République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي

ESI ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

المدرسة الوطنية العليا للإعلام الآلي
(المعهد الوطني للتكوين في الإعلام الآلي سابقا)
Ecole nationale Supérieure d' Informatique
ex. INI (Institut National de formation en Informatique)

Mémoire de fin d'études

Pour l'obtention du diplôme d'Ingénieur d'Etat en Informatique

Option : Systèmes Informatiques

Thème

APPRENTISSAGE AUTOMATIQUE POUR LA DÉTECTION D'ANOMALIES DANS UN RÉSEAU IOT.

Encadré par

Dr Meziani Lila

Réalisé par

Mahamdi Mohammed

Promotion : 2019/2020

Résumé

L'Internet des objets (IoT) est apparu comme la prochaine grande révolution technologique de l'informatique ces dernières années, avec un réseau en continuelle expansion. Les appareils IoT sont utilisés dans diverses types d'applications telles que : les voitures connectées, les maisons intelligentes, les soins de santé, le commerce de détail intelligent, la gestion de la chaîne logistique, etc. Les chercheurs prévoient près de 38 milliards d'appareils connectés à Internet en 2020.

La sécurisation des objets connectés est l'un des défis de la communauté scientifique. Insérer un module de sécurité constitue une tâche complexe étant donné que la plupart des objets connectés sont contraints en puissance et en mémoire. De ce fait, l'utilisation d'un système de détection d'anomalie se révèle comme la solution de rechange. La détection d'anomalies se base sur la comparaison du comportement de l'entité observée avec le comportement normal établie antérieurement.

Le présent travail a pour objectif, de concevoir et réaliser un outil basé sur l'apprentissage automatique pour la détection des anomalies dans un réseau Iot. Tout d'abord, nous avons introduit la sécurité en générale puis, nous avons défini les concepts de bases de l'architecture Iot, ses caractéristiques et l'aspect de sécurité lié à l'Iot. Ensuite, nous avons présenté l'apprentissage automatique. Nous avons cité les travaux existants et les techniques utilisées pour détecter les anomalies dans un réseau IoT essentiellement celle basées sur l'apprentissage automatique. A la fin, nous avons présenté la conception et la réalisation de notre modèle de détection. Nous évaluons ce modèle de détection sur le jeu de données AWID.

Mot clés : Apprentissage automatique, Iot, sécurité, Anomalies dans un réseau IoT.

Abstract:

The Internet of Things (IoT) has emerged as the next major information technology revolution in recent years, with a continually expanding network. IoT devices are used in a variety of applications such as: connected cars, smart homes, healthcare, smart retail, supply chain management, and more. Researchers predict nearly 38 billion devices connected to the Internet in 2020.

Securing connected objects is one of the challenges of the scientific community. Inserting a security module is a complex task since most of the connected objects are constrained in power and memory. As a result, the use of an anomaly detection system is proving to be the alternative. The detection of anomalies is based on the comparison of the behavior of the observed entity with the previously established normal behavior.

The present work aims to design and produce a tool based on machine learning for the detection of anomalies in an Iot network. In this work, we introduced security in general then we defined the basic concepts of Iot architecture, its characteristics and the security aspect linked to Iot then we presented machine learning. We cited existing work and the techniques used to detect anomalies in an IoT network, essentially that based on machine learning. At the end we showed the design and realization of our solution.

Key words: Automatic learning, Iot, security, Anomalies in an IoT network.

الملخص

برزت إنترنت الأشياء باعتبارها الثورة التكنولوجية الرئيسية التالية في مجال الحوسبة في السنوات الأخيرة ، وهي تعد شبكة آخذة في التوسع. تستخدم أجهزة إنترنت الأشياء في أنواع مختلفة من التطبيقات مثل: السيارات المتصلة ، والمنازل الذكية ، والرعاية الصحية ، وما إلى ذلك. يتوقع الباحثون ما يقرب من 38 مليار جهاز متصل بالإنترنت بحلول عام 2020.

يعد تأمين إنترنت الأشياء أحد تحديات المجتمع العلمي. يعد إدخال وحدة الحماية مهمة معقدة لأن معظم الاشياء المتصلة مقيدة بالطاقة والذاكرة. لذلك تبين أن استخدام نظام الكشف عن الاختراق هو البديل. يعتمد هذا النظام على مقارنة سلوك الاجهزة المرصودة بالسلوك العادي الذي تم إثباته سابقًا.

يهدف العمل الحالي إلى تصميم وإنتاج أداة تعتمد على التعلم الآلي لاكتشاف الحالات الشاذة في شبكة إنترنت الأشياء. في هذا العمل قدمنا الأمن بشكل عام ثم حددنا المفاهيم الأساسية لهندسة إنترنت الأشياء وخصائصها والجانب الأمني المرتبط بـها، ثم قدمنا التعلم الآلي. والتقنيات المستخدمة للكشف عن الحالات الشاذة في شبكة إنترنت الأشياء ، والتي تعتمد أساسًا على التعلم الآلي. في النهايةقدمنا حلنا المقترح.

الكلمات المفتاحية : التعلم الآلي ، الأمان ، الشذوذ في شبكة إنترنت الأشياء.

Table de matières

List de figures

Liste des tableaux

Liste des abréviations

AP : (Access point en anglais), le terme désigne un point d'accès sans fil, qui est un dispositif qui permet aux périphériques sans fil de se connecter à un réseau câblé ou au réseau Internet.

API : (Application Programming Interface en Anglais), le terme désigne une interface de programmation d'application, qui est une interface informatique avec un composant logiciel ou un système, qui définit comment d'autres composants ou systèmes peuvent l'utiliser. Il définit les types d'appels ou de requêtes qui peuvent être effectués, comment les effectuer, les formats de données à utiliser, les conventions à suivre, etc.

AWID : (Aegean Wi-Fi Intrusion Dataset), c'est une base de données axées sur la détection d'intrusion.

BSSID : (Basic service set identifiers), est utilisé pour décrire des sections d'un réseau local sans fil ou WLAN.

CPU : (Central Processing Unit en Anglais), le terme référence le processeur central qui est le circuit électronique qui exécute des instructions constituant un programme informatique.

RTS / CTS : (Clear To Send / Ready To Send) est un mécanisme utilisé par le protocole réseau sans fil 802.11 pour réduire les collisions de trames.

DA : (Detection Accuracy), c'est la Précision de détection.

DNS : (Domain Name System en Anglais), c'est le service informatique distribué utilisé pour traduire les noms de domaine Internet en adresse IP ou autres enregistrements.

Dos : (Denial of Service Attack en anglais), c'est une attaque informatique ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser.

EPROM : (Erasable Programmable Read-Only Memory en Anglais), c'est un type de mémoires électroniques mortes programmables par l'utilisateur et dont la programmation n'est pas irréversible.

ESSID : (Extended Service Set Identifier en anglais), c'est l'identifiant pour réseaux.

FN : (False Negative), le faux négative.

FP : (False Positif), le faux positive.

FRAM : (Ferroelectric Random-access Memory), c'est type de mémoire qui combine le stockage de données non volatile avec les hautes performances de la RAM.

FTP : (File Transfer Protocol en anglais), c'est un protocole de communication destiné au partage de fichiers sur un réseau TCP/IP.

GPRS : (General Packet Radio Service), c'est une norme pour la téléphonie mobile dérivée du GSM et complémentaire de celui-ci, permettant un débit de données plus élevé.

GPS : (Global Positioning System), c'est le système mondial de positionnement.

GSM : (Global System for Mobile Communications), c'est une norme numérique de seconde génération pour la téléphonie mobile.

IDS : (Intrusion Detection System), c'est Un système de détection d'intrusion.

IP : Internet Protocol.

IPv6 : la version 6 du protocole IP.

KNN : (k-Nearest Neighbors), c'est la méthode des k plus proches voisins.

LAN : (Local Area Network), c'est un réseau local.

LOOCV : (Leave-One-Out Cross-Validation), c'est La validation croisée.

LReLU : (Leaky ReLU), est une modification de ReLU qui remplace la partie zéro du domaine dans [-∞, 0] par une faible pente.

LTE : (Long Term Evolution) est une évolution de plusieurs normes de téléphonie mobile.

LTE_A : (LTE-Advanced) est une norme de réseau de téléphonie mobile de quatrième génération.

OISF : (Open Information Security Foundation), c'est l'agence nationale de la sécurité des systèmes information.

PReLU : (Parametric ReLU), Basé sur les mêmes idées que LReLU, PReLU a les mêmes objectifs : augmenter la vitesse d'apprentissage en ne désactivant pas certains neurones.

R2L, **U2R**: (Remote To Local Attack / User To Root Attack), ce sont des types d'attaques.

ReLU : (Rectified Linear Unit), c'est une fonction d'activation définie comme la partie positive de son argument : où x est l'entrée d'un neurone.

Réseaux 802.11 : est un ensemble de normes concernant les réseaux sans fil locaux (le Wi-Fi). Il a été mis au point par le groupe de travail 11 du comité de normalisation LAN/MAN de l'IEEE (IEEE 802).

RFID : (Radio-Identification) est une méthode pour mémoriser et récupérer des données à distance.

SAE: (Sparse Autoencoder)

SOHO: (Small Office and Home Office).

SSH : (Secure Shell), est à la fois un programme informatique et un protocole de communication sécurisé.

SSID : (Service Set Identifier), c'est le nom d'un réseau sans fil selon la norme IEEE 802.11.

SVM : (Support Vector Machine), c'est un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression.

TN : (True Negative), le vrai négatif.

TP : (True Positive), le vrai positive.

TPR : (True positive rate), le taux de vrai positive.

UIT : Union internationale des télécommunications.

USB : (Universal Serial Bus), est une norme relative à un bus informatique en série qui sert à connecter des périphériques informatiques.

VoIP : (Voice Over Internet Protocol) c'est la transmission de la voix via Internet.

WEB : (World Wide Web), est un système hypertexte public fonctionnant sur Internet.

WEP : (Wired Equivalent Privacy), est un protocole pour sécuriser les réseaux sans fil de type Wi-Fi.

WIFI : (Wireless Fidelity) désigne un protocole de communication sans fil.

WPA, **WPA2** :(Wi-Fi Protected Access) ce sont des mécanismes pour sécuriser les réseaux sans-fil de type Wi-Fi.

WSN : (Wireless Sensor Network), c'est un réseau ad hoc d'un grand nombre de nœuds, qui sont des micro-capteurs capables de recueillir et de transmettre des données d'une manière autonome.

XGBoost : (eXtreme Gradient Boosting) , c'est un algorithme d'apprentissage automatique.

Introduction générale

Contexte

L'Internet des objets a pour objectif de permettre aux différents objets de se connecter n'importe quand, n'importe où en utilisant n'importe quel chemin et n'importe quel service. L'Internet n'a jamais connecté autant de systèmes informatiques qu'aujourd'hui, et le nombre des objets connectés est en croissance.

Ces objets n'intégrant que rarement des mécanismes de protection contre les attaques. Donc la sécurité de leurs utilisateurs, qu'ils soient des individus ou des entreprises est en menace.

Les objets connectés peuvent prendre des formes diverses, allant des systèmes d'ouverture des portes à des systèmes de gestion des usines, ils ont des ressources parfois très limitées, ce qui ne permet pas toujours de mettre en place des mécanismes pour gérer la sécurité.

Contrairement à l'informatique traditionnelle, l'absence d'un système d'exploitation centrale ou un micrologiciel commun rend l'implémentation des protocoles de sécurité très difficile.

Problématique

La mise en place d'une solution de sécurité applicable à l'ensemble des objets intelligents est très difficile à cause de variétés des protocoles, différence des capacités hardware et l'absence de mises à jour par les constructeurs dans la majorité des cas.

Les travaux effectués en termes de détection d'intrusion se concentrent essentiellement sur des analyses du flux réseau, puisqu'il n'y a pas besoin de considérer les ressources matérielles des objets ni même les applications qui sont exécutées dessus.

Objectifs

Afin de répondre à la problématique posée ci-dessus, une solution doit être proposée tout en répondant aux objectifs suivants :

1. Comprendre les caractéristiques de l'écosystème des objets connectés.
2. La mise en place d'un programme capable d'analyser le flux réseau en utilisant des

techniques d’apprentissage automatique.

3. Implémenter et optimiser divers algorithmes d’apprentissage machine pour la classification et la détection des attaques.
4. Tester l’efficacité du mécanisme de sécurité proposé.

Chapitre I

Sécurité et internet des objets

1 Définitions et concepts de base

Afin d'étudier le domaine de l'internet des objets, en premier temps, nous allons aborder les concepts de base. Dans cette section, nous définissons les concepts essentiels à la compréhension de ce document.

1.1 La sécurité informatique

On peut définir la sécurité d'une manière générale, comme étant une situation tranquille qui résulte de l'absence réelle de danger. Pour la sécurité informatique, c'est l'ensemble de techniques et d'outils collaboratifs permettant de garantir les objectifs essentiels de la sécurité : confidentialité, intégrité, et disponibilité. Ces outils peuvent être organisationnels, matériels, logiciels, ou juridiques dont le but est de protéger les informations et les systèmes contre l'accès, l'utilisation malveillante ou non autorisée, la modification, la divulgation, et la destruction des données et connaissances.

1.1.1 Buts de la sécurité

Améliorer la sécurité face aux risques identifiés. Pour pouvoir assurer :

• **La disponibilité** *:* aptitude du système à remplir une fonction dans des conditions prédéfinies d'horaires, de délai ou de performance.

• **L'intégrité** : garantit que l'information n'est pas modifiée sauf par une action volontaire et autorisée.

• **La confidentialité** : l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé.

1.1.2 Types de sécurité

1. Sécurité des données : concerne exclusivement les données à l'intérieur d'un système ; (cryptographie et théorie des codes).

2. Sécurité des réseaux : concerne les données quand elles transitent entre des systèmes, dans un environnement distribué ou par un réseau.
3. La sécurité physique.
4. La sécurité personnelle.
5. La sécurité procédurale (procédures informatiques...).
6. La sécurité des émissions physiques (écrans, câbles d'alimentation, courbes de consommation de courant...).
7. La sécurité des systèmes d'exploitation.

1.2 La Menace

En informatique, une menace est une cause potentielle d'incident, qui peut résulter en un dommage au système ou à l'organisation (définition selon la norme de sécurité des systèmes d'information ISO/CEI 27000).

1.3 Intrusion

Il s'agit d'intrusion lorsqu'une personne pénètre dans un espace qui lui est normalement interdit d'accès. Dans les systèmes informatiques, nous référerons à intrusion par toute pénétration dans un système informatique ayant pour but de mettre à mal sa confidentialité, son intégrité ou sa disponibilité.

La détection d'intrusion rassemble toutes les techniques mises en œuvre pour alerter les utilisateurs du système informatique visé par une intrusion sur le fait qu'ils sont en train d'être ciblés par un attaquant.

1.3.1 Les types d'intrusion

Selon les classifications les plus connues (Pharate, Bhat, Shilimkar, & Mhetre, 2015), il existe deux types d'intrusions :

A) Intrusion interne / externe

Une intrusion est dite interne si elle a été commise par un attaquant interne ayant plus de privilèges qu'un utilisateur ordinaire externe du système. Elle est dite externe si elle a été commise par un attaquant externe du système.

Les intrusions internes sont les intrusions les plus difficiles à détecter, car les ressources du système ne sont pas altérées par ce type d'intrusion.

B) Intrusion passive / active

Les intrusions passives sont celles qui visent à avoir accès à des ressources confidentielles sans les modifier, à titre d'exemple, le « **Sniffing** » qui consiste à analyser le trafic d'un réseau.

Les intrusions actives, elles sont les intrusions qui touchent l'intégrité des ressources du système (elles modifient l'état des ressources), par exemple une intrusion qui modifie le contenu d'un fichier.

1.4 L'Attaque

L'attaque est une action malveillante qui consiste à tenter de contourner les fonctions et les mesures de sécurité d'un système informatique.

1.4.1 Les types d'attaques

Dans cette partie, nous allons aborder au premier lieu les menaces les plus courantes puis nous présenterons les menaces dans le domaine de l'Iot, c'est-à-dire les menaces réseaux.

1.4.1.1 Les attaques les plus courantes

Ces menaces sont variées, mais peuvent être facilement identifiées par leur mode d'opération. Nous les présentons dans ce qui suit :

A) Logiciel malveillant (Malware)

Un logiciel malveillant est un terme générique englobant ces différentes menaces informatiques visant toutes à nuire à un système. Un logiciel malveillant peut corrompre, effacer ou voler les données des appareils et réseaux d'une entreprise. Il peut subtiliser des données confidentielles, comme les numéros de carte de crédit de clients.

B) Virus

Un virus informatique est un type de logiciel malveillant caché dans un logiciel légitime. Chaque fois qu'un utilisateur ouvre le logiciel infecté, il permet au virus de se propager. Il agit

discrètement et se réplique à une vitesse fulgurante grâce aux échanges de données, que ce soit par une clé USB ou un réseau informatique.

C) Ver

Un ver informatique est un logiciel malveillant qui se reproduit sur plusieurs ordinateurs en utilisant un réseau informatique comme Internet. Il a la capacité de se dupliquer une fois qu'il a été exécuté.

D) Cheval de Troie

Un cheval de Troie est un type de programme malveillant se faisant passer bien souvent pour un logiciel authentique (Iliev, Kyurkchiev, Rahnev, & Terzieva, 2019). Les chevaux de Troie peuvent être utilisés par des cybercriminels et des pirates informatiques pour accéder aux systèmes des utilisateurs. Ces derniers sont généralement incités, par le biais d'une technique d'ingénierie sociale, à charger et exécuter des chevaux de Troie sur leurs systèmes. Une fois activés, les chevaux de Troie peuvent permettre aux cybercriminels d'espionner, de dérober les données sensibles et d'accéder au système à l'aide d'un backdoor. Ces actions peuvent être les suivantes :

- Suppression de données
- Blocage de données
- Modification de données
- Copie de données
- Perturbation des performances des ordinateurs ou des réseaux informatiques

Contrairement aux virus et aux vers informatiques, les chevaux de Troie ne s'auto-répliquent pas.

E) Hameçonnage

L'hameçonnage est un type de fraude sur Internet visant à obtenir par tromperie des informations sur les destinataires. Cela inclut le vol de mots de passe, de numéros de carte de crédit et d'autres données confidentielles.

Les messages d'hameçonnage revêtent généralement la forme de notifications envoyées par des banques, des fournisseurs, des systèmes de paiement en ligne ou d'autres organismes. La

notification va inciter le destinataire, pour une raison ou pour une autre, à saisir ou à mettre à jour d'urgence ses données personnelles.

F) Spam

D'après Kaspersky Lab, un spam est un e-mail **anonyme**, **indésirable** et **envoyé en masse**.

Examinons de plus près les termes de cette définition :

Anonyme : le vrai spam est envoyé sous des adresses volées à l'insu d'utilisateurs tiers pour masquer le véritable expéditeur.

Mailing de Masse : le vrai spam est envoyé en très grand quantité. Les spammeurs font de l'argent grâce au petit pourcentage de réponses. Pour que le spam soit rentable, le mail initial doit être envoyé en masse.

Indésirable : les listes de mailing, newsletters et autres matériels publicitaires auxquels les internautes ont souscrit, peuvent ressembler à des spams mais sont en fait des e-mails légitimes. En d'autres termes, un mail peut être considéré comme spam ou comme mail légitime selon que l'utilisateur ait choisi de le recevoir ou non.

1.4.1.2 Les menaces dans les réseaux Wi-Fi

Dans cette section, nous présentons les menaces réseaux (Kolias, Kambourakis, Stavrou, & Gritzalis, 2015).

A) Deauthentication

Elle est considérée comme l'attaque la plus puissante des réseaux 802.11 en raison de sa simplicité et efficacité, elle est basée sur le fait que les paquets de deauthentication sont transmis sans protection et qu'ils peuvent facilement être usurpés par une entité mal motivée.

Lors de la réception de tels paquets, le client doit immédiatement abandonner le réseau sans aucune action supplémentaire, La figure 1 résume le schéma de cette attaque :

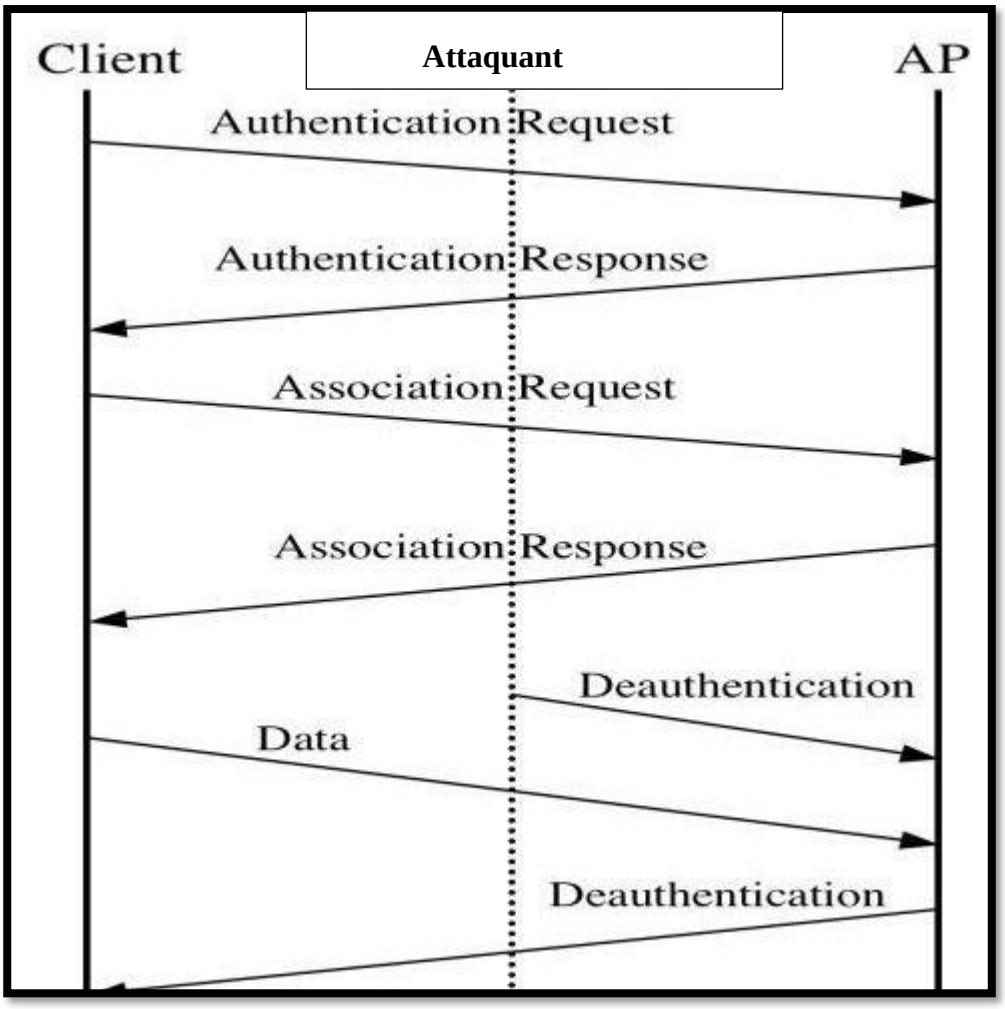

Figure 1 Explication de l'attaque Deauthentication , source (Noman, Abdullah, & Mohammed, 2015)

B) Disassociation

L'attaque de dissociation est très similaire à celle de deauthentication dans la méthodologie, la facilité d'utilisation et les effets.

La différence est la durée de la perte de service pour le client. Elle est très courte et l'attaque est moins efficaces car le client peut revenir d'un état non associé à un état associé.

C) Authentication Request Flooding Attack

Elle est basée sur le fait que le nombre maximum de clients pouvant être maintenus dans la table d'association du point d'accès du client est limité.

L’attaquant devra émuler un grand nombre de clients et envoyer un paquet d'authentification au nom de chacun. Après que la table d'association de clients du point d’accès déborde de fausses entrées, le point d’accès ne pourra plus associer de clients légitimes.

D) Fake Power Saving

En abusant du mécanisme d'économie d'énergie, cette attaque incite fondamentalement le point d’accès à penser qu'une station spécifique est tombée en mode veille.

Le mécanisme de gestion de l'alimentation dans les réseaux 802.11 permet de réduire la consommation d'énergie d'une station en plaçant leurs adaptateurs réseau en mode d'économie d'énergie. La transition vers le mode veille se fait généralement lorsque le client passe un certain temps sans communication.

Cette attaque a lieu en envoyant une trame de données nulle en mettent le champ ’ **Power Save**’ à 1. Le point d’accès acceptera ce message et commencera immédiatement à mettre en mémoire tampon toutes les trames de données destinées à cette station.

E) CTS Flooding

En exploitant le mécanisme RTS/CTS, cette attaque a lieu en envoyant des trames CTS, l’attaquant envoies des trames à lui-même ou à une autre station, forçant le reste des stations du réseau à reporter continuellement leur transmission.

F) RTS Flooding

Cette attaque exploite aussi le mécanisme RTS/CTS. Elle fonctionne de manière opposée à celle de CTS Flooding. L’attaquant transmet un grand nombre de trames RTS usurpées avec éventuellement une grande fenêtre de durée de transmission, dans l'espoir de monopoliser le support sans fil d'une manière qui forcera éventuellement les autres stations à se retirer de la transmission.

G) Beacon Flooding

Ce scénario d'attaque se base sur la création d’une confusion de connectivité pour un client donné, Il y a transmission d’un flux constant de fausses balises qui annoncent des ESSID non existants faite par l’attaquant. Après un certain temps, les réseaux sans fil disponibles sont si nombreux que l'utilisateur est totalement confus et perdu dans une grande liste de réseaux.

H) Probe Request Flooding

Cette attaque est basée sur le fait que selon la norme 802.11, un point d'accès est obligé de répondre à chaque message **'probe request'** avec un message de **'probe response'**, ces messages contiennent des détails sur le réseau et les capacités du point d'accès.

L'attaque a lieu en envoyant un flux constant de faux paquets de **'probe request'**. Si cela se fait en grand volume et pendant des périodes prolongées, le point d'accès ne pourra pas servir ses clients légitimes.

I) Probe Response Flooding

Cette fois, l'attaquant surveille les messages de **''probe request'** provenant de clients valides et en agissant comme un point d'accès. Ensuite, il transmet un flux de **'probe response'** faux et inexactes aux stations. Ces messages contiennent de fausses informations sur le réseau, ce qui induit la station en erreur de recevoir la réponse du point accès valide et l'empêche de se connecter à un point d'accès.

J) Evil_twin

Cette attaque est possible du fait que :

- Plusieurs points d'accès avec le même ESSID peuvent exister dans la même zone.
- Dans de telles situations, le client préférera se connecter à celui avec le signal le plus fort sans tenir compte du BSSID du point d'accès légitime.

L'attaque commence par le clonage d'un SSID réseau et fait semblant d'être un point d'accès local. Un utilisateur sans méfiance se connecte alors au point d'accès en le croyant être le vrai. À l'insu de l'utilisateur, un attaquant intercepte tout le trafic entre l'utilisateur et l'hôte, tout en volant des données personnelles. Cela peut entraîner le vol d'informations d'identification et d'informations sensibles.

La figure 2 explique le schéma de cette attaque :

K) Rogue Access Point

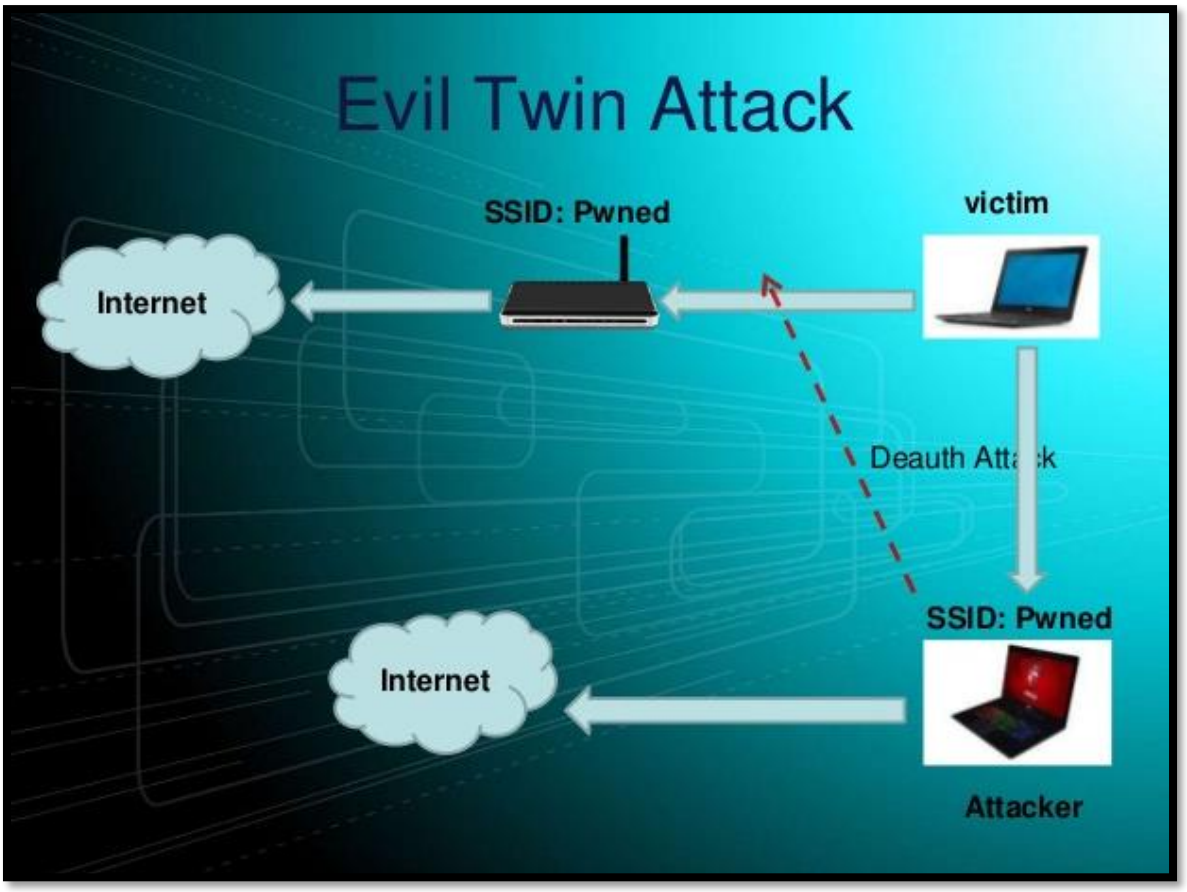

Figure *2* ***Explication de l'attaque Evil_twin, source [web7]***

Il s'agit d'un AP non autorisé activé dans les locaux de l'entreprise, de la maison ou du bureau.de tels paramètres peuvent être générés par des utilisateurs sans l'autorisation de l'administrateur réseau afin de rendre une politique de sécurité plus pratique pour eux.

1.4.2 Les contres mesures

Plusieurs outils permettant la détection, l'anticipation et la désinfection des intrusions existent dans la littérature et sont utilisés dans la pratique. Dans cette partie, nous allons citer les outils les plus connus :

1.4.2.1 Antivirus

Un antivirus est un logiciel informatique destiné à identifier et à effacer des logiciels malveillants, L'antivirus analyse les fichiers entrants (fichiers téléchargés ou courriers électroniques) et, périodiquement, la mémoire vive de l'ordinateur et les périphériques de stockage comme les disques durs, internes ou externes, les clés USB et les cartes à mémoire Flash.[Web1].

La détection d'un logiciel malveillant peut reposer sur trois méthodes :

- Reconnaissance d'un code déjà connu (appelé signature) et mémorisé dans une base de données.
- Analyse du comportement d'un logiciel (méthode heuristique).
- Reconnaissance d'un code typique d'un virus.

1.4.2.2 Pare-feu

Érigé entre un ordinateur et sa connexion à un réseau externe ou sur le Web, un pare-feu décide quel trafic réseau est autorisé à traverser et quel trafic est jugé dangereux. Sa fonction principale est de filtrer le bon du mauvais, les éléments fiables des éléments non sécurisés.

Le filtrage des paquets par un Pare-feu se fait généralement selon l'adresse IP source et destination et les propriétés de protocole de transport (les Port source et destination) et les informations des différents protocoles.

1.4.2.3 Les IDS

Selon (Debar, Dacier, & Wespi, 1999), "un IDS C'est un système dont la mission principale est d'analyser les évènements générés dans un environnement donné afin de décider s'ils sont susceptibles de produire une attaque ou non".

Il existe deux types D'IDS, Les systèmes de détection d'intrusions machine (HIDS) et les systèmes de détection d'intrusions réseau (NIDS). Sur les systèmes HIDS, des applications de protection, comme le pare-feu, les antivirus et les programmes de détection des logiciels espions, sont installées sur tous les ordinateurs d'un réseau à accès bidirectionnel à un environnement extérieur, comme Internet. Et Sur les systèmes NIDS, un logiciel de protection est installé uniquement à des points spécifiques, comme les serveurs qui établissent l'interface entre l'environnement extérieur et le segment de réseau à protéger.

1.4.2.3.1 Pourquoi utiliser un IDS

En peut utiliser un IDS dans les cas suivants :

- Détecter les attaques et faire des contremesures pour y remédier.
- Aider les experts de sécurité à connaitre les activités suspectes d'un système.
- Identifier les vulnérabilités d'un système.
- Automatiser la tâche de l'opérateur de sécurité.

1.4.2.3.2 Étapes suivies dans la détection d'intrusions

La détection d'une intrusion par l'IDS se fait selon trois étapes principales à savoir la collecte des données, la normalisation et enfin la reconnaissance de l'intrusion. La figure 3 illustre le fonctionnement d'un IDS.

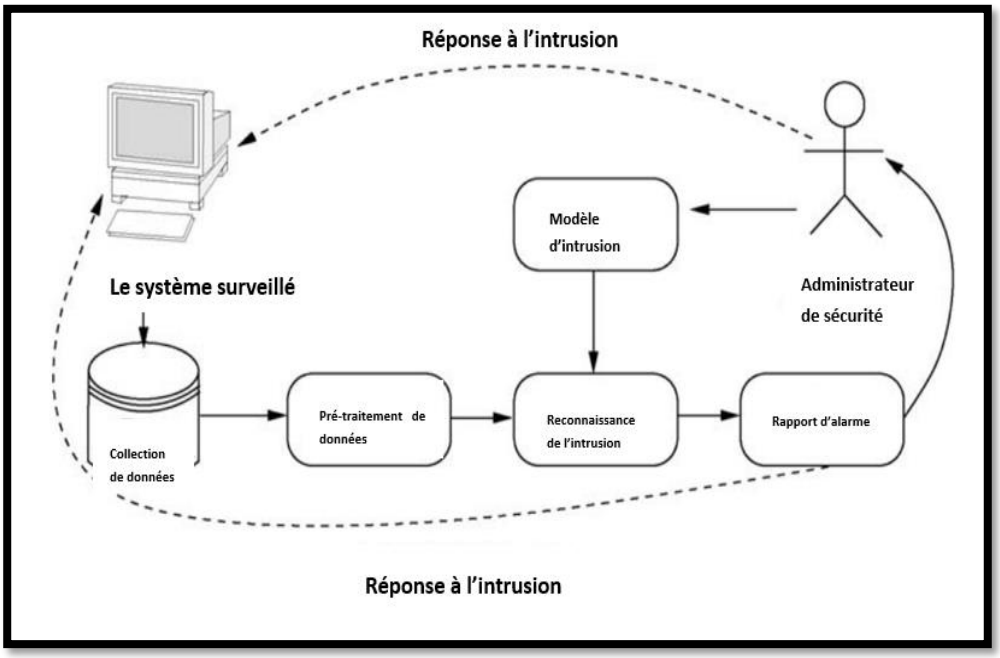

Figure 3 Étapes suivies dans la détection d'intrusions, source (Wu & Banzhaf, 2010)

1. **La collecte des données** : dans cette étape, nous collectons les données transitant par le système surveillé en utilisant des capteurs placés aux des points sensibles du système.
2. **La normalisation (prétraitement)** : Les données collectées par un IDS, dans la première étape, sont souvent de nature hétérogène, car elles proviennent de sources

différentes. Ainsi, pour faciliter le traitement et la manipulation de ces données, l'IDS les normalise toutes dans un format unique.

3. **Reconnaissance de l'intrusions** : En utilisant des modèles d'intrusion qui indiquent comment détecter une intrusion, l'IDS analyse les données normalisées. S'il trouve une donnée ou une séquence de données qui correspond à l'un des modèles d'intrusion, alors une nouvelle alerte est signalée dans le rapport d'alertes.

1.4.2.4 L'évaluation des IDSs

Il est nécessaire de savoir quelle sont les critères utilisés pour mesurer l'efficacité pour la détection d'intrusions d'un IDS.

Selon (Porras & Valdes, 1998) et (Debar, Dacier, & Wespi, 2000) il existe cinq critères pour évaluer un IDS :

a) La précision

Un IDS est considéré comme idéal lorsque celui-ci détecte toutes les intrusions parfaitement sans générer de faux positifs. La précision est le taux des alertes qui correspondent à des intrusions réelles, la non-précision est le taux de faux positif produit par le système. La précision peut se calculer directement en utilisant la définition, comme elle peut se calculer à partir de la Non-précision, car elles sont reliées par l'équation 1

$$\boldsymbol{Precision = 1 - Nonprecision} \qquad \textbf{(Eq 1)}$$

b) La performance

C'est le débit avec lequel un IDS analyse les informations qui transitent par le système surveillé. Les IDS à temps réel doivent avoir une performance très élevée pour détecter les intrusions le plus rapidement possible.

c) La complétude

Un IDS complet est un IDS capable de détecter tout type d'attaque, en d'autres termes, un IDS complet est un IDS qui ne présente aucun faux négatif.

La non-complétude survient, lorsqu'un IDS ne parvient jamais à détecter un type d'attaque donné. Un IDS complet n'existe pas en pratique, car de nouvelles attaques apparaissent chaque jour.

Il est très difficile d'évaluer cette mesurer parce qu'il est impossible d'avoir une connaissance absolue sur tous les types d'attaques. En pratique cette mesure est évaluée avec une base de données considérée complète qui contient un nombre important de types d'attaques. Si l'IDS à évaluer arrive à détecter tous les types d'attaques qui figurent dans cette base, alors il est qualifié de complet.

d) La tolérance aux fautes

L'IDS lui-même doit être résistant aux attaques qui endommagent le système sur lequel il opère. En d'autres termes, l'IDS ne doit pas être affecté par les effets des attaques contre le système qu'il surveille. Cette mesure est essentielle pour tout IDS, car si ce dernier est vulnérable, alors il suffit pour l'attaquant d'exploiter les failles de l'IDS pour l'endommager et le paralyser, pour qu'il puisse lancer par la suite des attaques contre le système en question sans être détecté.

e) La rapidité

L'IDS ne doit pas seulement analyser rapidement le trafic qui circule dans le système à surveiller et détecter efficacement les intrusions, mais il doit aussi réagir aux intrusions le plus rapidement possible. Cette mesure permet de mesurer la rapidité de l'IDS pour fournir les alertes relatives aux différentes attaques.

1.4.2.5 Mécanismes de détection utilisé par les IDS

Les mécanismes de détection utilisé par les IDS sont de trois types : détection des abus, détection des anomalies et détection hybride. Dans l'approche de détection des abus, IDS maintient un ensemble de la base de connaissances (règles) pour détecter les types d'attaque connus.

Le tableau 1 montre la différence entre la détection d'abus et la détection d'anomalie (Mishra, Varadharajan, Tupakula, Pilli, & Tutorials, 2018) :

Tableau I La différence entre la détection d'abus et la détection d'anomalie

Détection d'abus	Détection d'anomalie
• Il modélise les schémas / signatures d'attaque connus pour détecter une activité malveillante. Une correspondance de modèle entrant avec des profils d'attaque existants est déclarée suspecte.	• Il utilise le profil de comportement normal établi du système. Une incompatibilité de modèle entrant avec le profil normal existant est déclarée suspecte.
• « Correspondance de signature » est une approche de détection des abus très populaire, qui connaît un succès commercial.	• « L'apprentissage statistique » est une approche de détection d'anomalies très populaire et les chercheurs travaillent toujours dans ce sens.
• Les approches d'apprentissage automatique supervisés sont bien adaptées pour ce type de détection.	• Les approches d'apprentissage machine semi ou non supervisées sont bien adaptées à la détection d'anomalies.
• Impossible de détecter les attaques inconnues.	• Capables pour détecter les attaques inconnues.
• Faible occurrence de faux positifs.	• Occurrences élevées de faux positifs.
• Le défi consiste à maintenir à jour une base de données de toutes les attaques et les signatures.	• Le défi consiste à différencier l'attaque et à développer un comportement normal.

Les techniques de détection des abus peuvent être largement classées en techniques basées sur les connaissances et d'autres basées sur l'apprentissage automatique. Dans la première technique, le trafic réseau ou les données d'audit d'hôte (telles que les traces d'appels système) sont comparés à des règles prédéfinies ou à des modèles d'attaque. Les techniques basées sur les connaissances peuvent être classées en trois types :

1. Correspondance des signatures
2. Analyse de la transition des États
3. Systèmes experts fondés sur des règles

Les approches de détection hybrides intègrent une approche de détection des abus et des anomalies pour détecter les attaques.

1.4.3 Le pare-feu et l'IDS

Un pare-feu contrôle le trafic entrant ou sortant d'un réseau en fonction de l'adresse source ou de destination. Il modifie le trafic selon les règles du pare-feu. Les pares-feux sont également limités au nombre d'états disponibles et à leur connaissance sur les hôtes. (Vasilomanolakis, Karuppayah, Mühlhäuser, & Fischer, 2015)

Un IDS est un type d'outil de sécurité qui surveille le trafic réseau et analyse le système pour détecter les activités suspectes et alerte le système ou l'administrateur réseau.

2 L'Internet des objets

L'IoT est l'acronyme de « Internet Of Things », Internet des Objets en français. Le terme IoT est apparu la première fois en 1999 dans un discours de Kevin ASHTON, un ingénieur britannique. Il servait à désigner un système où les objets physiques sont connectés à Internet. Il s'agit également de systèmes capables de créer et transmettre des données afin de créer de la valeur pour ses utilisateurs à travers divers services. (Aswale, Shukla, Bharati, Bharambe, & Palve, 2019).

2.1 Objets connectés

Selon l'UIT (Union Internationale des Télécommunications), l'Internet des Objets est défini comme « une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets physique ou virtuels grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution ».

Au fil du temps, le terme a évolué et il englobe maintenant tout l'écosystème des objets connectés. Cet écosystème englobe, des fabricants de capteurs, des éditeurs de logiciels, des opérateurs historiques ou nouveaux sur le marché, des intégrateurs…

2.2 Les caractéristiques de l'internet des objets

Les caractéristiques fondamentales de l'IoT peuvent être résumées par les points suivants (Zikria, Yu, Afzal, Rehmani, & Hahm, 2018) :

a) **Inter connectivité** : le tout peut être interconnecté avec l'information globale et l'Infrastructure de communication.

b) **Hétérogénéité** : les objets de l'Iot sont hétérogènes, basés sur différentes plates-formes

matérielles et réseaux. Ils peuvent interagir avec d'autres appareils via différents réseaux.

c) **Le changement dynamique** : l'état des appareils change de manière dynamique. Par exemple, changement d'état connecté / déconnecté.de plus, le nombre d'appareils peut changer dynamiquement.

d) **L'énorme échelle** : Le nombre d'appareils qui doivent être gérés est plus grand que les appareils connectés au réseau Internet actuel.

e) **Connectivité** : la connectivité permet aux objets de L'Iot d'accéder au réseau et échanger des informations.

f) **Sécurité** : la nécessité de prendre en charge une politique de sécurité, cela inclut la sécurité des personnes, des données et des réseaux.

g) **Les services liés aux objets** : L'Iot doit imposer des contraintes sur son utilisation. Elle doit garantir des services tel que protection de la vie privée.

2.3 Architecture de l'internet des objets

L'architecture d'un système IoT typique est composée de plusieurs couches ou niveaux qui communiquent entre eux(Patel & Patel, 2016). L'architecture Iot est constituée de différentes couches de technologies (Layers en Anglais). Elle sert à illustrer les relations entre les différentes technologies et à communiquer les configurations de déploiements dans les différents scénarios. Il existe plusieurs architectures proposées et tous les projets n'adoptent pas une même architecture. La figure 4 montre un des modèles proposés pour l'architecture d'un système IoT composé de 4 couches.

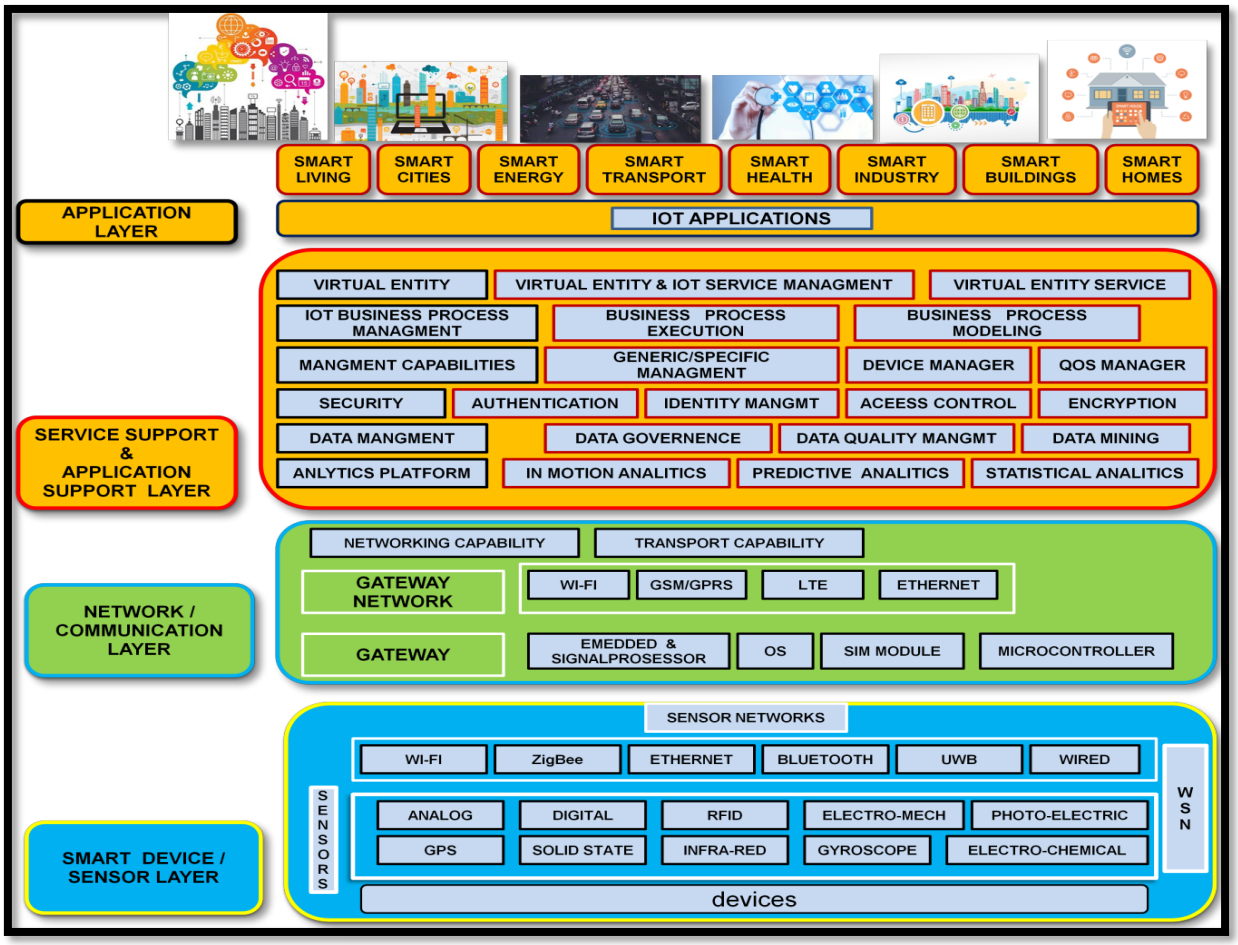

Figure *4* ***L'architecture de l'Iot en couche, source (Patel & Patel, 2016)***

Comme c'est présenté dans la figure 4 l'architecture de Iot est divisé en quatre couches, la couche Objet, la couche réseau, la couche gestion de service et la couche application. Nous les détaillons dans les sections qui suivent.

2.3.1 Couche Objet (Smart device)

C'est la couche la plus basse, elle regroupe l'ensemble des objets intelligents avec les capteurs. Les capteurs permettent de faire la liaison physique-numérique, ils ne sont pas tous similaires en termes de portée et de volume de données à transférer.

Les capteurs ont la capacité de prendre des mesures telles que la température, la qualité de l'air, l'humidité, la vitesse, la pression, le débit, le mouvement, l'électricité, etc. Dans certains cas, un capteur mesure la propriété physique et la convertit en signal.

La classification des capteurs peut se faire en fonction de leur objectif, par exemple les capteurs environnementaux, capteurs corporels, capteurs pour appareils ménagers et capteurs télématiques pour véhicules, etc.

Pour connecter ces capteurs, plusieurs technologies peuvent être utilisées, tel que : LAN, WIFI, Bluetooth…Etc. Les WSNs sont les plus utilisés.

2.3.2 La couche réseaux

Les données collectées par la première couche nécessitent un moyen de transport efficace par une infrastructure réseau pour arriver à la troisième couche.

Multiples réseaux avec diverses technologies et protocoles d'accès sont nécessaires pour travailler les uns avec les autres dans une configuration hétérogène. Ces réseaux peuvent prendre la forme de modèles privé, public ou hybride et sont conçus pour prendre en charge les exigences de communication en matière de latence, de bande passante, performance et sécurité.

2.3.3 La couche gestion de service

Le service de gestion rend possible le traitement des informations par l'analyse, les contrôles de sécurité, la modélisation des processus et la gestion des périphériques.

L'Iot associe la connexion et l'interaction d'objets et de systèmes en fournissant des informations sous la forme de données contextuelles ou d'événements.

Le rôle de cette couche est la formulation des logiques de décision et déclencher des processus interactifs et automatisés pour permettre un système Iot plus réactif.

2.3.4 La couche application

La couche application couvre plusieurs domaines tel que : le transport, le bâtiment, l'agriculture, l'industrie, la santé, l'interaction utilisateur, la culture et le tourisme, l'environnement et énergie, etc.

2.4 Les technologies à base de l’internet des objets

Avec l'Internet des objets, la communication est étendue via Internet à toutes les choses qui nous entourent. Les objets connectés sont beaucoup plus que la communication machine à machine, réseaux de capteurs sans fil, réseaux de capteurs, 2G / 3G / 4G, GSM, GPRS, RFID, WI-FI, GPS, etc.

Ses derniers sont considérés comme étant des technologies qui font des applications Iot possible. L’Iot se base sur plusieurs technologies clés qui peuvent être regroupées en trois catégories(Vorakulpipat, Rattanalerdnusorn, Thaenkaew, & Hai, 2018)

- Les technologies qui permettent aux objets d’acquérir des connaissances contextuelles.
- Les technologies qui permettent aux ces objets de traiter ces informations.
- Les technologies pour améliorer la sécurité et la confidentialité.

Les deux premières catégories peuvent être groupées dans une seule catégorie et la troisième catégorie n’est pas une exigence fonctionnelle.

L’Iot n’est pas une technologie unique, mais c’est un mélange de différentes technologies matérielles et logicielles. Elle fournit des solutions basées sur l'intégration de la technologie de l'information, qui fait référence au matériel et logiciel utilisé pour stocker, récupérer et traiter les données et la technologie de communication.(Xingmei, Jing, & He, 2013)

Pour répondre aux besoins de L’Iot, tel que : la vitesse, la sécurité et la fiabilité, l’efficacité énergétique, il est possible que le niveau de la diversité des technologies de communication soit réduit à un nombre de technologies bien définies, et répondant aux besoins des applications IoT. Parmi ces technologies, nous pouvons citer : le WI-FI, les réseaux câblés, Bluetooth, ZigBee, GSM et GPRS.

2.5 Classification des technologies Iot

On peut classifier les technologies en trois groupes (Aswale et al., 2019), selon les dispositifs, selon les réseaux ou encore selon la gestion des applications. Ces types sont décrit dans les sections qui suivent.

2.5.1 les dispositifs

Ce premier groupe de technologies concerne les dispositifs, les puces à microprocesseur :

- Capteur de faible puissance pour la conservation de l'énergie.

- les capteurs intelligents.

- la miniaturisation des puce électroniques (chipsets)

- Capteur sans fil pour un réseau de capteurs.

2.5.2 Réseaux

Ce deuxième groupe comprend les technologies prenant en charge le partage de réseau et les problèmes de capacité d'adresse et de latence :

- Technologies de partage de réseau.

- Technologies de réseau qui traitent des problèmes de capacité et de latence tels que LTE et LTE-A.

2.5.3 Gestion des applications

Ce troisième groupe concerne les services de gestion prenant en charge les applications Iot

- Technologies de prise de décision intelligentes telles que l'analyse prédictive, traitement d'événements complexes et analyse comportementale.

- Technologies de traitement rapide de données telles que l'analyse de la mémoire et la transmission en temps réel.

2.6 Les problèmes liés à l'Iot

L'internet des objets touches divers domaines cependant, plusieurs défis sont à relever. Les problèmes liés à l'internet des objets sont énumérés dans ce qui suit (Zikria et al., 2018) :

1. **La taille de mémoire**

Les efforts supplémentaires devraient être faits pour utiliser un faible encombrement de la mémoire tout en fournissant une API conviviale pour les développeurs et en ajoutant des

fonctionnalités sophistiquées, pouvant nécessiter l'ajout d'un nouveau langage de programmation ou d'extensions des langages existants. Un appareil IoT ne contient que quelques kilo-octets de mémoire. Par conséquent, les caractéristiques fondamentales d'un système d'exploitation IoT sont la réduction de la taille du code et l'utilisation efficace de la mémoire minimale.

2. Efficacité énergétique

Une autre orientation de recherche générale consiste à envisager un mécanisme d'efficacité énergétique permettant de prolonger la durée de vie de la batterie d'un dispositif Iot en concevant des protocoles réseau plus efficaces. De même, une utilisation plus intelligente des fonctionnalités matérielles peut améliorer l'efficacité énergétique.

3. Fiabilité des appareils Iot

Pour prendre en charge les déploiements complexes D'Iot, la fiabilité du système d'exploitation peut être obtenue en utilisant un micro-noyau, des unités de protection de la mémoire, une analyse de code statique, etc.

4. Support en temps réel

Les appareils Iot nécessitent diverses applications ; certains d'entre eux fournissent un fonctionnement en temps réel. Ces sensations en temps réel ont tendance à être sensibles au temps.

5. La planification

Le système d'exploitation Iot est également soumis à certaines limitations lors de l'exécution de tâches qui affectent le processeur et peuvent entraîner une charge supplémentaire et une charge supplémentaire pour le processeur. Donc il faut bien prévoir des plans d'exécution et planification.

6. Gestion des tampons réseaux

Le tampon réseau est le composant principal de L'Iot. Ce domaine nécessite des recherches approfondies pour allouer efficacement de la mémoire limitée aux paquets.

7. Coexistence

Avec les scénarios d'application de plus en plus nombreux des réseaux Iot dans un spectre de fréquences limité, les technologies de coexistence constituent un problème de recherche récurrent pour les concepteurs de systèmes d'exploitation et de radio. Il s'agit d'un domaine de recherche diversifié qui nécessite une conception optimale des couches physique, MAC et réseau.

L'atteinte de la coexistence sans fil peut permettre le partage des ressources du spectre, le déchargement du trafic et une connectivité optimale pour divers services Iot.

2.7 Développement futur et besoins en recherche

Le tableau 2 montre l'évolution future et les besoins en matière de recherche pour L'Iot (Patel and Patel 2016, Aswale, Shukla et al. 2019).

Tableau II Développement future et besoins en recherche pour les technologies Iot

Technologie	Développements future	Besoins de recherche
Périphériques matériels	• Nanotechnologie • Miniaturisation des chipsets • Circuits à très basse consommation	• Dispositifs modulaires à faible coût • Ultra basse consommation EPROM / FRAM • Circuits autonomes
Capteurs	• Capteurs intelligents • Plus de capteurs • Capteurs de faible puissance • Réseau de capteurs sans fil	• Capteurs autoalimentés • Intelligence des capteurs
La communication	• Protocole unifié sur large spectre • Puces reconfigurables multifonctionnelles	• Protocoles d'interopérabilité • Puces multi protocole • Convergence des passerelles sur les réseaux à puce • Portée plus longue (fréquences plus élevées-dixièmes de GHz) • Développements 5G

Technologie de réseau	• Des réseaux autoorganisés • Evolutivité activée par IPv6 • Basé sur IPv6 omniprésent • Déploiement IoT	• Réseau Grid / Cloud • Réseaux définis par logiciel • Réseau de service
Logiciel et algorithmes	• Logiciel axé sur les objectifs • Résoudre le problème de l'intelligence distribuée • Logiciel orienté utilisateur	• Logiciel sensible au contexte • Logiciel évolutif • Logiciel auto réutilisable • Les objets autonomes : (auto configurable, auto guérison, autogestion)
Traitement de données, traitement de signal	• Traitement cognitif et optimisation • Analyse de données complexe • Visualisation intelligente des données	• Ontologie commune des capteurs • Efficacité énergétique distribuée • Traitement de l'information • Informatique autonome
Découverte et moteur de recherche	• Centres de gestion automatique et d'identification des routes • Découverte / intégration de services à la demande	• Services de découverte évolutifs pour connecter des objets avec des services
Technologies de sécurité et de confidentialité	• Politique de confidentialité • Traitement de données sensible à la vie privée • Sélection de profils de sécurité et de confidentialité en fonction des besoins en matière de sécurité et de confidentialité	• Dispositifs d'identification / authentification à faible coût, sécurisés et hautes performances • Approches décentralisées de la protection de la vie privée par la localisation de l'information

Comme le tableau 2 montre, pour chaque technologie de L'Iot des besoins de recherche doivent être résolu pourque cette technologie soit développée. Prenons par exemples les technologies de sécurité et confidentialité qui sont liée directement avec notre sujet, des nouvelles procédures d'identification doivent être établis car les procédures courantes contient des vulnérabilités peuvent être exploité par les attaquants, pour cela les développements futures se concentrent sur la conception des politiques de sécurité et de confidentialité.

2.8 Les exigences de sécurité pour l'IoT

Etant donné l'émergence des technologies de l'IoT dans tous les domaines de la vie quotidienne, cela pose la question sur la sécurité de l'IoT. Ceci peut être étudié selon la couche considérée de l'architecture de l'IoT. Le tableau 3 présente les exigences de sécurité d'un système IoT, selon chaque couche du modèle. (Patel & Patel, 2016)

Tableau III Les exigences de sécurité pour chaque couche

La Couche	Exigences de sécurité
La couche application	● Minimisation de données spécifique à l'application ● Protection de la confidentialité et gestion des politiques ● Authentification ● Autorisation, assurance ● Cryptage spécifique à l'application, cryptographie.
La couche gestion de service	● Gestion et traitement des données protégées (recherche, agrégation, corrélation, calcul) ● Stockage de données cryptographiques ● Calcul sécurisé, traitement de données en réseau, agrégation de données, cloud computing

La couche réseau	● Interaction capteur / cloud sécurisée ● Gestion de la sécurité des données entre domaines ● Sécurité de la communication et de la connectivité
Couche objet	● Contrôle d'accès aux nœuds ● Chiffrement léger ● Format de données et structures ● Ancres de confiance et attestation

Conclusion

Dans ce premier chapitre, nous avons traité la question de la sécurité et de l'internet des objets. Dans la première partie, nous avons présenté les définitions et les concepts de base ; nous avons abordé plusieurs notions à savoir : la sécurité informatique, ses buts et ses types, la menace, l'intrusion et ses types, l'attaques est ses variantes, ainsi que le pare-feu et l'IDS.

Tandis que dans la deuxième partie, nous l'avons consacré pour l'Iot. Nous avons introduit ses caractéristiques générales, son architecture en couche et puis ses technologies que nous avons classé en trois groupes : le groupe des dispositifs, le groupe réseaux et le groupe gestion des applications.

A la fin de ce chapitre, nous avons discuté des problèmes liés à l'Iot, le développement futur et les besoins en recherche avec les exigences de sécurité.

Le chapitre suivant traite de l'intelligence artificielle, de l'apprentissage automatique et de l'apprentissage profond.

Chapitre II

L’apprentissage automatique

Introduction

Le terme apprentissage automatique est souvent lié au termes suivant: l’intelligence artificielle et L’apprentissage profond.

L’intelligence artificielle peut être vue comme une série de tests si, sinon ou comme un modèle statistique complexe pour catégoriser les données. Par exemple : jeu d’échecs et les systèmes expert.

L’apprentissage automatique fait partie de l’intelligence artificielle, c’est un traitement dynamique qui n’a pas besoin d’une intervention humain. L’approche représente la capacité à apprendre sans programmation explicite. L’apprentissage automatique est une manière d'entraîner un algorithme pour l’amener à exécuter une tâche sans programmation explicite.

L’apprentissage profond est une partie de l’apprentissage automatique, c’est une technique récente inspirée du cerveau humain, et vise à imiter les mécanismes perceptifs et de raisonnement logique humain. Il nécessite la précision et beaucoup de calculs. Par exemple : le moteur de jeu d’échecs Alpha zero réalisé par Google.

1. Apprentissage automatique

Cette section est consacrée pour l’apprentissage automatique et dans laquelle nous allons détailler ce concept, ses types et ses algorithmes.

1.1 Définition

L'apprentissage automatique fait référence au développement, à l’analyse et à l’implémentation de méthodes qui permettent à une machine d’évoluer grâce à un processus d’apprentissage, et ainsi de remplir des tâches qu’il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.(HAMDAD, 2017)

L’apprentissage automatique a une large valeur ajoutée dans beaucoup de domaines d’application tels que :

- Data mining pour analyser par exemple des résultats de traitements médicaux ou apprendre la règle d'affectation de crédit aux clients, contrôle du processus de fabrication lors du changement de stocks, reconnaissances faciales...
- Diagnostiquer une maladie : cancer, arythmie cardiaque.
- Prédire par exemple le prix d'une maison se basant sur ses caractéristiques.
- Analyse de sentiments : Avis des utilisateurs.
- Rechercher des documents intéressants
- Reconnaissance de forme

L'objectif principale est d'extraire et d'exploiter automatiquement l'information présente dans un jeu de données.

1.2 Types d'apprentissage

Il existe plusieurs types d'apprentissage, parmi les classifications les plus courantes on trouve celle de (Mitchell, 1999).

Les algorithmes d'apprentissage peuvent se catégoriser selon le type d'apprentissage qu'ils emploient, on distingue quatre types :

1. L'apprentissage supervisé
2. L'apprentissage non-supervisé
3. L'apprentissage semi-supervisé
4. L'apprentissage par renforcement

1.2.1 Apprentissage supervisé

Il s'agit d'apprendre à partir d'exemples pour pouvoir affecter de nouveaux exemples. Soit un ensemble d'apprentissage constitué de n données d'observations de type entrée-sortie

$D = \{(X1, Y1) \dots (Xn, Yn)\}$ Où les $Xi, i = 1, \dots, n$ représentent des entrées appartenant souvent à R_p (variables explicatives) et $Yi, i = 1, \dots, n$ sont des valeurs quantitatives ou qualificatives.

Classification supervisée : $Yi, i = 1, \ldots, n$ prennent comme valeur $-1{,}1$ ou plusieurs valeurs catégoriques.

Régression : $Yi, i = 1, \ldots, n$ sont numériques.

Le but est de construire à partir de l'ensemble d'apprentissage un modèle qui permettra de prévoir la sortie d'une nouvelle entrée.

Il existe deux types de classification : la classification multi-classe et la classification binaire, la première est le problème de la classification des instances dans l'une des trois classes ou plus, et la deuxième est le problème de la classification des instances dans l'une des deux classes.

1.2.2 Apprentissage non-supervisé

Les entrées sont connues et non les sorties, ce sont des méthodes dont le rôle est plus descriptif que prédictif. Il s'agit alors de construire un modèle permettant de représenter au mieux les observations $X_1, \ldots, Xn$, de manière précise.

• Le clustering est de l'apprentissage supervisée : Il s'agit de regrouper les données hétérogènes en classes de données homogènes et cela en se basant sur des indices de similarités.

1.2.3 Apprentissage semi-supervisé

Classe de techniques d'apprentissage automatique qui utilise un ensemble de données avec étiquettes(peu) et sans étiquettes (beaucoup).

Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées, ça veut dire la valeur de Yi est connus, et l'apprentissage non-supervisé qui n'utilise que des données non étiquetées (la valeur de Yi n'est pas connus).

Il a été démontré que l'utilisation de données non-étiquetées, en combinaison avec des données étiquetées, permet d'améliorer la qualité de l'apprentissage.

L'étiquetage de données nécessite l'intervention d'un utilisateur humain. Lorsque les jeux de données deviennent très grands, cette opération peut être fastidieuse. Dans ce cas, l'apprentissage semi-supervisé, qui ne nécessite que quelques étiquettes, devient d'un intérêt pratique évident.

Classification semi-supervisée

Entraîner sur des données avec labels et exploiter les données (beaucoup) sans labels.

Clustering semi-supervisé

Clustering des données sans labels en s'aidant des données avec labels.

1.2.4 Apprentissage par renforcement

Apprendre en interagissant avec son environnement. Ceci donne des informations de cause à effet, sur la conséquence d'une action et sur quoi faire pour améliorer l'objectif.

Il se compose de quatre éléments

- **La politique** : la voie choisie par l'agent apprenant à un instant donné ?
- **Fonction récompense** : l'objectif de l'apprentissage par renforcement.
- **Fonction valeur** : Ce qui est bon à long terme.
- **Modèle** : Imite le comportement de l'environnement. Etant donné un état et une action, le modèle peut par exemple prévoir l'état et la récompense suivante.

1.3 Les problèmes de l'apprentissage automatique

Selon (HAMDAD, 2017), parmi les problèmes de l'apprentissage automatique :

- Le théorème No free lunch stipule qu'un algorithme ne résout pas tous les problèmes. Un algorithme bon pour une instance donnée ne l'est pas forcément pour d'autres instances.
- Problème insoluble : problème non résolu en des temps raisonnable. La majorité des algorithmes ne convergent pas vers la solution optimale.
- Gérer le fléau de la dimensionnalité : Peu de données d'apprentissage lorsque le nombre de variables augmente.
- Sur-apprentissage : il s'agit de surapprentissage quand un modèle a trop appris les particularités de chacun des exemples fournis en exemple. Il présente alors un taux de succès très important sur les données d'entraînement (pouvant atteindre jusqu'à 100%), au détriment de ses performances générales réelles.

- Sous-apprentissage : Situation observée quand un algorithme d'apprentissage automatique ou un modèle statistique ne s'ajuste que grossièrement aux données d'entraînement, ce qui se traduit par une erreur élevée sur les données d'entraînement.

1.4 Les algorithmes de l'apprentissage

Dans ce qui suit, nous présentons les algorithmes de l'apprentissage automatique les plus utilisés :

1.4.1 Classifieur bayésien naïf

C'est un algorithme basé sur le théorème de Bayes qui permet de classifier un ensemble d'observations selon des règles déterminées par l'algorithme lui-même. Cet outil de classification doit dans un premier temps être entrainé sur un jeu de données d'apprentissage qui montre la classe attendue en fonction des entrées. Pendant la phase d'apprentissage, l'algorithme élabore ses règles de classification sur ce jeu de donnée, pour les appliquer dans un second temps à la classification d'un jeu de données de prédiction. [web2]

Le classificateur bayésien naïf implique que les classes du jeu de données d'apprentissage soient connues et fournies, il s'agit alors d'un algorithme d'apprentissage supervisé.

Il donne des bons résultats dans la classification de documents et l'élaboration de filtres anti-spam.

Parmi ses avantages :

- Son apprentissage rapide qui ne nécessite pas un gros volume de données.
- La rapidité d'exécution.

  L'inconvénient majeur est la forte hypothèse simplificatrice d'indépendance des variables.

Théorème de bayes :

$$\boldsymbol{P(A/B) = \frac{P(B/A)P(A)}{P(B)}} \ \ldots\ldots (\mathbf{eq\ 2})$$

P(A/B) désigne la probabilité conditionnelle de A sachant B.

Le théorème de Bayes est utilisé dans l'inférence statistique pour mettre à jour ou actualiser les estimations d'une probabilité ou d'un paramètre quelconque, à partir des observations et des lois de probabilité de ces observations.

1.4.2 K-Means

C'est un algorithme non supervisé utilisé pour constituer un nombre bien défini de classes homogènes d'observations sur la base de leur description par un ensemble de variables quantitatives. Les données similaires se retrouveront dans la même classe. Par ailleurs, une observation ne peut se retrouver que dans une classe à la fois (exclusivité d'appartenance). Une même observation, ne pourra donc, appartenir à deux classes différentes.

Pour pouvoir regrouper un jeu de données en K classe distinctes, l'algorithme K-Means a besoin d'un moyen de comparer le degré de similarité entre les différentes observations. Ainsi, deux données qui se ressemblent, auront une distance de dissimilarité réduite, alors que deux objets différents auront une distance de séparation plus grande.

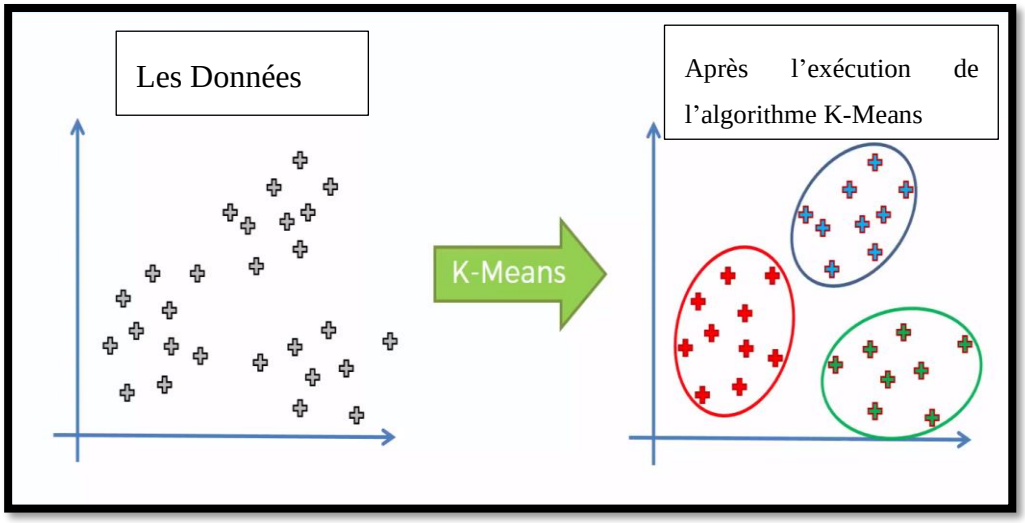

Figure *5* ***L'algorithme K-means***

La figure 5, montre un exemple de classification avec l'initialisation de nombre de classes à trois.

C'est un algorithme très simple, un inconvénient possible de k-means est que les classes dépendent de l'initialisation et de la distance initiale choisie (K).

1.4.3 A-Priori

L'algorithme A-priori est un algorithme d'exploration de données conçu en 1994, par Rakesh Agrawal et Ramakrishnan Sikrant, dans le domaine de l'apprentissage des règles d'association. Il sert à reconnaître des propriétés qui reviennent fréquemment dans un ensemble de données et d'en déduire une catégorisation [web3].

A-Priori détermine les règles d'association présentes dans un jeu de données, pour un seuil de support et un seuil de confiance fixés. Ces deux valeurs peuvent être fixées arbitrairement par l'utilisateur.

Le support d'un ensemble d'items est défini comme la fréquence d'apparition simultanée des items figurant dans l'ensemble des données.

La confiance d'une règle « si condition alors conclusion » est le rapport entre le nombre de données où les items de la condition et de la conclusion apparaissent simultanément et le nombre de données où les items de la condition apparaissent simultanément.

Exemple

Le tableau 4 illustre l'exemple de paniers d'achats de certains clients :

Tableau IV A-Priori : déroulement d'un exemple

	Article A	Article B	Article C	Article D
Client 1	**x**	**x**		
Client 2	**x**		**x**	
Client 3		**x**		
Client 4	**X**		**x**	**x**
Client 5		**x**		

Support (A, B) = 1/5, car A et B n'apparaissent simultanément que dans le panier du Client 1.

Support (A, C) = 2/5, car A et C apparaissent simultanément dans le panier des clients 2 et 4.

Confiance (si A, alors B) = 1/3, car A et B apparaissent simultanément dans le panier de 1 client et A apparait dans le panier de 3 clients.

Confiance (si A, alors C) = 2/3 car A et C apparaissent simultanément dans le panier de 2 clients et A apparait dans 3 clients.

L'inconvénient majeur de cet algorithme est le nombre considérable d'accès à la base de données.

1.4.4 Régression linéaire

(Cours de Marie Chavent, Université de Bordeaux)

On cherche à modéliser la relation entre deux variables quantitatives continues. Un modèle de régression linéaire simple est de la forme suivante :

$$\boldsymbol{y = \beta_0 + \beta_1 x + \varepsilon} \qquad (\textbf{eq 3}) \quad \textbf{,} \text{ Où :}$$

- y est la variable à expliquer (à valeurs dans R).
- x est la variable explicative (à valeurs dans R).
- ε est le terme d'erreur aléatoire du modèle.
- β_0 et β_1 sont deux paramètres à estimer.

Le modèle est représenté graphiquement dans la figure 6. La désignation « simple » fait référence au fait qu'il n'y a qu'une seule variable explicative x pour expliquer y.

La désignation « linéaire » correspond au fait que le modèle (3) est linéaire en β_0 et β_1.

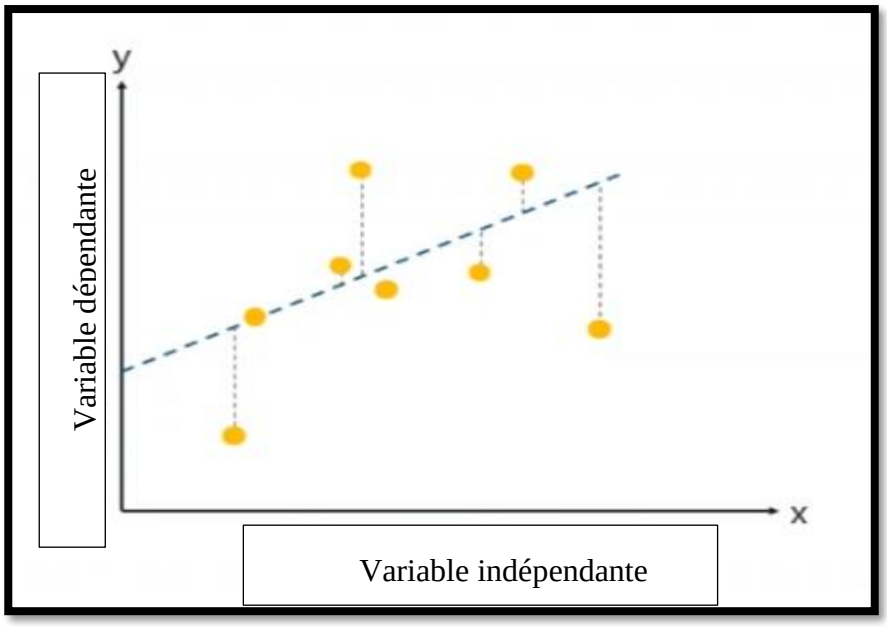

Figure *6* ***La représentation graphique de l'équation (3)***

1.4.5 Les K plus proches voisins

La méthode des K plus proches voisins (KNN) est une approche de classification supervisée qui a pour but de classifier des classes méconnues en fonction de leurs distances par rapport à des points dont la classe est connue a priori.

La figure 7 illustre un exemple où le nombre de classe résultant après l'exécution de l'algorithme est 3.

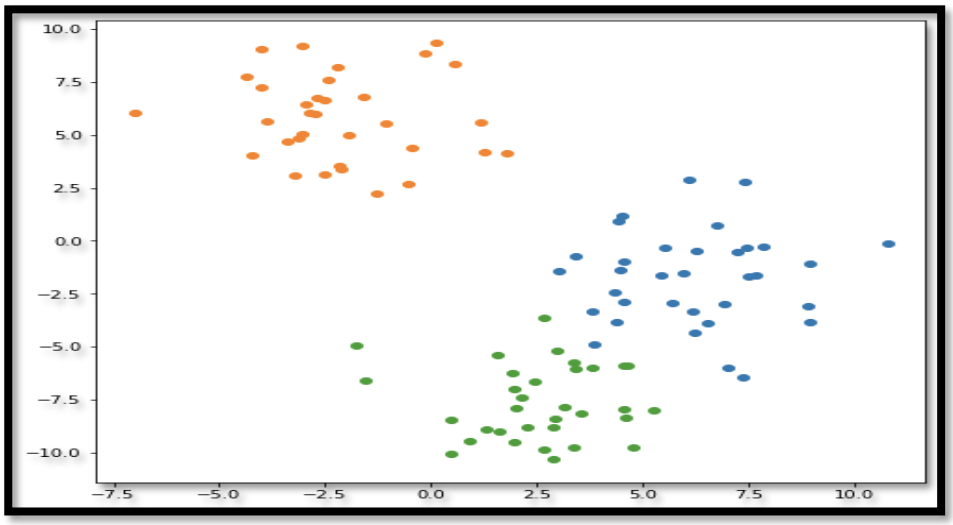

Figure *7* ***KNN classification***

1.4.6 Les forêts aléatoires

(Genuer, 2010)

Les forêts aléatoires sont une méthode d'apprentissage statistique qui fait aujourd'hui partie des outils centraux des statisticiens. Introduites par Leo Breiman en 2001, elles sont depuis intensément utilisées dans de nombreux domaines d'application (comme l'écologie, la prévision de la pollution ou encore la santé), du fait des très bonnes performances de l'algorithme en prédiction, mais aussi de leur généralité, n'imposant que très peu de restrictions sur la nature des données. En effet, elles sont adaptées aussi bien à des problèmes de classification supervisée qu'à des problèmes de régression. De plus, elles permettent de prendre en compte un mélange de variables explicatives qualitatives et quantitatives. Enfin, elles sont capables de traiter des données standards pour lesquelles le nombre d'observations est plus élevé que le nombre de variables, mais se comportent également très bien dans le cas de données de grande dimension où le nombre de variables est très important.

Une forêt aléatoire est un ensemble d'arbres de décision binaire dans lequel a été introduit de l'aléatoire. Les forêts aléatoires consistent à faire tourner en parallèle un grand nombre (environ de 400) d'arbres de décisions construits aléatoirement, avant de les moyenner.

1.4.7 XGBoost

Le Boosting de Gradient est un algorithme d'apprentissage supervisé dont le principe et de combiner les résultats d'un ensemble de modèles plus simple et plus faibles afin de fournir une meilleure prédiction. L'idée est donc simple : au lieu d'utiliser un seul modèle, l'algorithme va en utiliser plusieurs qui serons ensuite combinés pour obtenir un seul résultat.

Il gère efficacement une grande variété de types de données que le grand nombre d'hyperparamètres qui peuvent être modifiés et réglés à des fins d'amélioration. Cette flexibilité fait de XGBoost un choix solide pour les problèmes de régression, de classification (multi classe et binaire).

1.5 Les techniques de validation de l'apprentissage automatique

1.5.1 Re-substitution

Dans le cas où toutes les données sont utilisées pour l'apprentissage du modèle et que le taux d'erreur est évalué en fonction des résultats par rapport à la valeur réelle du même ensemble de données d'apprentissage [web4].

Cette technique est appelée technique de validation de la Re-substitution.

1.5.2 Hold-out

Pour éviter l'erreur de Re-substitution, les données sont divisées en deux ensembles de données différents étiquetés comme un ensemble de données d'apprentissage et de test. Il peut s'agir d'une répartition 60/40 ou 70/30 ou 80/20.

1.5.3 Validation croisée K-Fold

Dans cette technique, les éléments k-1 sont utilisés pour la formation et le reste est utilisé pour les tests comme indiqué dans la figure 8.

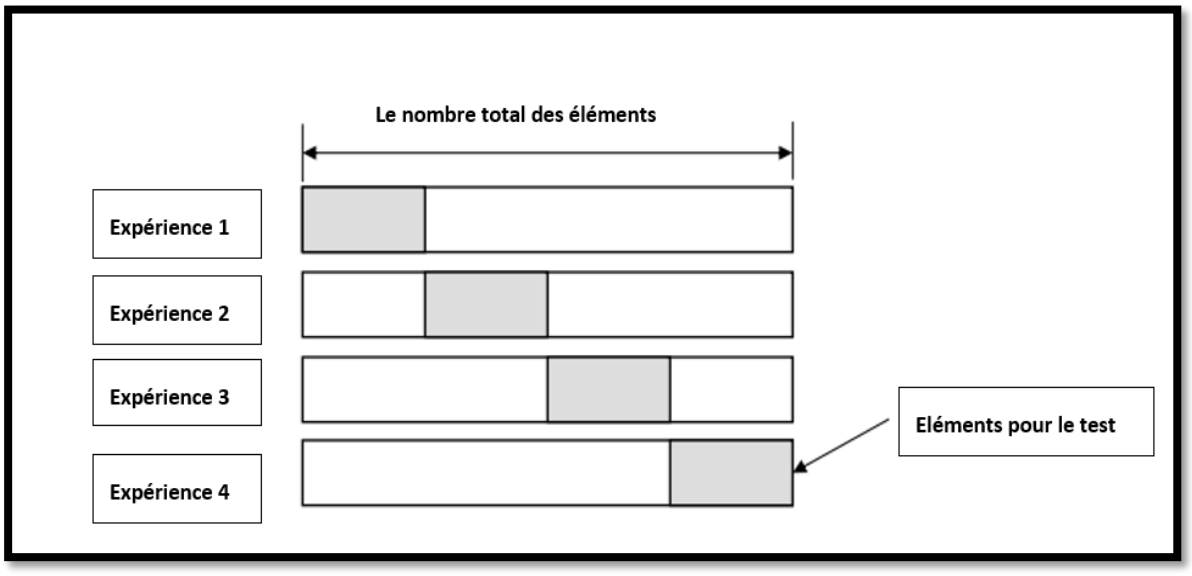

Figure *8* ***La Validation croisée K-Fold***

L'avantage est que des données entières sont utilisées pour la formation et les tests. Le taux d'erreur du modèle est la moyenne du taux d'erreur de chaque itération. Cette technique peut également être appelée une forme de méthode de Hold-out répété.

1.5.4 Leave-One-Out Cross-Validation (LOOCV)

Dans cette technique, toutes les données sauf un enregistrement sont utilisées pour l'entrainement et un enregistrement est utilisé pour les tests. Ce processus est répété N fois s'il y a N enregistrements. L'avantage est que des données entières sont utilisées pour l'entrainement et le test. Le taux d'erreur du modèle est la moyenne du taux d'erreur de chaque itération. La figure 9 représente la technique de validation LOOCV.

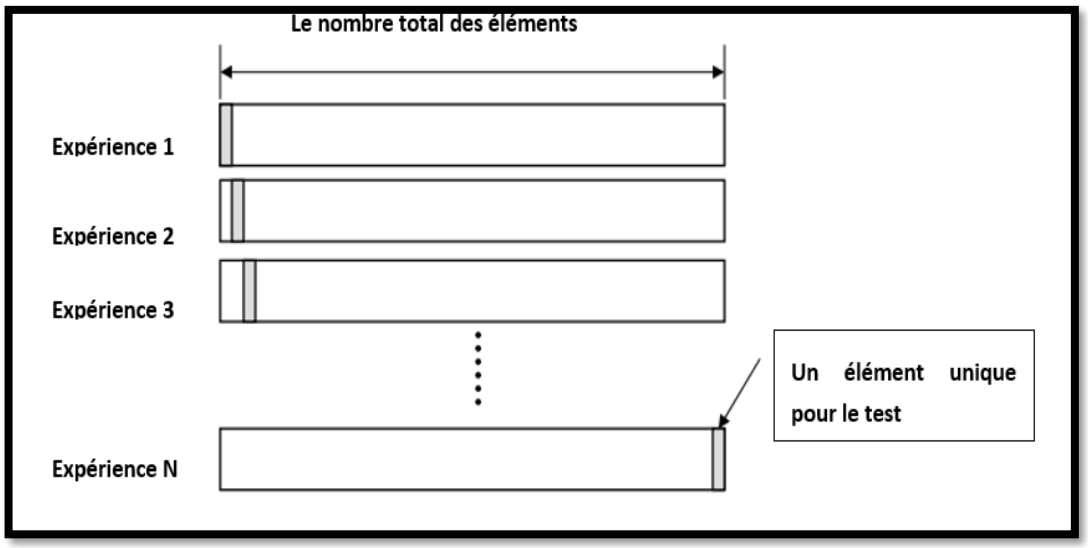

Figure *9* ***La technique de validation LOOCV***

2. Sélection d'attributs dans l'apprentissage automatique

Les deux éléments importants qui affectent fortement les performances d'un classificateur sont : la technique du classificateur et le sous-ensemble d'attributs sélectionnés. Les chercheurs ont proposé diverses combinaisons de classificateurs et de méthodes de sélection des caractéristiques. Le but de la sélection d'attributs est de sélectionner le sous-ensemble de fonctionnalités le plus important et optimal. La sélection d'attributs améliore la généralisation, la performance, réduit le coût de calcul du classificateur et rend le classificateur plus rapide pour détecter les données invisibles et simplifie la compréhension du traitement des données.

Il y a divers inconvénients, si on considère toutes les caractéristiques pour l'apprentissage

(Mishra et al., 2018)

- Cela augmentera la charge de calcul du système et ralentira la formation et les tests.

- Les besoins de stockage augmentent, car plus une base de données contient de fonctionnalités, plus elle nécessite d'espace pour stocker chaque fonctionnalité.
- Il limite la capacité de généralisation d'un classificateur qui utilise des techniques d'exploration de données pour la prédiction.
- Il augmente le taux d'erreur du classificateur, car les caractéristiques non pertinentes diminuent le pouvoir discriminant des caractéristiques pertinentes.

Conclusion

L'apprentissage automatique est un sujet en continuel évolution. Il est utilisé dans plusieurs domaines, grâce à sa capacité de résoudre des problèmes complexes.

Dans ce chapitre, nous avons abordé quelques algorithmes d'apprentissage automatique, et donné quelques techniques de validation de l'apprentissage automatique. La fin de ce chapitre traite de la sélection d'attributs dans l'apprentissage automatique. Dans le prochain chapitre, nous présenterons l'apprentissage automatique et la sécurité dans l'IoT.

Chapitre III

L’apprentissage automatique et la sécurité dans l’IoT

1 Introduction

Pour traiter les menaces et les problèmes existant dans les réseaux Iot, de nombreux travaux ont tenté de trouver des moyens efficaces pour détecter les intrusions dès qu’ils ont lieu.

Les IDS traditionnels ne sont pas adaptés pour l’IoT car les objets ont des ressources limitées qui rendent impossible l’utilisation de techniques de détection d’intrusion trop lourdes. De plus, les protocoles utilisés par les objets connectés sont nombreux et leur sont spécifiques, et les objets évoluent souvent dans des réseaux maillés, étant à la fois une source de données et un système capable de rediriger les données qu’il reçoit. (Zarpelão, Miani, Kawakani, & de Alvarenga, 2017)

Dans le domaine de la détection d'intrusions, l'apprentissage supervisé produit généralement des classificateurs pour la détection des abus à partir d'ensembles de données de formation étiquetés en classe. Les classificateurs sont essentiellement considérés comme une fonction mappant des échantillons de données aux étiquettes de classe correspondantes. L'apprentissage non supervisé se distingue de l'apprentissage supervisé par le fait qu'aucune donnée labellisée n'est disponible dans la phase de formation. Il regroupe les points de données en fonction de leurs similitudes. L'apprentissage non supervisé satisfait l'exigence de détection d'anomalies, il est donc généralement utilisé dans la détection d'anomalies.(Wu & Banzhaf, 2010)

Nous distinguons deux types de système de détection d’intrusions : le premier type se base sur des règles à propos du système, et le deuxième se base sur la détection d’anomalies du système.

2 Comparaison entre les solutions basées sur l’apprentissage automatique et les solutions traditionnelles

2.1 les systèmes experts et la détection par règle

L’utilisation d’un système expert, qui est l’un des outils utilisé pour la détection d’intrusion par des règles, peut reposer sur des signatures connues de menaces, sur des règles concernant les données collectées ou sur les transitions d’état du système. Il est évident que ce type de système

est incapable de détecter de nouvelles attaques, mais il reste toujours précis et il fonctionne efficacement sur des menaces connues.

Il est difficile de réaliser un IDS basé sur les anomalies pour la surveillance du comportement du système parce qu'une détection précise requière une précision sur chaque comportement du système.

Pour avoir les informations nécessaires pour la détection des anomales dans un réseaux Iot, l'utilisation de technique de traçage peut être exploité pour avoir :

- Une liste d'appels système.
- Leurs arguments.
- Diverses informations tel que le statut des processus, la séquence d'action de l'ordonnanceur, les information réseaux, etc.

Vu la difficulté d'avoir une connaissance totale et précise des méthodes d'attaque des réseaux et qu'il est peut-être difficile d'identifier des intrusions dont la méthode varie de celles connues.

Il est possible d'utiliser le modèle statistique pour la prédiction des anomalies lorsqu'elles sont facilement récoltables et qu'il n'est pas difficile de caractériser le comportement normal du système.

Les éléments statistiques sont :

- Le taux d'utilisation moyen de la mémoire vive.
- Le taux d'utilisation moyen du processeur.
- L'occupation classique du disque dur.

Sous la condition que les paramètres du programme n'évoluent pas beaucoup au cours du temps, alors un changement dans une de ces valeurs pourrait indiquer qu'une anomalie survient dans le système.

2.2 Les systèmes experts et les systèmes basés sur des réseaux de neurones artificiels

Le tableau 5 montre une comparaison entre les systèmes expert et les réseaux de neurones artificiels :

Tableau V Comparaison entre les systèmes expert et les réseaux de neurones

Les systèmes expert		**Les réseaux de neurones artificiels**	
Avantages	**Inconvénients**	**Avantages**	**Inconvénients**
• Simple pour la mise en œuvre.	• Ont besoin de nombreuses mises à jour pour rester efficace contre les menaces émergentes.	• Peuvent fonctionner même avec des données incomplètes. • Rapides. • Peuvent apprendre des attaques précédemment analysées.	• Cette technique requiert de très nombreuses données. • Il est compliqué de comprendre le fonctionnement exact du modèle entrainé.

2.3 IDS basés sur l'apprentissage automatique et Les IDS conventionnels

Selon (Mishra et al., 2018), certains des avantages de l'utilisation d'IDS basés sur l'apprentissage automatique par rapport aux IDS conventionnels basés sur les signatures sont les suivants :

- Il est facile de contourner les IDS basés sur les signatures en faisant de légères variations dans un modèle d'attaque tandis que les IDS basés sur l'apprentissage automatique basés sur des techniques supervisées peuvent facilement détecter les variantes d'attaques lorsqu'ils apprennent le comportement du flux de trafic.
- La charge CPU est faible à modérée dans les IDS basés sur l'apprentissage automatique car ils n'analysent pas toutes les signatures de la base de données de signatures comme cela est fait par les IDS basés sur les signatures.

- Certains IDS basés sur l'apprentissage automatique, en particulier basés sur des algorithmes d'apprentissage non supervisés, peuvent détecter de nouvelles attaques.
- Les IDS basés sur l'apprentissage automatique peuvent capturer les propriétés complexes du comportement d'attaque et améliorer la précision et la vitesse de détection par rapport aux IDS classiques basés sur les signatures.
- Différents types d'attaques continuent d'évoluer. Les IDS basés sur les signatures nécessiteront la maintenance de la base de données de signatures de temps en temps et la maintiendront à jour, tandis que les IDS basés sur l'apprentissage automatique basés sur le clustering et la détection des valeurs aberrantes ne nécessiteront pas une telle mise à jour.

3 Synthèse de travaux existants

Dans ce qui suit, nous présentons une forme de synthèse bibliographique de quelques travaux existant dans la littérature et traitant le problème de détection d'intrusion en se basant sur l'apprentissage automatique.

Les auteurs Rezvy Shahadate et Luo Yuan (Rezvy, Luo, Petridis, Lasebae, & Zebin, 2019) proposent un système de détection d'intrusion hybride basé sur l'apprentissage profond. Ils ont appliqué un algorithme de réseau de neurones type profond pour détecter les intrusions ou les attaques dans les réseaux 5G et IoT. Ils ont évalué l'algorithme avec le benchmark AWID **[web5]**. Ils ont obtenu une précision de détection globale de 99,9% pour les attaques de type Inondation (flooding), Usurpation d'identité et Injection. Ils ont également présenté une comparaison avec les approches récentes utilisées dans la littérature qui ont montré une amélioration substantielle en termes de précision et de vitesse de détection avec l'algorithme proposé.

Comme première étape, ils ont utilisé la version réduite du data set AWID à savoir **AWID-CLS-R-Trn**, **AWID-CLS-R-Tst**. Ce data set inclus trois types d'attaques : Usurpation d'identité, injection et flooding.

La figure 10 illustre, le modèle de détection d'intrusion proposé.

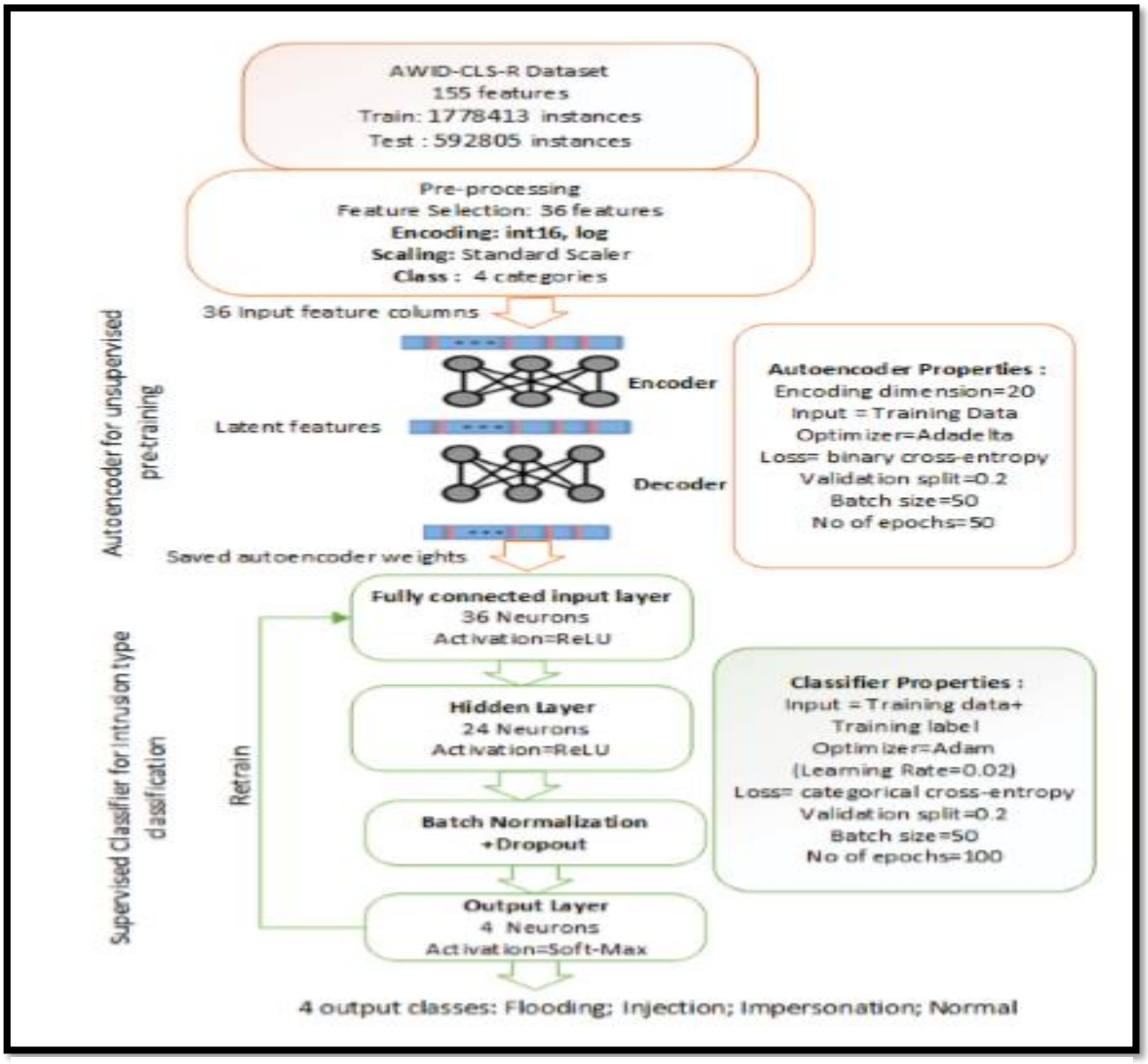

Figure 10 Le modèle proposé par S Rezvy

Comme le montre la figure 10, ils ont utilisé deux couches principales à savoir, une couche de prétraitement et un réseau de neurones dense supervisé pour la classification des types d'attaque.

Pour le pré-traitement, ils ont utilisé un auto-encodeur avec un codage et une couche de décodage qui a été formé pour minimiser l'erreur de reconstruction. Cela a incorporé les connaissances antérieures de l'ensemble de formation pour apprendre efficacement à partir de données et fournir de bonnes performances.

Après la couche d'auto-codeur, un réseau neuronal dense à trois couches est utilisé et formé en utilisant la première sortie de l'auto-encodeur comme entrées. Cette séquence de tâches est recyclée de manière supervisée avec les étiquettes de classe et la fonction d'entrée donnée au classificateur. Ils ont utilisé une couche d'activation softmax comme couche de sortie. La couche calcule la perte entre les valeurs prévues et les vraies valeurs, et les poids dans le réseau sont ajustés en fonction de la perte. Ils ont obtenu un taux de vrai positif supérieur à 99% pour toutes les catégories d'attaque. Pour l'attaque de type flooding seulement 1,2% de fausses prédictions sont faites par le modèle autrement dit 177 prédictions erronées sur 14 milles instances. Et pour les attaques de type injection et usurpation d'identité, le model donne une très bonne prédiction.

Les auteurs (Kolias et al., 2015) se sont intéressés aux attaques contre les réseaux Wifi 802.11. Ils ont rassemblé, catégorisé et évalué en profondeur les attaques les plus populaires contre 802.11 et analysé leurs signatures. Plusieurs vulnérabilités de sécurité ont été détectée dans pratiquement toutes les versions du protocole WIFI rendant l'intégration de mécanismes de protection externes une nécessité.

Ils ont commencé par définir l'architecture du standard 802.11 et montrer les mécanismes de sécurité disponible dans ce standard. Ensuite, ils ont illustré les attaques contre plusieurs versions du mécanisme de sécurité 802.11 (c'est-à-dire WEP, WPA, WPA2). Ils ont travaillé avec la dataset AWID. Ils ont testé plusieurs algorithmes à savoir : Naives Bayes, Random Forest, Random Tree et ZeroR. La meilleure précision est donnée par Random Forest qui est de 0,958.

Les auteurs (Shah & Issac, 2018) proposent une étude comparative entre deux IDS. En effet, l'article examine les performances de deux systèmes de détection d'intrusion (IDS) open source, à savoir Snort et Suricata, pour détecter avec précision le trafic malveillant sur les réseaux informatiques. Snort et Suricata ont été installés sur deux ordinateurs différents mais identiques et les performances ont été évaluées avec une vitesse de réseau de 10 Gbps. Les attaquants agissent comme des utilisateurs normaux. Ils génèrent des données et cachent leurs activités malveillantes sous la grande masse de données. Les auteurs ont reporté que L'IDS Snort possède une meilleure précision que Suricata. Ce dernier possède plutôt un meilleur débit quant au traitement des paquet de données, le taux de perte est moins important par rapport à Snort cependant consomme plus de ressource. Ils ont aussi noté que Snort génère un taux élevé de faux positifs. Ils ont intégré au module dans L'IDS basé sur l'apprentissage automatique pour améliorer ce taux. Pour sélectionner l'algorithme le plus performant pour le module à intégrer, une étude empirique a été réalisée avec

différents algorithmes d'apprentissage et support Vector Machine (SVM) a été sélectionné. Une version hybride de SVM et de logique floue a produit une meilleure précision de détection. Mais le meilleur résultat a été obtenu en utilisant un SVM optimisé avec l'algorithme Firefly avec un taux de faux positifs égale à 8,6% et un taux de faux négatifs égale à 2,2%. La contribution principale de cet article réside dans la comparaison des performances des deux IDS à 10 Gbps ainsi que l'intégration d'un module d'apprentissage automatique à Snort.

Dans l'article publié par (Thing, 2017), une analyse des menaces et des attaques ciblant le réseau IEEE 802.11 est fournie. Les auteurs ont également identifié les défis d'une classification précise des menaces et des attaques, en particulier dans les situations où les attaques sont nouvelles et n'ont jamais été rencontrées par le système de détection et de classification auparavant. (Thing, 2017) ont proposé une solution basée sur la détection et la classification des anomalies en utilisant une approche d'apprentissage profond. L'approche apprend automatiquement les caractéristiques nécessaires pour détecter les anomalies réseau et est capable d'effectuer une classification d'attaque avec précision. Les auteurs ont travaillé sur une architecture émulant une infrastructure SOHO. L'architecture était composée d'appareils mobiles et fixes, qui comprennent une machine de bureau, deux ordinateurs portables, deux téléphones intelligents, une tablette et une télévision connectée. Ces appareils ont été utilisés comme clients légitimes du réseau. Les téléphones intelligents ont été utilisés pour afficher des schémas de mobilité élevés (c'est-à-dire que leurs emplacements ont été changés fréquemment et conçus pour rejoindre ou quitter le réseau pendant toute la durée des expériences), tandis que les ordinateurs portables étaient semi-statiques (c'est-à-dire qu'ils changent rarement de Emplacements). Différents services fonctionnant sur les appareils produisaient du trafic légitime, par exemple via la navigation Web et la VoIP.

La couverture réseau a été fournie par un AP et protégée par le cryptage WEP, prenant en charge jusqu'à un taux de transfert de 54 Mbps. Les attaques ont été lancées à partir d'un nœud d'attaque, qui est un ordinateur portable exécutant le système d'exploitation 64 bits Kali Linux1.0.6.

Ils ont analysé les ***AWID-CLS-R-Trn*** et ***AWID-CLS-R-Tst***, qui sont l'ensemble d'entrainement et l'ensemble de données de test, respectivement.

Ils ont proposé d'utiliser une approche d'apprentissage profond pour dériver les caractéristiques complexes avec une meilleure capacité discriminante pour effectuer la détection d'anomalies et la classification des attaques.

Les auteurs ont utilisé un encodeur automatique empilé (SAE), qui est un réseau de neuro construit en empilant plusieurs couches d'auto-encodeurs clairsemés. La sortie de chaque couche forme l'entrée de la couche successive.

Deux étages sont proposés, qui sont composés de deux et trois couches cachées, respectivement. La première couche apprend les caractéristiques du premier ordre à partir des entrées brutes, tandis que la seconde couche apprend les caractéristiques correspondant aux motifs des caractéristiques du premier ordre.

La troisième couche dans le deuxième étage apprend les caractéristiques correspondant aux motifs des caractéristiques du second ordre. La première couche cachée se compose de 256 neurones, et la deuxième couche cachée se compose de 128 neurones, tandis que la troisième couche est composée de 64 neurones.

L'article explore l'utilisation de différentes techniques comme les fonctions d'activation pour les neurones cachés. Les résultats expérimentaux ont montré que l'approche proposée par les auteurs est capable d'effectuer la classification de 4 classes, en tenant compte des nouvelles attaques, avec une précision globale de 98,6688 %

Les auteurs de (Mishra et al., 2018)proposent une étude et une analyse détaillée des différentes techniques d'apprentissage automatique afin de déceler les problèmes associées aux différentes techniques apprentissage dans la détection d'activités malveillantes.

Ils ont commencé par introduire les IDS, donner leurs types et les avantages de l'utilisation des IDS basés sur l'apprentissage automatique par rapport aux IDSs classiques basés sur les signatures. Ils ont proposé une classification des attaques pour le dataset UNSW-NB en catégories. Les techniques d'apprentissage automatique ont été analysées et comparées en termes de capacité de détection pour détecter les différentes catégories d'attaques. Les limitations associées à chacune de ces catégories sont discutées. Divers outils d'exploration de données pour l'apprentissage automatique ont également été inclus.

- **Un tableau récapitulatif**

Le tableau 6 résume et compare les différents travaux présentés précédemment dans la section 3 par rapport à la source de données utilisée, au modèle ou type de l'algorithme, aux classes d'attaques dans le cas d'une détection multi-classes et enfin la précision obtenue.

Tableau VI Tableau récapitulatif de travaux

Auteurs Année	**Donnée**	**Modèle/Type Algorithme**	**Catégorie D'attaques**	**Précision**	
Rezvy Shahadate et Luo Yuan	AWID	Apprentissage profond/ Auto-encodeur dense	• Inondation (flooding), • Usurpation d'identité • Injection	0.999	
Constantinos Kolias, 2016	AWID	Random Tree	• Toutes les attaques présentes dans AWID	0.914	
		Random Forest		0.958	
		Naïve Bayes		0.891	
		ZeroR		0.85	
Syed Ali Raza Shah et Biju Issac	NSL-KDD (NSL- KDD, 2014)	SVM Decision Trees Fuzzy Logic BayesNet NaiveBayes	• MAC Spoofing • DNS Poisoning • IP Spoofing	SVM	0.956
				DT	0.82
				FL	0.923
				BN	0.73
				NB	0.7
	DARPA (DARPA IDDS, 2000)		• SSH Attacks • FTP Attacks • Scanning Attacks	SVM	0.942
				DT	0.85
				FL	0.94
				BN	0.712
				NB	0.71
	NSL-KDD IDS Dataset (NSLKDD, 2014)		• Denial of Service Attack (DoS) • User to Root Attack (U2R) • Remote to Local Attack (R2L)	SVM	0.954
				DT	0.812
				FL	0.94
				BN	0.74

			• Probing Attack	NB	0.79
(Thing, 2017)	AWID	Apprentissage profond/ SAE	• Énondation • Injection • Impersonation	0.9866	

Conclusion

Dans ce chapitre, nous avons comparé les approches basées sur l'apprentissage automatique et les approches traditionnelles pour la détection d'intrusion dans un réseau Iot. Pour cela, nous avons analysé plusieurs travaux de recherche que nous avons jugé importants et en liaison avec la problématique de sécurité dans le domaine Iot.

L'analyse des travaux existant montre que l'utilisation d'un algorithme unique ne donne pas les meilleurs résultats toujours, pour arriver à une bonne prédiction, cela dépend des données elles-mêmes, la technique utilisé pour le traitement des données et l'algorithme choisi avec ses paramètres.

Le choix d'un IDS dépend de plusieurs facteurs : l'infrastructure de sécurité, l'objective, la fiabilité, la réactivité, la facilité de mise en œuvre et l'adaptabilité.

Chapitre IV

Conception

4.1 Introduction

Dans tout ce qui a précédé, nous avons cerné la question de la sécurité de l'internet des objets d'une manière théorique en présentant les définitions des concepts liés à l'Iot.

L'étude bibliographique que nous avons menée, nous a permis d'étudier les différentes architectures d'apprentissage utilisées dans le domaine de la détection d'anomalies.

Notre objectif à travers ce travail est de mettre en place un modèle de détection d'anomalies qui doit : assurer un taux de détection élevé et un taux de fausses alarmes faible, en classant chaque type d'attaque en sa vraie classe. Le modèle réalisé est de type système de détection d'intrusions réseau (NIDS).

A travers ce chapitre, nous présentons notre approche étape par étape, pour répondre à la problématique posée auparavant en donnant notre conception.

4.2 Description du modèle proposé

Les modèles de l'apprentissage automatique offrent plus de performances dans le domaine de la détection d'anomalies, nous proposons une solution pour la détection d'anomalies qui met en œuvre une architecture de classification qui répond aux exigences du problème que nous traitons.

Donc, il s'agit d'une classification multi classes. Nous allons utiliser plusieurs modèles de classification pour sélectionner le plus performant parmi ces modèles.

4.3 Architecture du modèle proposé

La figure 11 montre les phases principales du système proposé. En commence par récupérer les données et les traiter. Les techniques de prétraitement sont utilisées pour adapter et faciliter l'application des algorithmes sur les données. Ensuite, nous avons choisi quelques algorithmes de classification. Ces algorithmes prennent comme entrée les données traitées pour ensuite entrainer à classifier ses données en plusieurs catégories. Ensuite, l'évaluation de ses modèles se fait par des

métriques d'évaluations, en fin, nous sélectionnons le meilleur classifieur adéquat pour ces données.

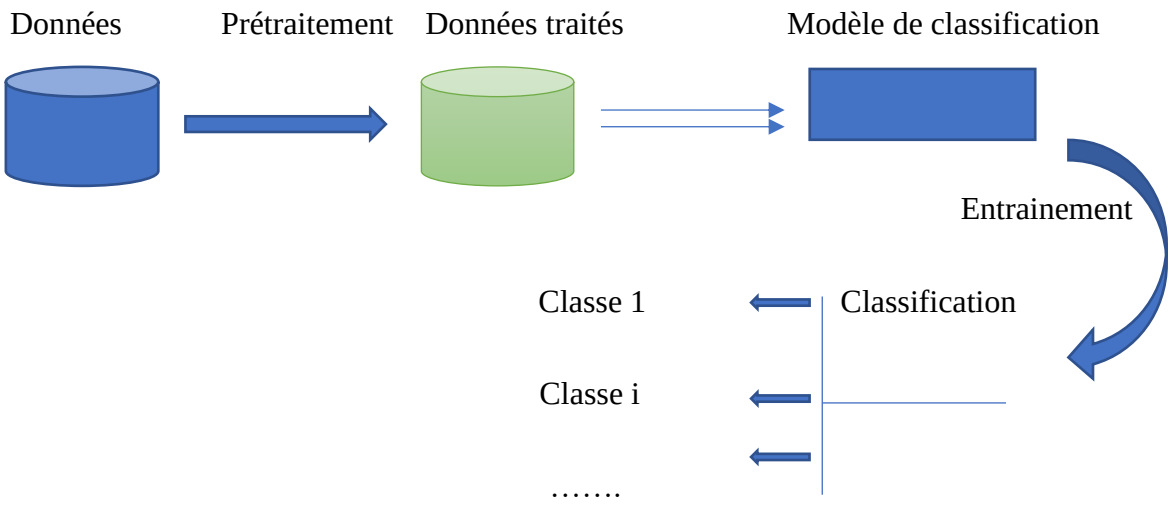

Figure 11 Le modèle proposé

A la fin de la classification, pour chaque instance le modèle donne une prédiction, qui est la classe de l'attaque prédite.

Dans ce qui suit nous détaillons, ces différentes phases.

4.3.1 le choix du dataset

Nous avons choisi de travailler sur le dataset AWID, qui est une collection d'ensembles de données accessibles au public dans un format facilement distribuable. Il comprend des données de réseau Wi-Fi collectées à partir de l'environnements réseau**,** pour cela, un laboratoire physique qui émule de manière réaliste une infrastructure SOHO typique est créé. Un certain nombre de stations mobiles et fixes ont été utilisées comme des clients du réseau, tandis qu'un seul attaquant mobile lançait diverses attaques.

Plus précisément, le réseau valide se composait d'un ordinateur de bureau, de deux ordinateurs portables, de deux smartphones, d'une tablette et d'un smart TV. La position de l'ordinateur bureau et de la Smart TV est restée statique tout au long de toutes les expériences. Les smartphones ont changé de position à l'intérieur des installations du laboratoire et ont rejoint / quitté le réseau à plusieurs reprises au cours des expériences. Enfin, les ordinateurs portables étaient semi-

statiques, c'est-à-dire qu'ils changeaient rarement de position. Les services exécutés sur les clients qui étaient responsables de la production de trafic étaient la navigation Web, la VoIP et le téléchargement de fichiers.

La figure 12 illustre les plans du laboratoire et les positions relatives des nœuds à l'intérieur des installations du laboratoire tout au long de la collecte de l'ensemble de données.

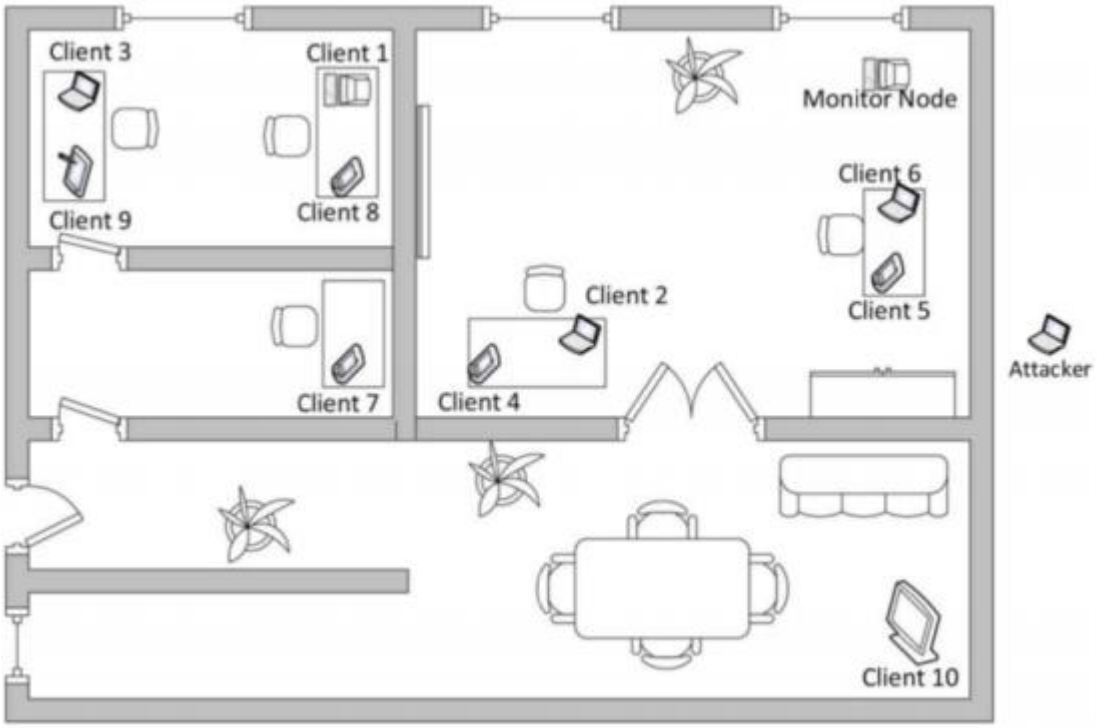

Figure 12 Environnement de collecte de données du dataset AWID

L'annexe A montre les spécifications détaillées des équipements utilisés pour construire le dataset AWID (Kolias et al., 2015)

Nous avons travaillé sur le dataset **AWID-CLS-R.** Ce dataset contient deux fichiers **AWID-CLS-R-Trn** et **AWID-CLS-R-Tst**, qui sont l'ensemble d'entrainement et l'ensemble de données de test, respectivement, chaque enregistrement est représenté par un vecteur de 156 attributs, la signification de quelques attributs est donnée dans l'annexe D.

Nous avons choisi AWID, car ce dataset est le premier ensemble de données accessible au public qui respecte le protocole réseau 802.11 (Kolias et al., 2015) et il est orienté vers la détection d'intrusion et plus spécifiquement la détection d'intrusion dans les réseaux sans fil, aussi les données dans AWID sont équilibrées c'est-à-dire la distribution des attaques est presque égale (2

jusqu'à 4 % du donnée pour chaque attaque), ce qui justifie notre choix. Le tableau 7 montre la structure du dataset AWID :

Tableau VII La structure du dataset AWID

Le nom du fichier	**Classes**	**Nombre totale d'enregistrements**	**Enregistrements normaux**	**Enregistrements de types attaque**
AWID-CLS-F-Trn	4	37817835	36732463	1085372
AWID-CLS-F-Tst	4	4570463	4373934	196529
AWID-CLS-R-Trn	4	1795575	1633190	162385
AWID-CLS-R-Tst	4	575643	530785	44858
AWID-ATK-F-Trn	16	37817835	36732463	1085372
AWID-ATK-F-Tst	16	4570463	4373934	196529
AWID-ATK-R-Trn	16	1795575	1633190	162385
AWID-ATK-R-Tst	16	575643	530785	44858

La colonne Classes du tableau fait référence au nombre de classes contenues dans cette version particulière de l'ensemble de données (les classes représentent le type d'attaque), par exemple dans le fichier **AWID-CLS-R-Tst** on trouve 4 classes : flooding, impersonation, injection et normal. Donc, il s'agit d'une classification multi-classes.

Le tableau 8 donne le nombre d'enregistrements pour chaque classe dans **AWID-CLS-R-Trn** et **AWID-CLS-R-Tst**.

Tableau VIII Le nombre d'enregistrement pour chaque classe dans AWID-CLS-R.

La classe	Nombre d'enregistrement	
	AWID-CLS-R-Trn	**AWID-CLS-R-Tst**
Flooding	48484	8097
Impersonation	48522	20079
Injection	65379	16682
Normal	1633190	530785

4.3.2 Pré-traitement des données

La phase de traitement des données est une phase essentielle dans la classification. Elle consiste à structurer les données et les adapter à une forme utilisable et traitable par le modèle d'apprentissage proposé. Pour notre cas, en premier lieu, nous devons ajouter les entêtes aux données, la liste des entêtes est référencée dans [web6].

Dans **AWID-CLS-R** tous les attributs de l'ensemble de données ont des valeurs numériques ou nominales à l'exception de la valeur SSID qui prend des valeurs de chaîne de caractère. Donc, les échelles des attributs de l'ensemble de données sont fortement déséquilibrées.

Apres l'ajout des entêtes, nous procédons au traitement des données manquantes dans le dataset. Comme stratégie, nous allons remplacer les données manquantes d'une colonne par la médiane.

D'après (Kuhn & Johnson, 2013), Il existe de nombreuses options que nous pourrions envisager lors du remplacement d'une valeur manquante, par exemple :

- Une valeur constante qui a une signification dans le domaine, telle que 0, distincte de toutes les autres valeurs.

- Une valeur d'un autre enregistrement sélectionné au hasard.
- Une valeur moyenne, médiane ou mode pour la colonne.
- Une valeur estimée par un autre modèle prédictif.

Nous rappelons que la valeur médiane correspond à l'observation qui se trouve au point milieu de cette liste ordonnée. Elle correspond plus précisément à un pourcentage cumulé de 50 % (c'est-à-dire que 50 % des valeurs sont supérieures à la médiane et 50 % lui sont inférieures). La position de la médiane est :la valeur à la position [(n + 1) ÷ 2], le n désignant le nombre de valeurs dans un ensemble de données.

Pour calculer la médiane, il faut d'abord ordonner les données (les trier dans l'ordre ascendant). La médiane est le nombre qui se situe au point milieu.

Par exemple, dans le dataset, la colonne data.len qui représente la longueur des données, a un taux de valeur manquante égale à 0.502914, la médiane de cette colonne est égale à 1488, l'annexe B présente le taux des valeurs manquantes pour toutes les colonnes.

Nous avons opté pour l'utilisation de la stratégie qui consiste d'utiliser la médiane pour la gestion des données manquante car les données comprennent des valeurs aberrantes.

L'objectif de ce traitement est d'éviter les problèmes causés par des données manquantes pouvant survenir lors de la phase de l'apprentissage.

Après avoir fixé le problème posé par les valeurs manquante, la prochaine étape est la normalisation des données. Cette étape est une méthode de prétraitement des données qui permet de réduire la complexité des modèles. C'est également un préalable à l'application de certains algorithmes.

La normalisation standardise la moyenne et l'écart-type de tout type de distribution de données, ce qui permet de simplifier le problème d'apprentissage en s'affranchissant de ces deux paramètres. Pour effectuer cette transformation, on soustrait aux données leur moyenne empirique μ on les divisent par leur écart-type δ ,voir l'équation 4

$$X_{normalisé} = \frac{X - \mu}{\delta} \dots (eq4)$$

4.3.3 Classification

Une fois le pré- traitement des données est fait. Nous procédons d'abord à la sélection d'attribut.

Notre objectif à travers la sélection d'attributs est de sélectionner le sous-ensemble d'attributs le plus important et optimal afin d'améliorer la généralisation, la performance et réduire le coût de calcul des classificateurs.

Il existe trois types de méthodes de sélection d'attribut :

1. **Méthodes de filtrage**

Elles appliquent une mesure statistique pour attribuer un score à chaque attribut, chaque attribut du jeu de données sera noté en fonction de sa corrélation avec la variable cible. Les attributs sont classés en fonction de leurs scores et sélectionnés, sur la base de ce résultat, pour être conservés ou supprimés de l'ensemble de données.

2. **Méthodes d'emballage**

Dans ce type de méthodes un sous-ensemble d'attributs est sélectionné au hasard, ensuite vérifier l'efficacité du modèle, si les précisions sont satisfaisantes alors le processus est terminé sinon quelques attributs seront ajoutés et autres seront supprimés, et ainsi de suite jusqu'à obtenir la précision souhaitée. Vu le nombre important d'itérations qu'il fallait faire afin d'aboutir à la précision désirée, ces méthodes coûtent assez cher en temps de calcul.

3. **Méthodes incorporées**

Elles représentent un hybride des méthodes de filtrage et celle d'emballage, elles combinent les deux précédentes afin d'obtenir un modèle plus performent.

Pour notre cas, nous allons utiliser l'algorithme Random Forest qui est une méthode incorporée. Nous allons l'exécuter avec différents hyperparamètres pour former 3 ensembles d'attributs en variant ces paramètres à chaque exécution.

En ajoutant l'ensemble de tous les attributs, on va tester nos modèles avec 4 ensembles d'Attributs. Le tableau 9 donne notre approche :

Tableau IIX les ensembles d'attributs choisis

L'ensemble d'attributs	Le nombre d'attributs	Comment il construit
Ensemble 1	153	En prenant toutes les colonnes sauf celle qui ont un taux de valeur manquant égale à 1.
Ensemble 2	9	En choisissant les 9 plus importantes colonnes dans le modelé généré par l'algorithme Random Forest avec un seul estimateur.
Ensemble 3	21	En choisissant les 21 plus importantes colonnes dans le modelé générer par l'algorithme Random Forest avec 100 estimateurs.
Ensemble 4	18	En choisissant les colonnes qui ont une importance plus de 0.03 dans le modelé générer par l'algorithme Random Forest avec 200 estimateurs.

Pour la classification, nous allons travailler avec trois algorithmes de classification à savoir Random Forest, Naive-bays et XGBoost, que nous détaillons dans ce qui suit :

1. Random Forest

Plusieurs arbres de décision sont créés indépendamment pendant l'entrainement pour notre cas en a choisi d'utiliser 100 arbres de décision. Chaque arbre est créé dans un sous-ensemble de l'ensemble de données avec un nombre aléatoire d'attributs. Chaque arbre votera alors pour une classe, et la classe avec le nombre maximum d'électeurs sera associée à l'entrée.

La figure 13 montre le fonctionnement de l'algorithme Random Forest :

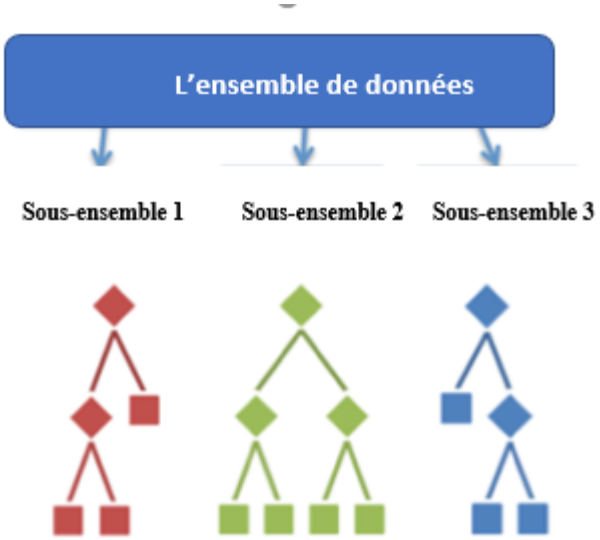

Figure 13 fonctionnement de l'algorithme Random Forest

Pseudo code :

Début

1.Sélectionnez au hasard m attributs parmi les attributs totales.

2.Sélectionnez le nœud racine en utilisant le meilleur fractionnement et formez divers arbres de décision.

3.Prédisez le résultat à l'aide de ces arbres de décision.

4.Calculez le vote pour chacune des classes prédites par chaque arbre.

5.La cible avec le vote le plus élevé est considérée comme la prédiction finale de l'algorithme de forêt aléatoire.

Fin

2. **Naive-bays**

C'est une technique de classification basée sur le théorème de Bayes (chap.2: Seq : 3. 1.4.1 Eq 2).

Pseudo code :

1. Convertir l'ensemble de données en une table de fréquences.

2. Créer une table de vraisemblance.

3. Utiliser l'équation bayésienne naïve (Equation 2) pour calculer la probabilité postérieure de chaque classe. La classe avec la probabilité postérieure la plus élevée est le résultat de la prédiction.

3. XGBoost

XGBoost est une implémentation d'arbres de décision à gradient amélioré conçus pour la vitesse et les performances.

XGBoost est l'une des implémentations du concept Gradient amélioré, mais ce qui rend XGBoost unique, c'est qu'il utilise "une formalisation de modèle plus régularisée pour contrôler le surapprentissage, ce qui lui donne de meilleures performances", selon l'auteur de l'algorithme, **Tianqi Chen**. Par conséquent, cela aide à réduire le surapprentissage.

4.3.4 Métriques d'évaluations

L'évaluation d'un algorithme d'apprentissage automatique est une partie essentielle dans tout projet. Le modèle peut donner des résultats satisfaisants lorsqu'il est évalué à l'aide d'une métrique, par exemple la précision, mais peut donner des résultats médiocres lorsqu'il est évalué par rapport à d'autres métriques. La plupart du temps, nous utilisons la précision de la classification pour mesurer les performances de modèles, mais cela ne suffit pas pour évaluer le modèle.

Nous avons opté pour les mesures d'évaluation suivantes pour évaluer nos modèles de classification :

La précision

Il s'agit du rapport entre le nombre de prédictions correctes et le nombre total d'échantillons d'entrée.

AUC

AUC est l'une des mesures d'évaluation les plus utilisées. L'aire sous la courbe est souvent utilisée comme mesure de la qualité des modèles de classification. Un classificateur aléatoire a une aire sous la courbe de 0,5, tandis que l'AUC pour un classificateur parfait est égal à 1. En pratique, la plupart des modèles de classification ont une AUC comprise entre 0,5 et 1.

L'AUC d'un classificateur est égale à la probabilité que le classificateur classe un échantillon positif choisi au hasard plus haut qu'un échantillon négatif choisi au hasard.

Sensibilité (TPR / Recall)

Elle est définie comme $\boldsymbol{TPR} = \frac{\boldsymbol{TP}}{\boldsymbol{FN + TP}} \ldots\ldots (\boldsymbol{Eq5})$. Le TPR correspond à la proportion de points de données positifs qui sont correctement considérés comme positifs, par rapport à tous les points de données positifs.

F1-score

F1 Score est la moyenne harmonique entre la précision et le rappel.

$$\boldsymbol{F1 - score} = \frac{\mathbf{1}}{\frac{\mathbf{1}}{\boldsymbol{Précision}} + \frac{\mathbf{1}}{\mathbf{Sensibilité}}} \ldots\ldots\ldots (\boldsymbol{Eq\ 6})$$

4.3 Conclusion

A travers ce chapitre, nous avons présenté l'aspect conceptuel de notre solution qui répond à la problématique de ce projet. Nous avons décrit notre modèle en fournissant son architecture globale, résumant ses différentes étapes.

Dans le chapitre suivant, nous entamerons la partie réalisation, dans laquelle nous expliquerons notre démarche suivie pour l'implémentation du modèle de classification.

Chapitre V

5.Réalisation

5.1 Introduction

Après avoir présenté la conception de notre modèle, nous passons à sa réalisation. En effet, cette étape de mise en œuvre de notre solution va permettre de mettre en place les différents concepts que nous avons utilisés afin de concevoir le modèle de détection d'anomalies. Dans ce chapitre, nous allons citer en premier temps toutes les technologies et les bibliothèques utilisées au cours de ce travail, en justifiant leurs choix. Ensuite, nous présenterons les différentes étapes de réalisation du modèle en citant les fonctions et méthodes implémentées.

5.2 les outils utilisés

Dans cette partie, nous décrivons les outils, les bibliothèques ainsi que les technologies utilisées pour la réalisation de notre solution.

1. Python

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions.

Le langage Python fonctionne sur la plupart des plates-formes informatiques, il être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

2. Google Colab

Colab est un environnement de bloc-notes Jupyter gratuit qui s'exécute entièrement dans le cloud. Plus important encore, il ne nécessite aucune configuration. Colab prend en charge de nombreuses bibliothèques d'apprentissage automatique populaires qui peuvent être facilement chargées dans votre ordinateur portable.

Les fonctionnalités de Google Colab :

- Écrire et exécuter du code en Python
- La documentation du code qui prend en charge les équations mathématiques
- Créer / télécharger / partager des cahiers
- Importer / enregistrer des blocs-notes de / vers Google Drive
- Importer / publier des blocs-notes depuis GitHub
- Importez des jeux de données externes, par exemple de Kaggle
- Intégrez PyTorch, TensorFlow, Keras, OpenCV
- Service Cloud gratuit avec GPU gratuit

3. **Numpy**

NumPy est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Cette bibliothèque fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

4. **Pandas**

Pandas est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.

5. **Scikit-learn**

Scikit-learn est une bibliothèque libre Python destinée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python, notamment NumPy et SciPy.

5.3 les fonctions réalisées

add_header : Cette fonction ajoute les entêtes aux données.

str_to_cat : Cette fonction remplace les colonnes de type chaînes de caractère par une colonne de valeurs catégorielles. Cela applique les changements en place.

Exemple :

```
>>> df = pd.DataFrame({'col1' : [1, 2, 3], 'col2' : ['a', 'b', 'a']})
>>> df
   col1 col2
0     1    a
1     2    b
2     3    a
# le type de col2 est : String
>>> str_to_cat(df)
>>> df
   col1 col2
0     1    a
1     2    b
2     3    a
# maintenant le type de col2 est : category
```

fix_missing **:** Cette fonction remplace les données manquantes dans une colonne avec la médiane, et ajoute une nouvelle colonne ***{name} _na*** qui spécifie que c'est une donnée manquante.

Exemple :

```
>>> df = pd.DataFrame({'col1' : [1, np.NaN, 3], 'col2' : [5, 2, 2]})
>>> df
   col1 col2
0     1    5
1   nan    2
2     3    2
>>> fix_missing(df, df['col1'], 'col1', {})
>>> df
   col1 col2 col1_na
0     1    5   False
1     2    2    True
2     3    2   False
```

proc_df : Cette fonction prend une trame de données et sépare la variable de réponse (dans notre cas la colonne class), et la transforme en une trame de données entièrement numérique. Pour chaque colonne les valeurs nulles sont remplacées par la valeur médiane de la colonne.

***scale_vars* :** pour la normalisation de données avec la formule (Equation 4).

***class_report* :** pour calculer et afficher toutes les métriques d'évaluation utilisés.

***plot_ROC_curve* :** pour afficher le graphe de la courbe ROC pour un modèle donnée, elle permet d'afficher la courbe ROC pour chaque classe ou en peut afficher une seule courbe qui est la moyenne de ces courbes.

***roc_comp* :** cette fonction permet de faire une comparaison entre déférents modèles, en affichant leur courbe ROC dans un seul graphe.

5.4 Conclusion

Ce chapitre a permis de décrire la phase d'implémentation du modèle proposé de manière détaillée. Nous avons présenté les différentes techniques et bibliothèques utilisées, ainsi que l'architecture technique du modèle avec toutes les fonctions et les méthodes établies.

Dans le prochain chapitre, nous évaluerons les résultats de notre modèle en utilisant le jeu de données **AWID-CLS-R**.

Chapitre VI

6. Résultats et tests

6.1 Introduction

Dans cette partie, nous allons donner les résultats que nous avons obtenu. L'exécution est faite avec Google Colab sur une machine GPU.

Dans la première étape de la phase de sélection d'attributs, nous avons calculé l'importance de chaque colonne avec l'algorithme Random Forest puis dans la deuxième étape, nous avons construit 4 ensembles d'attributs pour les utiliser dans les tests.

Notre objectif est de sélectionner le sou ensemble le plus représentatif des 156 attributs du data set.

6.2 L'importance de chaque colonne

L'importance d'une colonne est définie comme sa contribution relative à la prise de décision de l'algorithme. L'algorithme Random Forest est capable de tirer des conclusions sur les caractéristiques qui contribuent le plus à la prise de décision dans le modèle.

L'algorithme attribue à chaque colonne du jeu de donnée une importance sous forme d'un nombre compris entre 0 et 1. Les valeurs qui se rapproches de 1 sont les plus significatives. c'est-à-dire qui contribuent le plus et celle qui sont nulle n'ont aucune contribution dans le processus de classification.

Nous avons utilisé l'algorithme pour trouver l'importance de chaque colonne, les résultats sont illustrés dans le tableau 10. Nous avons présenté uniquement les 13 importantes colonnes, l'importance de toutes les colonnes de l'ensemble de donnée est donnée dans l'annexe C. Ce tableau est très utile pour la phase de sélection d'attributs.

Tableau X L'importance de chaque colonne

La colonne	L'importance	La colonne	L'importance
wlan.ta	0.057	wlan.sa	0.041
wlan.wep.icv	0.053	wlan.ra	0.04
wlan.fc.subtype	0.051	frame.time_delta_displayed	0.04
wlan.da	0.049	frame.time_relative	0.038
wlan.duration	0.046	frame.cap_len	0.037
frame.len	0.043	Data.len	0.037
wlan.fc.pwrmgt	0.036		

6.4 La sélection d'attributs

Nous avons testé nos modèles avec quatre ensembles d'attributs pour tirer le sous-ensemble d'attributs qui généralise ce dataset.

6.4.1 L'ensemble d'attributs 1

C'est l'ensemble de toutes les colonnes sans :

- frame.dlt
- wlan.qos.buf_state_indicated
- radiotap.mactime
- wlan.ba.bm
- wlan.ba.control.ackpolicy
- wlan.ba.control.cbitmap
- wlan.ba.control.multitid
- wlan.bar.compressed.tidinfo
- wlan.bar.type
- wlan_mgt.fixed.fragment
- wlan_mgt.fixed.sequence
- wlan_mgt.tcprep.link_mrg
- wlan_mgt.tcprep.trsmt_pow

Les colonnes mentionnées précédemment ont un taux de valeurs manquantes égale à 1, c'est-à-dire sont toutes nulles, donc pas besoin de les inclure.

Le taux de valeurs manquantes varie entre 0 et 1, il représente le rapport entre le nombre d'observation qui ont des valeur nulles (non enregistrés) et le nombre total d'observation dans une colonne.

6.4.2 L'ensemble d'attributs 2

Cet ensemble est représenté par les colonnes suivantes :

- radiotap.datarate
- wlan.fc.type_subtype
- wlan.seq
- wlan_mgt.fixed.capabilities .preamble
- wlan_mgt.fixed.timestamp
- wlan_mgt.fixed.beacon
- wlan_mgt.tim.dtim_period
- data.len
- class

6.4.3 L'ensemble d'attributs 3

- frame.time_epoch
- frame.time_relative
- radiotap.datarate
- wlan.fc.type_subtype
- wlan.fc.ds
- wlan.fc.retry
- wlan.fc.protected
- wlan.ta
- wlan.seq
- wlan_mgt.fixed.capabilities .ess
- wlan_mgt.fixed.capabilities .preamble
- wlan_mgt.fixed.capabilities .short_slot_time
- wlan_mgt.fixed.timestamp
- wlan_mgt.fixed.beacon
- wlan_mgt.fixed.auth.alg
- wlan_mgt.fixed.auth_seq
- wlan_mgt.tim.dtim_period
- wlan_mgt.rsn.akms.type
- wlan.wep.key
- data.len
- class

6.4.4 L'ensemble d'attributs 4

Cet ensemble représente les colonnes les plus importantes définies grâce au tableau 8.

- frame.time_epoch
- frame.time_delta
- frame.time_delta_displayed
- frame.time_relative
- frame.len
- frame.cap_len
- wlan.fc.subtype
- wlan.fc.ds
- wlan.duration
- wlan.ra
- wlan.da
- wlan.ta
- wlan.sa
- wlan.seq
- wlan.wep.iv
- wlan.wep.icv
- data.len
- class

6.5 Résultats de la classification

Dans la classification, chaque classifieur est entrainé sur les séquences de données **AWID-CLS-R-Trn**. Et son évaluation est effectuée sur la séquence de test **AWID-CLS-R-Tst.** Nous avons utilisé la technique de validation Hold-out pour la phase de validation.

Les résultats sont donnés dans les tableaux 11,12 et 13. Ils sont interprétés dans la section Interprétations des résultats. Les figures 14, 15 et 16 montrent la courbe ROC de chaque modèle réalisé. La figure 17 représente toutes les courbes dans un seul graphe.

6.5.1 Classification avec Random Forest

L'algorithme est exécuté avec les paramètres suivants :

n_estimators = 100 ,le nombre d'estimateur représente le nombre d'arbre de décision utilisé dans l'algorithme Random Forest, pour des valeurs supérieurs à 100, nous ne trouvons pas une influence sur les résultats.

Bootstrap = True, pour choisir aléatoirement des échantillons de l'ensemble d'entrainement lors de la construction d'arbres. Si la valeur est False, l'ensemble de données complet est utilisé pour construire chaque arbre.

n_jobs = -1, *n_jobs* est le nombre de travaux à exécuter en parallèle, la valeur -1 signifie l'utilisation de tous les processeurs.

Tableau XI Classification avec Random Forest

L'ensemble d'attributs	La classe	La précision	F1-score	recall	AUC	Temps d'entrainement
Ensemble 1	flooding	0.999642	0.815849	0.689144	0.817278	4min 17s
	impersonation	0.999321	0.136513	0.073261	0.891697	
	injection	1	0.905847	0.827898	0.993055	
	normal	0.956747	0.977893	0.999994	0.962115	
Ensemble 2	flooding	0.200221	0.160639	0.134124	0.77687	2min 38s
	impersonation	0.291129	0.116809	0.073061	0.491589	
	injection	0.47101	0.60143	0.831735	0.911724	
	normal	0.94698	0.951364	0.955788	0.680717	
Ensemble 3	flooding	0.999734	0.633364	0.463505	0.859379	2min 16s
	impersonation	1	0.042315	0.021615	0.443923	
	injection	1	0.905093	0.826639	0.999899	
	normal	0.951797	0.975302	0.999998	0.882968	
Ensemble 4	flooding	0.999313	0.700048	0.538718	0.87787	1min 20s
	impersonation	1	0.120296	0.063997	0.970586	
	injection	1	0.905847	0.827898	0.99477	
	normal	0.954331	0.976629	0.999994	0.969111	

La courbe ROC

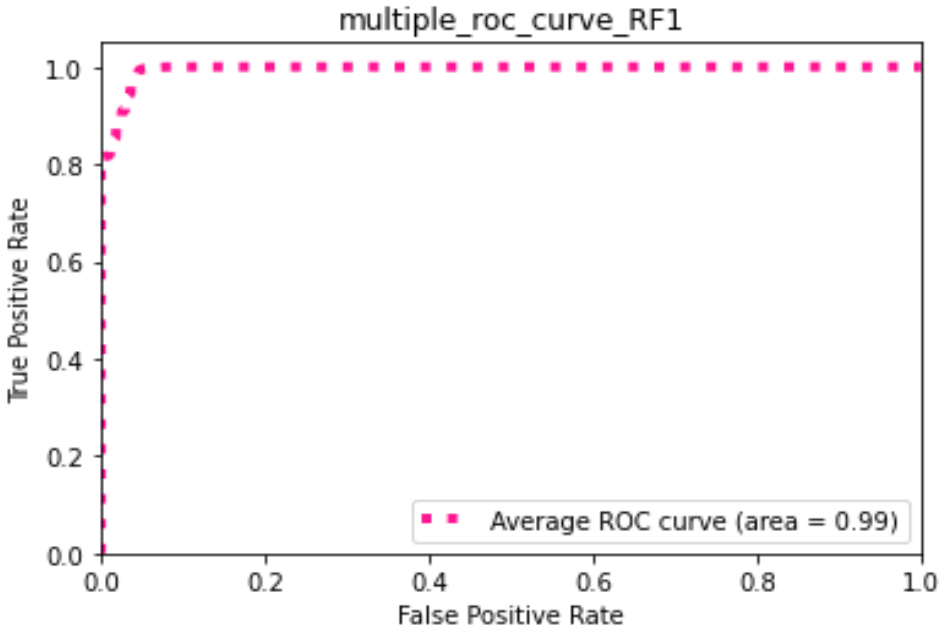

Figure 14 La courbe ROC pour le modèle Random Forest

La courbe dans la figure 14 montre que le classifieur Random Forest donne une très bonne séparation entre les classes.et il est très performant.

6.5.2 Classification avec l'algorithme Naive-Bayes

Tableau XII Classification avec Naive-Bayes

L'ensemble d'attributs	La classe	La précision	F1-score	recall	AUC	Temps d'entrainement
Ensemble 1	**flooding**	**0.031278**	**0.060659**	**1**	**0.925626**	**4.67 s**
	impersonation	**0**	**0**	**0**	**0**	
	injection	**1**	**0.899822**	**0.817888**	**0.47955**	
	normal	**0.929785**	**0.675953**	**0.530992**	**0.326341**	
Ensemble 2	**flooding**	**0.073225**	**0.136338**	**0.987279**	**0.909473**	**439 ms**
	impersonation	**0**	**0**	**0**	**0.924841**	
	injection	**0.293944**	**0.43803**	**0.85919**	**0.945397**	
	normal	**0.999783**	**0.866319**	**0.764292**	**0.918663**	
Ensemble 3	**flooding**	**0.073924**	**0.13767**	**0.999876**	**0.935695**	**813 ms**
	impersonation	**0**	**0**	**0**	**0**	
	injection	**0.997965**	**0.902102**	**0.823043**	**0.476322**	
	normal	**0.953138**	**0.877113**	**0.812321**	**0.139732**	
Ensemble 4	**flooding**	**0.030791**	**0.059743**	**1**	**0.919588**	**791 ms**
	impersonation	**0**	**0**	**0**	**0**	
	injection	**1**	**0.899822**	**0.817888**	**0.479513**	
	normal	**0.928824**	**0.669426**	**0.523284**	**0.33007**	

La courbe ROC :

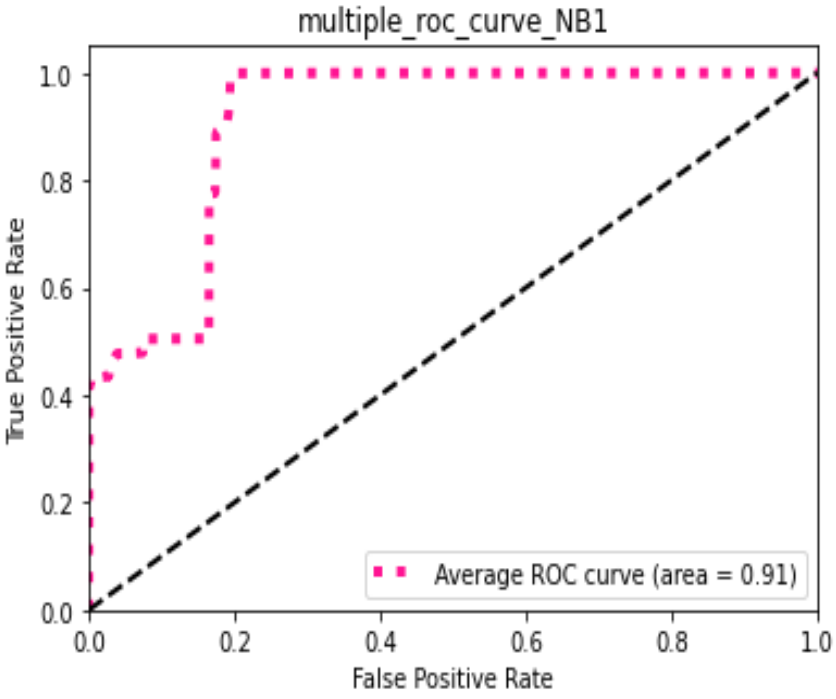

Figure 15 La courbe ROC pour le modèle Naïve Bayes

La figure 15 montre que ce classifieur ne donne pas une bonne séparation entre les classes, el il n'est pas efficace dans la détection.

6.5.3 Classification avec l'algorithme XGBoost

L'algorithme est exécuté avec les paramètres suivants :

objective='multi:softprob , configurer XGBoost pour effectuer une classification multiclasse. C'est à dire, pour chaque instance de test, l'algorithme va prédirai sa classe qui prend 4 modalités : Injection, Impersonation, Flooding, Normal.

n_estimators = 100.

max_depth = 3, fixer la profondeur maximale d'un arbre à trois niveaux . L'augmentation de cette valeur rendra le modèle plus complexe.

Tableau XIIII Classification avec XGBoost

L'ensemble d'attributs	La classe	La précision	F1-score	recall	AUC	Temps d'entrainement
Ensemble 1	flooding	0.999108	0.817226	0.691367	0.954478	16min 57s
	impersonation	0.902769	0.135182	0.073061	0.6945	
	injection	0.437639	0.572595	0.827898	0.980604	
	normal	0.980147	0.985722	0.99136	0.980374	
Ensemble 2	flooding	0.971119	0.233786	0.132889	0.912126	1min 49s
	impersonation	0.63534	0.131052	0.073061	0.826249	
	injection	0.471599	0.596456	0.811234	0.961205	
	normal	0.947738	0.958981	0.970495	0.902289	
Ensemble 3	flooding	0.990122	0.732944	0.58182	0.942393	3min 18s
	impersonation	0.084637	0.07776	0.071916	0.924001	
	injection	0.998698	0.905313	0.827898	0.980778	
	normal	0.953903	0.962108	0.970455	0.942216	
Ensemble 4	flooding	0.998478	0.723478	0.567247	0.894714	4min 7s
	impersonation	1	0.119768	0.063698	0.07561	
	injection	1	0.905452	0.827239	0.991358	
	normal	0.954698	0.976818	0.999987	0.972548	

La courbe ROC :

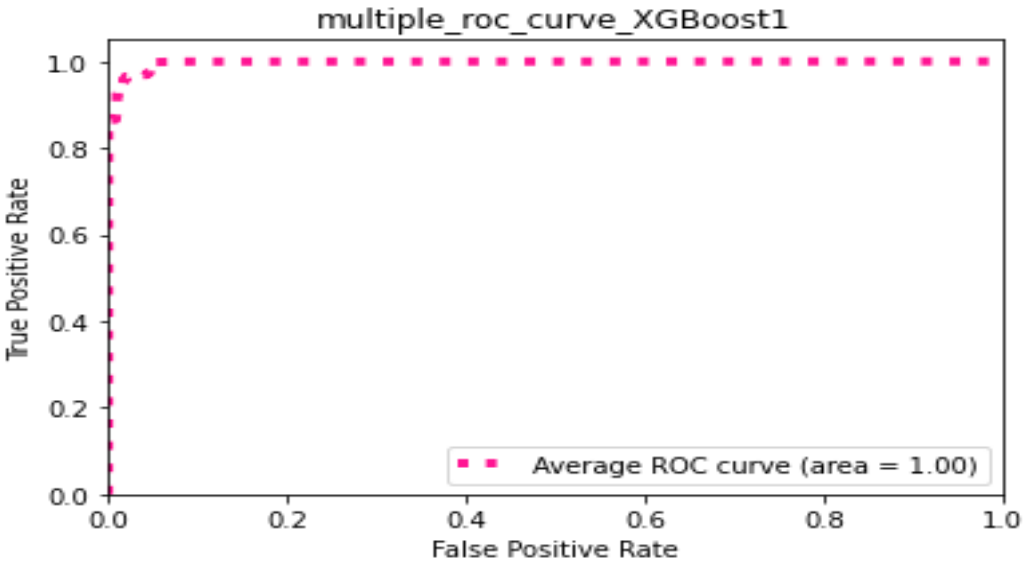

Figure 16 La courbe ROC pour le modèle XGBoost

La figure montre que ce classifieur est très précis et performant, il donne une très bonne séparation entre les classes.

6.5.4 Les courbes ROC de tous les modèles

Une courbe ROC peut montrer plusieurs résultats :

- Elle montre le compromis entre sensibilité et spécificité (toute augmentation de sensibilité s'accompagnera d'une diminution de spécificité).
- Plus la courbe se rapproche de la bordure gauche puis de la bordure supérieure de l'espace ROC, plus le test est précis.
- Plus la courbe se rapproche de la diagonale de 45 degrés de l'espace ROC, moins le test est précis.
- L'aire sous la courbe est une mesure de la précision.

La figure 17 regroupe les courbes ROC présentée précédaient dans un seul graphe :

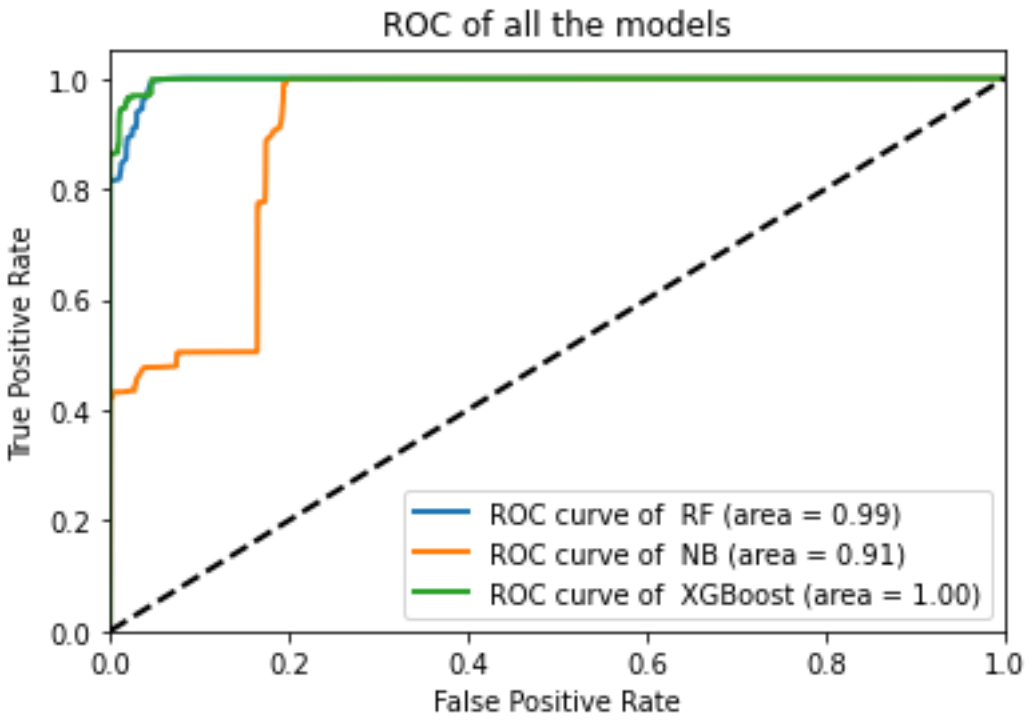

Figure 17 Les courbes ROC des 3 modèles

A Partir de ce graphe, la courbe ROC du modèle naïf bayes indique que ce modèle est moins capable de distinguer entre les classes par rapport aux deux autres modèles RF et XGBOOST, ce qu'est vrai car ce modèle est incapable de détecter les attaques de type impersonation.

La courbe ROC du modèle Random Forest est presque idéal, elle est très proche du coin supérieur gauche ce qui indique une meilleure performance. La même chose pour la courbe ROC qui correspond au modèle XGBoost, cette courbe est presque congruente au courbe du modèle Random Forest donc en peut dire la même chose pour la performance de ce modèle, et il donne aussi une bonne séparation entre les classes.

6.6 Interprétations des résultats

- Il est clair que l'ensemble 4 regroupe les fonctionnalités les plus importantes, il donne presque les mêmes résultats que l'ensemble de tous les attributs.

- L'algorithme NaiveBayes ne prédit pas l'attaque de type impersonation, et sa précision de détecter l'attaque flooding est très faible.

- Les deux algorithmes RF et XGBoost donnent des résultats similaires.

- Les deux algorithmes RF et XGBoost détectent les deux catégories : le flux normal et l'attaque impersonation avec une précision de 1.

- L'algorithme RF produit une bonne séparation entres les classes (la valeur de AUC est supérieure à 0.96 pour trois classes).

- Avec la métrique Recall, le Classifieur RF classe mieux le flux normal comme un flux normale avec une Précisons de 99.9999 %.

- Vu que les faux positifs et les faux négatifs ont un coût similaire donc La précision fonctionne mieux comme une métrique d'évaluation, Donc comme résultat final de ce projet, l'algorithme Random Forest est le meilleur algorithme pour la classification des attaques dans le dataset AWID.

6.7 Limites de notre solution

Comme toute technologie ou approche, l'utilisation de l'apprentissage automatique dans la prédiction des attaques trouve des difficultés d'application sur plusieurs niveaux et plus particulièrement au niveau des jeux de données. Dans ce qui suit, nous citerons ces limites.

1. **Limites liées aux jeux de données**

La source de données, comme mentionné ci-dessus, est un élément majeur de tout modèles d'apprentissage automatique. Cet élément peut présenter des limites soit à cause de sa structure ou sa façon d'utilisation. Les limites les plus répandues dans ce cas sont comme suit :

— Les jeux de données d'évaluation : Si les données utilisées pour la validation des modèles d'apprentissage ne sont pas généralement une partie de la base d'apprentissage alors cette partie d'évaluation n'est pas toujours un bon représentant des données en entrées.

— Le prétraitement des données : Ce problème est généralement posé dans le cas des jeux de données non équilibrées, c'est-à-dire lorsque certaines catégories de la variable à prédire sont beaucoup plus rares que d'autres, par conséquence, les méthodes basées sur ces données auront des difficultés de classification.

2. **Le temps d'apprentissage**

Certains algorithmes d'apprentissage nécessitent un nombre très élevé d'itération pour atteindre un bon niveau d'apprentissage.

3. **La nécessité de réapprentissage**

En cas d'ajoute d'une nouvelle classe d'attaque un réapprentissage complet est nécessaire sinon le fonctionnement du modèle change significativement.

6.8 Conclusion

Nous avons réalisé des modèles basés sur l'apprentissage machine pour la classification et la détection des attaques. Une comparaison de ces modèles en termes d'efficacité est fournie pour sélectionner la meilleure approche pour résoudre la problématique.

D'après les tests effectués, en utilisant les métriques d’évaluation définis, le classifieur Random Forest donne de très bons résultats dans la prédiction et classification des attaques.

Les résultats obtenus par ce dernier classificateur sont presque idéals ; il donne une très bonne séparation entre les classes, et une haute précision de détection.

Conclusion générale

Les travaux de recherche visant à améliorer les systèmes de détection d'intrusions sont très actifs ces dernières années. Leur utilisation est devenue indispensable puisqu'elle permet de connaitre toute activité anormale qui peut présenter un danger pour notre système ou notre réseau informatique. D'un autre côté, l'apprentissage automatique est une technique très brillante ayant de très grandes capacités de simulation des problèmes, de calcul et d'apprentissage à partir de grande quantité de données.

Le travail présenté dans ce mémoire consiste en une étude pour la mise en œuvre d'une solution basée sur l'apprentissage automatique permettant la détection d'anomalies dans un réseau Iot. Notre travail a été réalisé en deux grandes phases. Dans la première phase, nous avons effectué une étude bibliographique sur les réseaux IoT, aux systèmes de détection d'intrusions, à l'apprentissage profond et les différents travaux de ce dernier appliqué aux IDS dans les réseaux IoT. L'objectif de cette partie était d'avoir une vue globale sur ces domaines et les travaux existant afin de diriger nos choix pour la conception de notre solution. Ensuite, en se basant sur les concepts acquis à travers la première phase, nous avons opté pour les classifieurs Random Forest, XGboost et Naive-Bayes. Nous estimons que la solution de prédiction que nous avons réalisée permettra de prédire un flux de données anormale avec une haute précision.

Comme contribution, nous avons réalisé un outil capable de prédire avec une grande précision toutes les instances dans le data set AWID-CLS-R qui présente des attaques de type impersonation et injection et nous avons obtenu un taux de précision de 0.9993 pour la détection des attaques de type flooding.

Plusieurs perspectives d'évolution peuvent être envisagées, le travail réalisé peut être complété et amélioré en ajoutant les fonctionnalités suivantes :

- Entrainer les modèles sur un dataset qui contient plusieurs catégories d'attaques.
- Implémenter cette solution sur un IDS pour assurer une détection d'intrusions en temps réel.

Bibliographie

Aswale, P., Shukla, A., Bharati, P., Bharambe, S., & Palve, S. (2019). An Overview of Internet of Things: Architecture, Protocols and Challenges. In *Information and Communication Technology for Intelligent Systems* (pp. 299-308): Springer.

Debar, H., Dacier, M., & Wespi, A. (2000). *A revised taxonomy for intrusion-detection systems.* Paper presented at the Annales des télécommunications.

Debar, H., Dacier, M., & Wespi, A. J. C. n. (1999). Towards a taxonomy of intrusion-detection systems. *31*(8), 805-822.

Genuer, R. (2010). *Forêts aléatoires: aspects théoriques, sélection de variables et applications.*

HAMDAD, L. (2017). Présentation cours apprentissage. In (pp. 28).

Kolias, C., Kambourakis, G., Stavrou, A., & Gritzalis, S. (2015). Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. *IEEE Communications Surveys & Tutorials, 18*(1), 184-208.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26): Springer.

Mishra, P., Varadharajan, V., Tupakula, U., Pilli, E. S. J. I. C. S., & Tutorials. (2018). A detailed investigation and analysis of using machine learning techniques for intrusion detection. *21*(1), 686-728.

Mitchell, T. M. J. C. o. t. A. (1999). Machine learning and data mining. *42*(11), 30-36.

Noman, H. A., Abdullah, S. M., & Mohammed, H. I. J. I. J. o. C. S. I. (2015). An automated approach to detect deauthentication and disassociation dos attacks on wireless 802.11 networks. *12*(4), 107.

Patel, K. K., & Patel, S. M. (2016). Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing, 6*(5).

Pharate, A., Bhat, H., Shilimkar, V., & Mhetre, N. J. I. J. o. C. A. (2015). Classification of intrusion detection system. *118*(7).

Porras, P. A., & Valdes, A. (1998). *Live traffic analysis of TCP.* Paper presented at the IP gateways, Networks and Distributed Systems Security Symposium (San Diego, CA, USA).

Rezvy, S., Luo, Y., Petridis, M., Lasebae, A., & Zebin, T. (2019). *An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks.* Paper presented at the 2019 53rd Annual Conference on Information Sciences and Systems (CISS).

Shah, S. A. R., & Issac, B. J. F. G. C. S. (2018). Performance comparison of intrusion detection systems and application of machine learning to Snort system. *80*, 157-170.

Thing, V. L. (2017). *IEEE 802.11 network anomaly detection and attack classification: A deep learning approach.* Paper presented at the 2017 IEEE Wireless Communications and Networking Conference (WCNC).

Vasilomanolakis, E., Karuppayah, S., Mühlhäuser, M., & Fischer, M. J. A. C. S. (2015). Taxonomy and survey of collaborative intrusion detection. *47*(4), 1-33.

Vorakulpipat, C., Rattanalerdnusorn, E., Thaenkaew, P., & Hai, H. D. (2018). *Recent challenges, trends, and concerns related to IoT security: An evolutionary study.* Paper presented at the 2018 20th International Conference on Advanced Communication Technology (ICACT).

Wu, S. X., & Banzhaf, W. J. A. s. c. (2010). The use of computational intelligence in intrusion detection systems: A review. *10*(1), 1-35.

Xingmei, X., Jing, Z., & He, W. (2013). *Research on the basic characteristics, the key technologies, the network architecture and security problems of the internet of things.* Paper presented at the Proceedings of 2013 3rd International Conference on Computer Science and Network Technology.

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. *Journal of Network and Computer Applications, 84*, 25-37.

Zikria, Y. B., Yu, H., Afzal, M. K., Rehmani, M. H., & Hahm, O. (2018). Internet of Things (IoT): Operating System, Applications and Protocols Design, and Validation Techniques. In: Elsevier.

Web Bibliographié

Web1 *: (Sources https://www.futura-sciences.com/tech/definitions/informatique-antivirus-10999,consulté le 16/01/2020).*

Web2 : *(Source : https://www.xlstat.com/fr/solutions/fonctionnalites/classifieur-bayesien-naifwww.edureka.co ,consulté le 16/01/2020).*

Web3 *:(Source : http://blog.khaledtannir.net/2011/05/apriori/ ,consulté le 16/01/2020).*

Web4 *: (Source : https://dzone.com/articles/machine-learning-validation-techniques, consulté le 17/01/2020).*

Web5 *: (Source : http://icsdweb.aegean.gr/awid/, consulté le 04/12/2020).*

Web6 *: (Source :* http://icsdweb.aegean.gr/awid/attributes.html, *consulté le 03/01/2020).*

Web7 *: (Source :* https://linuxsecurityblog.com/2018/10/08/the-evil-twin-attack/, *consulté le 12/05/2020).*

Annexes

Annexe A : Les spécifications des équipements utilisées pour construire le AWID

Le nœud	Le type	La marque	Système d'exploitation	La carte réseau	CPU
Client1	Desktop	Custom	Ubuntu Linux 12.04 LTS	Netgear WNA3100 N300	Intel Core i7 3.2GHz
Client2	Laptop	Fujitsu-Siemens	Ubuntu Linux 12.04 LTS	Intel 3945ABG	Intel Core Duo T2050 1.6GHz
Client3	Laptop	Acer	Ubuntu Linux 12.04 LTS	Qualcomm Atheros AR9462	Intel Core i5 1.7GHz
Client4	Smartphone	iPhone 3G	iOS 4.2	NA	Samsung 32-bit RISC ARM 620MHz
Client5	Other	iPod Touch	iOS 3.1	NA	Samsung 32-bit RISC ARM 533MHz
Client6	Laptop	Acer Aspire 5750G	Windows 7	Broadcom BCM943227HM4L	Intel Core i5 2.8GHz
Client7	Smartphone	HTC Diamond	Windows Phone 6.1	NA	528 MHz ARM 11
Client8	Smartphone	Samsung Nexus	Android 4.2	NA	dual-core ARM Cortex-A9 1.2 GHz
Client9	Tablet	Samsung Galaxy Tab	Android 2.2	NA	Cortex-A8 1 GHz
Client10	Smart TV	LG 42LM7600S	Linux	NA	NA
L'attaquant	Laptop	Acer Aspire 5750G	Kali Linux 1.0.6	D-Link DWA-125/Linksys WUSB54GC	Intel Core i5 2.8GHz
Nœud de surveillance	Desktop	Custom	Linux Debian 7.3	Alpha AWUS036H	Core i7 2.4Ghz

Annexe B : Le taux de valeur manquante dans AWID-CLS-R-Trn

La colonne	Le taux des valeurs manquantes
class	0
data.len	0.502914
frame.cap_len	0
frame.dlt	1
frame.ignored	0
frame.interface_id	0
frame.len	0
frame.marked	0
frame.offset_shift	0
frame.time_delta	0
frame.time_delta_displayed	0
frame.time_epoch	0
frame.time_relative	0
radiotap.antenna	0.000434
radiotap.channel.freq	0.000434
radiotap.channel.type.2ghz	0.000434
radiotap.channel.type.5ghz	0.000434
radiotap.channel.type.cck	0.000434
radiotap.channel.type.dynamic	0.000434
radiotap.channel.type.gfsk	0.000434
radiotap.channel.type.gsm	0.000434
radiotap.channel.type.half	0.000434
radiotap.channel.type.ofdm	0.000434
radiotap.channel.type.passive	0.000434
radiotap.channel.type.quarter	0.000434
radiotap.channel.type.sturbo	0.000434
radiotap.channel.type.turbo	0.000434
radiotap.datarate	0
radiotap.dbm_antsignal	0.000434
radiotap.flags.badfcs	0.000434
radiotap.flags.cfp	0.000434
radiotap.flags.datapad	0.000434
radiotap.flags.fcs	0.000434
radiotap.flags.frag	0.000434
radiotap.flags.preamble	0.000434
radiotap.flags.shortgi	0.000434
radiotap.flags.wep	0.000434
radiotap.length	0
radiotap.mactime	0.000434
radiotap.pad	0
radiotap.present.ampdu	0

radiotap.present.antenna	0
radiotap.present.channel	0
radiotap.present.db_antnoise	0
radiotap.present.db_antsignal	0
radiotap.present.db_tx_attenuation	0
radiotap.present.dbm_antnoise	0
radiotap.present.dbm_antsignal	0
radiotap.present.dbm_tx_power	0
radiotap.present.ext	0
radiotap.present.fhss	0
radiotap.present.flags	0
radiotap.present.lock_quality	0
radiotap.present.mcs	0
radiotap.present.rate	0
radiotap.present.reserved	0
radiotap.present.rtap_ns	0
radiotap.present.rxflags	0
radiotap.present.tsft	0
radiotap.present.tx_attenuation	0
radiotap.present.vendor_ns	0
radiotap.present.vht	0
radiotap.present.xchannel	0
radiotap.rxflags.badplcp	0.000434
radiotap.version	0
wlan.ba.bm	0.999957
wlan.ba.control.ackpolicy	0.999944
wlan.ba.control.cbitmap	0.999944
wlan.ba.control.multitid	0.999944
wlan.bar.compressed.tidinfo	0.999987
wlan.bar.type	0.999987
wlan.bssid	0.252954
wlan.ccmp.extiv	0.998291
wlan.da	0.253618
wlan.duration	0.000664
wlan.fc.ds	0
wlan.fc.frag	0
wlan.fc.moredata	0
wlan.fc.order	0
wlan.fc.protected	0
wlan.fc.pwrmgt	0
wlan.fc.retry	0
wlan.fc.subtype	0
wlan.fc.type	0
wlan.fc.type_subtype	0
wlan.fc.version	0

wlan.fcs_good	0.000434
wlan.frag	0.253618
wlan.qos.ack	0.631126
wlan.qos.amsdupresent	0.631678
wlan.qos.bit4	0.918333
wlan.qos.buf_state_indicated	1
wlan.qos.buf_state_indicated.1	0.712793
wlan.qos.eosp	0.712793
wlan.qos.priority	0.631126
wlan.qos.tid	0.631126
wlan.qos.txop_dur_req	0.918333
wlan.ra	0.000664
wlan.sa	0.253618
wlan.seq	0.253618
wlan.ta	0.252187
wlan.tkip.extiv	0.982223
wlan.wep.icv	0.526193
wlan.wep.iv	0.526193
wlan.wep.key	0.506707
wlan_mgt.country_info.environment	0.997238
wlan_mgt.ds.current_channel	0.912646
wlan_mgt.fixed.aid	0.992714
wlan_mgt.fixed.auth.alg	0.991167
wlan_mgt.fixed.auth_seq	0.991167
wlan_mgt.fixed.beacon	0.915296
wlan_mgt.fixed.capabilities.agility	0.899336
wlan_mgt.fixed.capabilities.apsd	0.899336
wlan_mgt.fixed.capabilities.cfpoll.ap	0.899781
wlan_mgt.fixed.capabilities.del_blk_ack	0.899336
wlan_mgt.fixed.capabilities.dsss_ofdm	0.899336
wlan_mgt.fixed.capabilities.ess	0.899336
wlan_mgt.fixed.capabilities.ibss	0.899336
wlan_mgt.fixed.capabilities.imm_blk_ack	0.899336
wlan_mgt.fixed.capabilities.pbcc	0.899336
wlan_mgt.fixed.capabilities.preamble	0.899336
wlan_mgt.fixed.capabilities.privacy	0.899336
wlan_mgt.fixed.capabilities.radio_measurement	0.899336
wlan_mgt.fixed.capabilities.short_slot_time	0.899336
wlan_mgt.fixed.capabilities.spec_man	0.899336
wlan_mgt.fixed.category_code	0.999984
wlan_mgt.fixed.chanwidth	0.99999
wlan_mgt.fixed.current_ap	0.999949
wlan_mgt.fixed.fragment	0.999943
wlan_mgt.fixed.htact	0.99999
wlan_mgt.fixed.listen_ival	0.991326

wlan_mgt.fixed.reason_code	0.87651
wlan_mgt.fixed.sequence	0.999943
wlan_mgt.fixed.status_code	0.983876
wlan_mgt.fixed.timestamp	0.915296
wlan_mgt.rsn.akms.count	0.957149
wlan_mgt.rsn.akms.type	0.957159
wlan_mgt.rsn.capabilities.gtksa_replay_counter	0.957149
wlan_mgt.rsn.capabilities.mfpc	0.957149
wlan_mgt.rsn.capabilities.mfpr	0.957149
wlan_mgt.rsn.capabilities.no_pairwise	0.957149
wlan_mgt.rsn.capabilities.peerkey	0.957149
wlan_mgt.rsn.capabilities.preauth	0.957149
wlan_mgt.rsn.capabilities.ptksa_replay_counter	0.957149
wlan_mgt.rsn.gcs.type	0.957148
wlan_mgt.rsn.pcs.count	0.957148
wlan_mgt.rsn.version	0.957148
wlan_mgt.ssid	0.906886
wlan_mgt.tagged.all	0.893905
wlan_mgt.tcprep.link_mrg	0.999978
wlan_mgt.tcprep.trsmt_pow	0.999978
wlan_mgt.tim.bmapctl.multicast	0.927659
wlan_mgt.tim.bmapctl.offset	0.927659
wlan_mgt.tim.dtim_count	0.927659
wlan_mgt.tim.dtim_period	0.927659

Annexe C : L'importance de chaque colonne dans AWID-CLS-R

La colonne	L'importance
wlan.ta	0.062
wlan.da	0.049
wlan.wep.icv	0.048
wlan.fc.subtype	0.046
frame.cap_len	0.046
frame.time_delta_displayed	0.042
data.len	0.042
frame.len	0.041
frame.time_relative	0.04
wlan.ra	0.039
wlan.fc.ds	0.039
wlan.duration	0.039
frame.time_epoch	0.039
frame.time_delta	0.039
wlan.sa	0.037
wlan.wep.iv	0.031
wlan_mgt.fixed.reason_code	0.03
wlan.fc.pwrmgt	0.03
wlan.fc.type_subtype	0.029
radiotap.dbm_antsignal	0.027
wlan.seq	0.024
radiotap.datarate	0.02
wlan.fc.protected	0.014
wlan.fc.type	0.013
wlan.bssid	0.013
wlan.wep.key	0.011
wlan.fc.retry	0.009
radiotap.channel.type.ofdm	0.008
radiotap.channel.type.cck	0.008
wlan_mgt.fixed.timestamp	0.007
wlan_mgt.tagged.all	0.004
wlan_mgt.fixed.beacon	0.004
wlan_mgt.ds.current_channel	0.004
wlan.qos.ack	0.004
wlan_mgt.ssid	0.003
wlan_mgt.fixed.capabilities.short_slot_time	0.003
wlan_mgt.fixed.capabilities.privacy	0.003
wlan_mgt.fixed.capabilities.preamble	0.003
wlan.qos.priority	0.003
wlan.frag	0.003
wlan_mgt.tim.dtim_period	0.002

wlan_mgt.tim.dtim_count	0.002
wlan_mgt.fixed.status_code	0.002
wlan_mgt.fixed.capabilities.spec_man	0.002
wlan_mgt.fixed.capabilities.pbcc	0.002
wlan_mgt.fixed.capabilities.imm_blk_ack	0.002
wlan_mgt.fixed.capabilities.ibss	0.002
wlan_mgt.fixed.capabilities.ess	0.002
wlan_mgt.fixed.capabilities.dsss_ofdm	0.002
wlan_mgt.fixed.capabilities.del_blk_ack	0.002
wlan_mgt.fixed.capabilities.apsd	0.002
wlan_mgt.fixed.capabilities.agility	0.002
wlan_mgt.fixed.auth.alg	0.002
wlan.tkip.extiv	0.002
wlan.qos.tid	0.002
wlan.qos.eosp	0.002
wlan.qos.amsdupresent	0.002
wlan_mgt.tim.bmapctl.offset	0.001
wlan_mgt.tim.bmapctl.multicast	0.001
wlan_mgt.fixed.listen_ival	0.001
wlan_mgt.fixed.capabilities.radio_measurement	0.001
wlan_mgt.fixed.capabilities.cfpoll.ap	0.001
wlan_mgt.fixed.auth_seq	0.001
wlan.qos.buf_state_indicated.1	0.001
wlan_mgt.rsn.version	0
wlan_mgt.rsn.pcs.count	0
wlan_mgt.rsn.gcs.type	0
wlan_mgt.rsn.capabilities.ptksa_replay_counter	0
wlan_mgt.rsn.capabilities.preauth	0
wlan_mgt.rsn.capabilities.peerkey	0
wlan_mgt.rsn.capabilities.no_pairwise	0
wlan_mgt.rsn.capabilities.mfpr	0
wlan_mgt.rsn.capabilities.mfpc	0
wlan_mgt.rsn.capabilities.gtksa_replay_counter	0
wlan_mgt.rsn.akms.type	0
wlan_mgt.rsn.akms.count	0
wlan_mgt.fixed.htact	0
wlan_mgt.fixed.current_ap	0
wlan_mgt.fixed.chanwidth	0
wlan_mgt.fixed.category_code	0
wlan_mgt.fixed.aid	0
wlan_mgt.country_info.environment	0
wlan.qos.txop_dur_req	0
wlan.qos.bit4	0
wlan.fcs_good	0
wlan.fc.version	0

wlan.fc.order	0
wlan.fc.moredata	0
wlan.fc.frag	0
wlan.ccmp.extiv	0
radiotap.version	0
radiotap.rxflags.badplcp	0
radiotap.present.xchannel	0
radiotap.present.vht	0
radiotap.present.vendor_ns	0
radiotap.present.tx_attenuation	0
radiotap.present.tsft	0
radiotap.present.rxflags	0
radiotap.present.rtap_ns	0
radiotap.present.reserved	0
radiotap.present.rate	0
radiotap.present.mcs	0
radiotap.present.lock_quality	0
radiotap.present.flags	0
radiotap.present.fhss	0
radiotap.present.ext	0
radiotap.present.dbm_tx_power	0
radiotap.present.dbm_antsignal	0
radiotap.present.dbm_antnoise	0
radiotap.present.db_tx_attenuation	0
radiotap.present.db_antsignal	0
radiotap.present.db_antnoise	0
radiotap.present.channel	0
radiotap.present.antenna	0
radiotap.present.ampdu	0
radiotap.pad	0
radiotap.length	0
radiotap.flags.wep	0
radiotap.flags.shortgi	0
radiotap.flags.preamble	0
radiotap.flags.frag	0
radiotap.flags.fcs	0
radiotap.flags.datapad	0
radiotap.flags.cfp	0
radiotap.flags.badfcs	0
radiotap.channel.type.turbo	0
radiotap.channel.type.sturbo	0
radiotap.channel.type.quarter	0
radiotap.channel.type.passive	0
radiotap.channel.type.half	0
radiotap.channel.type.gsm	0

radiotap.channel.type.gfsk	0
radiotap.channel.type.dynamic	0
radiotap.channel.type.5ghz	0
radiotap.channel.type.2ghz	0
radiotap.channel.freq	0
radiotap.antenna	0
frame.offset_shift	0
frame.marked	0
frame.interface_id	0
frame.ignored	0

Annexe D : la signification de quelques colonnes dans AWID

Le nom de la colonne	La signification	Le type
comment	Commentaire.	Chaîne de caractères.
frame.cap_len	Longueur de trame stockée dans le fichier de capture.	Entier non signé, 4 octets.
frame.coloring_rule.name	Coloration Nom de la règle.	Chaîne de caractères.
frame.coloring_rule.string	Chaîne de règles à colorier.	Chaîne de caractères.
frame.comment	Commentaire.	Chaîne de caractères.
frame.encap_type	Type d'encapsulation.	Entier signé, 2 octets.
frame.ignored	La trame est ignorée.	Booléen.
frame.interface_description	Description de l'interface.	Chaîne de caractères.
frame.len	Longueur de trame sur le fil.	Entier non signé, 4 octets.
frame.md5_hash	Le hachage MD5 de la trame.	Chaîne de caractères.
frame.packet_flags_crc_error	L’erreur CRC.	Chaîne de caractères.
frame.packet_flags_reception_type	Type de réception.	Entier non signé, 4 octets.
frame.protocols	Les protocoles utilisés.	Chaîne de caractères.

frame.time	Heure et date d'arrivée.	Date.