RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ DE 8 MAI 1945 - GUELMA -

FACULTÉ DES MATHÉMATIQUES, D'INFORMATIQUE ET DES SCIENCES DE LA MATIÈRE

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Systèmes Informatiques

Thème

Detection des intrusions basée sur l'apprentissage automatique dans les systèmes IdO (Internet des Objets)

Encadré Par :

DR. CHOHRA Chemseddine

Présenté par :

BOUKERTOUTA Mohammed Amin

Juin 2022

Remerciements

Alhamdoulillah qui m'a facilité mon périple, et qui m'a fait sortir des ténèbres de l'ignorance à la lumière de la science et de la connaissance. Alhamdoulillah que j'ai pu finalement arriver ici, et que j'ai pu aboutir à un de mes objectifs dans la vie. Arriver ici n'était pas vraiment facile pour moi, mais grâce à l'aide des bonnes personnes que j'ai la chance de les avoir dans ma vie, je suis finalement ici.

Je commence par remercier sincèrement mon encadreur Dr. Chemseddine CHOHRA pour ses efforts géants. Ce jeune docteur me motivait, m'encourageait, et même influençait ma personne avec ses pensées et ses principes. Je lui remercie infiniment.

Mon encadreur n'était pas le seul à m'aider dans mon parcours, j'avais de la chance d'avoir des enseignants respectés, certains d'entre eux m'ont même donné plus que la simple information, leurs conseils étaient éclairants à plusieurs stages de mon parcours, et je suis certain que ces conseils vont encore m'aider dans ma vie poste-universitaire.

Je tiens à remercier aussi le staff administratif, pour tous qu'ils font pour garder la forteresse du département.

Le meilleur pour la fin ! Ma mère la raison de mon existence, ma flamme inextinguible, ma source qui ne draine jamais. Je lui dédie mon succès et tous mes accomplissements. Soyez fière.

RÉSUMÉ

Le système de détection d'intrusion (IDS) est défini comme un dispositif ou une application logicielle qui surveille les activités du réseau ou du système et détecte toute activité malveillante. La croissance exceptionnelle et l'utilisation d'Internet soulèvent des préoccupations sur la façon de communiquer et de protéger les informations numériques. Dans le monde d'aujourd'hui, les pirates utilisent différents types d'attaques pour obtenir des informations précieuses. De nombreuses techniques, méthodes et algorithmes de détection d'intrusion aident à détecter ces différentes attaques. L'objectif de ce travail est d'utiliser les algorithmes d'apprentissage automatique dans le but de créer un système de détection d'intrusion dans les environnement IdO (Internet des Objets). Nous étudions plusieurs algorithmes pour conclure nos résultats.

Mots Clé : système de détection d'intrusion (IDS), L'apprentissage profond (DL), L'apprentissage automatique (ML), Classification, Régression.

ABSTRACT

Intrusion Detection System (IDS) is defined as a tool or software application which monitors the network or system activities and finds if there is any malicious activity. Outstanding growth and usage of internet raises concerns about how to communicate and protect the digital information safely. In today's world hackers use different types of attacks for getting the valuable information. Many of the intrusion detection techniques, methods and algorithms help to detect those several attacks. The aim of this paper is to provide a complete study about the intrusion detection, types of intrusion detection methods, types of attacks, different tools and techniques to protect an IoT (Internet of Things) system from intrusion. We focus on using machine learning algorithms to detect intrusion, several algorithms have been studied to reach our conclusion.

Key Words : Intrusion Detection System (IDS), Deep Learning (DL), Machine Learning (ML), Classification, Regression

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

Avec l'évolution des réseaux et en particulier les réseaux internet, les techniques de l'information et de communication nous offrent actuellement des facilites incontournables en matière de l'apprentissage a distance.

L'Internet des objets (IoT) est un sujet important dans les milieux de l'industrie technologique, de la politique et de l'ingénierie. et fait la une de la presse spécialisée et des médias populaires. Cette technologie s'incarne dans un large éventail de produits, de systèmes et de capteurs en réseau, qui tirent parti des avancées en matière de de la puissance de calcul, de la miniaturisation de l'électronique et de l'interconnexion des réseaux pour offrir des capacités inédites auparavant impossibles. Une multitude de conférences, de rapports et d'articles de presse discutent et débattent de l'impact potentiel de la "révolution de l'IdO". Certains observateurs voient dans l'IdO un monde "intelligent" révolutionnaire, entièrement interconnecté, porteur de progrès, d'efficacité et d'opportunités, et susceptible d'ajouter des milliards de dollars de valeur à l'industrie et à l'économie mondiale. D'autres avertissent que l'IdO représente un monde plus sombre de surveillance, de violations de la vie privée et de la sécurité, et de verrouillage des consommateurs. Les gros titres qui attirent l'attention sur le piratage des voitures connectées à l'internet, la surveillance des préoccupations liées aux fonctions de reconnaissance vocale des téléviseurs "intelligents", et les craintes pour la vie privée liées à l'utilisation potentiellement abusive des données de l'IdO ont capté

l'attention du public. Ce débat sur le thème "promesses et dangers", ainsi que l'afflux d'informations par le biais des médias populaires et du marketing, peuvent faire de l'IdO un sujet complexe à comprendre.

Le taux croissant de cyber-attaques sur les réseaux de systèmes ces dernières années exacerbe la confidentialité et la sécurité des infrastructures informatiques et des ordinateurs personnels. Les systèmes de détection et de prévention des intrusions sont en train de devenir une partie importante des réseaux informatiques et de la cybersécurité. Diverses techniques sont proposées par des particuliers pour atténuer ce problème. Cela soulève un autre problème, celui de savoir quelle technique doit être utilisée pour un scénario donné. Dans cette étude, nous apportons une solution au problème mentionné et présentons une solution basé sur l'apprentissage automatique pour la détection d'intrusion dans les système IdO. Nous allons étudier plusieurs algorithmes d'apprentissage automatique afin d'atteindre les meilleurs performances et précision possible. Ce mémoire se compose de quatre chapitres :

- **Chapitre 01 :** nous ici une vision générale sur l'internet des objets, les protocoles utilisés et les problème de sécurité rencontrés dans ce contexte avec quelques solutions possible.
- **Chapitre 02 :** ce chapitre est consacré à la présentation des différentes techniques et algorithmes d'apprentissage automatique.
- **Chapitre 03 :** dans ce chapitre nous allons expliquer en détails notre méthodologie de travail.
- **Chapitre 04 :** est dans ce dernier chapitre nous allons aborder les détails techniques de notre travail, et nous allons aussi présenter et discuter les résultats obtenus.

CHAPITRE 1

SÉCURITÉ DANS LES ENVIRONNEMENT IDO

1.1 Introduction

Comment serait le monde sans Internet ? Il est difficile d'imaginer un tel scénario que nous n'avons jamais vu. Aujourd'hui, l'internet devient de plus en plus important pour tout le monde, tant dans la vie personnelle que dans la vie professionnelle. Différents appareils tels que les téléphones intelligents, les capteurs, les ordinateurs mobiles et d'autres objets intelligents sont des exemples de choses que nous utilisons quotidiennement. Ces technologies et d'autres technologies liées à l'IdO (Internet des Objets) ont une incidence considérable sur les nouvelles technologies des TIC (Technologie de l'information et de la communication) et des systèmes d'entreprise [14]. Au début, il était connu sous le nom d'"Internet des ordinateurs", puis il est devenu l'"Internet des gens" et, récemment, avec le développement rapide des TIC, il est devenu l'"Internet des objets". Dans l'IdO, différents dispositifs et objets intelligents sont inclus pour étendre l'internet et devenir accessibles et identifiés de manière unique. La connectivité est améliorée, passant de "n'importe quand, n'importe où" pour "n'importe qui" à "n'importe quand, n'importe où" pour "n'importe quoi" [12]. Dans le cadre des innovations en matière de TIC et de l'évolution de l'économie, l'accent a

été mis sur les technologies liées à l'IdO, qui sont largement considérées comme l'une des infrastructures les plus importantes pour leur promotion et l'une des stratégies prometteuses pour l'avenir. L'objectif principal est de permettre l'interaction et l'intégration du monde physique et du cyberespace [36]. L'IdO est considéré comme un pilier de l'internet du futur et devrait permettre des opérations intelligentes et des communications avancées de dispositifs, d'objets intelligents, de systèmes et de services. En effet, il s'agit d'une nouvelle révolution dans la technologie de l'information et de la communication qui signifie que tout, du pneu à la brosse à cheveux, se verra attribuer un identifiant unique afin de pouvoir être adressé, connecté à d'autres objets et échanger des informations.

1.2 Internet des objets

L'internet des objets (IdO) est un réseau intelligent qui connecte tous les objets à l'internet dans le but d'échanger des informations selon des protocoles convenus [10]. Ainsi, n'importe qui peut accéder à n'importe quoi, à tout moment et de n'importe où [5]. Dans le réseau Ido, les choses ou les objets sont connectés sans fil avec de minuscules capteurs intelligents. Les dispositifs IdO peuvent interagir entre eux sans intervention humaine [34].

L'IdO utilise des schémas d'adressage uniques pour interagir avec d'autres objets ou choses et coopérer avec les objets pour créer de nouvelles applications ou de nouveaux services. L'Ido introduit diverses applications telles que les maisons intelligentes, les villes intelligentes, la surveillance de la santé, l'environnement intelligent et l'eau intelligente [51]. Avec le développement des applications IdO soulève de nombreuses questions. Parmi les nombreuses autres questions, celle de la sécurité de l'IdO ne peut être ignorée. Les dispositifs IdO sont accessibles de n'importe où via un réseau non fiable comme l'internet. Les réseaux IdO ne sont donc pas protégés contre un large éventail d'attaques malveillantes.

Si les problèmes de sécurité ne sont pas résolus, les informations confidentielles peuvent être divulguées à tout moment. Le problème de la sécurité doit donc être résolu.

1.2.1 La confidentialité

Un attaquant peut facilement intercepter le message passant de l'expéditeur au destinataire, de sorte que la confidentialité peut être divulguée et le contenu modifié. la confidentialité peut être divulguée et le contenu peut être modifié [42]. Le passage de messages sécurisés est donc nécessaire dans l'IdO.

1.2.2 Intégrité

Le message ne doit pas être altéré 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 [42].

1.2.3 Disponibilité

Les données ou les ressources doivent être disponibles au moment voulu [42]. Les attaquants peuvent inonder la bande passante des ressources pour nuire à la disponibilité. ressources pour nuire à la disponibilité. La disponibilité peut être endommagée par des attaques malveillantes telles que le déni de service (DOS), l'inondation, le trou noir, le brouillage $\cdots$ etc.

1.2.4 Authenticité

L'authenticité implique la preuve de l'identité. Les utilisateurs doivent être capables d'identifier l'identité des autres avec lesquels ils interagissent. Elle peut être vérifiée par le processus d'authentification afin que l'entité non autorisée ne puisse pas participer à la communication [29].

1.2.5 Non-répudiation

La non-répudiation garantit que l'expéditeur et le destinataire ne peuvent pas nier avoir envoyé et reçu le message respectivement [11].

1.2.6 Fraîcheur des données

Les données doivent être récentes chaque fois que cela est nécessaire. Elle garantit qu'aucun ancien message n'est rejoué par un adversaire.

1.3 Les Environnements d'utilisation

Les progrès de l'IdO motivent l'adoption de plus en plus d'applications de cette technologie innovante. Les applications de l'IdO ont de plus en plus envahi les industries et les organisations des secteurs publics et privés, ce qui nous permet d'économiser du temps, des ressources et des efforts. Les applications de l'IdO ont été classées dans la littérature sur la base de différents critères et facteurs de classification. On distingue trois grands domaines d'applications de l'IdO : l'industrie, l'environnement et la société (voir tableau 1.1). Ces domaines sont cohésivement liés et interdépendants les uns des autres et ne peuvent pas être isolés. Au sein de chaque grand domaine, de plus en plus d'applications peuvent être identifiées. Les exigences de base de ces applications dans ces domaines sont souvent les mêmes, avec une différence marginale selon la fonctionnalité principale de l'application. Dans cette section, nous examinons brièvement certaines des applications courantes et largement utilisées de l'IdO.

Les systèmes de surveillance et de contrôle sont des applications courantes de l'IdO. Les données relatives à l'environnement ou aux objets en réseau sont collectées (détectées ou calculées), puis envoyées à un système intelligent (centralisé ou non), et une bonne décision est alors prise. Cela permet de suivre en permanence le comportement de travail, de reconfigurer les paramètres de fonctionnement et donc d'ajuster automatiquement les performances du système. Les WSN (Wireless Sensor

Networks) ont été adoptés très tôt dans de tels scénarios et ont toujours été une technologie principale dans les systèmes de sécurité et de contrôle climatique. Aujourd'hui, les WSN basés sur IP sont identifiés comme un sous-réseau des réseaux IdO offrant une flexibilité, une interaction et une dynamique accrues aux applications de surveillance environnementale [20]. Cela inclut la mesure des phénomènes naturels tels que le vent, les tempêtes, les précipitations, la température, la pollution, la hauteur des rivières $\cdots$ etc, le suivi des objets mobiles en temps réel, partout et à tout moment [32], La surveillance et la sécurité dans différents scénarios tels que les maisons, les marchés, les centres commerciaux, les entreprises $\cdots$ etc sont également adoptées de manière notable grâce aux technologies WSN basées sur l'IdO.

Dans le domaine commercial, les systèmes intelligents sont utilisés pour découvrir et résoudre les problèmes commerciaux afin d'apporter une réponse appropriée et d'obtenir la satisfaction du client [32]. Dans la chaîne d'approvisionnement/de livraison et la logistique de distribution, par exemple, le suivi des articles et des denrées périssables est l'une des applications courantes de la technologie RFID. Grâce aux informations recueillies, l'acheteur ou le fournisseur à distance est en mesure de surveiller en permanence l'état et le mouvement des marchandises, par exemple leur emplacement actuel, leur quantité, les conditions environnementales et le moment prévu de leur disponibilité sur le marché [39]. Ainsi, les clients ont automatiquement accès à ces informations. Dans l'industrie manufacturière, les usines intelligentes sont devenues proéminentes et contribuent à améliorer le processus de production. Dans ces systèmes, l'intelligence est intégrée dans les machines et les équipements. Ainsi, ils sont capables d'améliorer leurs performances grâce à des capacités d'autogestion. En outre, ces composants sont reliés entre eux par un système de coordination et de contrôle robuste. L'intervention humaine est largement réduite, ce qui se traduit par un certain nombre d'avantages clés tels que des délais de production/livraison plus rapides, des coûts moindres, une qualité améliorée et des environnements de travail plus sûrs [52].

Domaine	Description	Exemples indicatifs
Industrie	Activités impliquant des transactions financières ou commerciales entre entreprises, organisations et autres entités	Industrie manufacturière, logistique, secteur des services, banques, autorités gouvernementales financières, intermédiaires, etc.
Environnement	Activités concernant la protection, la surveillance et le développement de toutes les ressources naturelles	Agriculture et élevage, recyclage, services de gestion environnementale, gestion de l'énergie, etc.
Société	Activités/initiatives concernant le développement et l'inclusion des sociétés, des villes et des personnes	Services gouvernementaux envers les citoyens et les autres structures de la société (eparticipation), einclusion (par exemple, les personnes âgées, les handicapés), etc.

TABLE 1.1 – Domaines d'application de l'IdO - Description et exemples [52]

1.3.1 Ville intelligente

Une ville intelligente va au-delà de l'utilisation des technologies numériques pour une meilleure utilisation des ressources et une réduction des émissions. Elle implique des réseaux de transport urbain plus intelligents, une amélioration de l'approvisionnement en eau et des installations d'élimination des déchets, ainsi que des moyens plus efficaces d'éclairer et de chauffer les bâtiments. Cela signifie également une administration municipale plus interactive et plus réactive, des espaces publics plus sûrs et la satisfaction des besoins d'une population vieillissante.

1.3.2 Maison connectée

Il est intéressant de noter que la première génération de maisons intelligentes n'avait pas grand-chose à voir avec l'intelligence et concernait plutôt le contrôle à distance et l'automatisation. Il y a dix ans, un espace futuriste où vous pouviez actionner des stores depuis votre smartphone ou apprendre à votre thermostat à se souvenir de la température que vous préférez était suffisant pour l'appeler une maison intelligente. En 2021, ce concept implique bien plus que cela.

Qu'est-ce que la domotique à l'aide de l'IdO aujourd'hui ? Aujourd'hui, une maison intelligente est à la hauteur des attentes du consommateur, et parfois même les dépasse. Grâce à des capteurs, les dispositifs, les appareils et l'ensemble des espaces de votre maison collectent en permanence des données sur la façon dont vous les utilisez. Ils apprennent à connaître vos habitudes et déterminent des modèles de consommation à l'aide d'algorithmes complexes. Ces informations permettent ensuite de personnaliser votre expérience à un niveau granulaire. Grâce aux solutions IdO pour les maisons intelligentes, nous obtenons un tout nouveau niveau de contrôle sur notre foyer. Nous pouvons non seulement allumer et éteindre nos appareils à distance, mais aussi contrôler l'ensemble des fonctionnalités sur des applications mobiles ou web.

Les systèmes intelligents peuvent vous donner une image de la façon dont les choses fonctionnent dans votre foyer. Les appareils connectés collectent des données,

les traitent et fournissent des informations utiles tel que : la quantité d'électricité consommée par chaque appareil, les dépenses liées aux services publics, l'humidité et les autres conditions de qualité de l'air dans votre espace $\cdots$ etc.

1.4 Protocoles de communication

Dans l'IdO, des milliards de dispositifs et d'objets intelligents, ayant des capacités différentes, nécessitent un moyen d'échanger et de transmettre les informations collectées ou générées au niveau du dispositif. Cependant, les dispositifs IdO sont censés être connectés entre eux et capables de communiquer d'une manière ou d'une autre. Un objet IdO doit pouvoir communiquer avec d'autres dispositifs : identifier le chemin approprié vers la destination, comprendre les messages reçus et, par conséquent, répondre de manière appropriée. Ainsi, les protocoles standard deviennent des exigences essentielles pour le monde de l'IdO. Il est ainsi facile d'obtenir la pleine fonctionnalité de ces dispositifs limités tout en maintenant le niveau souhaité de performance du réseau. La mobilité dans l'IdO est l'un des principaux problèmes. Un appareil mobile se déplace fréquemment d'un endroit à l'autre. Dans la plupart des cas, il doit être transféré du point d'attache actuel à un autre. Les protocoles de communication doivent être conscients de cette nature dans la majorité des dispositifs IdO [4]. Des mécanismes intelligents sont nécessaires pour assurer un transfert transparent et réduire le délai imposé aux différentes couches

1.4.1 IEEE 802.15.4

L'IEEE 802.15.4 est une norme conçue par le groupe de travail IEEE 802.15 de l'IETF qui définit les couches physique (PHY) et de contrôle d'accès au support (MAC) pour les réseaux personnels sans fil à faible débit de données, à faible puissance et à courte portée (LR-WPAN) [3]. La version originale est fournie en 2003 et prend en charge des débits de données de 20, 40 et 250 kb/s avec une portée de 10 mètres pour une communication omniprésente entre les appareils. Par la suite, la norme EEE

802.15.4a/c/d a été améliorée pour étendre la couche PHY avec plusieurs bandes de fréquences et techniques de transmission supplémentaires. La norme IEEE 802.15.4-2011, une révision des amendements précédents, est fournie pour les réunir en une seule norme prenant en charge un débit de données maximal de 850 kb/s et mettant l'accent sur les exigences techniques d'interopérabilité. Plus tard, un certain nombre d'amendements ont été introduits tels que IEEE 802.15.4e, IEEE 802.15.4f et IEEE 802.15.4g. La norme IEEE 802.15.4e a été publiée afin d'améliorer et d'ajouter des fonctionnalités à la sous-couche MAC. Une stratégie de saut de canal est adoptée pour renforcer la prise en charge des marchés industriels et améliorer la robustesse pour surmonter l'évanouissement par trajets multiples et les interférences externes. Dans la norme IEEE 802.15.4f, le PHY est amélioré pour prendre en charge la flexibilité et de meilleures performances dans les déploiements très denses de dispositifs autonomes et de systèmes RFID actifs partout dans le monde. Cet amendement prend en charge une large gamme d'applications caractérisées par plusieurs contraintes telles que le faible coût, la faible consommation d'énergie, la durée de vie de la batterie sur plusieurs années, la fiabilité des communications, la précision de la localisation et les options de lecture [3]. La norme IEEE 802.15.4g prend en charge les exigences des réseaux sans fil intelligents à faible débit en extérieur et offre une plus grande portée de transmission égale à 1 km et une grande taille de paquet de 2047 octets [3].

1.4.2 6LoWPAN

L'IPv6 sur les réseaux personnels sans fil à faible puissance (6LoWPAN) est une norme pour la couche d'adaptation permettant l'envoi et la réception de paquets IPv6 sur les liaisons IEEE 802.15.4 [31]. Elle concrétise l'idée d'appliquer le protocole Internet aux petits dispositifs autonomes, comme seule solution disponible pour les réseaux d'objets intelligents ou LLN. Ainsi, ces dispositifs limités peuvent être connectés en très grand nombre à l'Internet [3]. En outre, 6LoWPAN prend en charge la mobilité, les dispositifs étant tout au plus déployés de manière ad hoc, sans emplacement

prédéfini, et se déplaçant en permanence. Pour la mise en correspondance du réseau IPv6 avec celui de l'IEEE 802.15.4, 6LoWPAN remplit trois fonctions essentielles :

1. La compression de l'en-tête IPv6.
2. la fragmentation des paquets IPv6.
3. le transfert de couche 2 [3].

Pour chacun d'eux, un en-tête 6LoWPAN distinct est inclus si nécessaire. Dans le premier cas, l'en-tête IPv6 est compressé, les champs qui peuvent être obtenus à partir du contexte étant omis et les autres étant envoyés sans modification. Dans la deuxième, les paquets plus grands que la MTU de l'IEEE 802.15.4 sont fragmentés à l'expéditeur et réassemblés à la destination. Dans la troisième fonction, appelée "mesh-under" et adaptée aux petits réseaux locaux, le routage IP n'est pas effectué. Les paquets sont transmis à la destination par la couche d'adaptation sur de multiples sauts radio. Ce routage est effectué au niveau de la couche liaison en fonction de l'en-tête 6LoWPAN et de la trame IEEE 802.15.4 [3, 22].

1.4.3 RPL (Routing Protocol for LLN)

Le protocole de routage pour LLNs (Low-power and lossy networks) est un protocole de couche réseau conçu pour les réseaux à faible puissance et à pertes [47]. RPL a été développé dans le but de répondre aux exigences spécifiques des applications pour les LLNs identifiées par le groupe de travail ROLL (Routing Over LLNs) de l'IETF (Internet Engineering Task Force). Ces exigences sont définies pour, mais sans s'y limiter, un ensemble de domaines d'application : l'industrie, l'automatisation des bâtiments, la domotique et les réseaux de capteurs urbains. Selon l'évaluation, ROLL a constaté que les protocoles existants ne satisfont pas toutes les exigences spécifiées. Ces protocoles, qui n'utilisent que des métriques de liens statiques, ne prennent pas en compte les états des dispositifs tels que les ressources de traitement, la mémoire, l'énergie résiduelle ou les défaillances matérielles lors de la création du meilleur/plus court chemin. Le RPL est un protocole de vecteur de distance IPv6

proactif extensible qui prend en charge les environnements de routage maillé, le routage basé sur les contraintes du plus court chemin (sur les liens et les nœuds) et différents modèles de trafic, notamment MP2P, P2MP et P2P. Il prend en compte les objectifs d'optimisation du routage indépendamment du traitement et de la transmission des paquets et peut être exécuté sur différentes couches de liaison. Cela inclut les couches de liaison contraintes ou celles utilisées en conjonction avec des dispositifs très contraints tels que, mais sans s'y limiter, les technologies WPAN (802.15.4) ou PLC (Power Line Communication) à faible puissance [47]. En outre, la RPL comprend des mesures d'économie d'énergie telles que l'adaptation du taux d'envoi des messages de contrôle et la mise à jour de la topologie uniquement lorsque des paquets de données doivent être envoyés.Sur un réseau, plus d'une instance de RPL peut être exécutée simultanément. Chacune de ces instances peut considérer un ensemble de contraintes ou d'objectifs d'optimisation différents et potentiellement antagonistes [47].

1.4.4 D. CoAP

Le protocole d'application contraint (CoAP) est un protocole de couche d'application basé sur le Web conçu par le groupe de travail Constrained RESTful Environments (CoRE) de l'IETF [33]. Il offre des communications M2M (Machine to Machine) interactives pour les dispositifs autonomes et les objets intelligents via l'Internet standard. Il est destiné à être utilisé dans les réseaux à faible puissance et contraints tels que LLNs/IoT et 6LoWPAN qui nécessitent une surveillance et une manipulation à distance. CoAP est une version allégée de HTTP qui prend en charge la simplicité, la faible surcharge des messages, la complexité réduite de l'analyse syntaxique et le besoin limité de fragmentation des paquets dans ces environnements et dispositifs contraints. De plus, il s'agit d'une plateforme qui fournit un modèle d'interaction demande/réponse entre les applications et qui facilite l'intégration des réseaux embarqués avec le Web existant [49, 57]. De plus, il possède plus de fonctionnalités pour

le M2M telles que la découverte intégrée, le support du mode proxy, le support multicast, la livraison fiable et les échanges de messages asynchrones [3, 49]. Les paquets dans le CoAP sont beaucoup plus petits, plus simples à générer et plus faciles à analyser, avec moins de mémoire utilisée. CoAP est basé sur les datagrammes et fonctionne sur UDP, et non sur TCP. Cependant, il peut être utilisé au-dessus des SMS et d'autres protocoles de communication par paquets .

1.5 Sécurité

Un aspect très souvent exposé et discuté dans les objets connectés, et plus généralement dans l'Internet des Objets, est celui de la sécurité. Beaucoup de travaux scientifiques insistent sur le manque de sécurité de ces nombreux objets. Au vu des caractéristiques présentées précédemment ainsi que des réflexions autour des environnements connectés, nous jugeons également qu'un certain nombre de problématiques de sécurité peuvent se poser vis-à-vis de l'intégration d'objets dans un environnement. Cette section présente donc les éléments nécessaires pour comprendre ce qu'est la sécurité dans le contexte des environnement IdO [46].

1.5.1 Sûreté de fonctionnement

La sûreté de fonctionnement d'un système informatique est définie comme "la propriété qui permet aux utilisateurs d'un système de placer une confiance justifiée dans le service qu'il leur délivre". Le service délivré correspond au comportement du système perçu par ses utilisateurs. La non-sûreté de fonctionnement correspond quant à elle à une perte de confiance, qui ne peut plus ou ne pourra plus être placée dans le service délivré. Les causes ou résultats de celle-ci sont les circonstances indésirables définies comme étant les entraves [46] :

— Une défaillance survient lorsque le service délivré dévie de l'accomplissement de la fonction du système.

— Une erreur est la partie de l'état du système qui est susceptible d'entraîner une défaillance.
— Une faute est la cause adjugée ou supposée d'une erreur.

1.5.2 Malveillances

Une malveillance est une faute qui est intentionnellement nuisible, c'est-à-dire créée ou commise délibérément pour nuire au fonctionnement normal du système. Les malveillances peuvent être divisées en plusieurs classes, néanmoins, nous nous concentrons ici sur les intrusions qui sont associées à deux causes [46] :

— Un acte malveillant ou une attaque essayant d'exploiter une faiblesse du système. Une attaque étant une faute d'interaction, dont le but est de violer un ou plusieurs des attributs de sécurité.
— Une faiblesse ou une vulnérabilité placée dans les exigences, la spécification, la conception ou la configuration du système, ou dans la manière dont il est utilisé. Une vulnérabilité étant une faute accidentelle, ou une faute intentionnellement malveillante ou non malveillante.

L'intrusion se définit donc comme étant une faute externe nuisible qui résulte d'une attaque ayant réussi à exploiter une vulnérabilité.

1.6 Attaques visant les environnements IdO

Les dispositifs IdO présentent de nombreuses vulnérabilités. Comme il est simple et facile d'effectuer des cyberattaques contre les dispositifs IdO, les pirates les exécutent souvent afin de pouvoir capturer des informations sensibles. La plupart des menaces de sécurité de l'IdO peuvent entraîner une fuite d'informations et/ou une perte de services. Ces risques de sécurité peuvent également présenter des risques de sécurité physique qui peuvent être nuisibles pour les personnes.

Les failles de sécurité de l'IdO peuvent ouvrir la voie à de nombreux pirates malveillants qui souhaitent exploiter les faiblesses des systèmes IdO pour accéder à nos

informations personnelles pour leur propre bénéfice. Cette section aborde les différents types d'attaques de cybersécurité contre les systèmes IdO. Le tableau 2 présente les différents types d'attaques de cybersécurité contre les dispositifs IdO.

1.6.1 Attaques physiques contre les dispositifs IdO

L'IdO étant distribué et fragmenté par nature, il présente une surface d'attaque plus importante pour un accès physique aux dispositifs. Un pirate peut être en mesure de modifier un nœud ou les données d'un capteur, ce qui peut mettre en danger l'ensemble du réseau de capteurs. Les attaques physiques sont liées aux composants matériels des dispositifs IdO et l'adversaire doit accéder physiquement au système Ido pour exécuter son attaque [45]. Ces attaques peuvent nuire à la fonctionnalité du matériel Ido.

Falsification de nœuds

L'altération des nœuds est une attaque physique contre les dispositifs Ido qui peut endommager un nœud de capteur. Un adversaire remplacera physiquement le nœud entier ou une partie de celui-ci afin de pouvoir accéder et modifier des informations sensibles telles que les clés cryptographiques partagées [2].

Analyse des canaux latéraux

Un exemple d'attaque physique contre l'IdO est un pirate qui utilise l'analyse des canaux latéraux pour voler le secret de la norme de cryptage avancée (AES), les clés secrètes utilisées dans les lampadaires connectés. L'analyse des canaux latéraux est une attaque non invasive impliquant qu'un intrus observe la signature électrique ou le rayonnement électromagnétique émis par un circuit intégré afin d'extraire des informations sensibles telles que des clés secrètes [37]. Les lampadaires connectés utilisent le chiffrement AES afin de s'assurer que seuls les utilisateurs autorisés qui connaissent

Classification	Attaques de sécurité	Impacts sur la sécurité
Attaques physiques	Températion des nœuds, analyse des canaux latéraux, brouillage radioélectrique	Ces attaques permettront aux pirates de modifier les données d'un nœud ou d'un capteur et d'endommager physiquement le matériel de l'IdO.
Attaques de réseau	Attaque par analyse du trafic, transfert sélectif, attaque sybille, attaque trou d'eau, attaque botnet, attaque Hello-flood, attaque Man in the middle.	Ces attaques permettront aux pirates d'avoir un accès à distance et d'envoyer des instructions erronées pour prendre le contrôle des appareils IdO.
Attaques des applications	Injection de code, dépassement de tampon, injection SQL, détournement de session, attaques d'authentification et d'autorisation.	Ces attaques permettront aux pirates de voler des données sensibles en fournissant un accès non autorisé au niveau des applications de l'IdO.

TABLE 1.2 – Différentes attaques de cybersécurité contre les dispositifs IdO.

les clés secrètes peuvent transmettre les informations en toute sécurité. Si un adversaire peut voler ces clés secrètes, il sera en mesure de détourner le réseau de lampadaires. Pour exécuter ces attaques, un pirate doit se trouver à proximité de l'appareil. Ces types d'attaques peuvent également être utilisés pour compromettre la sécurité des cartes bancaires, des appareils mobiles ou des appareils médicaux.

1.6.2 Attaques réseau contre les dispositifs Ido

Ces attaques sont généralement exécutées au niveau du réseau de l'IdO. Les attaquants peuvent exécuter ces attaques à distance et il n'a pas besoin d'être proche du réseau [6].

Attaque par analyse de trafic

L'attaque par analyse du trafic est un type d'attaque de réseau où un adversaire peut intercepter et examiner les messages pour déduire des informations à partir de modèles de communication. Dans une attaque par analyse de trafic, un adversaire peut examiner la fréquence et le timing des paquets du réseau IdO pour obtenir des informations importantes. Par exemple, un attaquant peut tenter d'exécuter une attaque de synchronisation sur un dispositif IdO qui utilise SSH pour l'authentification. Il utilisera les informations de synchronisation pour déduire les mots de passe, car SSH transmet chaque frappe sous forme de message pendant la session interactive.

Attaque Sybil

Dans cette attaque, un nœud malveillant, connu sous le nom de nœud Sybil, peut usurper l'identité d'un grand nombre de nœuds, ce qui permet à l'attaquant de se trouver à plusieurs endroits à la fois.

Attaque de type "sinkhole"

Un attaquant peut réaliser une attaque de type sinkhole contre des dispositifs IdO pour attirer tout le trafic des nœuds voisins et prendre le contrôle d'un nœud à l'intérieur d'un réseau.

1.6.3 Attaques applicatives contre les dispositifs IdO

Les attaques d'applications permettent aux pirates de cibler les données sensibles des utilisateurs pour y accéder sans autorisation. Différents types de vulnérabilités applicatives, comme le dépassement de tampon ou l'injection de code, sont exploités par les adversaires afin d'obtenir un accès non autorisé à différentes applications IdO. Les attaquants peuvent violer la sécurité des applications IdO en raison d'une mauvaise configuration du code ou d'une API non sécurisée. En outre, les logiciels malveillants tels que les virus, les vers, les chevaux de Troie, les rootkits, les ransomwares $\cdots$ etc peuvent également cibler les applications exécutées sur les appareils IdO pour y accéder sans autorisation.

Injection de code

Cette attaque exploite les erreurs de programme pour introduire un code malveillant dans le système. Les adversaires peuvent utiliser l'attaque par injection de code pour voler des données sensibles aux utilisateurs, prendre le contrôle total d'un système ou diffuser des logiciels malveillants [9]. L'injection de shell et l'injection de script HTML sont les types les plus courants d'attaques par injection de code.

Détournement de session (Session Hijacking)

Cette attaque peut permettre à un pirate de révéler des informations personnelles sensibles des utilisateurs. Dans une attaque par détournement de session, un pirate exploite les failles de sécurité dans l'authentification et la gestion des sessions pour se faire passer pour un véritable utilisateur [33].

1.7 Solutions

L'objectif principal de l'atténuation de la sécurité est de garantir la vie privée, la confidentialité, la protection des utilisateurs, des infrastructures, des données et des dispositifs de l'IdO, ainsi que la disponibilité des services fournis par une infrastructure IdO. Pour renforcer la sécurité des dispositifs IdO, nous devons authentifier chaque dispositif communiquant sur un réseau, maintenir la confidentialité et l'intégrité des connexions entre les dispositifs, chiffrer les données et les stocker dans un endroit sécurisé. Cette section traite des solutions de sécurité que nous pouvons employer pour protéger les dispositifs IdO contre les attaques de sécurité. Le tableau 3 présente différents types de solutions de sécurité pour sécuriser les dispositifs IdO.

1.7.1 Authentification

L'authentification est la méthode d'identification et de vérification des utilisateurs et des dispositifs de l'IdO afin qu'elle puisse fournir un accès aux utilisateurs et aux dispositifs autorisés dans le réseau. L'IdO étant constitué d'un grand nombre de dispositifs interconnectés et distribués qui communiquent entre eux, l'authentification joue un rôle important dans la sécurité. Il est nécessaire de contrôler et d'authentifier correctement chaque dispositif IdO pour s'assurer qu'il est authentique et empêcher les dispositifs non autorisés d'accéder au réseau. L'authentification forte peut nous aider à atténuer plusieurs attaques de sécurité IdO telles que les attaques par écoute, les attaques par relecture, les attaques par l'homme du milieu, les attaques par dictionnaire et les attaques par force brute [21].

Une meilleure façon d'authentifier les dispositifs IdO est d'utiliser la cryptographie asymétrique ou à clé publique. La cryptographie à clé publique peut fournir une authentification forte pour l'IdO. L'algorithme de signature numérique à courbe elliptique (ECDSA) est un autre algorithme asymétrique qui peut être utilisé pour l'authentification. La cryptographie à clé publique nécessite l'accès à une infrastructure à

Classification	Solutions de sécurité
Authentification	Algorithme symétrique, cryptographie asymétrique ou à clé publique, sécurité de la couche transport (TLS), signature numérique..
Solutions de communication sécurisées	Réseau privé virtuel (VPN), fonctions de hachage cryptographiques, clé privée pré-partagée (PPSK), pare-feu et IDS/IPS, confidentialité des messages de bout en bout.
Sécurité des applications	Codage sécurisé, démarrage sécurisé, liste de contrôle d'accès (ACL), pare-feu et IDS, mises à jour logicielles sécurisées.

TABLE 1.3 – Solutions de sécurité pour sécuriser les dispositifs Ido

clé publique (PKI). Par conséquent, une mise en œuvre sécurisée de la PKI est également nécessaire pour la sécurité de l'IdO. La mise en œuvre sécurisée de la PKI pour l'authentification implique la génération d'une paire de clés sur la puce qui utilise un générateur de nombres aléatoires réels pour générer la clé, et l'exécution d'opérations cryptographiques telles que le cryptage, le décryptage, la signature et la vérification de la signature dans un environnement contrôlé.

1.7.2 Solutions de communication sécurisées

Les dispositifs IdO peuvent utiliser les technologies et protocoles internet existants qui peuvent leur fournir des solutions de communication sécurisées pour protéger les données des utilisateurs. Par exemple, les dispositifs IdO peuvent utiliser un réseau privé virtuel (VPN) basé sur des protocoles tels que : Secure Socket Layer/-Transport Layer Security (SSL/TLS), Media Access Control security (MACsec) ou Datagram Transport Layer Security (DTLS) pour chiffrer la connexion, ce qui offre une meilleure sécurité aux utilisateurs IdO. Les fonctions de hachage cryptographique

peuvent également être utilisées pour confirmer l'intégrité des données. Des mécanismes de vérification des erreurs peuvent être introduits pour atténuer le problème des données falsifiées [33].

Une autre méthode de sécurisation des communications entre les dispositifs IdO consiste à utiliser une clé privée pré-partagée (PPSK) [33]. Le domaine d'accès de chaque type d'appareil peut être facilement défini en fournissant différentes clés uniques. Des politiques de mots de passe forts et un changement périodique des mots de passe doivent être utilisés pour protéger la communication. En outre, des technologies telles que le pare-feu et le système de détection des intrusions/système de prévention des intrusions (IDS/IPS) doivent être utilisées pour empêcher les intrus d'accéder au réseau.

1.7.3 Sécurité des applications

Les applications IdO doivent être sécurisées en utilisant diverses techniques. Les dispositifs IdO doivent exécuter le code de l'application de manière sécurisée afin qu'il ne puisse pas être modifié ou corrompu et qu'il ne révèle pas de données sensibles. Le codage sécurisé est particulièrement important lors de l'utilisation de données sensibles comme les clés ou fonctions cryptographiques, les applications de paiement et les informations de santé.

Une liste de contrôle d'accès (ACL) doit être mise en place pour définir les politiques et les autorisations qui permettront de déterminer qui peut accéder à l'application IdO et la contrôler . Cela permettra également de garantir la confidentialité des données. L'ACL a la capacité d'autoriser et de bloquer les connexions entrantes et sortantes et de garantir que seuls les utilisateurs autorisés peuvent accéder au réseau.

Les systèmes de pare-feu et de détection des intrusions peuvent également être utilisés pour sécuriser les applications IdO. Le pare-feu dispose de jeux de règles pour autoriser les connexions autorisées et bloquer les connexions malveillantes. Le système de détection d'intrusion peut analyser les modèles de trafic pour détecter les actions malveillantes afin de pouvoir signaler une alarme lorsqu'une attaque est détectée.

1.7.4 Système de Prévention d'intrusion

Le système de prévention des intrusions est également connu sous le nom de système de détection et de prévention des intrusions. Il s'agit d'une application de sécurité réseau qui surveille les activités du réseau ou du système pour détecter les activités malveillantes. Les principales fonctions des systèmes de prévention des intrusions consistent à identifier les activités malveillantes, à recueillir des informations sur ces activités, à les signaler et à tenter de les bloquer ou de les arrêter.
Les systèmes de prévention des intrusions sont considérés comme une augmentation des systèmes de détection des intrusions (IDS).

Les IPS enregistrent généralement les informations relatives aux événements observés, notifient les administrateurs de sécurité des événements importants observés et produisent des rapports. De nombreux IPS peuvent également répondre à une menace détectée en essayant de l'empêcher de réussir. Ils utilisent diverses techniques de réponse, qui impliquent que l'IPS arrête l'attaque elle-même, modifie l'environnement de sécurité ou change le contenu de l'attaque.

1.7.5 Classification des systèmes de prévention des intrusions

Les systèmes de prévention des intrusions (IPS) peuvent être classés en 4 types [23].

Système de prévention des intrusions basé sur le réseau (NIPS)

Il surveille l'ensemble du réseau à la recherche de trafic suspect en analysant l'activité des protocoles.

Système de prévention des intrusions sans fil (WIPS)

Il surveille le trafic suspect d'un réseau sans fil en analysant les protocoles de réseau sans fil.

Analyse du comportement du réseau (NBA)

Elle examine le trafic réseau pour identifier les menaces qui génèrent des flux de trafic inhabituels, comme les attaques par déni de service distribué, les formes spécifiques de logiciels malveillants et les violations de politiques.

Système de prévention des intrusions basé sur l'hôte (HIPS)

Il s'agit d'un logiciel intégré qui exploite un seul hôte pour détecter toute activité douteuse en analysant les événements qui se produisent au sein de cet hôte.

1.7.6 Méthode de détection du système de prévention des intrusions (IPS)

Détection par signature

Les IDS basés sur les signatures exploitent les paquets dans le réseau et les comparent avec des modèles d'attaque préétablis et prédéfinis, appelés signatures [30].

Détection statistique d'anomalies

Les IDS basés sur les anomalies surveillent le trafic réseau et le comparent à une base de référence établie. La ligne de base permet d'identifier ce qui est normal pour ce réseau et quels protocoles sont utilisés. Toutefois, il peut déclencher une fausse alarme si les lignes de base ne sont pas configurées intelligemment [26].

Détection d'analyse de protocole avec état

Cette méthode IDS reconnaît la divergence des protocoles énoncés en comparant les événements observés avec des profils préétablis de définitions généralement acceptées de l'activité non nuisible [23].

1.7.7 Système de Détection d'intrusion

est un système qui surveille le trafic réseau à la recherche d'activités suspectes et émet des alertes lorsqu'une telle activité est découverte. Il s'agit d'une application logicielle qui scanne un réseau ou un système à la recherche d'une activité nuisible ou d'une violation de politique. Toute activité malveillante ou violation est normalement signalée soit à un administrateur, soit collectée de manière centralisée à l'aide d'un système de gestion des informations et des événements de sécurité (SIEM). Un système SIEM intègre les sorties de plusieurs sources et utilise des techniques de filtrage des alarmes pour différencier les activités malveillantes des fausses alarmes.

Bien que les systèmes de détection d'intrusion surveillent les réseaux pour détecter les activités potentiellement malveillantes, ils sont également disposés à recevoir de fausses alarmes. C'est pourquoi les organisations doivent régler avec précision leurs produits IDS dès leur première installation. Cela signifie qu'il faut configurer correctement les systèmes de détection d'intrusion pour qu'ils puissent reconnaître le trafic normal sur le réseau par rapport à l'activité malveillante.

Les IDS sont classés en 5 types

Système de détection des intrusions dans le réseau (NIDS)

Les systèmes de détection d'intrusion dans le réseau (NIDS) sont installés à un point planifié du réseau pour examiner le trafic de tous les appareils du réseau. Ils effectuent une observation du trafic passant sur l'ensemble du sous-réseau et font correspondre le trafic qui passe sur les sous-réseaux à la collection d'attaques connues. Lorsqu'une attaque est identifiée ou qu'un comportement anormal est observé, l'alerte peut être envoyée à l'administrateur. Un exemple de NIDS consiste à l'installer sur le sous-réseau où se trouvent les pare-feu afin de voir si quelqu'un essaie de percer le pare-feu [8].

Système de détection d'intrusion dans l'hôte (HIDS)

Les systèmes de détection d'intrusion sur l'hôte (HIDS) fonctionnent sur des hôtes ou des dispositifs indépendants sur le réseau. Un HIDS surveille les paquets entrants et sortants du périphérique uniquement et alerte l'administrateur si une activité suspecte ou malveillante est détectée. Il prend un instantané des fichiers système existants et le compare avec l'instantané précédent. Si les fichiers système analysés ont été modifiés ou supprimés, une alerte est envoyée à l'administrateur pour qu'il les enquête. Un exemple d'utilisation du HIDS peut être observé sur des machines critiques, qui ne sont pas censées modifier leur disposition [8].

Système de détection d'intrusion basé sur le protocole (PIDS)

Le système de détection d'intrusion basé sur le protocole (PIDS) comprend un système ou un agent qui résiderait constamment à l'extrémité frontale d'un serveur, contrôlant et interprétant le protocole entre un utilisateur/dispositif et le serveur. Il tente de sécuriser le serveur web en surveillant régulièrement le flux de protocole HTTPS et en acceptant le protocole HTTP correspondant. Comme le HTTPS n'est pas crypté et avant d'entrer instantanément dans sa couche de présentation web, ce système doit résider dans cette interface, entre l'utilisation du HTTPS [8].

Système de détection d'intrusion basé sur le protocole d'application (APIDS)

Le système de détection des intrusions basé sur les protocoles d'application (APIDS) est un système ou un agent qui réside généralement dans un groupe de serveurs. Il identifie les intrusions en surveillant et en interprétant la communication sur des protocoles spécifiques aux applications. Par exemple, il surveille le protocole SQL explicite à l'intergiciel lorsqu'il transige avec la base de données dans le serveur Web [8].

Système hybride de détection des intrusions

Le système de détection d'intrusion hybride est réalisé par la combinaison de deux ou plusieurs approches du système de détection d'intrusion. Dans le système de détection d'intrusion hybride, les données de l'agent hôte ou du système sont combinées avec les informations du réseau pour développer une vue complète du système de réseau. Le système de détection d'intrusion hybride est généralement plus efficace en comparaison avec les autres systèmes de détection d'intrusion [8].

1.8 Conclusion

Dans ce premier chapitre de notre mémoire, nous avons présenté une définition générale ainsi que quelques exemples des domaines d'application et protocole de communication pour l'internet des objets. L'un des aspects les plus discuté dans les objets connectés, et plus généralement dans l'Internet des Objets, est celui de la sécurité. plusieurs attaques sont possible contre les systèmes IdO, nous avons présentés les techniques utilisées pour ces différentes attaques ainsi que les solutions possible.

CHAPITRE 2
APPRENTISSAGE AUTOMATIQUE

2.1 Introduction

L'apprentissage automatique, par définition, est un domaine de l'informatique qui a évolué à partir de l'étude de la reconnaissance des formes et de la théorie de l'apprentissage informatique (computational learning) en intelligence artificielle. Il s'agit de l'apprentissage et de la construction d'algorithmes capables d'apprendre à partir de jeux de données et de faire des prédictions sur ces derniers. Ces procédures fonctionnent par construction d'un modèle à partir d'exemples d'entrées afin de faire des prédictions ou des choix basés sur des données plutôt que de suivre des instructions de programme statiques et fermes.

"On dit d'un programme informatique qu'il apprend de l'expérience $\boldsymbol{E}$ en ce qui concerne une certaine tâche $\boldsymbol{T}$ et une certaine mesure de performance $\boldsymbol{P}$, si sa performance sur $\boldsymbol{T}$, telle que mesurée par $\boldsymbol{P}$, s'améliore avec l'expérience $\boldsymbol{E}$." [Tom Mitchell, Université Carnegie Mellon].

Ainsi, si nous voulons que notre programme prévoie, par exemple, les formes de

trafic à un nœud occupé (tâche ***T***), nous pouvons le soumettre à un processus d'apprentissage automatique avec des données sur les formes de trafic précédentes (expérience ***E***) et, s'il a "appris" avec succès, il sera alors plus performant pour prédire les formes de trafic à venir (mesure de performance ***P***).

2.2 Définitions

Arthur Samuel, un pionnier dans le domaine de l'intelligence artificielle et des jeux vidéo, a inventé le terme "apprentissage automatique". Il a défini l'apprentissage automatique comme "un domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés".
L'apprentissage automatique ou machine learning (ML) désigne la capacité d'un système à acquérir et à intégrer des connaissances par le biais d'observations à grande échelle, ainsi qu'à s'améliorer et à s'étendre en apprenant de nouvelles connaissances plutôt qu'en étant programmé avec ces connaissances [53].

2.3 Apprentissage supervisé

L'apprentissage supervisé est une forme importante de ML. Il est appelé supervisé, car le processus d'apprentissage est effectué sous l'étiquette vue des variables d'observation. Dans l'apprentissage supervisé, les ensembles de données sont entraînés avec les ensembles d'entraînement pour construire la ML, et seront ensuite utilisés pour étiqueter les nouvelles observations de l'ensemble de test. En ce qui concerne l'ensemble d'apprentissage, les variables d'entrée sont les caractéristiques qui influencent la précision de la variable prédite. Elles contiennent à la fois des variables quantitatives et qualitatives, la variable de sortie est la classe d'étiquetage que l'apprentissage supervisé utilisera pour étiqueter les nouvelles observations. En fonction des différents types de variables de sortie, les tâches d'apprentissage supervisé peuvent être divisées en deux catégories : les tâches de classification et les tâches de régression.

Paramètres	Apprentissage automatique supervisé	Apprentissage automatique non supervisé
Données d'entrée	Les algorithmes sont formés à l'aide de données étiquetées.	Les algorithmes sont utilisés contre des données qui ne sont pas étiquetées.
Complexité informatique	Une méthode plus simple	Complexe sur le plan informatique
Précision	Haute précision	Moins précis
Nombre de classes	Le nombre de classes est connu	Le nombre de classes n'est pas forcément connu

TABLE 2.1 – Différence entre l'apprentissage supervisé et non supervisé.

Les variables de sortie de la tâche de classification sont des variables catégorielles, et celles de la tâche de régression sont des variables continues [58]. Par exemple, la classification des images chaudes actuelles est une tâche de classification, et la prédiction du prix des actions est une tâche de régression.

La procédure d'apprentissage supervisé peut être décrite comme suit : nous utilisons x(i) pour désigner les variables d'entrée, et y(i) pour désigner la variable de sortie. Une paire (x(i), y(i)) est un exemple d'apprentissage, et l'ensemble d'apprentissage que nous allons utiliser pour apprendre est (x(i), y(i)), i = 1, 2, ..., m. (i) dans la notation est un index dans l'ensemble d'apprentissage. Nous utilisons X pour désigner l'espace des valeurs d'entrée, et Y pour désigner l'espace des valeurs de sortie. Le but est, étant donné un ensemble d'apprentissage, d'apprendre une fonction h : X $\rightarrow$ Y afin que h(x) soit un bon prédicteur de la valeur correspondante de y.

2.3.1 Classification

La classification est le processus qui consiste à trouver ou à découvrir un modèle ou une fonction qui aide à séparer les données en plusieurs classes catégorielles, c'est-à-dire en valeurs discrètes. Dans la classification, les données sont classées sous

différentes étiquettes en fonction de certains paramètres donnés en entrée, puis les étiquettes sont prédites pour les données. Le processus de classification traite les problèmes dans lesquels les données peuvent être divisées en étiquettes discrètes binaires ou multiples.

2.3.2 Régression

La régression est un processus consistant à trouver un modèle ou une fonction permettant de distinguer les données en valeurs réelles continues au lieu d'utiliser des classes ou des valeurs discrètes. La régression peut également identifier le mouvement de distribution en fonction des données historiques.

En d'autres termes, la régression sert à trouver la relation d'une variable par rapport à une ou plusieurs autres, dans le but d'estimer une valeur (numérique) de sortie à partir des valeurs d'un ensemble de caractéristiques en entrée. Par exemple, estimer le prix d'une maison en se basant sur sa surface, nombre des étages, son emplacement $\cdots$ etc.

2.4 Apprentissage non supervisé

L'apprentissage non supervisé consiste à former une machine en utilisant des informations qui ne sont ni classées ni étiquetées et à permettre à l'algorithme d'agir sur ces informations sans aucune aide. Dans ce cas, la tâche de la machine consiste à regrouper des informations non triées en fonction de similitudes, de modèles et de différences, sans formation préalable des données.

Contrairement à l'apprentissage supervisé, aucun étiquetage n'est fourni. Par conséquent, la machine est limitée pour trouver la structure cachée dans les données. Cela permet au modèle de travailler par lui-même pour découvrir des modèles et des informations qui n'étaient pas détectés auparavant. Il traite principalement des données non étiquetées [27] .

L'apprentissage non supervisé est classé en deux catégories d'algorithmes :

2.4.1 Regroupement (Clustering)

Une méthode d'apprentissage non supervisée est une méthode dans laquelle nous tirons des références d'ensembles de données constitués de données d'entrée sans réponses étiquetées. En général, elle est utilisée comme un processus pour trouver une structure significative, des processus sous-jacents explicatifs, des caractéristiques génératives et des regroupements inhérents à un ensemble d'exemples.

Un problème de regroupement consiste à découvrir les regroupements inhérents aux données, par exemple en regroupant les clients par comportement d'achat [50].

2.4.2 Association

Un problème d'apprentissage de règles d'association consiste à découvrir des règles qui décrivent de grandes parties de vos données, telles que les personnes qui achètent X ont également tendance à acheter Y.

2.5 Algorithmes d'apprentissage automatique

Les scientifiques des données utilisent de nombreux types d'algorithmes d'apprentissage automatique pour découvrir des modèles dans les données volumineuses qui mènent à des informations exploitables. À un niveau élevé, ces différents algorithmes peuvent être classés en deux groupes en fonction de la manière dont ils "apprennent" les données pour faire des prédictions : l'apprentissage supervisé et non supervisé. Nous citons ci-dessous certain des algorithmes d'apprentissage les plus utilisés.

2.5.1 Régression logistique

La régression logistique est essentiellement un algorithme de classification supervisée. Dans un problème de classification, la variable cible (ou sortie), y, ne peut prendre que des valeurs discrètes pour un ensemble donné de caractéristiques (ou

entrées), X. Le modèle construit un modèle de régression pour prédire la probabilité qu'une entrée de données donnée appartienne à la catégorie numérotée "1". Tout comme la régression linéaire suppose que les données suivent une fonction linéaire, la régression logistique modélise les données à l'aide de la fonction sigmoïde. La régression logistique ne devient une technique de classification que lorsqu'un seuil de décision est introduit dans l'image. La fixation de la valeur du seuil est un aspect très important de la régression logistique et dépend du problème de classification lui-même. La décision concernant la valeur du seuil est principalement affectée par les valeurs de précision et de rappel. Idéalement, nous souhaitons que la précision et le rappel soient égaux à 1, mais c'est rarement le cas.

2.5.2 Machine à vecteur de support

L'algorithme SVM (Support Vector Machine) est principalement utilisé pour les problèmes de classification, mais aussi pour les modèles de régression. Il s'agit d'un modèle linéaire qui fournit des solutions aux problèmes linéaires et non linéaires. Il fonctionne sur le concept du calcul de la marge. Fondamentalement, il sépare l'ensemble de données en différentes classes et dessine l'hyperlane entre elles [25].

2.5.3 Random Forest

Cet algorithme est également utilisé pour les problèmes de régression et de classification. L'algorithme crée aléatoirement une forêt avec plusieurs arbres. Ainsi, dans le classificateur aléatoire Forest, plus le nombre d'arbres dans la forêt est élevé, plus la précision des résultats est grande. Donc, en d'autres termes, nous pouvons dire qu'une forêt aléatoire construit des arbres de décision collectifs appelés forêt et les intègre ensemble pour obtenir une prédiction plus précise et stable et la forêt qu'il construit dans la collection d'arbres de décision et chaque arbre de décision est construit seulement sur une partie de l'ensemble de données définies et il est formé avec la

méthode de bagging. La forêt aléatoire regroupe plusieurs arbres de décision pour aboutir à la décision finale.

2.5.4 Régression linéaire

La régression linéaire est un algorithme d'apprentissage automatique basé sur l'apprentissage supervisé. Il effectue une tâche de régression. La régression modélise une valeur de prédiction cible basée sur des variables indépendantes. Elle est principalement utilisée pour découvrir la relation entre les variables et les prévisions. Les différents modèles de régression diffèrent en fonction du type de relation entre les variables dépendantes et indépendantes qu'ils considèrent, et du nombre de variables indépendantes utilisées. La régression linéaire a pour but de prédire la valeur d'une variable dépendante (y) en fonction d'une variable indépendante donnée (x). Ainsi, cette technique de régression trouve une relation linéaire entre x (entrée) et y (sortie). D'où le nom de régression linéaire. Dans la figure ci-dessus, X (entrée) est l'expérience professionnelle et Y (sortie) est le salaire d'une personne. La ligne de régression est la meilleure ligne d'ajustement pour notre modèle.

$$y = \Theta 1 + \Theta 2 \cdot x$$

Lors de l'apprentissage du modèle, on nous donne :

— x : données d'entrée pour l'apprentissage (univarié - une variable d'entrée (paramètre)).

— y : étiquettes des données (apprentissage supervisé).

Lors de l'apprentissage du modèle il ajuste la meilleure ligne pour prédire la valeur de y pour une valeur donnée de x. Le modèle obtient la meilleure ligne d'ajustement de régression en trouvant les meilleures valeurs $\Theta 1$ et $\Theta 2$. Une fois que nous avons trouvé les meilleures valeurs, nous obtenons la meilleure droite d'ajustement. Ainsi, lorsque nous utilisons finalement notre modèle pour la prédiction, il prédit la valeur de y pour la valeur d'entrée de x [59].

2.5.5 Analyse discriminante linéaire

L'analyse discriminante linéaire ou analyse discriminante normale ou analyse de la fonction discriminante est une technique de réduction de la dimensionnalité couramment utilisée pour les problèmes de classification supervisée. Elle est utilisée pour modéliser les différences entre les groupes, c'est-à-dire pour séparer deux ou plusieurs classes. Elle est utilisée pour projeter les caractéristiques d'un espace de dimension supérieure dans un espace de dimension inférieure. Par exemple, nous avons deux classes et nous devons les séparer efficacement. Les classes peuvent avoir plusieurs caractéristiques. L'utilisation d'une seule caractéristique pour les classer peut entraîner un certain chevauchement, comme le montre la figure ci-dessous. Nous allons donc continuer à augmenter le nombre de caractéristiques pour une classification correcte [54].

2.5.6 K-Nearest Neighbors (KNN)

Les méthodes d'apprentissage basées sur les voisins sont de deux types : supervisées et non supervisées. L'apprentissage supervisé basé sur les voisins peut être utilisé pour les problèmes prédictifs de classification et de régression, mais il est principalement utilisé pour les problèmes prédictifs de classification dans l'industrie. Les méthodes d'apprentissage basées sur les voisins n'ont pas de phase de formation spécialisée et utilisent toutes les données pour la formation pendant la classification. Elles ne supposent rien non plus des données sous-jacentes. C'est la raison pour laquelle elles sont paresseuses et non paramétriques par nature. Le principe des méthodes du plus proche voisin est le suivant Trouver un nombre prédéfini d'échantillons d'entraînement proches en distance du nouveau point de données. Prédire l'étiquette à partir de ce nombre d'échantillons d'apprentissage. Ici, le nombre d'échantillons peut être une constante définie par l'utilisateur comme dans l'apprentissage du K-plus proche voisin ou varier en fonction de la densité locale du point comme dans l'apprentissage du voisin basé sur le rayon [7].

2.5.7 Gaussian Naive Bayes

Un classificateur Naive Bayes est basé sur la logique probabiliste qui utilise des algorithmes basés sur le théorème de Bayes. Le théorème de Bayes est une technique mathématique probabiliste qui permet de calculer les probabilités conditionnelles d'un événement. Naive Bayes est une technique de classification basée sur le théorème de Bayes avec une hypothèse d'indépendance entre les prédicteurs. Il s'agit de l'un des algorithmes d'apprentissage automatique les plus simples et les plus efficaces, capable de faire des prédictions rapides sur la base de la probabilité des données. Il est essentiellement utilisé pour la classification de textes et de questions comportant de nombreuses classes [1].

2.5.8 K-Means

L'algorithme K-Means est un algorithme de clustering, qui regroupe l'ensemble de données non étiquetées en différents clusters. Ici, K définit le nombre de clusters qui doivent être créés dans le processus, comme si K = 2, il y aura deux clusters, et pour K=3, il y aura trois clusters, et ainsi de suite. Il nous permet de regrouper les données en différents groupes et constitue un moyen pratique de découvrir les catégories de groupes dans l'ensemble de données non étiquetées sans avoir besoin d'entraînement. Il s'agit d'un algorithme basé sur les centroïdes, où chaque groupe est associé à un centroïde. L'objectif principal de cet algorithme est de minimiser la somme des distances entre le point de données et les clusters correspondants. L'algorithme prend l'ensemble de données non étiquetées comme entrée, divise l'ensemble de données en un nombre k de clusters, et répète le processus jusqu'à ce qu'il ne trouve pas les meilleurs clusters. La valeur de k doit être prédéterminée dans cet algorithme [35].

2.5.9 Decision Tree

L'arbre de décision est l'outil le plus puissant et le plus populaire pour la classification et la prédiction. Un arbre de décision est une structure arborescente de type organigramme, où chaque nœud interne représente un test sur un attribut, chaque branche représente un résultat du test, et chaque nœud feuille (nœud terminal) contient une étiquette de classe [44].

2.5.10 L'apprentissage profond

L'apprentissage profond est une branche de l'apprentissage automatique qui est entièrement basée sur les réseaux neuronaux artificiels. Comme les réseaux neuronaux vont imiter le cerveau humain, l'apprentissage profond est également une sorte d'imitation du cerveau humain. Dans l'apprentissage profond, nous n'avons pas besoin de tout programmer explicitement. Le concept d'apprentissage profond n'est pas nouveau. Il existe depuis quelques années maintenant. Il fait l'objet d'un engouement aujourd'hui parce qu'auparavant, nous ne disposions pas d'une aussi grande puissance de traitement ni de beaucoup de données. Au cours des 20 dernières années, la puissance de traitement a augmenté de manière exponentielle, et l'apprentissage profond et l'apprentissage automatique ont fait leur apparition. Une définition formelle de l'apprentissage profond est la suivante : neurones.
Le cerveau humain compte environ 100 milliards de neurones, et chaque neurone est connecté à des milliers de ses voisins. La question est de savoir comment recréer ces neurones dans un ordinateur. Nous créons donc une structure artificielle appelée réseau neuronal artificiel où nous avons des nœuds ou des neurones. Nous avons certains neurones pour la valeur d'entrée et d'autres pour la valeur de sortie et entre les deux, il peut y avoir beaucoup de neurones inter-connectés dans la couche cachée [17].

Architectures :

1-Réseau neuronal profond - Il s'agit d'un réseau neuronal présentant un certain niveau de complexité (ayant plusieurs couches cachées entre les couches d'entrée et de sortie). Ils sont capables de modéliser et de traiter des relations non linéaires. **2-Réseau de croyance profond (DBN)** : Il s'agit d'une classe de réseau neuronal profond. Il s'agit de réseaux de croyance multicouches. Étapes pour réaliser un DBN : a. Apprendre une couche de caractéristiques à partir des unités visibles à l'aide de l'algorithme Contrastive Divergence. b. Traiter les activations des caractéristiques précédemment formées comme des unités visibles et ensuite apprendre les caractéristiques des caractéristiques. c. Enfin, l'ensemble du DBN est formé lorsque l'apprentissage de la dernière couche cachée est réalisé [28].

3-Réseau neuronal récurrent (exécute la même tâche pour chaque élément d'une séquence) - Permet le calcul parallèle et séquentiel. Similaire au cerveau humain (grand réseau de neurones connectés). Ils sont capables de se souvenir d'éléments importants concernant les données qu'ils ont reçues, ce qui leur permet d'être plus précis [38].

2.6 Types de réseaux de neurones

Les types de réseaux neuronaux sont les concepts qui définissent comment la structure du réseau neuronal fonctionne dans le calcul, à l'instar de la fonctionnalité du cerveau humain pour la prise de décision . Il existe plusieurs types de réseaux neuronaux :

2.6.1 Feed-Forward Neural Network

Il s'agit d'un réseau neuronal de base qui peut exister dans tout le domaine des réseaux neuronaux. Comme son nom l'indique, le mouvement de ce réseau est uniquement vers l'avant, et il se déplace jusqu'au point où il atteint le nœud de sortie. Il n'y a pas de rétroaction pour améliorer les nœuds des différentes couches et peu de mécanisme d'auto-apprentissage [48].

2.6.2 Radial Basis Function (RBF) Neural Network

L'intuition principale de ces types de réseaux neuronaux est la distance des points de données par rapport au centre. Ces réseaux neuronaux comportent généralement deux couches (l'une est la couche cachée et l'autre la couche de sortie). La couche cachée possède une fonction de base radiale typique. Cette fonction permet une interpolation raisonnable lors de l'adaptation des données. Cela vient avec l'intuition que les points les plus proches sont similaires en nature et ont une similitude avec le k-NN. L'intuition est la suivante : "La sortie cible prédite d'un élément se comportera de manière similaire à d'autres éléments qui ont une ressemblance proche des variables prédicteurs [41]."

2.6.3 Multilayer Perceptron

Maintenant, nous passerions lentement à des réseaux neuronaux ayant plus de 2 couches, c'est-à-dire plus d'une couche cachée. Dans un perceptron multicouche, l'intuition principale de l'utilisation de cette méthode est lorsque les données ne sont pas linéairement séparables. Chaque nœud d'une couche est constitué d'une fonction d'activation non linéaire à traiter. Ces fonctions sont généralement la fonction Sigmoïde/Logistique, tanh/Fonction tangente hyperbolique, ReLU (unité linéaire rectifiée), Softmax. Ce réseau neuronal est entièrement connecté et a également la capacité d'apprendre par lui-même en modifiant les poids de connexion après le traitement de chaque point de données et la quantité d'erreur qu'il génère [40] .

2.6.4 Convolutional Neural Network

Pour en venir au réseau neuronal convolutif, ce type de réseau neuronal est une version avancée du perceptron multicouche. Dans ce type, il y a une ou plusieurs couches convolutives. La question fondamentale est de savoir ce qu'est exactement une couche convolutive. La convolution n'est rien d'autre qu'un simple mécanisme de filtrage qui permet une activation. Lorsque ce mécanisme de filtrage est répété, il

donne l'emplacement et la force d'une caractéristique détectée. En raison de cette capacité, ces réseaux sont largement utilisés dans le traitement de l'image, le traitement du langage naturel, les systèmes de recommandation afin de produire des résultats efficaces de la caractéristique importante détectée [16].

2.6.5 Recurrent Neural Network

Comme son nom l'indique, dans ce réseau, quelque chose se répète. Pour mentionner ce réseau, la sortie d'une couche particulière est sauvegardée et est réintégrée dans l'entrée à nouveau. Ici, la première couche sera un simple réseau neuronal de type feed-forward et, par la suite, chaque nœud conservera les informations dans les couches suivantes. Ainsi, si la prédiction est erronée, le réseau essaiera de réapprendre et d'apprendre efficacement à faire la bonne prédiction. Cette méthode est largement utilisée dans la conversion texte-parole. Le principal élément constitutif de ce réseau est le stockage en mémoire qui influencera la meilleure prédiction de ce qui va suivre [38].

2.6.6 Modular Neural Network

Comme son nom l'indique, dans ce réseau, quelque chose se répète. Pour mentionner ce réseau, la sortie d'une couche particulière est sauvegardée et est réintégrée dans l'entrée à nouveau. Ici, la première couche sera un simple réseau neuronal de type feed-forward et, par la suite, chaque nœud conservera les informations dans les couches suivantes. Ainsi, si la prédiction est erronée, le réseau essaiera de réapprendre et d'apprendre efficacement à faire la bonne prédiction. Cette méthode est largement utilisée dans la conversion texte-parole. Le principal élément constitutif de ce réseau est le stockage en mémoire qui influencera la meilleure prédiction de ce qui va suivre [19].

2.7 Conclusion

Nous apprenons dans ce chapitre que les machines peuvent être entraînées à effectuer des activités humaines dans plusieurs domaines et peuvent aider les humains à vivre mieux. L'apprentissage automatique peut être supervisé ou non supervisé. Si vous avez moins de données et des données clairement étiquetées pour la formation, l'apprentissage non supervisé donne généralement de meilleures performances et de meilleurs résultats pour les grands ensembles de données. Si vous disposez d'un énorme ensemble de données, optez pour les techniques d'apprentissage profond.

CHAPITRE 3
ARCHITECTURE ET IMPLÉMENTATION

3.1 Introduction

Après avoir défini les concepts théoriques de base sur l'IdO et l'apprentissage automatique dans les deux premiers chapitres, nous passons à la deuxième partie de notre travail qui consiste à étudier les performances des plusieurs algorithmes d'apprentissage automatique pour la détection d'intrusions dans les systèmes IdO. Dans le cadre de notre étude comparative, nous avons utilisé la base de données IoTID20 détaillé dans [55]. Nous avons entrainé plusieurs modèles basés sur des algorithmes d'apprentissage différents, dans le but de choisir lequel est le plus adapté à notre travail en terme de précision ainsi qu'en terme de rapidité. Dans ce chapitre nous allons présenter les étapes qui ont abouti aux résultats de notre étude. Nous commençons dans la première partie par présentation de la base de données utilisée et la traitement à effectuer sur cette base pour pouvoir l'utiliser dans le cadre de notre travail. Puis nous passons aux choix des algorithmes à utiliser dans notre étude, et finalement, nous expliquons les différentes mesures de performance que nous allons prendre en compte pour la comparaison des algorithmes.

3.2 Jeu de données

L'ensemble de données choisi pour cette étude est IoTID20 dataset détaillée dans [55], elle est parmi les rares datasets qui sont publiquement disponibles. Elle se compose de 83 caractéristique du système, et 3 caractéristiques d'étiquetage qui prennent en compte différents scénarios de classification. Les caractéristiques d'étiquetage sont les suivantes :

1. ***Binary :*** qui permet de séparer les ligne de la dataset en deux catégories étiquetées "Normal" et "Anomaly".
2. ***Category :*** qui permet d'aller au dela de la classification binaire et donner plus de détails sur le type d'intrusion. Les différents type d'anomalies prises en charge sur la colonne "Category" sont : DoS, Mirai, MITM et Scan, s'ajoute à cela la catégorie "Normal" pour un total de 5 classes étiquetés sur cette colonne.
3. ***SubCategory :*** certaines parmi les catégories citées ci-dessus peuvent également être décomposées hiérarchiquement en d'autres sous-catégories. Nous mentionnons à titre d'exemple la catégorie "Mirai" qui comporte les sous-catégories : "Brute Force", "HTTP Flooding" et "UDP Flooding", ou la catégorie "Scan" qui comporte les deux sous-catégories : "Host Port" et "OS".

Dans le cadre de notre étude, nous allons nous intéresser seulement à la classification binaire, donc nous considérons seulement la colonne "Binary" pour les tests que nous allons effectuer.

Un pré traitement des données de la dataset IoTID20 est nécessaire, comme c'est souvent le cas pour la plupart des bases dans ce domaine. Ce pré traitement est nécessaire principalement car certains types de données ont un format qui n'est pas bien adapté aux algorithmes d'apprentissage automatique, nous citons par exemple le type "String" ou les valeurs "NaN" et "Inf". Nous expliquons dans la section ci-dessous le processus de pré traitement que nous avons suivi pour préparer les données avant de lancer l'apprentissage.

3.2.1 Pré-traitement

Afin de construire un modèle très précis, il est important d'effectuer des analyses exploratoires sur l'ensemble de données et ses caractéristiques. Le pré-traitement de l'ensemble de données est effectué avant d'appliquer un algorithme d'apprentissage automatique. Nous listons ci-dessous les étapes de pré-traitement que nous avons suivies pour préparer nos données.

1. Vu que nous nous intéressons à la classification binaire seulement nous avons commencé par la suppression des deux colonnes de catégories et sous catégorie de chaque ligne.
2. Par la suite, nous avons remplacer les étiquettes "Normal" et "Anomaly" dans la colonne "Binary" par les valeurs 1 et 0 respectivement, car les valeurs numérique sont plus facile à gérer.
3. Après, nous procédons à l'élimination dans un premier temps les colonnes qui contiennes des valeurs non numériques.
4. A cette étape, nous n'avons que des données numériques dans la matrice contenant le jeu de données, et donc nous pouvons passer à l'étape de la normalisation. Cette étape est très importante pour égaliser le poids de chaque caractéristique, sont normalisation, les caractéristiques ayant une plage de valeurs plus large pourrons avoir plus d'impact que les caractéristiques avec une plage de valeurs réduite.
5. Après la normalisation, nous passons à l'élimination des valeurs "NaN" et "Inf" qui pourrais soit être présent sur le jeu de données initial, soit apparaître après la normalisation dans certains cas.
6. Finalement, pour assurer que le jeu de données est équilibré, nous avons pris le même nombre de lignes ($\approx$ 40000) pour l'étiquette "Normal" (1) et "Anomaly" (0). Pour des jeux de données non équilibrés, nous risquons de finir avec un

modèle qui reconnais mieux une classe contrairement à l'autre, ce qui va réduire les scores pour certaines mesure de performance ("precision" et/ou "recall").

3.3 Choix de l'algorithme d'apprentissage

Déterminer quel algorithme d'apprentissage utiliser dans notre cas d'étude et sur le jeu de données est un choix très important ou souvent loin d'être évident. Dans la plus part des cas, l'approche empirique qui consiste à tester et évaluer plusieurs algorithmes d'évaluation pour en choisir le meilleur est la seule façon de garantir que le choix que nous avons effectué va dans le bon sens. Nous avons décider de tester quelques algorithmes qui sont généralement les plus utilisés dans la littérature. Ci-dessous la liste des algorithmes que nous allons tester et comparer pour choisir lequel est le mieux adapté nos objectifs.

— Régression Logistique (Expliqué dans la section 2.5.1).
— Analyse discriminante linéaire (Expliqué dans la section 2.5.5).
— Gaussian Naive Bayes (Expliqué dans la section 2.5.7).
— K-Nearest Neighbors (KNN) (Expliqué dans la section 2.5.6).
— Decision tree (Expliqué dans la section 2.5.9).
— Machine à vecteurs de support - SVM (Expliqué dans la section 2.5.2).
— Les réseau de neurones artificiels (Expliqué dans la section 2.5.10).

Nous expliquons dans le section suivante nos mésures et méthode d'évaluation pour ces algorithmes.

3.4 Méthode d'évaluation

L'objectif de n'importe quel algorithme de classification est de se baser sur des données étiquetées pour construire un modèle capable de reconnaître la classe d'appartenance de données non étiquetées avec la meilleure précision possible. Plusieurs

mesures de performance peuvent être utilisée, celle qui sont les plus utilisées pour mesurer l'efficacité d'un modèle sont les suivantes : "Accuracy", "Precision", "Recall", "F1". Certaines autres mesures sont pratiques est intéressantes à regarder pour avoir une idée sur l'usage du modèle en pratique. Nous nous intéressons en particulier au : temps d'apprentissage (fit time), et temps de prédiction.

3.4.1 Accuracy

C'est probablement la mesure la plus importante et la plus simple à la fois. La précision de prédiction (accuracy) c'est un pourcentage défini par le nombre de prédictions justes, divisé par le nombre de prédictions totales (multiplié par 100). Clairement un pourcentage élevé signifie que le modèle fait de bonnes prédictions, alors qu'une valeur faible signifie que le modèle n'est pas très efficace. Un pourcentage proche de 50% sur une classification binaire signifie que le modèle dans des résultats proche d'une affectation aléatoire des classes, en d'autres termes il est inutile.

3.4.2 Precision

A ne pas confondre avec "accuracy", même si les deux mots se traduisent à "précision" en français, ce sont deux mesures différentes. Dans notre cas la mesure "precision" est un ratio de nombre de lignes qui ont été "correctement" classées dans la catégories "Normal" "appelés : True Positives" divisé par toutes les ligne qui ont été classées dans la catégorie "Normal" correctement ou incorrectement. En d'autre terme, elle permet de mesurer le pourcentage des lignes classées dans la catégorie "Normal" par rapport à la somme de toutes les lignes considérées normal. Un mauvais score de "precision" signifie que le modèle se trompe souvent en classifiant des cas "Anomaly" dans la catégorie "Normal".

3.4.3 Recall

A l'enconre de la mesure "precision", la mesure "recall" est un ratio de nombre de lignes qui ont été "correctement" classées dans la catégories "Normal" divisé par le nombre de ligne qui auraient du être classées dans la catégorie "Normal" (même si elles ont été classées dans la catégorie "Anomaly"). En d'autre termes, cette mesure permet de reconnaître le pourcentage des lignes dans la catégorie "Normal" qui ont été classées correctement. Un mauvais score de la mesure "recall" signifie que le modèle se trompe souvent en classifiant des cas "Normal" dans la catégorie "Anomaly".

Dans le cas d'un jeu de données pas équilibré ou une classe à significativement plus d'exemple qu'une autre, il est possible d'avoir un score très élevé pour la mesure "accuracy" mais avec un très mauvais score pour la mesure "precision" et/ou "recall".

3.4.4 F1

Le score F1 est une métrique de classification qui mesure la capacité d'un modèle à bien prédire les individus positifs, tant en termes de "precision" qu'en termes de "recall". Il correspond en effet à la moyenne harmonique de ces indicateurs, qui doivent tous deux être élevés pour que le F1-score le soit aussi [15].

3.4.5 Temps d'apprentissage

Si le temps que prend le modèle durant la phase d'apprentissage pour converger vers la valeur minimale de la fonction objectif. Un temps d'apprentissage réduit signifie qu'en pratique, il est raisonnable de relancer l'apprentissage si de nouvelles informations se présente pour améliorer la dataset.

3.4.6 Temps de prédiction

Si le temps que prend le modèle durant la phase de test pour faire la classification d'un certain nombre de lignes. Un temps de prédiction réduit signifie qu'en pratique, le modèle est très réactif et peux être utilisé même sur des application avec des contraintes sur le temps d'exécution.

3.4.7 Validation des mesures

Durant la phase d'apprentissage, il n'est raisonnable d'utiliser les mêmes données pour l'apprentissage et le test à la fois. Dans ce cas, le modèle pourrais simplement mémoriser les cas qu'il a déjà vu, et donner des score très élevés, sans être capable de reconnaître de nouvelles données en pratique. C'est ce qu'on appelle le problème le "overfitting". Pour éviter cette situation, il est recommandé de diviser le jeu de données en deux parties, une partie pour l'apprentissage, et une autre pour le test. Pour valider scores obtenus pour chaque mesure nous utilisons la méthode de validation croisée (cross validation) [13], La partie du jeu de données consacrée à l'apprentissage est décomposé en "n" sous-parties et à chaque itération "n - 1" sous-parties sont utilisées pour l'apprentissage, et la partie restante (qui change à chaque itération) est celle considérée pour le test intermédiaire. Le test final n'est effectuer que pour valider le modèle obtenu comme le montre la figure 3.1

Pour chaque algorithme, nous effectuons 10 testes de cette façon, et nous considérons à la fin la moyenne des résultats obtenus pour chaque mesure. Par contre, pour le temps d'exécution, que cela soit le temps d'apprentissage ou de prédiction, nous prenons le minimum au lieu de la moyenne, vu que le temps d'exécution change très souvent à cause d'interruptions système qui n'ont rien à voir avec l'exécution du programme.

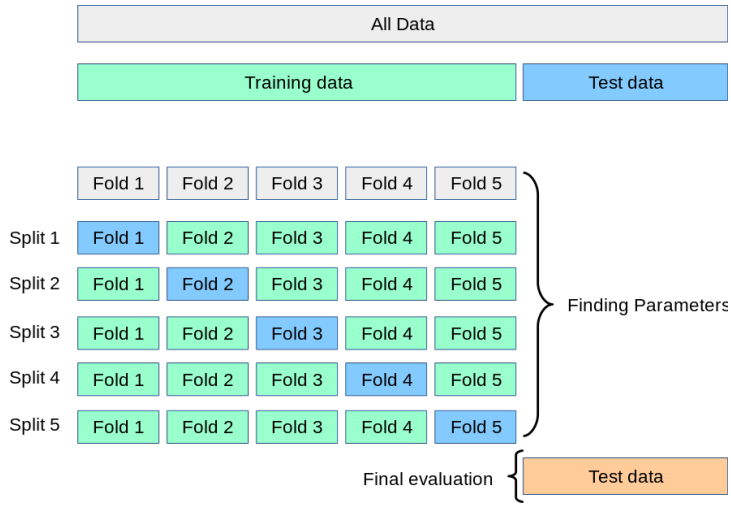


FIGURE 3.1 – La méthode Cross-Validation [13]

3.5 Conclusion

Dans ce chapitre, nous avons présenté notre approche consistant à tester plusieurs algorithmes d'apprentissage automatique pour pouvoir identifier lequel est le plus adapté à la base que nous avons choisie. Nous avons également présenté les mesures de performance sur lesquelles nous allons nous baser pour choisir le bon algorithme, comme la précision, le rappel, le F1, ce qui nous permet d'obtenir les meilleurs résultats possibles pour la détection d'intrusion. Ces algorithmes d'apprentissage ont des performances différentes selon les ensembles de données sélectionnés et les caractéristiques d'entrée.

CHAPITRE 4

RÉSULTATS ET DISCUSSIONS

4.1 Introduction

De nos jours, le système de détection d'intrusion (IDS), qui est de plus en plus un élément clé de la sécurité des systèmes, est utilisé pour identifier les activités malveillantes dans un système et un réseau informatique. Il y a différentes approches employées dans les systèmes de détection d'intrusion, mais malheureusement aucune technique n'est pas entièrement idéale. Dans ce chapitre, nous proposons nous procédons à l'implémentation et le test des plusieurs algorithmes d'apprentissage. Dans la deuxième section de ce chapitre, nous présentons les outils de développement utilisé pour accomplir ce travail. Dans la troisième section nous présentons une petite visite guidée de notre environnement de travail avec une explication de notre méthodologie, et dans la quatrième section nous allons présenter et discuter les résultats obtenus.

4.2 Les Outils de Développement

4.2.1 Python

Python est un langage de programmation interprété, orienté objet, de haut niveau et doté d'une sémantique dynamique. Ses structures de données intégrées de haut niveau, combinées au saisie dynamique et à la liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script ou de colle pour connecter des composants existants. La syntaxe simple et facile à apprendre de Python privilégie la lisibilité et réduit donc le coût de la maintenance des programmes. Python supporte les modules et les packages, ce qui encourage la modularité des programmes et la réutilisation du code. L'interpréteur Python et la bibliothèque standard étendue sont disponibles gratuitement sous forme de source ou de binaire pour toutes les principales plates-formes et peuvent être distribués librement [56].

4.2.2 Google Colab

Google Colab ou Colaboratory est un service cloud, proposé par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche en apprentissage automatique. Cette plateforme permet d'entraîner des modèles d'apprentissage automatique directement dans le cloud, donc sans avoir besoin d'installer quoi que ce soit sur notre ordinateur à part un navigateur [18].

4.2.3 SKLearn

Scikit-learn (Sklearn) est la bibliothèque la plus utile et la plus robuste pour l'apprentissage automatique en Python. Elle fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression, le regroupement et la réduction de la dimensionnalité, via une interface cohérente en Python. Cette bibliothèque, qui est en grande partie écrite en

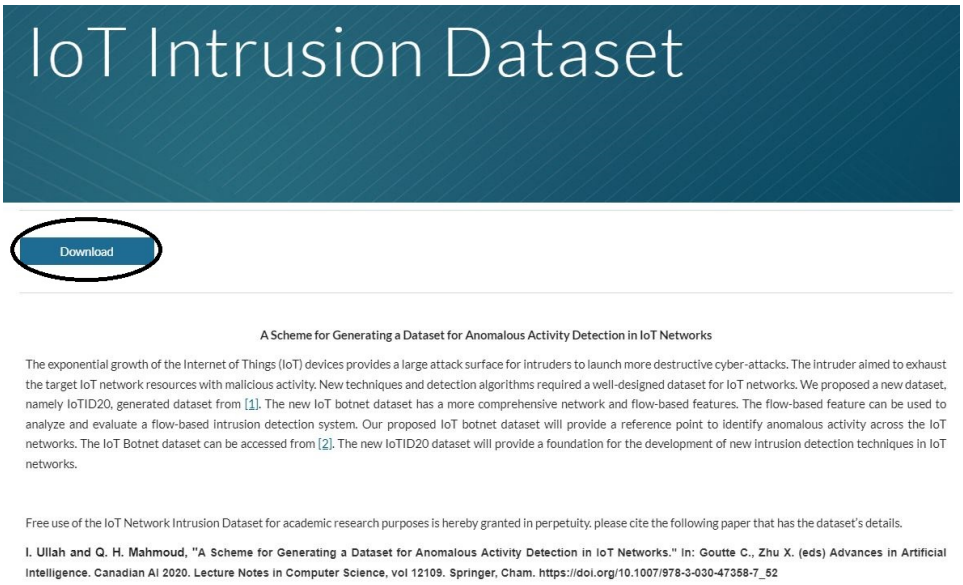


FIGURE 4.1 – Téléchargement de la base IoTID20

Python, s'appuie sur NumPy, SciPy et Matplotlib. Elle s'appelait à l'origine scikits-learn et a été initialement développée par David Cournapeau dans le cadre d'un projet Google summer of code en 2007. Plus tard, en 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort et Vincent Michel, de FIRCA (Institut français de recherche en informatique et en automatique), ont porté ce projet à un autre niveau et ont fait la première version publique (v0.1 beta) le 1er février 2010 [43].

4.3 Présentation de l'environnement

Comme nous avons déjà expliqué, nous allons travailler sur l'environnement offert par Google Colab, cela nous évite d'installer tous les outils nécessaires sur notre machine (interpréteur python, sklearn $\cdots$ etc), et aussi, cela nous évite de télécharger des jeux de données (souvent volumineux) que nous allons plutôt importer à partir d'un fichier partager sur Google Drive.

Nous montrons ici comment comment importer la base IoTID20 à partir du site officiel [24] vers l'environnement Google Colab. La figures 4.1 montre le site officiel

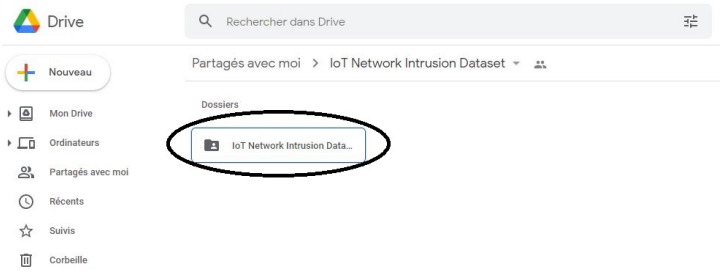


FIGURE 4.2 – Le fichier partagé sur Google Drive

de la base que nous allons utiliser, après avoir cliqué sur le bouton ""Download", un fichier CSV sera partagé sur notre espace Google drive comme le montre la figure 4.2.

Par la suite, il va falloir importer le contenu de l'espace Google Drive sur Google Colab en utilisant l'instruction "Drive.mount" comme le montre la figure 4.3.

FIGURE 4.3 – Lire le jeu de données à partir de Google Drive

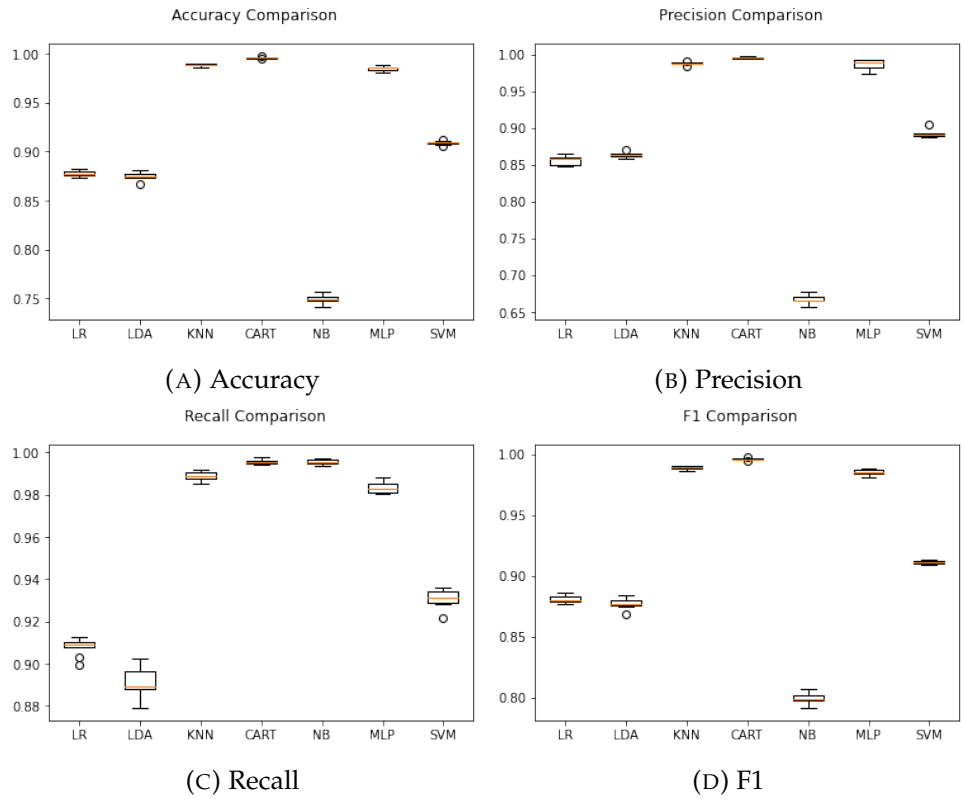
Cette procédure nous évite de devoir télécharger le jeu de données sur une machine locale. Tous les résultats présentés ci-dessous sont générés sur le même environnement cloud équipé d'un processeur "Intel Xeon" ayant une fréquence de 2.2Ghz et 12GB de RAM. Nous n'avons utilisé aucun GPU ou autre forme d'accélération dans ce travail.

4.4 Résultats et Discussion

Dans cette section, nous allons montrer et interpréter les résultats que nous avons obtenus à travers nos tests. Nous commençons d'abord par les résultats de précision puis nous passons aux comparaisons des temps d'exécution pour l'apprentissage et la prédiction.

4.4.1 Précision des Modèles

Nous montrons respectivement les résultats des mesures "Accuracy", "precision", "recall" et "F1" respectivement dans les figures 4.4a, 4.4b, 4.4c et 4.4d. Ces figures montrent à la fois la moyenne et l'écart type des résultats pour dix (10) tests effectuer pour chaque algorithme.



(A) Accuracy (B) Precision

(C) Recall (D) F1

FIGURE 4.4 – Résultats de précision sous forme visuelle

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

TABLE 4.1 – Moyenne des résultats de précision sous forme numérique

Dans le tableau 4.1 la moyenne seulement est montrée sous forme de pourcentage avec arrondi au plus proche en deux chiffres après la virgule.

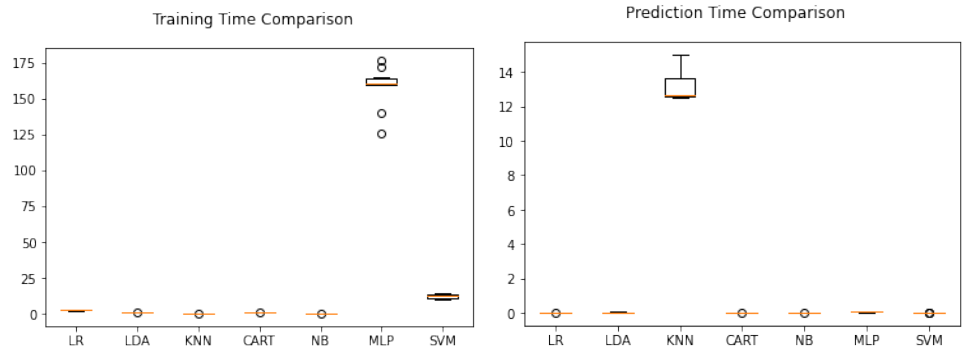
Les trois algorithmes qui ont la précision la plus élevée (98 à 99%) sont "Arbre de décision (CART)", "k voisins les plus proches (KNN)" et "Réseau de neurones (MLP)", nous croyons que la précision de ce dernier pourrais encore être améliorer en personnalisant la configuration du réseau et le taux d'apprentissage, nous avons utilisé ici la configuration par défaut qui consiste en une seule couche intermédiaire de "100" neurones et un taux d'apprentissage constant qui vaut "0.001". Malheureusement, suite aux contraintes de temps, nous n'avions pas pousser ces tests.

Un peu derrière, arrivent les algorithmes basés sur des modèles linéaires, les trois algorithmes "regression logistique", "analyse discriminante linéaire" et "machine à vecteur de support" montrent une précision proche de 90%. Les autres mesures de précision pour tous les algorithmes cités ci-dessus ne soin pas très loin de "Accuracy" (une différence maximale de 3%), ce qui montre que les résultats de prédiction ne favorisent pas une classe par rapport à une autre.

Finalement, l'algorithme le moins performant parmi ceux qui sont testés est la classification naïve bayésienne (NB). Son mesure accuracy ne dépasse pas les 75%, nous observons aussi qu'il a un score "precision" très bas accompagné d'un score "Recall" très élevé, ce qui montre qu'il a tendance à considérer plus de cas d'anomalies dans la classe "Normal", inversement, il se trompe rarement en considérant un cas normal dans la classe "Anomaly".

4.4.2 Temps d'exécution

Dans cette section, nous allons montrer et discuter les temps d'exécution des modèles que nous avons étudiés. Nous nous intéressons à deux types de temps d'exécution qui sont le temps d'apprentissage, et le temps de prédiction.



(A) Temps d'apprentissage (B) Temps de prédiction

FIGURE 4.5 – Résultats de temps d'exécution sous forme visuelle

Comme pour les résultats précédents, les deux figures 4.5a et 4.5b montrent à la fois la moyenne et l'écart type du temps d'exécution sur les 10 tests effectués en secondes. Alors que le tableau 4.2 montre le minimum de ces 10 tests effectués pour chaque algorithme en millisecondes arrondi à l'entier le plus proche.

Nous avons estimé qu'il nécessaire de partager les deux figures malgré le fait qu'elles soient très difficile à lire et interpréter, principalement à cause de la grande

Algorithme	Temps d'apprentissage	Temps de prédiction
Régression Logistique	2115	25
Analyse discriminante linéaire	982	25
Méthode des k plus proches voisins	34	12472
Arbres de décision	938	18
Classification naïve bayésienne	102	24
Réseaux de neurones	125621	44
Machine à vecteurs de support	9500	25

TABLE 4.2 – Minimum des résultats de temps d'exécution sous forme numérique

différence dans les ordres des grandeur entre les algorithmes. Nous allons nous concentrer beaucoup plus sur le tableau 4.2.

Nous observons premièrement que, contrairement à tous les algorithmes, l'algorithme KNN est plus rapide à la phase d'apprentissage, mais très lent à la prédiction. En effet ce phénomène est lié à la nature même de l'algorithme qui n'a aucune phase d'apprentissage, par contre il fait une indexation des données pour que la recherche des k voisins les plus proches ($\boldsymbol{k=5}$ par défaut dans notre cas) soit facile à la phase de prédiction. Toujours contrairement aux autres algorithmes, aucun modèle n'est prêt à la prédiction, et donc le temps d'exécution dépend largement du nombre d'exemples à traiter, le nombre de caractéristiques et la valeur de $\boldsymbol{k}$. Malgré la précision trop élevée de cet algorithme, nous estimons qu'il n'est pas pratique à utiliser dans un scénario réel vu que le temps de prédiction va jusqu'à 13 secondes, ce qui est plus que 500 fois plus lent par rapport aux algorithmes comparés.

Parmi les algorithmes les plus précis, l'algorithme CART est le plus rapide en phase de prédiction, il répond en 18 millisecondes seulement, ce qui est presque 3 fois plus rapide que l'algorithme MLP qui a presque la même précision. Les algorithmes restants, ont presque la même vitesse que CART, mais ce dernier est beaucoup plus précis (différence moyenne de 10%) ce qui le rend plus pratique.

En phase d'apprentissage, l'algorithme KNN est le plus rapide pour des raisons que nous venons d'expliquer, mais cet avantage est rapidement ignoré en regardant le temps de prédiction de ce dernier. Le deuxième algorithme plus rapide est NB, mais comme nous l'avons déjà dit, c'est l'algorithme le moins précis. Ce qui nous pousse à considérer encore l'algorithme CART qui est beaucoup plus rapide que l'algorithme MLP qui a une précision similaire, et beaucoup plus précis que les algorithmes à base de modèle linéaire qui ont un temps d'apprentissage similaire.

4.5 Conclusion

La classification ML nécessite un réglage fin des paramètres et, en même temps, un nombre important d'instances pour l'ensemble des données. Ce n'est pas seulement une question de temps pour construire le modèle de l'algorithme mais de précision et de classification correcte. Par conséquent, le meilleur algorithme d'apprentissage pour un ensemble de données particulier ne garantit pas la précision et l'exactitude (accuracy) pour un autre ensemble de données dont les attributs sont logiquement différents. Cependant, la question clé lorsqu'on traite de la classification ML n'est pas de savoir si un algorithme d'apprentissage est supérieur aux autres, mais dans quelles conditions une méthode particulière peut significativement surpasser les autres sur un problème d'application donné.

CONCLUSION GÉNÉRALE

Ces dernières années, la recherche sur les techniques d'apprentissage automatique pour en examinant le comportement du réseau ainsi que celui des menaces ont largement évolué. Le grand volume de bases de données augmente rapidement, ce qui entraîne une augmentation progressive des attaques de sécurité.

Le système de détection d'intrusion est la méthode la plus connue pour la détection des attaques de réseau. Il s'agit d'un système qui surveille le trafic réseau à la recherche d'activités suspectes et émet des alertes lorsqu'une telle activité est découverte.

Dans notre travail nous avons commencé par la sélection de données ou nous avons choisi de travailler avec une dataset nommé IoTID20 pour la détection des anomalies, nous avons décider de tester quelques algorithmes qui sont généralement les plus utilisés dans la littérature, ci-dessous la liste des algorithmes que nous avons testé et comparé pour choisir lequel est le mieux adapté à nos objectifs.

— Régression Logistique
— Analyse discriminante linéaire
— Gaussian Naive Bayes
— K-Nearest Neighbors (KNN)
— Decision tree
— Machine à vecteurs de support (SVM)

— Les réseau de neurones artificiels.

Notre objectif est de trouver le meilleur algorithme avec la meilleure précision possible, ainsi qu'avec le temps d'exécution minimal pour être utilisable en pratique. Plusieurs mesures de performance peuvent être utilisées, celle qui sont les plus utilisées pour mesurer l'efficacité d'un modèle sont les suivantes : "Accuracy", "Precision", "Recall" et "F1". Nous avons également inclu les mesure de "fit time" et "score time" pour comparer les algorithmes en terme de performances aussi. Les résultats que nous avons obtenus montrent que les deux meilleurs algorithmes pour notre jeu de données sont l'arbre de décision et le MLP (réseau de neurones), ils ont tous deux la meilleure précision avec le temps de prédiction le plus rapide, nous croyons que la précision de ce dernier (MLP) pourrais encore être améliorer en personnalisant la configuration du réseau et le taux d'apprentissage. Il est également possible dans des futurs travaux de regarder avec plus de soin la configuration de chacun des autres algorithmes aussi, dans le but de trouver les meilleurs paramètres permettant d'améliorer la précision, en réduisant le temps d'exécution.

BIBLIOGRAPHIE

[1] N Ben AMOR, Salem BENFERHAT et Zied ELOUEDI. « Réseaux bayésiens naïfs et arbres de décision dans les systèmes de détection d'intrusions ». In : *TSI-Technique et Science Informatiques* 25.2 (2006), p. 167-196.

[2] Ioannis ANDREA, Chrysostomos CHRYSOSTOMOU et George HADJICHRISTOFI. « Internet of Things : Security vulnerabilities and challenges ». In : *2015 IEEE symposium on computers and communication (ISCC)*. IEEE. 2015, p. 180-187.

[3] J APCAR. « The internet of things : Routing and related protocols ». In : *Cisco Live, https ://www. ciscolive. com/online/connect/se ssionDetail. ww* ().

[4] Luigi ATZORI, Antonio IERA et Giacomo MORABITO. « The internet of things : A survey ». In : *Computer networks* 54.15 (2010), p. 2787-2805.

[5] Raja BENABDESSALEM, Mohamed HAMDI et Tai-Hoon KIM. « A survey on security models, techniques, and tools for the internet of things ». In : *2014 7th International Conference on Advanced Software Engineering and Its Applications*. IEEE. 2014, p. 44-48.

[6] Ismail BUTUN, Patrik ÖSTERBERG et Houbing SONG. « Security of the Internet of Things : Vulnerabilities, attacks, and countermeasures ». In : *IEEE Communications Surveys & Tutorials* 22.1 (2019), p. 616-644.

[7] Faïcel CHAMROUKHI. « Classification supervisée : Les K-plus proches voisins ». In : *mémoire de fin d'étude, Université du Sud Toulon–Var* (2013).

[8] Rashmi Ravindra CHAUDHARI et Sonal Pramod PATIL. « Intrusion detection system : classification, techniques and datasets to implement ». In : *Int. Res. J. Eng. Technol.(IRJET)* 4.02 (2017), p. 1860-1866.

[9] Kejun CHEN et al. « Internet-of-Things security and vulnerabilities : Taxonomy, challenges, and practice ». In : *Journal of Hardware and Systems Security* 2.2 (2018), p. 97-110.

[10] Shanzhi CHEN et al. « A vision of IoT : Applications, challenges, and opportunities with china perspective ». In : *IEEE Internet of Things journal* 1.4 (2014), p. 349-359.

[11] VU CHEZHIAN, Dr RAMAR et Zaheer Uddin KHAN. « Security requirements in mobile ad hoc networks ». In : *International Journal of Advanced Research in computer and communication engineering* 1.2 (2012), p. 45-49.

[12] Louis COETZEE et Johan EKSTEEN. « The Internet of Things-promise for the future ? An introduction ». In : *2011 IST-Africa Conference Proceedings*. IEEE. 2011, p. 1-9.

[13] *Cross-validation : evaluating estimator performance*. `https://scikit-learn.org/stable/modules/cross_validation.html/`. Accessed : 03-06-2022.

[14] Li DA XU, Wu HE et Shancang LI. « Internet of things in industries : A survey ». In : *IEEE Transactions on industrial informatics* 10.4 (2014), p. 2233-2243.

[15] *F1-score, la synthèse entre precision et recall*. `https://kobia.fr/classification-metrics-f1-score/`. Accessed : 06-06-2022.

[16] Rafael C GONZALEZ. « Deep convolutional neural networks [Lecture Notes] ». In : *IEEE Signal Processing Magazine* 35.6 (2018), p. 79-87.

[17] Ian GOODFELLOW, Yoshua BENGIO et Aaron COURVILLE. *Deep learning*. MIT press, 2016.

[18] *Google Colab*. https://www.tutorialspoint.com/google_colab/what_is_google_colab.htm. Accessed : 12-06-2022.

[19] Frederic GRUAU. « Automatic definition of modular neural networks ». In : *Adaptive behavior* 3.2 (1994), p. 151-183.

[20] Jayavardhana GUBBI et al. « Internet of Things (IoT) : A vision, architectural elements, and future directions ». In : *Future generation computer systems* 29.7 (2013), p. 1645-1660.

[21] Wan Haslina HASSAN et al. « Current research on Internet of Things (IoT) security : A survey ». In : *Computer networks* 148 (2019), p. 283-294.

[22] Jonathan HUI, Pascal THUBERT et al. « Compression format for IPv6 datagrams over IEEE 802.15. 4-based networks ». In : (2011).

[23] *Intrusion Prevention System (IPS)*. https://www.geeksforgeeks.org/intrusion-prevention-system-ips/. Accessed : 26-05-2022.

[24] *IoT Intrusion Dataset*. https://sites.google.com/view/iot-network-intrusion-dataset/home. Accessed : 06-06-2022.

[25] Vikramaditya JAKKULA. « Tutorial on support vector machine (svm) ». In : *School of EECS, Washington State University* 37.2.5 (2006), p. 3.

[26] VVRPV JYOTHSNA, Rama PRASAD et K Munivara PRASAD. « A review of anomaly based intrusion detection systems ». In : *International Journal of Computer Applications* 28.7 (2011), p. 26-35.

[27] ROLAND KRAUSE et DAVID L WILD. « Gatsby Computational Neuroscience Unit, University College London, London, WC1N 3AR, UK E-mail : chuwei, zoubin (Qgatsby. ucl. ac. uk ». In : *Biocomputing 2006-Proceedings Of The Pacific Symposium*. World Scientific. 2005, p. 231.

[28] Alex KRIZHEVSKY et Geoff HINTON. « Convolutional deep belief networks on cifar-10 ». In : *Unpublished manuscript* 40.7 (2010), p. 1-9.

[29] Shyam Nandan KUMAR. « Review on network security and cryptography ». In : *International Transaction of Electrical and Computer Engineers System* 3.1 (2015), p. 1-11.

[30] Vinod KUMAR et Om Prakash SANGWAN. « Signature based intrusion detection system using SNORT ». In : *International Journal of Computer Applications & Information Technology* 1.3 (2012), p. 35-41.

[31] Nandakishore KUSHALNAGAR, Gabriel MONTENEGRO et Christian SCHUMACHER. « IPv6 over low-power wireless personal area networks (6LoWPANs) : overview, assumptions, problem statement, and goals ». In : (2007).

[32] In LEE et Kyoochun LEE. « The Internet of Things (IoT) : Applications, investments, and challenges for enterprises ». In : *Business horizons* 58.4 (2015), p. 431-440.

[33] Engin LELOGLU. « A review of security concerns in Internet of Things ». In : *Journal of Computer and Communications* 5.1 (2016), p. 121-136.

[34] Shancang LI, Li Da XU et Shanshan ZHAO. « The internet of things : a survey ». In : *Information systems frontiers* 17.2 (2015), p. 243-259.

[35] Yingqiu LIU, Wei LI et Yunchun LI. « Network traffic classification using k-means clustering ». In : *Second international multi-symposiums on computer and computational sciences (IMSCCS 2007)*. IEEE. 2007, p. 360-365.

[36] Hua-Dong MA. « Internet of things : Objectives and scientific challenges ». In : *Journal of Computer science and Technology* 26.6 (2011), p. 919-924.

[37] Francesca MENEGHELLO et al. « IoT : Internet of threats ? A survey of practical security vulnerabilities in real IoT devices ». In : *IEEE Internet of Things Journal* 6.5 (2019), p. 8182-8201.

[38] Tomas MIKOLOV et al. « Recurrent neural network based language model. » In : *Interspeech*. T. 2. 3. Makuhari. 2010, p. 1045-1048.

[39] Daniele MIORANDI et al. « Internet of things : Vision, applications and research challenges ». In : *Ad hoc networks* 10.7 (2012), p. 1497-1516.

[40] Leonardo NORIEGA. « Multilayer perceptron tutorial ». In : *School of Computing. Staffordshire University* (2005).

[41] Mark JL ORR et al. *Introduction to radial basis function networks*. 1996.

[42] Manish M PATEL et Akshai AGGARWAL. « Security attacks in wireless sensor networks : A survey ». In : *2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*. IEEE. 2013, p. 329-333.

[43] Fabian PEDREGOSA et al. « Scikit-learn : Machine learning in Python ». In : *the Journal of machine Learning research* 12 (2011), p. 2825-2830.

[44] J Ross QUINLAN. « Learning decision tree classifiers ». In : *ACM Computing Surveys (CSUR)* 28.1 (1996), p. 71-72.

[45] Syed RIZVI et al. « Securing the Internet of Things (IoT) : A security taxonomy for IoT ». In : *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE. 2018, p. 163-168.

[46] Jonathan ROUX. « Détection d'intrusion dans des environnements connectés sans-fil par l'analyse des activités radio ». Thèse de doct. Université Paul Sabatier-Toulouse III, 2020.

[47] S SANKAR et P SRINIVASAN. « Fuzzy logic based energy aware routing protocol for Internet of Things ». In : *International Journal of Intelligent Systems and Applications* 10.10 (2018), p. 11.

[48] Murat H SAZLI. « A brief review of feed-forward neural networks ». In : *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering* 50.01 (2006).

[49] Zach SHELBY, Klaus HARTKE et Carsten BORMANN. « The constrained application protocol (CoAP) ». In : (2014).

[50] Gur Amrit Pal SINGH et PK GUPTA. « Performance analysis of various machine learning-based approaches for detection and classification of lung cancer in humans ». In : *Neural Computing and Applications* 31.10 (2019), p. 6863-6877.

[51] P Gokul Sai SREERAM et Chandra Mohan Reddy SIVAPPAGARI. « Development of Industrial Intrusion Detection and Monitoring Using Internet of Things ». In : *International Journal of Technical Research and Applications* (2015).

[52] Harald SUNDMAEKER et al. « Vision and challenges for realising the Internet of Things ». In : *Cluster of European research projects on the internet of things, European Commision* 3.3 (2010), p. 34-36.

[53] Andrea Kidd TAYLOR, Kyle ESDAILLE et Jennifer AMES. « Integrated Pest Management Policies in America's Schools : Is Federal Legislation Needed ? » In : *The Toxic Schoolhouse* (2016).

[54] Alaa THARWAT. « Linear vs. quadratic discriminant analysis classifier : a tutorial ». In : *International Journal of Applied Pattern Recognition* 3.2 (2016), p. 145-180.

[55] Imtiaz ULLAH et Qusay H MAHMOUD. « A scheme for generating a dataset for anomalous activity detection in iot networks ». In : *Canadian Conference on Artificial Intelligence*. Springer. 2020, p. 508-520.

[56] Guido VAN ROSSUM et Fred L DRAKE JR. *Python tutorial*. T. 620. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[57] Berta Carballido VILLAVERDE et al. « Constrained application protocol for low power embedded networks : A survey ». In : *2012 sixth international conference on innovative mobile and internet services in ubiquitous computing*. IEEE. 2012, p. 702-707.

[58] Lizhi WANG et al. « A Deep-forest based approach for detecting fraudulent online transaction ». In : 120 (2021), p. 1-38.

[59] Kelly H ZOU, Kemal TUNCALI et Stuart G SILVERMAN. « Correlation and simple linear regression ». In : *Radiology* 227.3 (2003), p. 617-628.