes d'apprentissage automatique sont classés en trois catégories : supervisés, non supervisés et par renforcement, en fonction de l'expérience utilisée pour apprendre. [21]

2.2.1.Apprentissage supervisé

L'apprentissage supervisé est une méthode qui estime une fonction reliant les données d'entrée et de sortie à l'aide d'un jeu de données d'entraînement. Le modèle apprend à prédire les sorties en minimisant une fonction de coût et peut généraliser ses prédictions à de nouvelles données. Une variante utilise une fonction de distribution de probabilité pour prédire la classe la plus probable, utilisée dans la classification et la régression. [22]

2.2.2.Apprentissage non supervisé

En apprentissage non supervisé, les données sont fournies sans valeurs de sortie de référence. Le but est de découvrir des motifs utiles dans la structure des données, permettant le regroupement en sous-groupes similaires (clustering) ou la réduction de dimension. Le clustering est utilisé pour la segmentation de la clientèle et les systèmes de recommandation en ligne, tandis que la réduction de dimension simplifie les algorithmes en préservant les principales informations des données d'origine. [22]

2.2.3.Apprentissage semi-supervisé

L'apprentissage semi-supervisé se situe entre l'apprentissage supervisé et non supervisé. Il utilise un jeu de données partiellement annotées pour surmonter le manque d'étiquetage complet. Il est utile pour la classification et le regroupement contraint. [22]

2.3. Détection d'anomalies

La détection d'anomalies identifie les modèles de données qui ne sont pas conformes à un comportement normal. Les anomalies peuvent être des valeurs aberrantes ou des exceptions dans un ensemble de données. Cette technique est essentielle pour détecter des informations critiques, telles qu'une intrusion dans un réseau IoT à partir d'un trafic anormal. La détection d'anomalies est largement utilisée dans de nombreux domaines d'application. [23]

2.3.1.Type d'anomalies

Trois types d'anomalies peuvent être détectés, parmi lesquels on retrouve :

a. **Anomalies ponctuelles :**

Une anomalie ponctuelle se produit lorsque des données individuelles sont très éloignées du reste des données, comme dans le cas d'une carte de crédit volée. Ces anomalies sont représentées par des points éloignés du nuage de données principal dans un espace à deux dimensions. [24]

b. **Anomalies contextuelles :**

Les anomalies contextuelles sont des valeurs anormales dans un contexte donné mais normales dans un autre. Elles sont représentées dans un schéma à deux dimensions avec l'axe principal pour

la valeur et l'axe secondaire pour le contexte. Par exemple, une température identique peut être considérée comme une anomalie en été mais normale en hiver. [24]

c. Anomalies collectives

Les anomalies collectives se réfèrent à des situations où la répétition d'une valeur, bien qu'elle puisse être normale individuellement, devient anormale collectivement. Par exemple, une brève absence d'activité entre deux battements de cœur est normale, mais si cela se produit de manière prolongée, cela devient une anomalie dans un électrocardiogramme. [24]

2.3.2.L'aide fondamentale de la détection d'anomalie

La détection d'anomalies repose sur l'utilisation d'un modèle représentant les données normales et le calcul d'un score pour chaque point de données afin d'évaluer l'écart par rapport à cette normale. Si le score dépasse un seuil prédéfini, le point est considéré comme une anomalie. La conception précise du modèle et du score est essentielle pour une détection efficace des anomalies. [25].

2.4. Algorithme d'apprentissage automatique utiliser

- **Machines à Vecteurs de Support (SVM)**

Le SVM (Support Vector Machine) est un concept simple à comprendre, même sans utiliser de formules mathématiques. Son objectif est de définir un hyperplan qui maximise la marge entre les classes pour améliorer la généralisation. Lorsque les classes ne peuvent pas être séparées linéairement, le SVM utilise des transformations de données avec des noyaux pour les rendre séparables [26]. Les méthodes à noyaux sont optimisées pour des calculs rapides et une faible consommation de mémoire [27]. Cependant, le temps d'entraînement des SVM à noyaux est plus élevé pour les corpus de grande taille [27]. En plus de la classification, le SVM peut être utilisé pour la régression en trouvant un hyperplan avec une marge maximale qui englobe autant de données que possible.

- **Les k- plus proches voisins (KNN) :**

L'algorithme des k plus proches voisins (k-NN) est un classifieur non paramétrique basé sur les distributions de probabilité. Il effectue une classification basée sur les instances, sans phase d'entraînement préalable. Les nouvelles instances sont comparées aux données d'entraînement en calculant les distances par rapport aux instances existantes. Les k plus proches voisins sont sélectionnés et la classe majoritaire de cet ensemble est attribuée à l'instance à classer. Le choix de la valeur de k est crucial, car il influence la classification. Différentes métriques de distance,

telles que Manhattan, euclidienne, Minkowski et Mahalanobis, sont utilisées par l'algorithme k-NN pour comparer les vecteurs de caractéristiques. [28]

- **L'algorithme J48**

L'algorithme J48, également connu sous le nom d'algorithme C4.5, est utilisé en apprentissage automatique pour l'analyse catégorique et continue des ensembles de données. Il se compose de deux phases principales : l'apprentissage, où l'algorithme est entraîné à classer les données, et la classification, où il étiquette de nouvelles données. En utilisant la théorie de l'information, l'algorithme C4.5/J48 génère des arbres de décision, une extension de l'algorithme ID3. L'implémentation J48 propose des fonctionnalités supplémentaires telles que la gestion des valeurs manquantes, l'élagage des arbres et la dérivation de règles. L'algorithme construit des arbres de décision en utilisant l'entropie de l'information et choisit l'attribut offrant le gain d'information le plus élevé pour diviser les échantillons en sous-groupes. Il utilise une approche gloutonne basée sur le gain d'information normalisé pour séparer les échantillons en sous-groupes enrichis en une classe spécifique. [29]

- **Random Forest**

La forêt aléatoire est une méthode qui crée plusieurs arbres de décision et les combine pour obtenir des prédictions plus précises et plus stables. Comme mentionné précédemment, une forêt aléatoire est un ensemble d'arbres de décision, mais il existe quelques différences. Lorsque vous fournissez un ensemble de données d'apprentissage avec des entités et des étiquettes à un arbre de décision, celui-ci génère un ensemble de règles utilisées pour effectuer des prédictions. Par exemple, si vous souhaitez prédire si une personne va cliquer sur une publicité en ligne, vous pouvez utiliser des caractéristiques décrivant les publicités sur lesquelles la personne a précédemment cliqué, ainsi que d'autres informations pertinentes. L'arbre de décision génère alors des règles permettant de prédire si une publicité sera cliquée ou non.

En revanche, l'algorithme de la forêt aléatoire sélectionne aléatoirement des observations et des caractéristiques pour créer plusieurs arbres de décision. Ensuite, il combine les résultats de ces arbres en effectuant une moyenne [30]. Cela permet d'améliorer les performances de prédiction en réduisant le surajustement et en augmentant la robustesse de l'algorithme.

- **L'algorithme NB**

L'algorithme NB (Naïve Bayes) est un algorithme d'apprentissage automatique couramment utilisé pour la classification. Il se base sur le théorème de Bayes et suppose une indépendance naïve entre les caractéristiques du modèle. Il utilise les probabilités conditionnelles pour prédire la classe d'un nouvel exemple en se basant sur les caractéristiques observées. Bien qu'il fasse l'hypothèse de

l'indépendance entre les caractéristiques, ce qui peut être irréaliste, l'algorithme NB peut fournir de bonnes performances dans de nombreux cas. Il est rapide à entraîner et à utiliser, même avec de grandes quantités de données, et fonctionne bien avec des données textuelles. Cependant, il peut être sensible aux caractéristiques corrélées et peut avoir des performances réduites dans de tels cas. Malgré cela, l'algorithme NB reste populaire pour la classification rapide et efficace dans divers domaines. [31]

3. Conclusion

Dans ce chapitre, nous avons commencé par définir l'apprentissage automatique en présentant ses trois catégories principales : supervisée, non supervisée et semi-supervisée. Ensuite, nous avons abordé la détection d'anomalies en décrivant les différents types d'anomalies ainsi que le concept fondamental de la détection d'anomalies. Nous avons également présenté les algorithmes et les plateformes d'apprentissage automatique que nous avons utilisés dans notre étude. Le prochain chapitre sera entièrement consacré à notre travail, qui consiste en la proposition d'une approche spécifique pour la détection des intrusions dans IoT.

3. Introduction

Pour assurer une utilisation aisée de l'Internet des objets (IoT), il est essentiel de dissimuler la complexité technologique sous-jacente et de permettre une manipulation sans souci afin de prévenir les menaces et les risques potentiels. Les chapitres précédents ont mis en évidence que, dans l'IoT, tout objet peut être connecté à Internet et capable de communiquer avec d'autres objets, ce qui engendre plusieurs risques, notamment en ce qui concerne la sécurité des données échangées entre les objets, en particulier les attaques des botnets. Nous avons étudié les solutions existantes pour garantir la sécurité, mais celles-ci se révèlent parfois inefficaces voire inapplicables pour les objets ayant des ressources limitées. Le défi consiste donc à concevoir une approche capable de détecter les différentes attaques des botnets dans les réseaux IoT sans perturber leur fonctionnement. Dans ce chapitre, nous présentons notre approche de détection des attaques des botnets dans les réseaux IoT, ainsi que sa mise en œuvre concrète. Nous détaillons les différentes étapes et outils utilisés pour le prétraitement des données brutes de la base de données utilisée. Ensuite, nous exposons les études expérimentales que nous avons réalisées dans le cadre de ce travail, dont l'objectif principal est de générer des classifieurs permettant de détecter les attaques des botnets dans les réseaux IoT en utilisant diverses approches d'apprentissage automatique. Enfin, nous présentons et discutons des résultats de l'évaluation des performances obtenues.

3.1. Description général de l’approche

Dans cette section, nous examinons les bénéfices de l'analyse du trafic pour détecter les attaques de botnets, puis nous présentons notre approche et notre modèle d'analyse de flux, qui repose sur l'analyse du trafic.

3.1.1. Analyse du trafic

Dans notre travail, nous utilisons l'analyse du trafic pour repérer les attaques de botnets dans les réseaux IoT. Notre approche repose sur l'idée que le trafic généré par les objets connectés présente généralement une certaine régularité, que nous pouvons identifier en utilisant un ensemble d'attributs spécifiques.

Les systèmes de détection basés sur l'analyse du trafic examinent l'ensemble du trafic réseau de manière globale. Dans cette étude, nous présentons une méthode de détection des attaques de botnets basée sur l'analyse du trafic, qui nous permet d'identifier l'activité des attaques de botnets dans les réseaux IoT. Pour ce faire, nous exploitons des caractéristiques particulières du trafic réseau légitime de l'IoT.

3.1.2. Approche proposée

Plusieurs études ont démontré la possibilité de détecter certaines classes de trafic réseau en observant simplement leurs modèles de trafic. Étant donné que les appareils d'un réseau IoT sont spécialisés et exécutent des tâches simples, répétitives et bien définies, leur comportement devrait présenter un certain niveau de régularité tant qu'ils ne sont pas compromis. De même, les logiciels malveillants qui infectent un appareil IoT devraient altérer son comportement dans le réseau. Ainsi, notre approche tire parti des avantages des classifieurs pour modéliser le comportement légitime des appareils et détecter les événements anormaux, sans avoir à collecter et étiqueter des données malveillantes pour former le modèle de comportement, ce qui peut parfois être impossible à réaliser.

Le fonctionnement de notre approche de détection comprend deux phases :

1. La première phase est le prétraitement des données, qui comprend quatre étapes. La première étape consiste à extraire l'ensemble des caractéristiques du flux à partir de la capture brute du réseau à l'aide du logiciel CICFlowMeter. Cet outil traite, analyse et extraire le trafic réseau IoT. La deuxième et troisième étape concerne l'encodage des données catégorielles dans un format numérique et la normalisation des données. La dernière étape est la sélection des attributs, pour laquelle nous utilisons deux méthodes : le coefficient de corrélation de Spearman (SCC) et le coefficient de tau de Kendall (KTC), afin de choisir un sous-ensemble représentatif des attributs permettant une classification précise du trafic des attaques de botnets.

2. La deuxième phase concerne les résultats des modèles de classification construits en utilisant l'ensemble des caractéristiques. Dans le cadre de ce travail, nous nous concentrons sur cinq algorithmes de classification : SVM, KNN, J48, RF et NB.

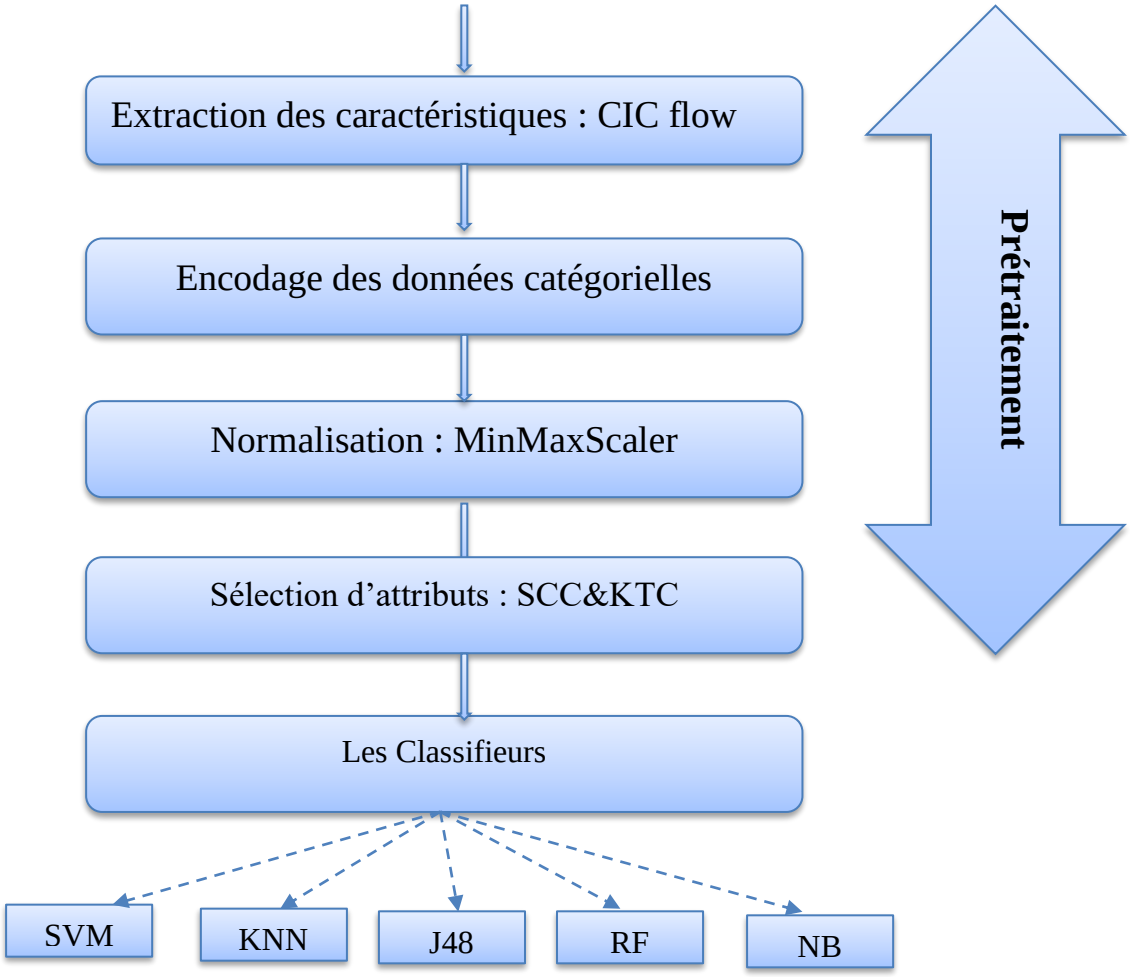

Figure 3.1. La méthodologie proposée

3.2. Capture réseau

Les données brutes utilisées dans notre approche proviennent de la base de données iot-network-intrusion, qui a pour objectif de fournir un large ensemble de données réelles et étiquetées sur les infections de logiciels malveillants IoT, ainsi que sur le trafic IoT légitime.

La base de données d'intrusion de réseau IoT a été publiée le vendredi 27/09/2019 à des fins académiques. Elle a été créée par des chercheurs pour regrouper différents types d'attaques réseau dans le contexte de l'Internet des Objets (IoT). La base de données comprend à la fois des activités normales liées à l'IoT et d'autres types de trafic réseau, ainsi que diverses formes de trafic d'attaque couramment utilisées par les botnets.

Deux appareils intelligents couramment utilisés, le SKT NUGU (NU 100) et la caméra Wi-Fi EZVIZ (C2C Mini O Plus 1080P), ont été utilisés dans cette base de données. Tous les appareils,

y compris certains ordinateurs portables et téléphones intelligents, étaient connectés au même réseau sans fil.

La base de données d'intrusion de réseau IoT se compose de 42 fichiers bruts de captures de paquets réseau (pcap) réalisées à différents moments. Ces fichiers ont été capturés en utilisant le mode moniteur de l'adaptateur réseau sans fil, et les en-têtes sans fil ont été supprimés à l'aide de l'outil Aircrack-ng. À l'exception de la catégorie des botnets, toutes les attaques sont basées sur des paquets capturés lors de simulations d'attaques à l'aide d'outils tels que Nmap.

Dans le cas de la catégorie des botnets Mirai, les paquets d'attaque ont été générés sur un ordinateur portable, puis manipulés pour donner l'impression qu'ils provenaient de l'appareil IoT.

En raison de la taille importante de l'ensemble de données iot-network-intrusion, nous avons sélectionné 7 scénarios, un pour chaque type de botnet, ainsi qu'un scénario d'appareils IoT réels non infectés. Voici les 8 scénarios sélectionnés :

1. benign-dec. pcap. pcap

2. dos-synflooding-1-dec. pcap_Flow

3. mirai-ackflooding-1-dec. pcap_Flow

4. mirai-hostbruteforce-1-dec. pcap_Flow

5. mirai-httpflooding-1-dec. pcap_Flow

6. mirai-udpflooding-1-dec. pcap_Flow

7. scan-hostport-1-dec. pcap_Flow

8. scan-portos-1-dec. pcap_Flow

3.3. Prétraitement des données

Le prétraitement des données est une méthode employée pour convertir les données brutes en un ensemble de données nettoyées. En d'autres termes, lorsque les données sont collectées à partir de diverses sources, elles sont d'abord recueillies sous une forme brute qui n'est pas adaptée à l'analyse. Cette méthode est réalisée avant d'entraîner les algorithmes d'apprentissage automatique, car ces algorithmes apprennent à partir des données et les résultats obtenus dépendent fortement de la qualité des données. Par conséquent, plusieurs étapes sont nécessaires pour transformer les données en un ensemble de données nettoyées, ce processus étant appelé prétraitement des données. Dans notre approche, la phase de prétraitement des données comprend quatre étapes : l'extraction des caractéristiques à l'aide de CICFlowMeter, l'encodage des données catégorielles,

la normalisation des données, et enfin la sélection des attributs en utilisant des filtres basés sur l'importance des attributs.

3.3.1. Extraction des caractéristiques

Il existe diverses méthodes pour extraire des flux à partir du trafic réseau, et l'une des approches les plus courantes consiste à utiliser des extracteurs de flux. Dans notre cas, nous avons utilisé l'outil CICFlowMeter.

Le CICFlowMeter est un outil open source qui prend des fichiers pcap en entrée et génère des biflux tout en extrayant les caractéristiques de ces flux. Il permet de générer des biflux bidirectionnels, où le premier paquet détermine les directions avant (de la source vers la destination) et arrière (de la destination vers la source), permettant ainsi le calcul séparé des caractéristiques statistiques liées au temps dans ces deux directions. Il offre également des fonctionnalités supplémentaires, telles que la sélection de fonctionnalités à partir d'une liste existante, la conversion des fichiers pcap au CSV, l'ajout de nouvelles fonctionnalités et le contrôle de la durée d'expiration du flux. [30]

3.3.2. Encodage des données catégorielles

CICFlowMeter génère 84 attributs, certains étant numériques et d'autres qualitatifs. Les variables numériques peuvent être directement utilisées par les algorithmes d'apprentissage automatique, tandis que les variables qualitatives nécessitent une transformation en données numériques. Pour cela, la méthode OneHotEncoder est utilisée, créant une représentation binaire pour chaque catégorie de la variable. Cette approche évite la création d'un grand nombre de nouvelles variables et facilite l'utilisation des algorithmes d'apprentissage automatique, en économisant de la mémoire. Il est important de noter que l'OneHotEncoder doit être appliqué uniquement aux variables catégorielles, tandis que les autres variables numériques nécessitent un traitement séparé, tel que la mise à l'échelle ou la normalisation.

3.3.3.Normalisation

La normalisation des données est souvent requise lorsque les chercheurs utilisent des techniques d'apprentissage automatique sur des données qui présentent des différences d'échelle entre leurs attributs. Dans notre étude, nous avons comparé les performances du modèle avec normalisation des caractéristiques.

Dans notre travail, nous avons utilisé la méthode MinMaxScaler pour normaliser les données. Cette technique a été mise en œuvre en utilisant la bibliothèque scikit-learn en Python. Son principe est assez simple : la valeur minimale de chaque caractéristique est transformée en 0, tandis que la valeur maximale est transformée en 1. Ainsi, toutes les autres valeurs sont converties en décimales comprises entre 0 et 1 en utilisant la formule suivante :

$$X_{normaliser} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (3.1)$$

3.3.4.Sélection d'attributs

Nous avons utilisé deux méthodes de sélection des caractéristiques : le coefficient de corrélation de Spearman (SCC) et le coefficient de Kendall-Tau (KTC),

1-Le coefficient de corrélation de Spearman est une mesure qui évalue la relation entre deux variables, sans supposer une relation linéaire entre elles. Il est spécialement conçu pour évaluer une relation monotone croissante ou décroissante entre deux ensembles de données. La formule utilisée pour calculer ce coefficient est la suivante :

$$r_s = 1 - \frac{6\sum d^2}{n(n^2 - 1)} \qquad (3.2)$$

Ou

di : correspond à la différence entre les rangs des observations des deux ensembles de données pour chaque individu.

n Est la taille de l'échantillon

En utilisant le SCC comme méthode de sélection des caractéristiques, nous sommes en mesure d'identifier les caractéristiques qui présentent une corrélation significative avec la variable cible. Cela nous permet de choisir les caractéristiques les plus pertinentes et de réduire la dimensionnalité des données en nous concentrant sur celles qui ont une relation étroite avec la variable cible.

De cette méthode, nous avons obtenu 10 attributs.

1. Pkt Len Std
2. Pkt Len Var
3. Fwd Pkt Len Mean
4. Fwd Seg Size Avg
5. Pkt Size Avg
6. Pkt Len Mean
7. Idle Min
8. Idle Mean

9. Bwd Pkt Len Max
10. TotLen Bwd Pkts

La figure 4.2. Présente la matrice de corrélation, Les attributs avec une corrélation élevée sont plus linéairement dépendants et ont donc presque le même effet sur la variable dépendante.

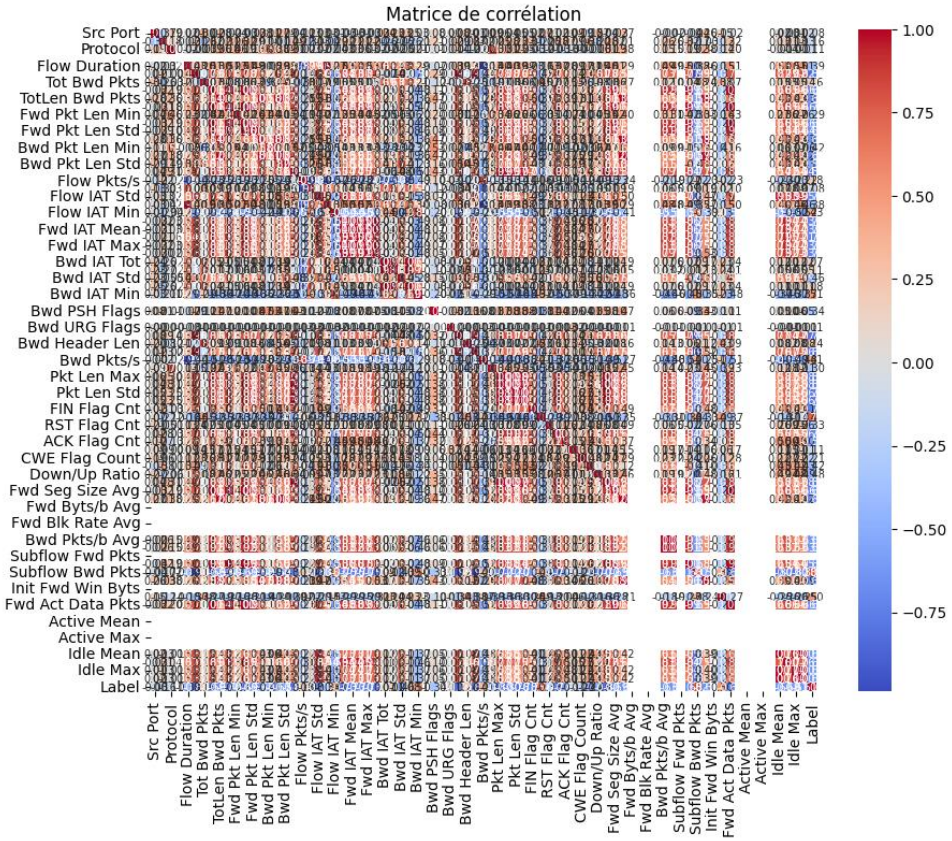

Figure 3.2. Matrice de corrélation (SCC)

2- Le coefficient de Kendall, connu sous le nom de tau, mesure l'association entre deux variables mesurées sur une échelle ordinale, où chaque observation peut être classée par ordre. De manière similaire au coefficient de Spearman, cette méthode statistique utilise les rangs des observations, par opposition au coefficient de corrélation de Pearson qui utilise les valeurs brutes.

Le tau est interprété de la même manière que les autres coefficients de corrélation. Sa valeur varie entre -1 et 1 : une valeur proche de 1 suggère une corrélation positive (les variables varient dans le même sens), une valeur proche de -1 indique une corrélation négative, tandis qu'une valeur proche de 0 suggère qu'il n'y a probablement aucune relation entre les deux variables. Sa formule est la suivante :

$$t = \frac{C-D}{\frac{1}{2}n(n-1)} \qquad (3.3)$$

C : est le nombre de paires concordantes

D : est le nombre de paires discordantes.

n: est le nombre total de paires.

Dans le but de le comparer aux valeurs de la loi normale centrée réduite, il est nécessaire d'appliquer la même transformation au tau. Comme l'espérance est nulle, le centrage est réalisé implicitement, il suffit donc de diviser le coefficient par son écart-type pour le réduire.

Comparé au coefficient de Spearman, le coefficient de Kendall présente l'avantage de mieux gérer les cas où les rangs sont ex-aequo.

De cette méthode, nous avons obtenu 15 attributs

1 Bwd Pkt Len Mean
2 Pkt Len Var
3 Pkt Len Std
4 Fwd Pkt Len Mean
5 Fwd Seg Size Avg
6 Pkt Len Mean
7 Pkt Size Avg
8 Idle Min
9 Idle Mean
10 TotLen Bwd Pkts
11 Bwd Pkt Len Max
12 Bwd IAT Max
13 Bwd IAT Tot
14 TotLen Fwd Pkts
15 Bwd Seg Size Avg

La Figure 4.3 présente la matrice de corrélation, Les attributs avec une corrélation élevée sont plus linéairement dépendants et ont donc presque le même effet sur la variable dépendante.

On outre après avoir obtenu le résultat des méthodes de sélection, l'entrainement a été réalisé avec les caractéstisque communes obtenues par ces deux méthodes.

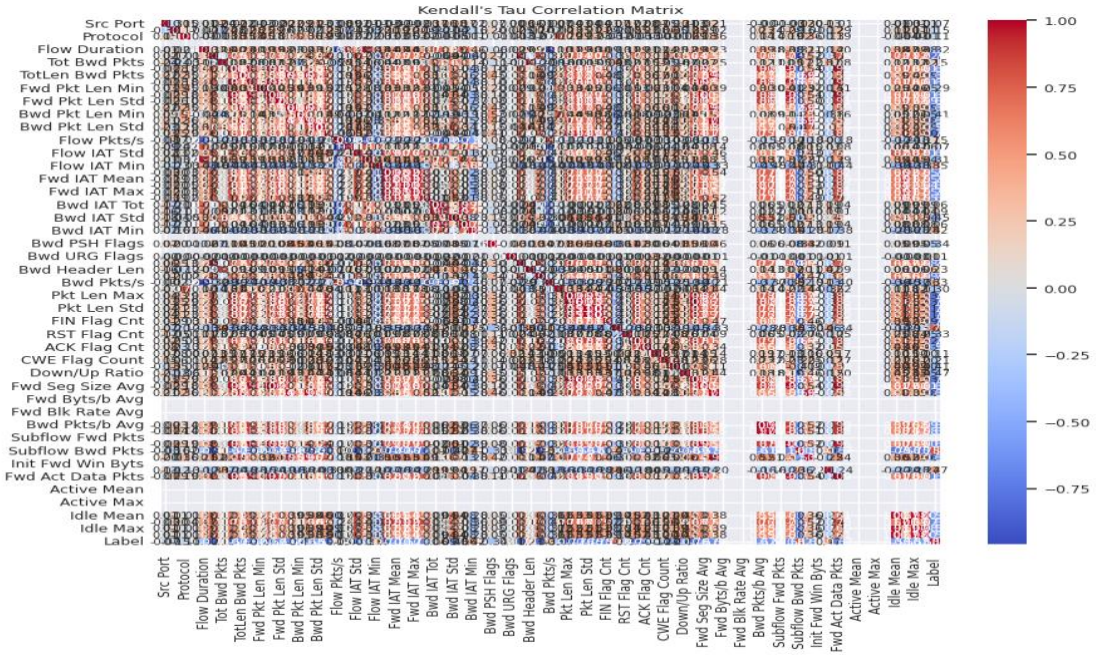

Figure 3.3: matrice de corrélation (KTC)

Et des deux méthodes, SCC et KTC, nous prenons les attributs communs

3.4. Evaluation des performances

Dans cette section, nous présentons l'implémentation de notre approche pour les tests. Pour commencer, nous avons utilisé Python pour mettre en œuvre notre approche dans le cadre du service cloud gratuit Google Colab (Colaboratory). Google Colab est basé sur Jupyter Notebook et est spécialement conçu pour la formation et la recherche en apprentissage automatique. Il offre un accès facile à différentes bibliothèques qui permettent d'utiliser les services fournis par Google.

L'avantage de Colab est que tous les scripts Jupyter sont enregistrés sur Google Drive, ce qui permet de les partager efficacement où que vous soyez. Afin de ne pas perdre nos données lorsque nous nous déconnectons du service, nous avons décidé de télécharger notre base de données contenant les fichiers CSV sur Google Drive. Nous les lisons à partir de cet emplacement chaque fois que nous en avons besoin.

Pour l'implémentation des classifieurs SVM, KNN, J48, RF et NB, nous avons utilisé la bibliothèque Python Scikit-learn.

Pour l'évaluation de notre approche proposée, nous avons utilisé Sept métriques de performances. Ces métriques sont calculées à l'aide de quatre mesures différentes : vrai positif (TP), vrai négatif (TN), faux positif (FP) et faux négatif (FN).

- **TP** : si une instance anormale est classée comme anormale, elle est acceptée comme TP.

- **FP** : si une instance normale est classée comme anormale, elle est acceptée comme FP.

- **TN** : si une instance normale est classée comme normale, elle est acceptée comme TN.

- **FN** : si une instance anormale est classée comme normale, elle est acceptée comme FN.

Précision (Accuracy)

La précision (précision) mesure la proportion d'instances anormales et normales correctement classées par rapport au nombre total d'instances. Elle évalue la capacité générale d'un modèle à être bien entraîné et à fonctionner correctement. Cependant, la précision ne fournit pas de détails spécifiques sur l'application du modèle à des problèmes particuliers. Elle permet d'évaluer l'exactitude globale du modèle, mais ne donne pas d'informations détaillées sur sa performance pour des cas ou des classes spécifiques.

$$\boldsymbol{Accuracy} = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (3.4)$$

Précision (Précision)

La précision (précision) est le rapport entre le nombre d'instances anormales correctement classées et le nombre total d'instances classées comme anormales par le modèle. Elle mesure à quelle fréquence le modèle est correct lorsqu'il fait une prédiction positive. Il est important de noter que la précision ici diffère de la précision décrite précédemment (Accuracy), qui mesure la précision globale du modèle dans toutes les classes, qu'elles soient anormales ou normales.

$$\boldsymbol{Précision} = \frac{TP}{(TP+FP)} \quad (3.5)$$

Rappel (Recall)

Le rappel (recall) est le rapport entre le nombre d'instances anormales correctement classées et le nombre total d'instances anormales réelles. Il évalue la capacité d'un modèle de classification à détecter toutes les instances pertinentes. Le rappel est particulièrement utile lorsque le taux de faux négatifs est élevé, car il mesure l'efficacité de l'approche dans l'identification des réseaux de botnet.

$$\boldsymbol{Recall} = TP\frac{TP}{(TP+FN)} \quad (3.6)$$

Taux de vrais positifs (TPR)

Le TPR, également appelé sensibilité ou rappel, est calculé en divisant les vrais positifs par le total des instances réelles positives. Il représente la proportion d'instances positives réelles correctement identifiées par le modèle.

$$\boldsymbol{TPR} = TP\,\frac{TP}{(TP+FN)} \quad (3.7)$$

Taux de faux positifs (FPR)

Le FPR est calculé en divisant les faux positifs par le total des instances réelles négatives, Il mesure la proportion d'instances négatives réelles incorrectement classées comme positives par le modèle.

$$\boldsymbol{FPR} = \frac{FP}{(FP+TN)} \quad (3.8)$$

F1score

Le F1-score est une mesure de précision d'un test qui prend en compte à la fois la précision et le rappel du test. Il est calculé comme la moyenne harmonique de la précision et du rappel, et sa valeur maximale est de 1. La formule du F1-score est donnée par

$$\boldsymbol{F1 - Score} = 2 \times \frac{Recall \times précision}{Recal + précision} \quad (3.9)$$

Courbe ROC (Receiver Operating Characteristic)

La courbe ROC est une représentation graphique du TPR en fonction du FPR pour différents seuils de classification. Elle permet d'évaluer la performance du modèle en mesurant le taux de vrais positifs par rapport au taux de faux positifs. La courbe ROC fournit une mesure globale de la capacité de discrimination du modèle.

3.5. Résultats et discussion

Dans cette section, nous présenterons et analyserons les résultats obtenus après avoir implémenté les algorithmes décrits dans le Chapitre 3. Pour évaluer les performances de classification, nous utiliserons plusieurs métriques telles que l'accuracy, la précision, le recall, le TPR, le FPR ,f1-score et la courbe ROC. Ainsi, nous prévoyons de réaliser deux expérimentations distinctes.

- nous allons diviser la base de données en deux ensembles distincts. La première base de données sera dédiée à une classification binaire, tandis que la deuxième base de données sera utilisée pour une classification multiclasse. Cette division nous permettra d'évaluer les performances de nos modèles pour différents types de classification et d'obtenir des insights spécifiques à chaque cas.

La première expérimentation

Dans le cadre de la première expérience, nous avons construit notre modèle en utilisant le trafic légitime provenant d'objets réels de manière séparée. Par la suite, ce modèle a été testé sur un ensemble de données binaires. L'objectif de cette expérience était d'évaluer la capacité de notre modèle à différencier le trafic légitime du trafic malveillant. Les résultats de cette expérience sont résumés dans ce tableau. Nous explorons plusieurs pistes pour améliorer les résultats de détection en termes de faux négatifs et de faux positifs, telles que la sélection des caractéristiques et l'optimisation des hyperparamètres des classificateurs.

	Classifieur	Accuracy	Precision	Recall	F1-score	TPR
le trafic légitime	SVM	81%	70%	47%	56%	33%
	KNN	78%	64%	41%	50%	38%
	J48	95%	91%	92%	91%	35%
	RF	93%	93%	79%	85%	22%
	NB	77%	65%	29%	41%	40%
Le tradic malveillent	SVM	81%	83%	92%	87%	75%
	KNN	78%	81%	91%	86%	58%
	J48	95%	97%	96%	97%	65%
	RF	93%	93%	97%	95%	89%
	NB	77%	65%	29%	86%	78%

Tableau 3.1. Résultat binaire avec méthode de sélection

Le tableau 4.1 présenté ci-dessus affiche les résultats des métriques pour l'évaluation des classifieurs dans le cas d'une base de données binaire, en utilisant une méthode de sélection des attributs.

Pour la classe "trafic légitime", le classifieur J48 obtient la meilleure exactitude (95%) par rapport aux autres classifieurs. Il a également une précision élevée (91%), ce qui signifie que la plupart des instances qu'il prédit comme étant du trafic légitime sont correctes. Cependant, son rappel

(92%) et son score F1 (91%) sont légèrement inférieurs, ce qui indique qu'il peut manquer certaines instances de trafic légitime.

Le classifieur RF présente également de bons résultats pour la classe "trafic légitime", avec une exactitude de 93% et une précision de 93%. Cependant, son rappel (79%) et son score F1 (85%) sont inférieurs à ceux de J48, ce qui suggère qu'il a plus de difficulté à détecter toutes les instances de trafic légitime.

Les classifieurs SVM et KNN ont des performances similaires pour la classe "trafic légitime", avec une exactitude d'environ 81%. Cependant, leur précision, leur rappel et leur score F1 sont inférieurs à ceux de J48 et RF.

En ce qui concerne la classe "trafic malveillant", le classifieur J48 obtient à nouveau la meilleure exactitude (95%) parmi tous les classifieurs. Il présente également une précision élevée (97%), ce qui indique qu'il est précis dans la prédiction des instances de trafic malveillant. Son rappel (96%) et son score F1 (97%) sont également élevés, ce qui suggère qu'il est capable de détecter la plupart des instances de trafic malveillant.

Le classifieur RF obtient des résultats similaires pour la classe "trafic malveillant", avec une exactitude de 93%, une précision de 93% et un rappel élevé (97%). Cependant, son score F1 (95%) est légèrement inférieur à celui de J48.

Les classifieurs SVM, KNN et NB ont des performances similaires pour la classe "trafic malveillant", avec une exactitude d'environ 81%. Cependant, leur précision, leur rappel et leur score F1 sont inférieurs à ceux de J48 et RF.

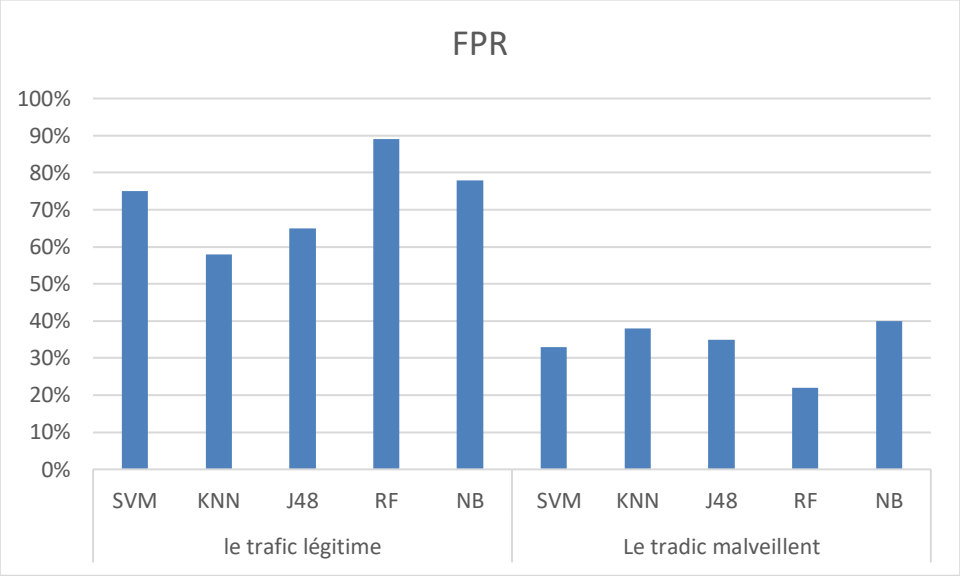

Figure 3.4. Performance globale de L'IDS proposé en termes de taux de faux positifs

En observant ces résultats, nous constatons que pour le trafic légitimt le classifieur RF présente le FPR le plus élevé, suivi par le classifieur NB, J48, KNN et SVM.

En revanche, pour le trafic malveillant, le classifieur RF affiche le FPR le plus bas, suivi par le J48, SVM, KNN et NB.

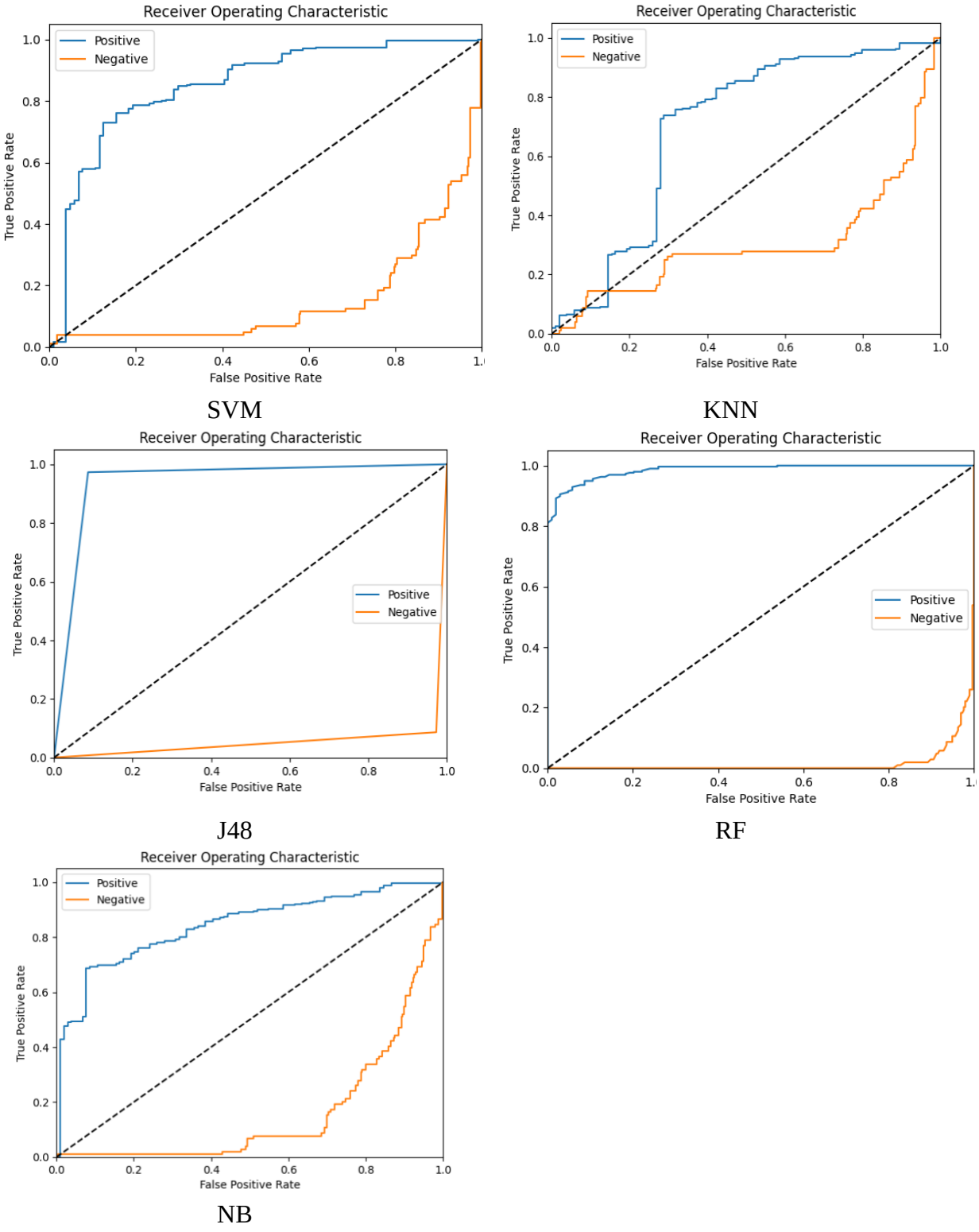

Figure 3.5. Les Courbe ROC avec méthode de sélection

	Classifieur	Accuracy	Precision	Recall	F1-score	TPR
Le trafic légitime	SVM	86%	99%	100%	68%	30%
	KNN	87%	83%	63%	72%	25%
	J48	94%	91%	87%	96%	37%
	RF	93%	95%	78%	54%	18%
	NB	54%	38%	94%	54%	35%
Le trafic malveillant	SVM	86%	86%	95%	91%	84%
	KNN	87%	88%	95%	91%	87%
	J48	94%	95%	98%	95%	65%
	RF	93%	92%	99%	95%	88%
	NB	59%	95%	46%	63%	48%

Tableau 3.2. Résultat binaire sans méthode de sélection

Le tableau présenté ci-dessus affiche les résultats des métriques pour l'évaluation des classifieurs dans le cas d'une base de données binaire, sans utilisation d'une méthode de sélection des attributs. Pour la classe "trafic légitime", le classifieur SVM obtient une exactitude de 86%. Il a une précision élevée de 99%, ce qui indique que la plupart des instances prédites comme étant du trafic légitime sont correctes. Son rappel (100%) et son score F1 (68%) montrent qu'il est capable de détecter toutes les instances de trafic légitime, mais il peut également générer un grand nombre de faux positifs. Le TPR (taux de vrais positifs) est faible à 30%, ce qui signifie qu'il a du mal à détecter la classe cible de manière satisfaisante.

Le classifieur KNN présente des performances légèrement meilleures pour la classe "trafic légitime" avec une exactitude de 87%. Sa précision (83%) et son rappel (63%) sont raisonnables, mais son score F1 (72%) et son TPR (25%) sont relativement faibles, indiquant qu'il peut manquer certaines instances de trafic légitime.

Le classifieur J48 obtient une exactitude élevée de 94% pour la classe "trafic légitime". Il présente également une précision de 91% et un rappel de 87%, ce qui indique qu'il est précis et capable de détecter la plupart des instances de trafic légitime. Son score F1 (96%) et son TPR (37%) montrent une bonne performance globale.

Le classifieur RF a une exactitude de 93% pour la classe "trafic légitime". Il présente une précision élevée de 95%, mais son rappel (78%) et son score F1 (54%) sont plus faibles, ce qui indique qu'il

peut manquer un certain nombre d'instances de trafic légitime. Le TPR est le plus bas parmi tous les classifieurs, à seulement 18%.

Le classifieur NB a une exactitude relativement basse de 54% pour la classe "trafic légitime". Il a une précision faible (38%) et un score F1 (54%) qui correspondent à l'exactitude globale. Cependant, son rappel (94%) est élevé, ce qui indique qu'il est capable de détecter la plupart des instances de trafic légitime. Le TPR est de 35%.

Pour la classe "trafic malveillant", les classifieurs SVM et KNN ont des performances similaires avec une exactitude de 86% et 87% respectivement. Leurs précisions, rappels, scores F1 et TPR sont également assez proches, montrant une capacité raisonnable à détecter le trafic malveillant.

Les classifieurs J48 et RF obtiennent de bons résultats pour la classe "trafic malveillant", avec une exactitude de 94% et 93% respectivement. Ils ont des précisions élevées (95% et 92%) et des rappels élevés (98% et 99%), indiquant une bonne capacité à détecter le trafic malveillant. Le TPR est plus élevé pour ces classifieurs, avec 65% pour J48 et 88% pour RF.

Le classifieur NB présente des performances moins satisfaisantes pour la classe "trafic malveillant". Il a une exactitude de 59% et un rappel de 46%, indiquant qu'il peut manquer un certain nombre d'instances de trafic malveillant. Cependant, sa précision est élevée (95%), ce qui signifie que la plupart des instances prédites comme étant du trafic malveillant sont correctes. Le TPR est de 48%.

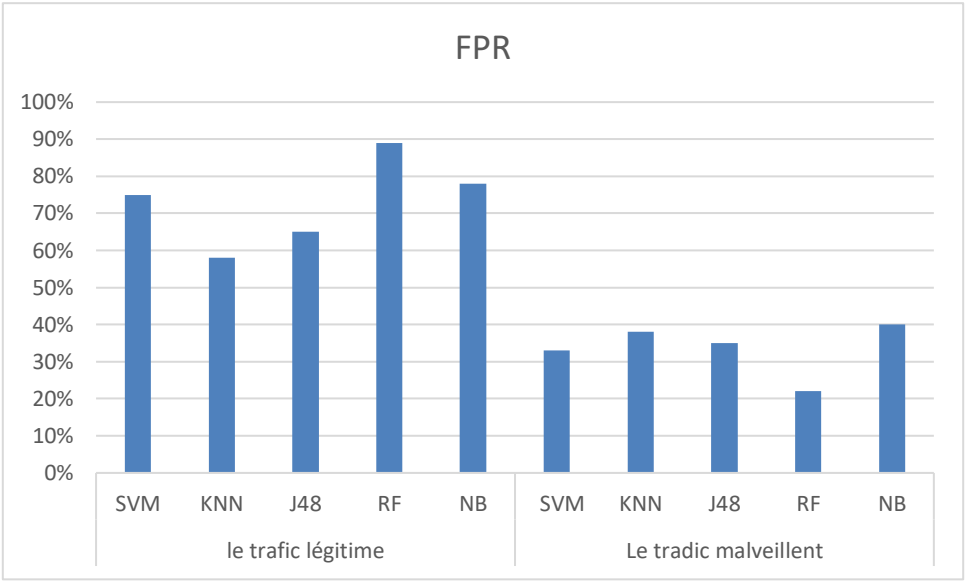

Figure 3.6. Représentation des performances des algorithmes sans la sélection des attributs.

Le classifieur RF présente le taux de faux positifs le plus élevé, tandis que le classifieur SVM affiche le taux le plus bas dans la classe négative. Pour la classe positive, le classifieur RF obtient le taux de faux positifs le plus bas, suivi par le classifieur J48.

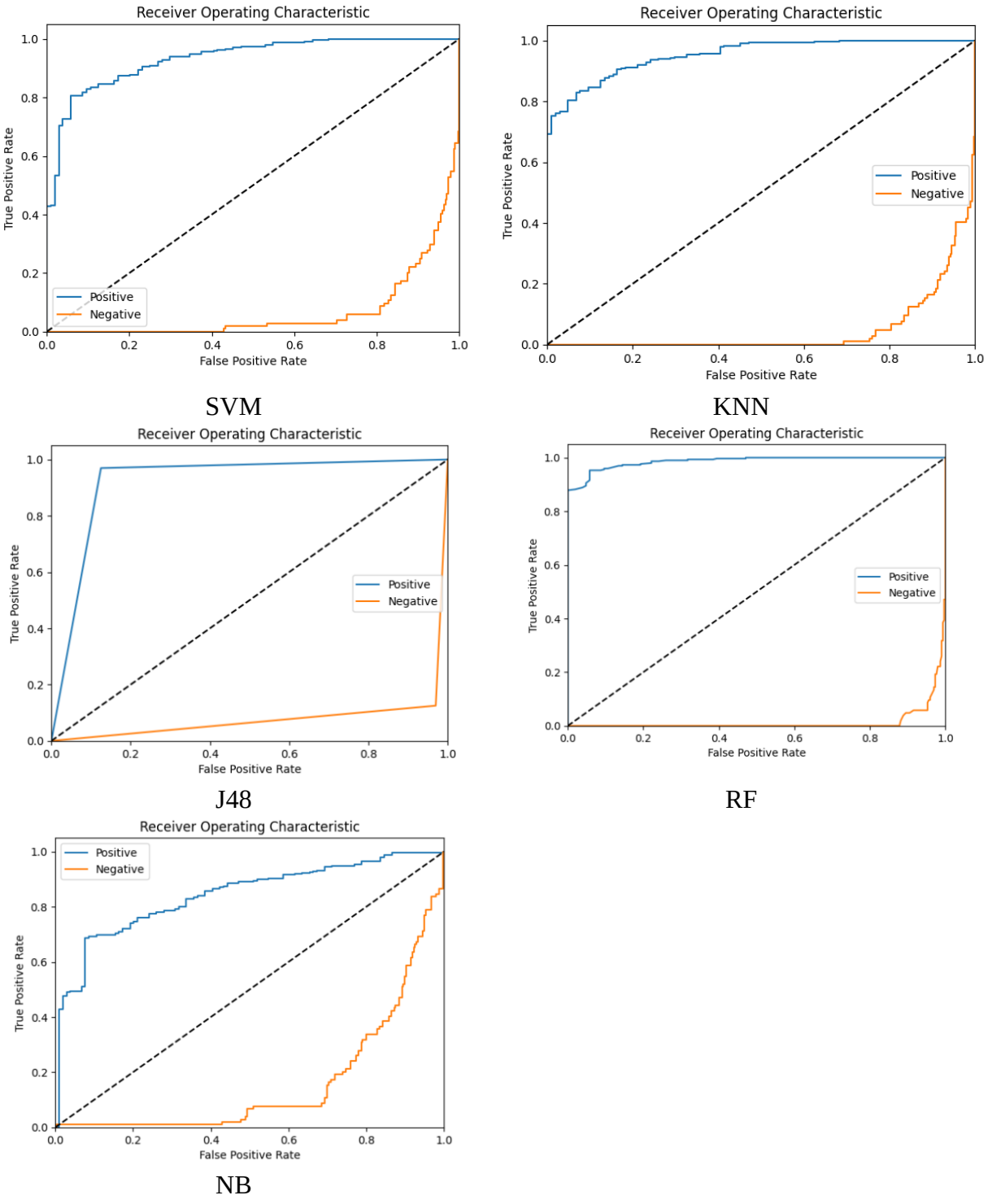

Figure 3.7. Les Courbe ROC sans méthode de sélection

La deuxième expérimentation

L'objectif de cette expérimentation était de mesurer la capacité de notre modèle à distinguer le trafic légitime spécifique à chaque objet du trafic malveillant, ainsi que d'évaluer son efficacité dans la détection des différents types de trafic

1- Avec méthode de sélection

Type de malveillants	Classifieur	Accuracy	Precision	Recall	TPR
Mirai – Httpflooding	SVM	65%	68%	65%	73%
	KNN	74%	51%	74%	55%
	J48	91%	87%	91%	63%
	RF	91%	84%	91%	73%
	NB	37%	67%	37%	67%
Mirai-udpflooding	SVM	64%	79%	64%	79%
	KNN	62%	50%	62%	61%
	J48	91%	42%	59%	63%
	RF	59%	53%	59%	78%
	NB	73%	24%	73%	67%
Mirai-hostbruteforce	SVM	48%	50%	48%	62%
	KNN	65%	50%	62%	62%
	J48	64%	63%	64%	57%
	RF	64%	62%	64%	84%
	NB	65%	71%	65%	71%
scan-portos	SVM	48%	95%	88%	89%
	KNN	65%	53%	41%	47%
	J48	84%	88%	84%	61%
	RF	84%	87%	84%	76%
	NB	13%	39%	13%	53%
scan-hostport	SVM	96%	99%	96%	94%
	KNN	84%	91%	84%	74%
	J48	93%	97%	93%	64%

	RF	92%	95%	92%	80%
	NB	100%	31%	5%	16%
Dos-synflooding (6)	SVM	99%	99%	99%	95%
	KNN	97%	97%	97%	81%
	J48	98%	94%	98%	66%
	RF	99%	95%	98%	83%
	NB	97%	41%	97%%	95%
Mirai-ackflodin	SVM	97%	99%	97%	95%
	KNN	97%	100%	99%	79%
	J48	99%	99%	99%	66%
	RF	99%	99%	99%	95%
	NB	99%	99%	99%	88%

Traffic normal					
Benign	SVM	99%	97%	97%	94%
	KNN	97%	97%	94%	79%
	J48	99%	99%	94%	94%
	RF	99%	99%	94%	95%
	NB	97%	97%	97%	94%

Tableau 3.3. Résultat de la classification Multiclasse avec méthode de sélection

Le tableau 4.2 ci-dessus révèle les résultats des métriques pour l'évaluation des classifieurs dans une base de données multiclasse, en utilisant une méthode de sélection des attributs. Chaque type d'attaque malveillante a été analysé individuellement.

Lors de l'analyse des performances des classifieurs pour chaque type d'attaque, nous observons des résultats variables. Pour l'attaque Mirai, le classifieur SVM présente une exactitude de 65% et une précision de 68%, tandis que le classifieur KNN affiche une exactitude de 74% et une précision de 51%. Les attaques Httpflooding sont mieux détectées par les classifieurs J48 et RF, qui obtiennent tous deux une exactitude de 91% et des précisions respectives de 87% et 84%. L'attaque udpflooding est bien identifiée par les classifieurs J48 et RF avec une exactitude de 91%, mais le classifieur RF se distingue avec une précision de 53% et un taux de vrais positifs (TPR) de 78%.

Pour l'attaque Mirai-hostbruteforce, les classifieurs J48, RF et NB ont des performances similaires avec une exactitude d'environ 64%. Le classifieur SVM se distingue avec une précision de 95% et un TPR de 89%. L'attaque scan-portos est mieux détectée par le classifieur SVM avec une exactitude de 96% et un TPR de 94%, tandis que le classifieur NB affiche une précision de 39% et un TPR de 53%. Les attaques scan-hostport sont efficacement détectées par les classifieurs J48 et RF, qui affichent des exactitudes de 93% et 92%, des précisions de 97% et 95%, et des TPR de 64% et 80% respectivement.

En ce qui concerne l'attaque Dos-synflooding, les classifieurs KNN, J48 et RF obtiennent des résultats similaires avec une exactitude d'environ 97%. Le classifieur NB présente une précision faible de 41% mais un TPR élevé de 95%, tandis que le classifieur SVM se démarque avec une précision de 99% et un TPR de 95%. Pour l'attaque Mirai-ackflooding, le classifieur KNN affiche une exactitude de 97%, une précision de 100% et un TPR de 79%, tandis que les classifieurs J48 et RF obtiennent des exactitudes élevées de 99% et des TPR respectifs de 66% et 95%.

Lors de l'analyse du trafic normal, les performances des classifieurs pour la détection de ce type de trafic ont été évaluées. Le classifieur SVM obtient une exactitude de 99% avec une précision de 97% et un rappel de 97%. Le classifieur KNN affiche une exactitude de 97% avec une précision de 97% et un rappel de 94%. Le classifieur J48 présente une exactitude de 99% avec une précision de 99% et un rappel de 94%. Le classifieur RF et le classifieur NB affichent des performances similaires, avec une exactitude de 99%, une précision de 99% et un rappel de 94% pour le classifieur RF, et une précision et un rappel de 97% pour le classifieur NB.

Type de malveillants	Classifieur	Accuracy	Precision	Recall	TPR
Mirai –	SVM	81%	49%	81%	67%
	KNN	75%	50%	75%	55%

Httpflooding	J48	91%	86%	91%	63%
	RF	89%	82%	89%	72%
	NB	37%	65%	37%	66%
Mirai-udpflooding	SVM	81%	68%	65%	67%
	KNN	65%	68%	65%	74%
	J48	59%	42%	59%	54%
	RF	59%	52%	59%	77%
	NB	72%	24%	72%	62%
Mirai-hostbruteforce	SVM	63%	74%	63%	76%
	KNN	63%	63%	63%	61%
	J48	61%	58%	61%	57%
	RF	61%	53%	61%	80%
	NB	63%	61%	63%	73%
scan-portos	SVM	81%	95%	81%	90%
	KNN	84%	53%	42%	45%
	J48	90%	83%	83%	64%
	RF	85%	85%	84%	73%
	NB	13%	38%	13%	52%
scan-hostport	SVM	50%	95%	81%	90%
	KNN	75%	91%	84%	75%
	J48	93%	97%	93%	64%
	RF	90%	96%	90%	80%
	NB	50%	32%	50%	17%
Dos-synflooding	SVM	96%	99%	96%	94%
	KNN	97%	97%	97%	81%
	J48	97%	93%	97%	65%

	RF	98%	95%	98%	85%
	NB	97%	41%	97%%	95%
Mirai-ackflodin	SVM	99%	99%	99%	95%
	KNN	99%	100%	99%	79%
	J48	99%	99%	99%	66%
	RF	99%	99%	99%	92%
	NB	99%	99%	99%	91%

Traffic normal					
Benign	SVM	97%	98%	97%	94%
	KNN	97%	98%	98%	94%
	J48	97%	99%	99%	78%
	RF	99%	99%	99%	95%
	NB	97%	98%	97%	96%

Tableau 3.4. Résultat Multiclasse sons méthode de sélection

Le tableau 4.4 fournit une évaluation des classifieurs dans une base de données multiclasse sans utilisation de méthode de sélection des attributs. Chaque attaque a été analysée individuellement, et les performances des classifieurs ont été discutées.

Lors de l'analyse des performances des classifieurs pour chaque type d'attaque malveillante, nous observons des résultats différents. Pour l'attaque Mirai - Httpflooding, le classifieur J48 obtient une exactitude élevée de 91% avec une précision de 86% et un rappel de 91%. Le classifieur RF présente également de bons résultats avec une exactitude de 89%, une précision de 82% et un rappel de 89%. Cependant, le classifieur SVM affiche une précision plus faible de 49% et un rappel de 81%.

Pour l'attaque Mirai - udpflooding ,le classifieur SVM obtient une exactitude de 81% avec une précision de 68% et un rappel de 65%. Le classifieur KNN présente des résultats similaires avec

une exactitude de 65%, une précision de 68% et un rappel de 65%. En revanche, le classifieur J48 affiche une précision plus faible de 42% et un rappel de 59%.

L'attaque Mirai-hostbruteforce est mieux détectée par le classifieur RF, qui affiche une exactitude de 61%, une précision de 53% et un rappel de 61%. Le classifieur SVM présente également de bons résultats avec une exactitude de 63%, une précision de 74% et un rappel de 63%. En revanche, le classifieur J48 a une précision inférieure de 58% et un rappel de 61%.

Pour l'attaque scan-portos, le classifieur SVM affiche une exactitude de 81% avec une précision de 95% et un rappel de 81%. Le classifieur J48 présente également de bons résultats avec une exactitude de 90%, une précision de 83% et un rappel de 83%. En revanche, le classifieur NB obtient une précision plus faible de 38% et un rappel de 13%.

L'attaque scan-hostport est mieux détectée par le classifieur J48, qui affiche une exactitude de 93%, une précision de 97% et un rappel de 93%. Le classifieur RF présente également de bons résultats avec une exactitude de 90%, une précision de 96% et un rappel de 90%. En revanche, le classifieur NB a une précision très faible de 32% et un rappel de 5%.

Pour l'attaque Dos-synflooding , le classifieur RF se démarque avec une exactitude de 98%, une précision de 95% et un rappel de 98%. Le classifieur SVM obtient également de bons résultats avec une exactitude de 96%, une précision de 99% et un rappel de 96%. En revanche, le classifieur NB présente une précision inférieure de 41% et un rappel de 97%.

Pour l'attaque Mirai-ackflodin, le classifieur KNN affiche une exactitude de 99% avec une précision de 100% et un rappel de 99%. Le classifieur J48 obtient des résultats similaires avec une exactitude de 99% et un rappel de 99%, tandis que le classifieur RF se distingue avec un rappel de 99% et une précision de 99%. Le classifieur NB présente également des résultats solides avec un rappel de 99% et une précision de 99%.

Lors de l'analyse du trafic normal

Le classifieur SVM présente une exactitude de 97% avec une précision de 98% et un rappel de 97%. Le classifieur KNN affiche une exactitude de 97% avec une précision de 98% et un rappel de 98%. Le classifieur J48 obtient une exactitude de 97% avec une précision de 99% et un rappel de 99%. Le classifieur RF se distingue avec une exactitude de 99%, une précision de 99% et un rappel de 99%., le classifieur NB présente une exactitude de 97% avec une précision de 98% et un rappel de 97%

D'après les résultats présentés, les algorithmes RF (Random Forest) semblent obtenir de bonnes performances en termes de précision, recall et F1-score. Par conséquent, il est conseillé de considérer RF comme les meilleurs candidats à partir de ces données.

5. Conclusion

Dans ce chapitre, nous avons exposé notre approche de détection des attaques des botnets dans les réseaux IoT, ainsi que son implémentation. Cette approche repose sur l'utilisation de classifieurs et intègre une méthode de sélection des attributs pertinents. Nous avons donné un aperçu de l'approche et exploré les avantages de l'analyse de trafic. Ensuite, nous avons extrait les caractéristiques catégorielles à partir des données à l'aide de l'outil CICFLOW. Nous avons également discuté de la méthode de sélection des attributs utilisée, notamment SCC et KTC. Dans la section consacrée à la génération des classifieurs, nous avons présenté le paramétrage et les outils utilisés pour implémenter les algorithmes de classification tels que SVM, KNN, J48, RF et NB. Enfin, nous avons présenté et discuté des performances des classifieurs obtenues, en analysant les résultats obtenus.

Conclusion et Perspectives

Aujourd'hui, l'Internet des objets (IoT) connaît une croissance exponentielle et les appareils IoT sont devenus omniprésents dans notre vie quotidienne. Cependant, leur connectivité croissante et leur niveau de sécurité insuffisant en font une cible attrayante pour les cyberattaquants, qui exploitent ces dispositifs pour lancer des attaques et former des botnets. Dans le cadre de notre recherche, notre objectif était de proposer une approche novatrice pour détecter les intrusions dans l'Internet des objets afin de les protéger sans perturber leur fonctionnement normal. Nous avons développé un système robuste et fiable capable de détecter les activités malveillantes au sein des réseaux d'objets connectés.

Notre démarche a d'abord consisté à analyser le trafic réseau afin de détecter les attaques des botnets. Pour mettre en œuvre cette approche, nous avons utilisé des outils puissants d'analyse et de traitement des données, tels que l'outil performant CICFLOW-METAR, ainsi que le langage Python avec ses bibliothèques. De plus, nous avons exploité deux méthodes de sélection des attributs de flux, à savoir SCC et KTC, afin d'améliorer les performances de notre approche.

Afin d'évaluer notre approche, nous avons testé cinq classifieurs différents : SVM, KNN, J48, RF et NB. Les résultats ont montré que l'algorithme RF a obtenu les meilleurs résultats.

Malgré ces résultats prometteurs, il reste encore des perspectives d'amélioration. Il serait intéressant d'étendre l'évaluation de notre approche à différents réseaux IoT et à d'autres types de botnets. De plus, une exploration de la détection d'intrusion en temps réel constituerait un sujet de recherche futur permettant d'améliorer notre travail.

.

Références

A. Références Bibliographiques

[1] BENYAHIA, S. (2017). *Vers un nouveau Système de détection d'intrusion hybride et hiérarchique basé sur les réseaux bayésiens* (Doctoral dissertation, Université Ibn Khaldoun-Tiaret-).

[2] Weill*, M., & Souissi**, M. (2010). L'Internet des objets: concept ou réalité?. *Réalités industrielles*, (4), 90-96.

[3] Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., ... & Doody, P. (2022). Internet of things strategic research roadmap. In *Internet of things-global technological and societal trends from smart environments and spaces to green ICT* (pp. 9-52). River Publishers.

[4] Roxin, I., & Bouchereau, A. (2017). Introduction aux technologies de l'écosystème de l'Internet des Objets.

[5] **CHALAL**, L., SIROUAKNE, S., & AISSANI, S. (2017). Gestion des clés dans l'internet des objets. *Master, A/Mira, Béjaia, Algerie*.

[6] Mercadet, F. (2021). *Solutions de sécurité pour l'internet des objets dans le cadre de l'assistance à l'autonomie à domicile* (Doctoral dissertation, Université de Limoges; Université Mouloud Mammeri (Tizi-Ouzou, Algérie)).

[7] Hammi, M. T. (2018). *Sécurisation de l'Internet des objets* (Doctoral dissertation, Université Paris-Saclay (ComUE)).

[8] BOUKERTOUTA, M. A. (2022). Detection des intrusions basée sur l'apprentissagautomatique dans les systèmes IdO (Internet des Objets).

[9] Bloch, L., Wolfhugel, C., Queinnec, C., Schauer, H., & Makarévitch, N. (2013). *Sécurité informatique : Principes et méthodes à l'usage des DSI, RSSI et administrateurs*. Editions

[10] BENYAHIA, S. (2017). *Vers un nouveau Système de détection d'intrusion hybride et hiérarchique basé sur les réseaux bayésiens* (Doctoral dissertation, Université Ibn Khaldoun-Tiaret-).

[11][CHIKOUCHE, S. (2012). *Système de détection d'intrusion basé sur la classification comportementale des processus* (Doctoral dissertation, Faculté des Mathématiques et de l'Informatique-UNIVERSITE MOHAMED BOUDIAF DE M'SILA).

[12] DJAAD, M., & DAHMANI, C. (2022). *Vers Un Système De Détection D'intrusion Basée Sur Un Skyline De Classifieurs Dans Un Environnement Iots (L'internet Des Objets)* (Doctoral dissertation, Université Ibn Khaldoun-Tiaret-).

[13] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, *16*(4), 1996-2018.

[14] Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

[15] Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.

[16] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273-297.

[117 Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

[19] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks : Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, *16*(4), 1996-2018.

[20] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks : Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, *16*(4), 1996-2018.

[21] Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

[22] Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York : springer.

[23] Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., & Sheth, A. P. (2018). Machine learning for Internet of Things data analysis: A survey. *Digital Communications and Networks*, *4*(3), 161-175.

[25] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1-58.

[26] Chandola, V., & Banerjee, A. V., K.(2009). Anomaly detection: A survey. *ACM Computing survey*, *41*.

[27] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273-297.

[28] Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

[29] Mitchell, T. M. (1997). Machine Learning McGraw-Hill International.

[30] Carine, N. M. A., Marc, Y. T., Mathunaise, S. V., Vincent, A. T., Germain, A. M., & Patrice, J. J. Dynamique d'occupation du sol du bassin versant de la volta, par la méthode de l'arbre de décision, à partir des images multispectrales de la génération Landsat de 1990 à 2020.

[31] Mitchell, T. M. (2007). *Machine learning* (Vol. 1). New York: McGraw-hill.

B. Références Web (Techniques)

[70] (Une anomalie Non, une information, 2012) https://www.dell.com/fr-fr/blog/une-anomalie-non-une-information/

Résumé

L'Internet des objets (IoT) est en train de révolutionner notre monde. Son concept principal consiste à connecter des objets physiques à Internet. Cependant, avec l'augmentation du nombre de dispositifs connectés et la croissance de l'IoT, de nouvelles menaces pour la sécurité des réseaux apparaissent en raison des vulnérabilités présentes dans ces dispositifs. Les botnets malveillants représentent une menace très répandue, utilisant les dispositifs IoT vulnérables pour mener des attaques informatiques. Face à ces menaces, il devient nécessaire de développer de nouvelles méthodes de détection des réseaux de botnets IoT.

Dans cette étude, nous proposons de mettre en place un système de détection d'intrusion spécifiquement conçu pour l'Internet des objets, en utilisant une approche d'apprentissage automatique pour identifier le trafic réseau malveillant des botnets. Notre modèle se base sur des classifieurs. Afin d'analyser le comportement des périphériques dans le réseau, nous avons utilisé l'ensemble de données iot-network-intrusion.

En raison de la taille considérable de cet ensemble de données, nous avons sélectionné neuf scénarios, un pour chaque type de botnet, ainsi que des scénarios représentant des appareils IoT non infectés. Nous avons évalué cinq algorithmes. Les résultats ont démontré que l'algorithme RF a obtenu les meilleurs résultats.

Mots clés : Apprentissage automatique, Détection d’anomalies, Internet des objets, système de détection d'intrusion, IDS

Abstract

Paragraph The Internet of Things (IoT) is revolutionizing our world. Its main concept is to connect physical objects to the internet. However, with the increasing number of connected devices and the growth of IoT, new threats to network security are emerging due to vulnerabilities present in these devices. Malicious botnets pose a widespread threat, using vulnerable IoT devices to carry out cyber-attacks. In response to these threats, it is necessary to develop new methods for detecting IoT botnet networks.

In this study, we propose the implementation of an intrusion detection system specifically designed for the Internet of Things, using a machine learning approach to identify malicious botnet network traffic. Our model is based on classifiers. To analyze the behavior of devices in the network, we used the iot-network-intrusion dataset.

Due to the significant size of this dataset, we selected nine scenarios, one for each type of botnet, as well as scenarios representing non-infected IoT devices. We evaluated five algorithms. The results demonstrated that the RF algorithm achieved the best performance,

Keywords: Machine Learning, Anomaly Detection, Internet of Things, Intrusion Detection System, IDS.…

Keywords: Anomaly Detection, Intrusion Detection System, Internet of Things, Machine Learning

ملخّص

أحدثت إنترنت الأشياء (IoT) ثورة في عالمنا. مفهومها الرئيسي هو توصيل الأشياء المادية بالإنترنت. ومع ذلك ، مع زيادة عدد الأجهزة المتصلة ونمو إنترنت الأشياء ، تظهر تهديدات جديدة لأمن الشبكة بسبب نقاط الضعف الموجودة في هذه الأجهزة. تمثل شبكات الروبوت الخبيثة تهديدًا واسع النطاق ، حيث تستخدم أجهزة إنترنت الأشياء الضعيفة لتنفيذ هجمات على الكمبيوتر. في مواجهة هذه التهديدات ، يصبح من الضروري تطوير طرق جديدة لاكتشاف شبكات الروبوتات الخاصة بإنترنت الأشياء.

في هذه الدراسة ، نقترح تنفيذ نظام كشف التسلل مصمم خصيصًا لإنترنت الأشياء ، باستخدام نهج التعلم الآلي لتحديد حركة مرور الشبكة الضارة من شبكات الروبوت. يعتمد نموذجنا على المصنفات. من أجل تحليل سلوك الأجهزة في الشبكة، استخدمنا مجموعة بيانات اختراق شبكة iot.

نظرًا للحجم الهائل لمجموعة البيانات هذه ، اخترنا تسعة سيناريوهات ، واحد لكل نوع من أنواع الروبوتات ، بالإضافة إلى سيناريوهات تمثل أجهزة إنترنت الأشياء غير المصابة. قمنا بتقييم خمس خوارزميات. أظهرت النتائج أن خوارزمية التردد الراديوي كانت أفضل أداء ،

الكلمات المفتاحية: تعلم الآلة ، كشف الشذوذ ، إنترنت الأشياء ، نظام كشف التسلل ، IDS