

REPUBLIQUE DU SENEGAL

Un Peuple-Un But-Une Foi

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE ET DE L'INNOVATION

UNIVERSITE GASTON BERGER DE SAINT-LOUIS



U.F.R. DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

DEPARTEMENT DE PHYSIQUE APPLIQUEE

Mémoire de fin d'études pour l'obtention du diplôme de

Master Ingénierie en Electronique et Télécommunications (MIETEL)

Sujet : Classification des données de trafic réseau dans l'Internet des Objets : Comparaison de Performances de Modèles de Machine Learning Avancées

Présenté par :

Mlle Pauline Djessiordine SENE

Sous la Direction de :

Pr. Cherif Diallo

Jury :

Président : Pr. Amadou Seidou Maiga

Examinateur : Pr. Abdoul Karim Mbodji

Examinateur : Jean Gabriel Philippe Tevi

Encadrant : Pr. Chérif Diallo

Année Académique 2022/2023

Résumé

L'Internet des Objets (IoT) représente une révolution technologique majeure, connectant une multitude d'appareils intelligents pour créer un écosystème dynamique et interconnecté. Au cœur de cette transformation, la gestion du trafic réseau est essentielle pour assurer une communication fluide entre les dispositifs connectés. Avec la croissance rapide des appareils, cette tâche devient de plus en plus complexe. La classification des données de trafic réseau est donc un défi crucial, nécessitant des solutions innovantes pour garantir la sécurité, la performance et la fiabilité des réseaux. Les méthodes traditionnelles de gestion du trafic tels que la détection d'intrusion, la prévention des attaques DDoS (Distributed Denial of Service), et la mise en œuvre de pare-feu pour protéger le réseau contre les menaces ne suffisent plus face à la diversité des applications et des flux de données générés par les dispositifs IoT. Ce mémoire se penche sur la classification multi-classe des données de trafic réseau dans l'Internet des Objets en utilisant des techniques avancées de Machine Learning (ML). L'objectif est de développer des modèles adaptatifs capables de comprendre et de catégoriser efficacement les données de trafic, améliorant ainsi la gestion et la sécurité des réseaux. Dans ce travail, nous avons proposé trois modèles de Machine Learning (Arbre de Décision, RandomForest et AdaBoost), un modèle de réseau de neurone artificiel ANN et un modèle de réseau de neurone récurrent RNN en utilisant le dataset CIC-IoT-dataset2022 Les résultats montrent que DecisionTree et RandomForest sur-apprennent aux données d'entraînement. Cela signifie qu'ils sont très performants sur ces données, mais généralisent mal sur de nouvelles données. En revanche, le fait qu'AdaBoost n'atteigne pas un score parfait sur les données d'entraînement indique qu'il ne sur-apprend pas autant. AdaBoost se concentre sur l'amélioration progressive de la performance, ce qui conduit souvent à une meilleure généralisation sur les données de test. Le modèle de réseau de neurones à 4 couches a montré une précision de 97,98%, tandis que le RNN simple a atteint 98%. En conclusion, nous avons mis en place un modèle IA appris à partir de données réelles qui nous permet de faire une classification efficace des données de trafic réseau.

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce mémoire. Le chemin jusqu'à la rédaction finale a été parsemé de défis, mais il a été rendu possible grâce au soutien et à l'encouragement de nombreuses personnes exceptionnelles.

En premier lieu, je tiens à remercier mon directeur de mémoire, Pr Cherif Diallo pour sa guidance éclairée, sa patience et son expertise. Ses conseils précieux et ses suggestions constructives ont grandement enrichi ce travail de recherche.

Je souhaite exprimer ma profonde gratitude envers mes éminents professeurs, dont les conseils éclairés et le dévouement ont été une source constante d'inspiration. Leur expertise et leur disponibilité ont joué un rôle déterminant dans ma progression académique. Je suis reconnaissante pour leur soutien inestimable qui a contribué de manière significative à la réussite de ce travail de recherche.

Je souhaite exprimer ma reconnaissance envers mes proches et ma famille, qui ont été d'un soutien inconditionnel tout au long de cette aventure académique. Leurs encouragements et leur compréhension ont été une source de réconfort dans les moments difficiles.

Enfin, je tiens à saluer l'ensemble de mes camarades de classe et amis qui ont partagé avec moi cette expérience académique. Leurs échanges, discussions et encouragements ont contribué à rendre ce parcours encore plus mémorable.

Merci à chacun d'entre vous pour votre contribution précieuse à la réalisation de ce mémoire. Votre soutien a été la pierre angulaire de cette réussite.

Table des Matières

Résumé i

Remerciements ii

Table Des Figures vi

Liste des Tableaux viii

Liste des Abréviations ix

Introduction générale 1

Chapitre 1 Les Fondamentaux sur L'IoT 2

1.1 Introduction 2
1.2 Quelques Définitions de l'IoT 2
1.3 Concepts de base de l'IoT 3
1.4 Architecture de l'IoT 4
1.4.1 Architecture à 3 Couches 4
1.4.2 Architecture à 5 couches 5
1.5 Applications de l'IoT 6
1.5.1 Les maisons intelligentes 6
1.5.2 Villes intelligentes : cas d'un transport intelligent 7
1.5.3 Santé intelligente 9
1.5.4 Environnement intelligent et agriculture 9
1.6 Sources de Menaces Pour la sécurité dans les applications IoT 9
1.6.1 Attaques par Déni de Service (DDoS) 9
1.6.2 Vulnérabilités de Sécurité des Périphériques 10
1.6.3 Attaque Main-in-the-middle 10
1.6.4 Attaque de site d'hameçonnage (phishing) 10
1.6.5 Attaque d'accès 10
1.6.6 Attaques de routage 10
1.7 Solutions de Défense pour la Sécurité de l'environnement IoT 11
1.7.1 L'Authentification 11
1.7.2 Confidentialité 11

1.7.3 Tolérance aux pannes…… 11
1.7.4 Gestion de confiance…… 11
1.7.5 Cryptologie…… 12
1.7.6 Système de Détection d'Intrusion……... 13
1.7.6.1 Méthodes de détection d'intrusion (IDS)…… 13
1.7.6.2 Catégories de Système de Détection D'Intrusion (IDS)……... 15
1.7.7 Système de Prévention D'Intrusion IPS…… 20
1.8 Conclusion…… 22

Chapitre 2 Machine Learning pour la Classification 23

2.1 Introduction…… 23
2.2 L'Intelligence Artificielle…… 23
2.3 Le Machine Learning (ML)…… 24
2.3.1 Quelques Modèles d'apprentissage du ML…… 25
2.3.2 L'apprentissage supervisé……... 25
2.3.3 Apprentissage non supervisée…… 26
2.3.4 Apprentissage par renforcement……... 26
2.4 Les Réseaux de neurones…… 27
2.5 Le Deep Learning…… 28
2.5.1 Quelques Méthodes de Deep Learning…… 29
2.5.2 Les réseaux de neurones convolutifs CNN)…… 29
2.5.3 Les Réseaux de neurones récurrents RNN (Recurrent Neural Networks)…. 33
2.5.4 Auto encodeurs AE…… 37
2.5.5 Exemples d'Auto encodeurs…… 38
2.6 Conclusion…… 39

Chapitre 3 Classification des données de trafic réseau dans l'Internet des Objets (IoT) 40

3.1 Introduction…… 40
3.2 Présentation des Outils utilisés…… 40
3.2.1 Python…… 40
3.2.2 Anaconda…… 41
3.2.3 Jupyter Notebook…… 41
3.2.4 Environnement de développement……... 41

3.2.5 Les librairies…… 42
3.3 Présentation de notre Dataset…… 42
3.4 Analyse Exploratoire des Données (EDA)……..... 43
3.4.1 Analyse de la forme…… 44
3.4.2 Analyse du fond…… 45
3.5 Prétraitement des données……..... 47
3.6 Résultats…… 50
3.6.1 Métriques d'évaluation…… 50
3.6.2 Algorithmes du Machine Learning…… 51
3.6.2.1 Evaluation de Decision-Tree…… 51
3.6.2.2 Evaluation du Modèle Random Forest…… 53
3.6.2.3 Evaluation du modèle d'Adaboost…… 55
3.6.3 Evaluation avec un modèle de Réseaux de neurones à 4 couches…… 56
3.6.4 Evaluation avec le Modèle de Réseaux de Neurones Récurrents (RNN)……. 57
3.7 Discussion…… 58
3.8 Conclusion…… 59
Conclusion générale…… 60
Bibliographie…… 61

Table Des Figures

1. 1 Architecture 3 couches..... 4
1. 2 Modèle Architecture à 5 couches..... 5
1. 3 Schéma fonctionnel d'un système de maison intelligente [1]..... 7
1. 4 Schéma fonctionnel d'un système de stationnement intelligent [1]...... 8
1. 5 Architecture pour la communication multimédia sécurisée dans l'environnement IoT[11].....12
1. 6 Classification des IDS en fonction de la méthodologie utilisée.....13
1. 7 Processus de détection d'un IDS basé sur les signatures.....14
1. 8 Processus de détection basé sur les anomalies.....15
1.9 NIDS déployé à l'extérieur du périmètre d'un pare-feu [17].....16
1. 10 NIDS en mode déploiement complet [17].....17
1. 11 IDS basé sur l'hôte (HIDS).....17
1. 12 IDS hybride.....18
1. 13 IDS centralisé.....19
1. 14 IDS distribué.....20
1. 15 Système de détection et de prévention d'intrusion.....21
2. 1 Les Branches de L'intelligence artificielle.....23
2. 2 Développement d'un modèle de machine Learning.....25
2. 3 Algorithmes machine Learning.....25
2. 4 Structure d'un réseau de neurone artificiel ou perceptron multicouche.....27
2. 5 Notion de fonction d'activation.....28
2. 6 Architecture typique d'un CNN.....29
2. 7 Schéma du parcours de la fenêtre de filtre sur l'image.....31
2. 8 Exemple de moyen et Max Pooling avec un filtre 2 x 2 en pas de 2.....32
2. 9 Fonction d'activation ReLU.....32
2. 10 Architecture d'une couche Fully-Connected.....33
2. 11 Illustration simplifiée d'un RNN.....34
2. 12 Version récurrente et version dépliée [31].....34
2. 13 Architecture LSTM.....36
2. 14 Architecture GRU.....37
2. 15 Structure d'une auto encodeur [23].....38
2. 16 Structure d'une auto encodeur profond.....39
3.1 Laboratoire intelligente [25].....43

3.2 Répartition de la totalité des données 44
3.3 Répartition des données de la variable Target......45
3.4 Nouvelle répartition des données de la variable Target.....46
3.5 Matrice de corrélation 46
3.6 Résultats de la modélisation avec Random Forest 52
3.7 Histogramme d'évaluation des caractéristiques importantes utilisées par Decision-Tree.....53
3.8 Résultats de la modélisation avec Random Forest.....54
3.9 Résultats de la modélisation avec AdaBoost 55
3.10 Performance du réseau de neurone par rapport aux données d'entrainement et de validation.....57
3.11 Performance du RNN par rapport aux données d'entrainement et de validation .58

Liste des Tableaux

Tableau 1 : Nouveaux Caractéristiques du trafic réseau 47
Tableau 2 : Répartition des données de la variable Target après division.....50
Tableau 3: Résultats de la modélisation avec Decision Tree 52
Tableau 4 : Résultats de la modélisation avec Random Forest 54
Tableau 5 : Résultats de la modélisation avec AdaBoost 55

Liste des Abréviations

IoT	Internet of Things
ML	Machine Learning
IA	Intelligence Artificielle
DDOS	Distributed Denial of Service
RFID	Radio Frequency Identification
IEEE	Institute of Electrical and Electronics Engineers
VM	Virtual Machine
DSE	Dossier de Santé Electronique
MQTT	Message Queuing Telemetry Transport
UIT	Union Internationales des Télécommunications
ISO/IEC	International Organization for Standardization / International Electrotechnical Commission
GPS	Global Positioning System
QR	Quick Response
IETF	Internet Engineering Task Force
MIT	Massachusetts Institute of Technology
DL	Deep Learning
SVM	Support Vector Machine
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
AE	Auto-Encodeur
MLP	Multi Layers Perceptron
ReLU	Rectified Linear Unit
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
CIC	Canadian Institute for Cybersecurity

DAE	Denoising AutoEncoder
EDA	Exploratory Data Analysis
CRC	Contrôle de Redondance Cyclique
IPS	Intrusion Prevention System
HIDS	Host Intrusion Detection System
NIPS	Network Intrusion Prevention System
IDPS	Intrusion Detection and Prevention System
NBA	Network Behavior Analysis
NIDS	Network Intrusion Detection System
HIPS	Host Intrusion Prevention System
WIPS	Wireless Intrusion Prevention System

1. Introduction générale

Internet of Things (IoT) c'est-à-dire Internet des Objets en français, a émergé comme une révolution technologique, interconnectant une multitude d'appareils intelligents pour créer un écosystème dynamique et informatisé. Au cœur de cette transformation numérique, le domaine du trafic réseau dans l'IoT joue un rôle crucial en facilitant la communication entre les dispositifs connectés. Cependant, avec la prolifération rapide des appareils IoT, la gestion efficace du trafic réseau devient une tâche complexe. Dans ce contexte, la classification des données de trafic réseau se présente comme un défi majeur, nécessitant des solutions novatrices pour garantir la sécurité, la performance et la fiabilité des réseaux IoT. Les approches traditionnelles de gestion du trafic ne suffisent plus à répondre à la diversité des applications et des flux de données générés par les dispositifs IoT. C'est dans ce contexte que les techniques de L'Intelligence Artificielle (IA) émergent comme des outils puissants pour aborder ces problématiques complexes. Ce mémoire explore les aspects fondamentaux de la classification des données de trafic réseau dans l'IoT en se concentrant sur l'intégration de techniques avancées de Machine Learning (ML). En combinant l'intelligence artificielle avec les défis spécifiques du trafic IoT, notre objectif est de concevoir des modèles adaptatifs capables de comprendre et de catégoriser efficacement les données de trafic, contribuant ainsi à une gestion plus efficace et sécurisée des réseaux IoT.

Le but de notre travail consiste à faire une classification à partir des données de trafic réseau dans l'IoT en utilisant les techniques du Machine Learning. Notre mémoire sera organisé comme suit : Dans le premier chapitre de notre travail, nous allons voir les fondamentaux de l'IoT : leurs concepts de base, ses architectures, ses applications, ainsi que les enjeux auxquels il est exposé. Le deuxième chapitre de notre mémoire portera sur le Machine Learning pour la classification partant de l'apprentissage automatique jusqu'à l'apprentissage profond (Deep Learning) en passant par les réseaux de neurones. Le troisième et dernier chapitre de notre travail consistera à présenter notre apport personnel : une proposition de classification intelligente des données de trafic réseau en utilisant des Modèles avancées de machine Learning. Nous présenterons aussi les outils et le Dataset utilisés.

Chapitre 1

Les Fondamentaux sur l'Internet des Objets (IoT)

1.1 Introduction

De nos jours, l'influence de l'Internet se fait ressentir dans tous les domaines de notre existence, des avancées en intelligence artificielle jusqu'aux appareils intelligents. Toutefois, cette évolution ne s'arrête pas là. Nous nous acheminons vers une ère de connectivité encore plus répandue, où une multitude d'appareils sera interconnectée au Web. C'est l'ère de l'Internet des Objets, où une diversité considérable d'objets se connectera et échangera des informations, ouvrant ainsi la voie à de nouvelles possibilités et transformations dans notre façon de vivre et d'interagir avec la technologie. [1].

L'appellation « Internet des Objets » a vu le jour en 1999, inventé par Kevin Ashton, un collaborateur qui voulait qualifier le lien entre la technologie RFID (Radio Frequency Identification) et Internet. Le terme IOT est formalisé par le centre MIT (Massachusetts Institute of Technology) Auto-ID. Jusqu'à présent, il n'y a pas de définition standard pour l'IoT [4]. Nous allons citer quelques définitions sur l'IoT définis par différents auteurs/instituts de différentes manières.

1.2 Quelques Définitions de l'IoT

Selon l'UIT (l'Union Internationales des Télécommunications) : « *Infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution* » [2].

Selon l'IEEE (Institute of Electrical and Electronics Engineers) : « *L'internet des Objets envisage une configuration automatique, adaptative et un réseau complexe qui interconnecte les objets à l'internet grâce à l'utilisation des protocoles de communication standard. Les objets interconnectés ont des propriétés physiques ou virtuelles dans le monde numérique, détection /actionnement capacité, une fonction programmable et sont uniquement identifiables.*

La représentation contient des informations y compris l'identité, le statut, l'emplacement ou toutes informations commerciales, sociales ou privées pertinentes. Les objets offrent des services, avec ou sans intervention humaine, à travers l'exploitation de l'identification unique, la capture de données et la capacité de communication et d'actionnement » [2].

Selon ISO/IEC (International Organization for Standardization / International Electrotechnical Commission) : « *C'est une infrastructure des objets, personnes, systèmes et ressources d'informations avec des services intelligents pour leur permettre de traiter l'information du monde physique et du monde virtuel et réagir* » [2].
Selon l'IETF (Internet Engineering Task Force) : « *Internet des Objets (IoT) fait référence aux appareils qui sont souvent limités dans la communication et capacités de calcul, devenant de plus en plus courantes connectées à Internet, et à divers services qui sont construits sur les capacités que ces appareils fournissent conjointement. Ceci s'attend à ce que cette évolution inaugure plus de communication de machine à machine utilisant internet sans aucun utilisateur humain qui n'est activement impliqué* » [2].

Certains chercheurs définissent l'IoT comme un modèle qui regroupe les technologies de communication sans fil comme les réseaux de capteurs sans fil, les réseaux mobiles et les actionneurs [4].

1.3 Concepts de base de l'IoT

Une solution IoT est généralement formée des composants suivants :

- **Capteurs**

Les capteurs jouent un rôle crucial dans l'univers de l'Internet des Objets (IoT). Ils sont conçus pour détecter et réagir aux informations de l'environnement en générant des signaux électriques correspondants. Les capteurs utilisés dans les dispositifs IoT sont généralement de petite taille et économes en énergie, ce qui les rend idéaux pour une intégration facile dans divers appareils. Le microcontrôleur est responsable du traitement des signaux produit par un capteur pour l'interprétation, l'analyse et la prise de décision [2]

- **Actionneurs**

Les actionneurs jouent un rôle complémentaire essentiel aux capteurs dans le domaine de l'Internet des Objets. Ils sont conçus pour provoquer des changements dans l'environnement, tels que le réglage de la température d'un climatiseur [1]. Ces dispositifs ont la capacité de transformer l'énergie qui leur est fournie en phénomènes physiques concrets, tels que le mouvement, le dégagement de chaleur, ou même l'émission de lumière

- **Passerelle (Gateway)**

Dispositif matériel et logiciel qui permet de connecter un réseau à un autre. L'opération consiste à relier un réseau local à Internet. Les Gateway sont donc utilisées pour la communication de données en collectant les mesures effectuées par les nœuds de capteurs et en les transmettant à l'infrastructure Internet [2].

❖ **Cloud computing**

Mode de fourniture à la demande de puissance de calcul, de stockage de base de données, d'applications et de ressources informatiques. Il permet aux organisations de consommer une ressource informatique comme une machine virtuelle (VM), au lieu de construire une infrastructure informatique sur place. Le cloud computing concerne la transmission des données à des Data Centers via Internet [3]**.**

1.4 Architecture de l'IoT

L'architecture IoT constitue plusieurs composants de systèmes IoT connectés pour assurer que les données générées par les capteurs sont collectées, stockées et traitées par les bases de données et que les actionneurs des objets exécutent leurs commandes envoyées via une application utilisateur. C'est un cadre qui définit les composants physiques, l'organisation fonctionnelle et la configuration du réseau, les procédures opérationnelles et les formats de données à utiliser, Cependant, il n'existe pas d'architecture de référence standard unique pour l'IoT [5]. Différentes architectures ont été proposées par différents chercheurs.

1.4.1 Architecture à 3 Couches

Elle est l'architecture la plus basique. Elle comporte 3 couches à savoir :

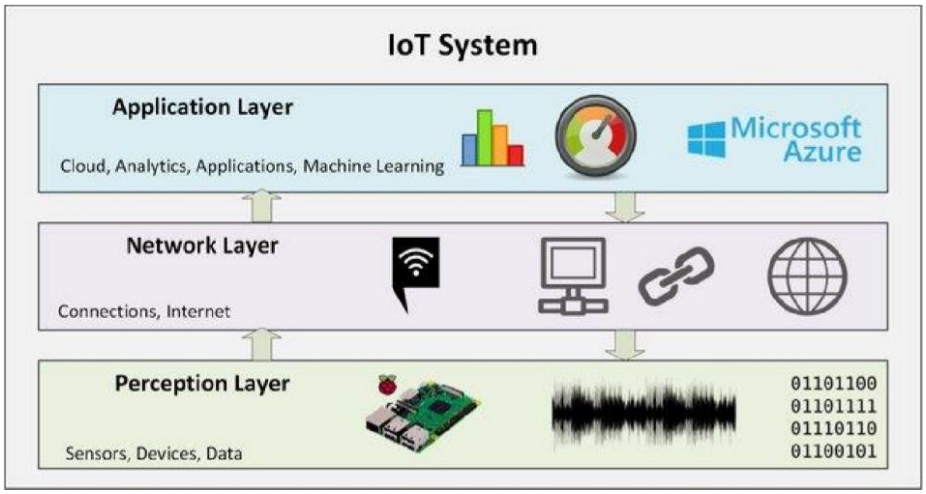


Figure 1. 1: Architecture 3 couches

- **La couche de perception**

C'est la couche physique, qui possède des capteurs pour détecter et collecter des informations sur l'environnement [1]. Cette couche est responsable de convertir des signaux analogiques en données numériques et vice versa [5].

- **La couche réseau**

Constitue le cœur de l'IoT. Elle transmet les informations recueillies par la couche de perception, elle contient les instrumentations logicielles et matérielles du réseau internet en plus des centres de gestion et d'information [4]. Elle est responsable de la connexion à d'autres objets intelligents, périphériques réseau et serveurs [1].

- **La couche d'application**

La couche d'application joue un rôle essentiel en offrant des services spécifiques aux utilisateurs. Elle englobe différentes applications dans lesquelles l'Internet des Objets peut être déployé, telles que les maisons intelligentes, les villes intelligentes, et bien d'autres domaines encore [1]. La couche application vise à concilier les besoins sociaux de l'Internet des Objets avec les technologies industrielles. En d'autres termes, elle joue un rôle intermédiaire crucial en mettant en relation les avancées technologiques de l'industrie avec leur application concrète pour répondre aux besoins humains [4].

1.4.2 Architecture à 5 couches

Du point de vue que la recherche se concentre souvent sur des aspects plus fins de l'Internet des Objets, l'architecture à 3 couches devient insuffisante. Par conséquent, une architecture à 5 couches est proposée.

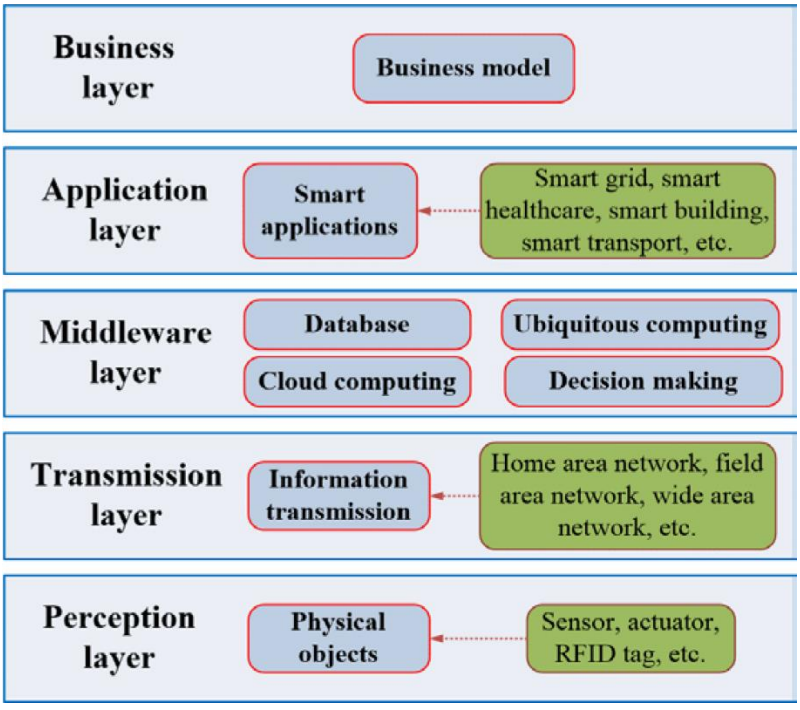


Figure 1. 2 : Modèle Architecture à 5 couches

Le rôle des couches de **perception** et **d'application** est le même que l'architecture à trois couches [1].

- **Couche de transmission (Network Layer)**

La couche de transmission, aussi connue sous le nom de couche réseau, joue un rôle similaire à celui de la couche réseau dans l'architecture en trois couches. Son principal objectif est d'assurer la transmission bidirectionnelle des informations entre la couche de perception et la couche de traitement. Cette couche est responsable de l'acheminement fluide des données collectées par les capteurs vers le niveau de traitement approprié et vice versa. De plus, un élément clé de cette couche est l'adressage de chaque composant du système à l'aide du protocole IPv6 [4].

- **Couche de traitement (couche middleware)**

Son rôle est de stocker, analyser et traiter les informations provenant de la couche de transport. Atteindre les objectifs de cette couche s'avère particulièrement difficile en raison du volume massif d'informations collectées sur les différents éléments du système. [4]. Elle utilise plusieurs technologies comme le traitement intelligent des données, le cloud computing, les bases de données…

- **Couche commerciale (couche métier)**

Elle est responsable de la confidentialité de l'utilisateur, de la gestion de l'ensemble du système IoT y compris les applications, les modules commerciaux et de profit.

1.5 Applications de l'IoT

Il existe plusieurs domaines dans lesquels des applications intelligentes ont été développés. L'IoT est couramment utilisé dans les activités suivantes : Réseaux électriques intelligents, villes intelligentes, la domotique, surveillance de la santé, les environnements industriels…

1.5.1 Les maisons intelligentes

Dans les maisons intelligentes, plusieurs capteurs sont développés qui vont fournir des services intelligents et automatisés à l'utilisateur. Les capteurs participent à la conservation de l'énergie en éteignant de manière automatique les lumières. Ils aident à automatiser les tâches quotidiennes et aident à maintenir une routine pour les personnes qui ont tendance à oublier. Les capteurs collectent des données (provenant de capteurs hétérogènes) sur l'environnement (température, gaz, lumière.), Ces données sont transmises à un agrégateur de contexte qui va transmettre les données collectées à un moteur de service sensible au contexte. Ce moteur sélectionne les services en fonction du contexte. Par exemple une application peut éteindre toutes les lumières en cas de fuite de gaz, activer automatiquement la climatisation lorsque

l'humidité augmente [1]. Dans les environnements intelligents, les applications de détection des chutes sont très utiles pour détecter si des personnes en âge avancée sont tombées. Les applications de maison intelligente sont très bénéfiques car elles participent à la surveillance de la santé des personnes, et informent immédiatement les proches en cas d'urgence. Malgré les nombreux avantages que présentent les applications de la maison intelligente, elles peuvent faire face à certains problèmes dont la sécurité et la confidentialité. Si la sécurité et la fiabilité du système ne sont pas garanties, un intrus peut attaquer le système et le faire se comporter de manière malveillante.

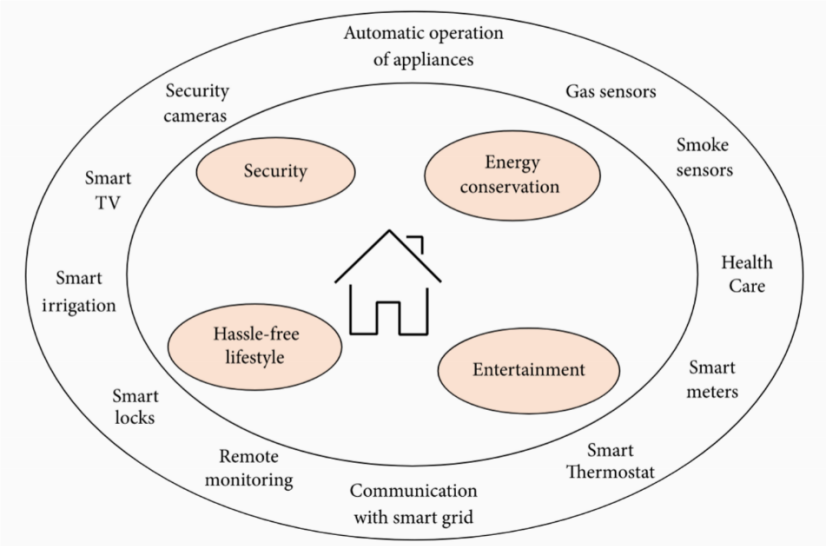


Figure 1. 3: Schéma fonctionnel d'un système de maison intelligente [1]

1.5.2 Villes intelligentes : cas d'un transport intelligent

A l'aide de capteurs et de système intelligent de traitement de l'information, les applications de transport intelligent peuvent gérer le trafic quotidien dans les villes. Ici, le principal objectif est de réduire les embouteillages, assurer un stationnement sans difficulté et éviter au maximum les accidents en dirigeant normalement la circulation. Les capteurs utilisés pour les applications de transport intelligent sont les capteurs Global Positioning System (GPS) pour la localisation, les gyroscopes pour la direction, les accéléromètres pour la vitesse, les capteurs RFID (Radio Frequency Identification) pour l'identification des véhicules, les caméras pour l'enregistrement des mouvements et du trafic des véhicules et les capteurs infrarouges pour le comptage des passagers et des véhicules [1].

On distingue plusieurs types d'applications dans ce domaine dont [1] :

- **Gestion intelligente du stationnement**

Dans un système de transport intelligent, on peut assurer un stationnement facile en vérifiant sur Internet pour savoir quel parking possède des places libres à l'aide des capteurs qui vont détecter si les emplacements sont libres ou occupés par des véhicules.

- **Feux de signalisation intelligents**

On appelle feux de signalisation intelligents, les feux de signalisation qui sont équipés de capacités de détection, de traitement et de communication. Ils détectent l'état de saturation du trafic à l'intersection et le nombre de trafic dans chaque sens. Ainsi, son objectif est destiné à réguler la circulation au niveau des intersections. Le feu de signalisation intelligent permet d'agir sur le comportement des conducteurs pour garantir leur sécurité.

- **Applications pour assurer la sécurité**

Plusieurs applications IoT ont été développées pour aider les conducteurs à conduire de manière plus sure. Ce sont des applications qui surveillent le comportement de conduite des conducteurs, et les accompagne à conduire en toute sécurité lorsqu'elles détectent des signes de fatigue, en les aidant à y faire face ou en leur suggérant ainsi de prendre du repos. Ces applications utilisent des technologies telles que la détection du visage, la détection de mouvement des yeux et la détection de pression sur la direction (utilisé pour mesurer l'adhérence des mains du conducteur sur la direction.

- **Applications de détection d'accident**

Des chercheurs ont développé une application pour smartphone qui détecte la survenance d'un accident à l'aide d'un accéléromètre et de données acoustiques.

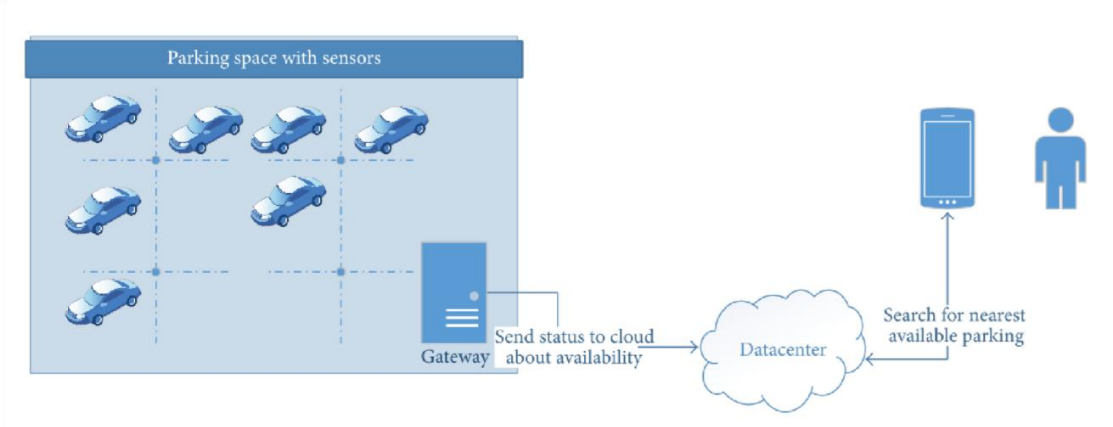


Figure 1. 4 : Schéma fonctionnel d'un système de stationnement intelligent [1].

1.5.3 Santé intelligente

Les appareils IoT se sont révélés très bénéfiques dans le domaine de la santé et du bien-être. Les applications de la santé intelligente sont devenues une aide précieuse notamment dans le suivi des maladies chroniques. Les capteurs IoT sont utilisés pour surveiller et enregistrer en continu l'état de santé des personnes malades et transmettre des indicateurs en cas d'avertissement anormaux. L'application IoT peut même proposer une prescription au patient en cas de problème mineur. A partir des applications de l'IoT on peut créer un dossier de santé électronique (DSE) pris en charge par le système de santé, qui contient tous les détails médicaux d'une personne. Un DSE peut être utilisé pour enregistrer les allergies, les augmentations de la glycémie et la tension artérielle [1].

1.5.4 Environnement intelligent et agriculture

Les agriculteurs utilisent des capteurs sur le terrain pour mesurer les paramètres tels que la température et l'humidité qui sont importants pour la production agricole. Pour une production efficace, les données de ces paramètres peuvent être utilisées. L'une des principales applications de l'IoT dans l'agriculture est la production sous serre. Les paramètres environnementaux mesurés en termes de température, d'informations sur le sol et d'humidité sont mesurés en temps réel et envoyés à un serveur pour analyse. Les résultats sont ensuite utilisés pour améliorer la qualité et le rendement des cultures. On peut aussi utiliser un code Quick Response (QR) pour identifier de manière unique un carton de produits agricoles [1]. Aujourd'hui, la pollution de l'air est une préoccupation importante car elle participe à la modification du climat de la terre et dégrade la qualité de l'air. Les capteurs électrochimiques de gaz toxique sont utilisés pour mesurer la pollution de l'air. Pour lutter contre les véhicules polluants, des étiquettes RFID permettant d'identifier les véhicules sont placés des deux côtés de la route avec les capteurs de gaz [1].

1.6 Sources de Menaces Pour la sécurité dans les applications IoT

L'évolution rapide de l'IoT a ouvert de nouvelles opportunités, mais elle a également introduit des défis significatifs en matière de sécurité. Les applications IoT, interconnectées et dépendantes des données de trafic réseau, sont vulnérables à diverses menaces qui peuvent compromettre la confidentialité, l'intégrité et la disponibilité des données. Voici une analyse approfondie des sources de menaces dans le contexte de la sécurité des applications IoT.

1.6.1 Attaques par Déni de Service (DDoS)

Les attaques DDoS restent l'une des menaces les plus courantes dans l'IoT. En surchargeant délibérément les ressources, elles peuvent paralyser les dispositifs connectés et perturber les

opérations normales. Cette vulnérabilité met en évidence la nécessité d'une surveillance constante du trafic réseau pour identifier et atténuer ces attaques.

1.6.2 Vulnérabilités de Sécurité des Périphériques

Les périphériques IoT peuvent devenir des points d'entrée pour les attaques si leurs protocoles de communication ne sont pas sécurisés. Les pirates peuvent exploiter ces vulnérabilités pour accéder aux données sensibles ou compromettre l'intégrité des informations échangées.

L'analyse du trafic réseau doit inclure la détection des modèles de comportement anormal liés à de telles tentatives d'exploitation.

1.6.3 Attaque Main-in-the-middle

Le protocole MQTT (Message Queuing Telemetry Transport) fonctionne sur les principes du modèle de publication / abonnement entre les clients et les abonnés à l'aide d'un agent de messagerie qui gère la communication entre les clients et les abonnés. Le rôle de l'agent est de filtrer tous les messages entrants des clients et de les distribuer correctement aux abonnés. Cela aide à découpler les clients de publication et d'abonnement les uns des autres. Si l'attaquant parvient à contrôler l'agent de messagerie et devenir un homme de milieu, il peut alors contrôler toutes les communications sans aucune connaissance des clients [6].

1.6.4 Attaque de site d'hameçonnage (phishing)

C'est une technique de manipulation psychologique qu'utilisent les attaquants pour amener la victime à divulguer des informations sensibles telles que le mot de passe. Le système IoT devient vulnérable aux attaques une fois que le mot de passe de l'utilisateur soit compromis. La couche réseau de l'IoT est très vulnérable aux attaques de sites de phishing [7].

1.6.5 Attaque d'accès

Également appelée menace persistante avancée (APT), est le type d'attaque où une personne non autorisée accède au réseau IoT. L'objectif de ce type d'attaque est d'usurper des données importantes. Les applications IoT sont très vulnérables aux attaques d'accès car elles envoient et reçoivent en permanence des données.

1.6.6 Attaques de routage

Au moment du transit des données, les nœuds malveillants d'une application IoT contrôlés par l'attaquant peuvent essayer de réorienter les chemins de routage. Un type spécifique d'attaque de routage est les attaques de gouffre (attack Sinkhole). Ces attaques sont effectuées en piratant un nœud de réseau ou en introduisant un nœud malveillant. Le nœud malveillant se présente comme le chemin le plus court vers la station de base, essaie d'éloigner le trafic des autres nœuds et attire des nœuds pour acheminer le trafic à travers ce chemin. Le nœud malveillant peut alors modifier les données, mettant en danger la sécurité du réseau [8].

1.7 Solutions de Défense pour la Sécurité de l'environnement IoT

Avec l'évolution des applications IoT, la sécurisation des appareils IoT devient de plus en plus difficile à cause des menaces courantes contre ces applications. Néanmoins, plusieurs solutions ont été proposées pour sécuriser le système IoT contre les attaques :

1.7.1 L'Authentification

L'authentification est un processus essentiel qui permet à un système informatique de vérifier l'identité d'un utilisateur ou d'un autre système. Son objectif est de gérer l'accès aux informations ou aux services de manière sécurisée. Dans le contexte des nombreux appareils connectés de l'Internet des Objets, l'authentification joue un rôle crucial pour assurer la sécurité des données et des interactions.

Ce mécanisme peut être réalisé à l'aide de diverses méthodes, telles que l'utilisation d'un jeton de sécurité de passerelle ou la mise en place d'une chaîne de confiance [8].

1.7.2 Confidentialité

La confidentialité est une démarche essentielle qui garantit que seules les personnes ayant les autorisations appropriées peuvent accéder aux données.

Dans le contexte des appareils IoT, il est primordial d'établir une politique de confidentialité claire et compréhensible. Ceci vise à minimiser les vulnérabilités face aux cyberattaques et à protéger la vie privée des utilisateurs [8]

1.7.3 Tolérance aux pannes

Pour que les appareils IoT puissent être tolérants aux pannes, des chercheurs ont identifié plusieurs conditions clés dans leur méthodologie. Premièrement, il est essentiel que la configuration soit complète et sécurisée pour tous les appareils dès le départ. Ensuite, afin de surveiller le réseau et les données de manière efficace, tous les appareils doivent être opérationnels. Enfin, pour une véritable tolérance aux pannes, chaque appareil doit être équipé de mécanismes de protection contre les interruptions de réseau et les menaces potentielles [8]. Les appareils doivent réagir immédiatement et récupérer/subir tout dommage lorsqu'un service est interrompu [9].

1.7.4 Gestion de confiance

La gestion de la confiance joue un rôle crucial dans la sécurité de l'Internet des Objets (IoT), aidant ainsi à surmonter les dangers et les incertitudes associés à cette technologie. Cette notion englobe à la fois la confidentialité et la sécurité des données. Selon des chercheurs [10], la confiance se réfère à la manière dont les individus se sentent lorsqu'ils interagissent avec les appareils IoT. Cela signifie que les utilisateurs doivent avoir accès à leurs ressources à tout

moment, tout en disposant d'appareils qui leur permettent de comprendre pleinement leurs interactions avec le système [8].

1.7.5 Cryptologie

La collecte de grandes quantités de données, notamment des vidéos, capturées par les appareils multimédia nécessite des ressources mémoire importantes et une large bande passante. Cela pose un défi particulier pour les appareils à faible consommation d'énergie et disposant de ressources limitées dans l'Internet des Objets (IoT). Pour traiter et transmettre ces données, ces appareils ont besoin d'une gestion efficace des ressources de mémoire et de bande passante. Dans ce contexte, il est primordial de garantir la sécurité du multimédia et la protection de la vie privée des utilisateurs. Cela implique de veiller à l'intégrité et à la confidentialité des informations de manière optimale. En optimisant l'utilisation des ressources disponibles et en mettant en place des mécanismes de sécurité robustes, les appareils IoT peuvent offrir une expérience multimédia fiable et sécurisée, tout en respectant la vie privée des utilisateurs et en protégeant leurs données sensibles. [11].

L'architecture de sécurité proposée est donnée par la figure suivante :

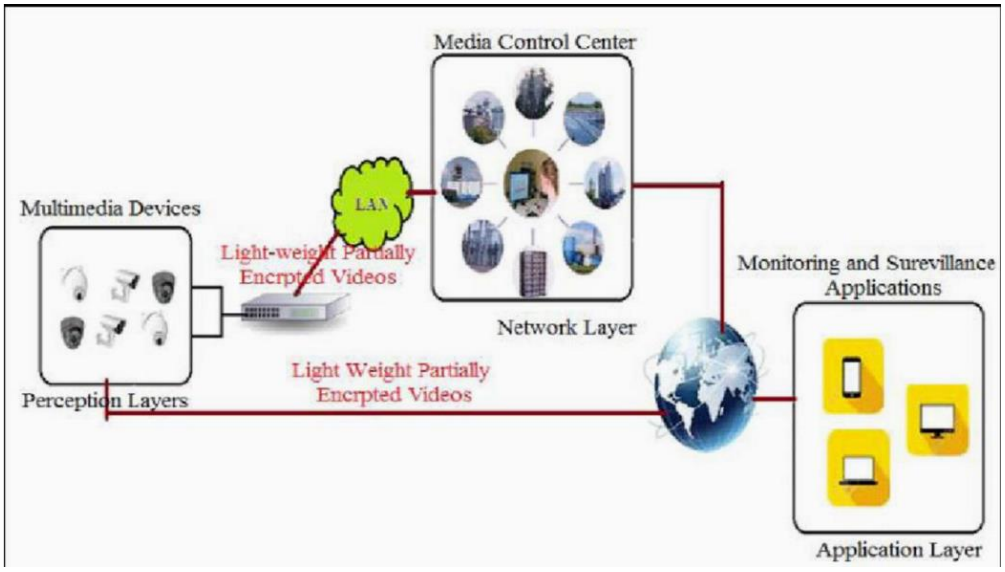


Figure 1. 5 : Architecture pour la communication multimédia sécurisée dans l'environnement IoT [11]

Pour mettre en œuvre la sécurité multimédia à différents niveaux, des techniques de cryptage partiel sont proposés. Seules quelques parties confidentielles des données seront cryptées. Dans cette architecture multimédia de sécurité proposée, les données que collectent les appareils multimédias et les capteurs seront d'abord cryptées de façon partielle avant d'être transmises sur le réseau pour que les derniers utilisateurs obtiennent ces données (Les vidéos) cryptées. Par contre, si un attaquant parvient à intercepter les données, la vidéo ne peut être examinée

que sous forme coder. Un autre avantage des schémas de cryptage proposés est le maintien des fonctionnalités d'encodage/ décodage des vidéos pour que les décodeurs ne tomberont pas en panne après la mise en œuvre des schémas proposés [11].

1.7.6 Système de Détection d'Intrusion

Dans un réseau ou un système, on appelle intrusion toute opération qui consiste à accéder sans autorisation aux données d'un système/ou réseau en contournant les dispositifs de sécurité mis en place. Dans [12], l'intrusion est décrite comme l'ensemble d'actions qui s'arrangent pour compromettre l'intégrité, la confidentialité ou la commodité d'une ressource. Un système de détection d'intrusion (IDS) est défini comme étant un mécanisme destiné à repérer des activités anormales ou suspects sur une cible analysée. Autrement dit, un IDS (Intrusion Detection System) est définit comme étant un système conçu pour automatiser le processus de détection d'intrusion. Un système de détection d'intrusion (IDS) se distingue d'un pare-feu en ce sens qu'un pare-feu ne signale pas une attaque à l'intérieur du réseau car il limite l'accès entre les réseaux pour prévenir les intrusions alors qu'un IDS décrit une intrusion suspectée une fois qu'elle a eu lieu et signale une alarme. Lorsqu'une intrusion est détectée, un IDS doit enregistrer les informations relatives à l'intrusion, de générer des alertes et prendre des mesures correctives.

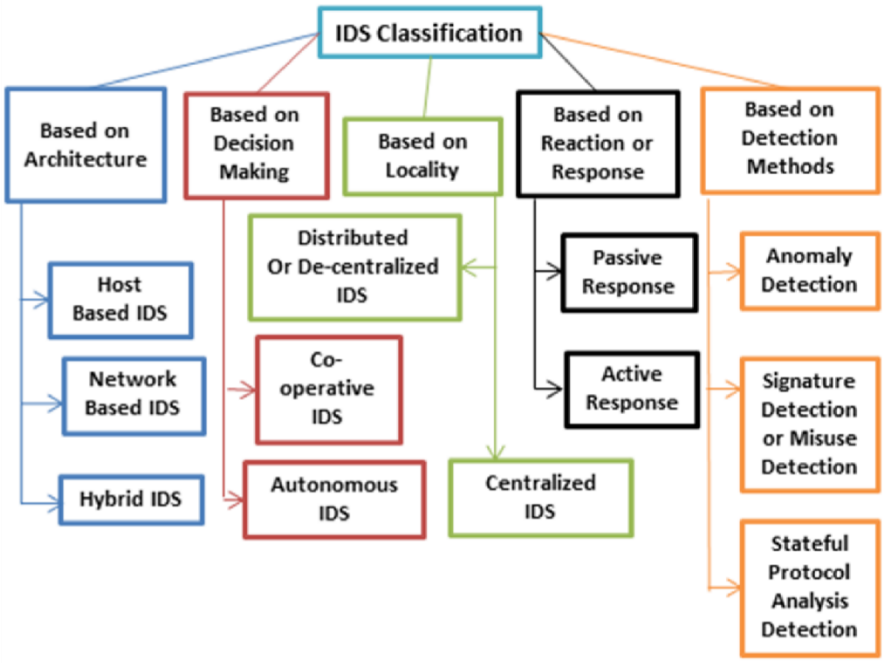


Figure 1. 6 : classification des IDS en fonction de la méthodologie utilisée

1.7.6.1 Méthodes de détection d'intrusion (IDS)

Les IDS différent en fonction de leur méthode de détection. On distingue principalement 3 méthodes de détection :

❖ **Détection d'intrusion basée sur les signatures**

Une signature est une chaine qui correspond à une attaque ou une menace connue. Autrement dit, Les signatures sont des modèles d'attaque préconfigurés et prédéterminés. Le concept derrière la détection de signature ou le schéma de détection d'abus est qu'il stocke la séquence du modèle, la signature de l'attaque ou de l'intrusion, dans la base de données. Lorsqu'un qu'une intrusion se produit, l'IDS fait correspondre les signatures d'intrusion avec la signature prédéfinie qui sont déjà stockées dans la base de données. En cas de correspondance réussie, le système génère une alarme. L'IDS basé sur la signature s'appuie sur des données de trafic connues pour analyser le trafic indésirable. Ce principe de détection d'intrusion ne détecte que les attaques répertoriées dont il possède l'empreinte, de ce fait, il nécessite des mises à jour fréquentes. L'avantage de cette méthode est la grande précision des attaques connues. Cependant, L'IDS basé sur les signatures est incapable de détecter des intrusions qui ne sont pas représentées dans sa base de connaissances [12].

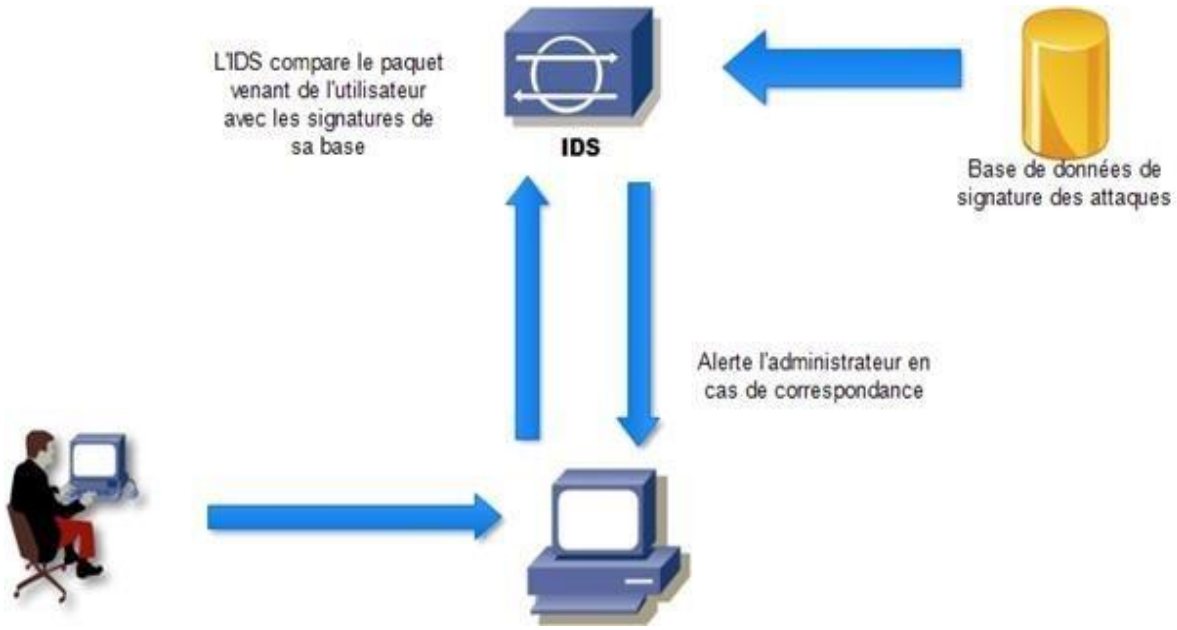


Figure 1. 7 : processus de détection d'un IDS basé sur les signatures

❖ **Détection d'intrusion basée sur les anomalies**

Un IDS qui examine le trafic réseau et détecte les données anormales est appelée détection basée sur les anomalies. Cette méthode dépend de l'identification des modèles et de leur comparaison avec le modèle de trafic normaux. Le système va reposer sur deux phases : Une phase d'apprentissage, au cours de laquelle ce dernier va étudier des comportements normaux de flux réseau et une phase de détection, le système analyse le

trafic et va chercher à identifier les événements anormaux en se basant sur ses connaissances. La précision contre les attaques inconnues est meilleure que par rapport aux IDS basés sur les signatures. Cependant, Elle souffre de faux positifs : une activité légitime antérieurement inconnue peut également être classée comme malveillante [13].

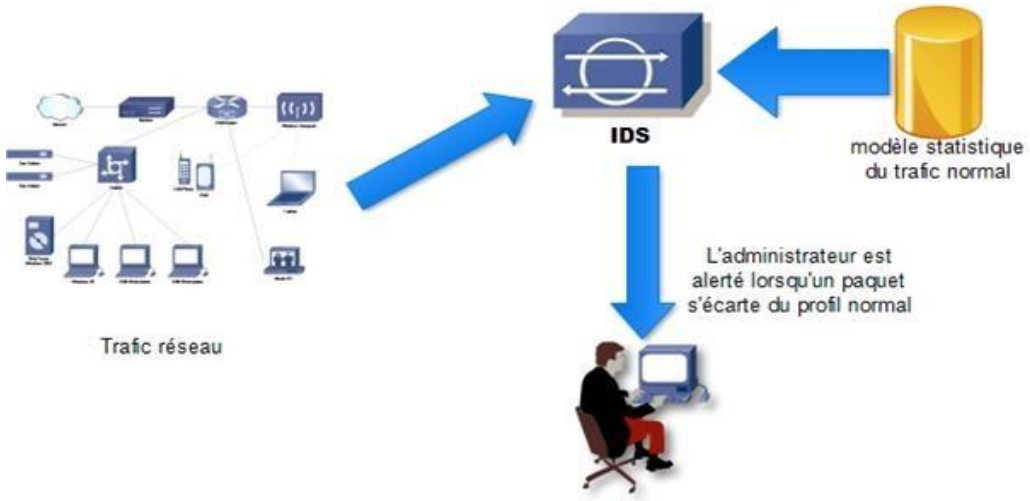


Figure 1. 8 : processus de détection basé sur les anomalies

❖ **Basée sur l'inspection de protocole avec état :**

Elle est similaire à la détection basée sur les anomalies mais, elle peut également analyser le trafic au niveau de la couche réseau et transport et le trafic spécifique au fournisseur au niveau de la couche application ce que la détection basée sur les signatures ne peut pas faire car elle utilise des profils spécifiques à l'hôte ou au réseau [14]. L'analyse de protocole avec état est capable de comprendre et de suivre l'état des protocoles qui ont une notion d'état, ce qui lui permet de détecter de nombreuses attaques que les autres méthodes ne peuvent pas.

Cependant, avec cette méthode il est très difficile voire impossible de développer des modèles de protocoles très précis car elle est gourmande en ressources, et est incapable de détecter les attaques qui ne violent pas les caractéristiques d'un comportement de protocole généralement acceptable [14].

1.7.6.2 Catégories de Système de Détection D'Intrusion (IDS)

On distingue principalement les catégories d'IDS suivants : L'IDS basé sur le réseau appelé NIDS, l'IDS basé sur l'hôte appelé (HIDS) et L'IDS hybride (NIDS + HIDS), IDS distribué et L'IDS centralisé**.**

❖ **IDS basé sur le réseau (NIDS : Network Intrusion Detection System)**

Ce type de système analyse le trafic réseau à toutes les couches du modèle OSI. Il permet à l'administrateur réseau, de comprendre en temps réel ce qui se passe sur son réseau et prendre des décisions en ayant toutes les informations. La plupart des NIDS sont facile à déployer sur un réseau et peuvent souvent visualiser le trafic de plusieurs systèmes à la fois. Les NIDS sont aussi appelés IDS passifs car ils informent le système administrateur qu'une attaque a ou a eu lieu, et il prend les mesures adéquates pour assurer la sécurité du système [14]. Le cryptage du trafic sur les réseaux commutés rend de plus en plus difficile l'analyse du segment réseau à analyser, car le contenu des paquets est crypté [15]. L'emplacement des NIDS est défini par la politique de sécurité menée. Nous allons voir comment les NIDS sont déployés dans un réseau donné.

✓ **NIDS déployés à l'extérieur du périmètre d'un pare-feu (Mode alerte précoce)**

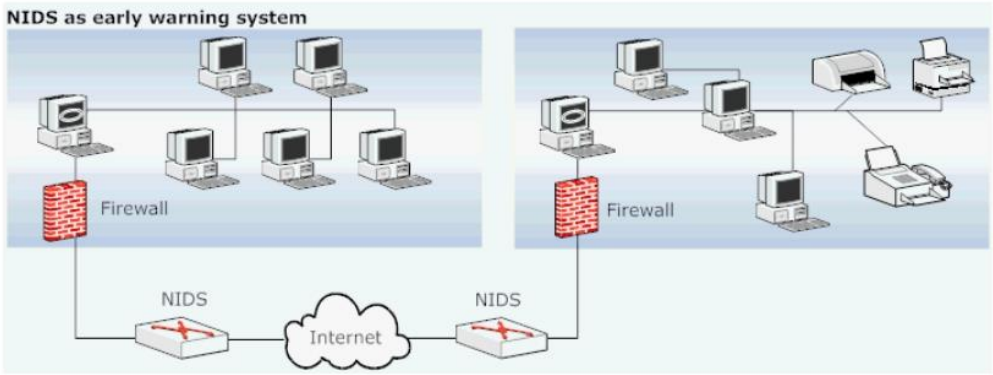


Figure 1.9 : NIDS déployé à l'extérieur du périmètre d'un pare-feu [16]

Dans un tel mode de fonctionnement, tout trafic entrant dans l'hôte ou dans le réseau local est analysé par le NIDS. L'avantage de ce mode de déploiement est que le NIDS reste à un emplacement unique en ayant une liaison à haut débit et à la capacité de desservir un grand nombre d'hôtes ainsi la gestion et la mise à jour des configurations sont beaucoup plus faciles. Par contre, les attaques lancées par les hôtes dans le périmètre du pare-feu ne seront pas détectées par le NIDS, ce qui représente un inconvénient. Avec ce mode de configuration, il est possible que le NIDS déclenche une alarme alors que le pare-feu bloque le trafic, rendant ainsi l'alarme fausse [16].

✓ **NIDS en mode déploiement complet**

Avec ce mode de configuration, le NIDS (Système de Détection d'Intrusion Réseau) est déployé à des emplacements stratégiques, à proximité des nœuds de commutation à l'intérieur du réseau

local et près des routeurs d'accès. Cela permet au NIDS de surveiller activement le trafic circulant à l'intérieur du réseau, renforçant ainsi la sécurité globale

Un avantage important de ce mode de déploiement est la réduction significative des taux de fausses alarmes, car le NIDS ne surveillera plus le trafic déjà bloqué par le pare-feu. Cela contribue à améliorer l'efficacité globale du système de détection des intrusions. Cependant, il convient de noter que ce mode de déploiement nécessite la mise en place de plusieurs instances de NIDS, ce qui peut rendre leur maintenance plus complexe et exigeante. Il est donc essentiel de prendre en compte ces considérations pratiques lors de la mise en œuvre de ce mode de configuration pour garantir une gestion efficace et continue de la sécurité du réseau [16].

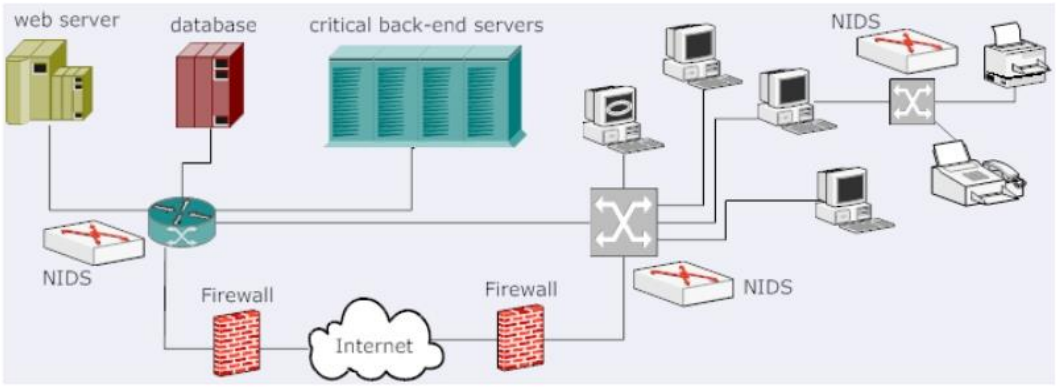


Figure 1. 10 : NIDS en mode déploiement complet [16]

- **IDS basé sur l'hôte (HIDS : Host Intrusion Detection System)**

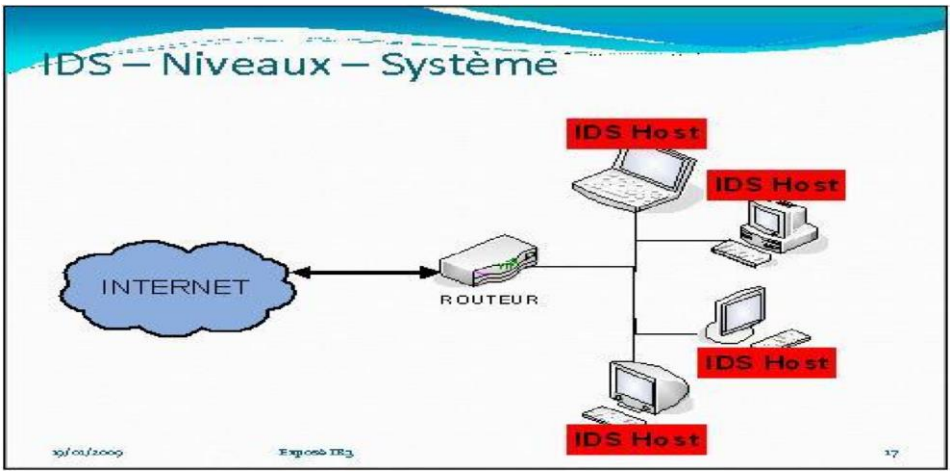


Figure 1. 11 : IDS basé sur l'hôte (HIDS)

Ce sont des IDS mis en place directement sur les hôtes à surveiller. Les HIDS analysent les paramètres spécifiques au système tels que les différents appels système et aussi les événements réseaux pour empêcher les intrus de compromettre la politique de sécurité du système. Par conséquent ces analyses sont strictement limitées à l'hôte sur laquelle l'HIDS est installé et n'ont

aucune vue sur le réseau [14,16]. L'HIDS se différencie de l'antivirus au sens que l'antivirus a pour rôle de surveiller toutes les activités à l'intérieur du système, n'est pas concerné par les attaques par débordement de mémoire tampon sur la mémoire système ni par le comportement malveillant du processus du système d'exploitation, alors que HIDS vérifie et collecte les données système, y compris le système de fichiers, les événements réseau et les appels système pour vérifier s'il y en a [17]. Le système HIDS s'appuie fortement sur la piste d'audit et les journaux systèmes pour détecter les activités anormales à l'intérieur du système. Dans [15], le système HIDS (Système de Détection d'Intrusion sur Hôte) offre diverses fonctionnalités :

- Il permet de détecter les compromissions de fichiers en contrôlant leur intégrité.
- Il effectue une analyse approfondie de la base de registre (sur Windows) ou des LKMs (sur Linux).
- Il est capable d'analyser et de corréler les journaux provenant de pare-feu de différentes natures.
- De plus, il peut également analyser les flux cryptés, une tâche impossible pour un NIDS (Système de Détection d'Intrusion Réseau). Les IDS basés sur l'hôte ne peuvent pas détecter les attaques provenant des couches réseau d'où leur inconvénient.

❖ **L'IDS Hybride**

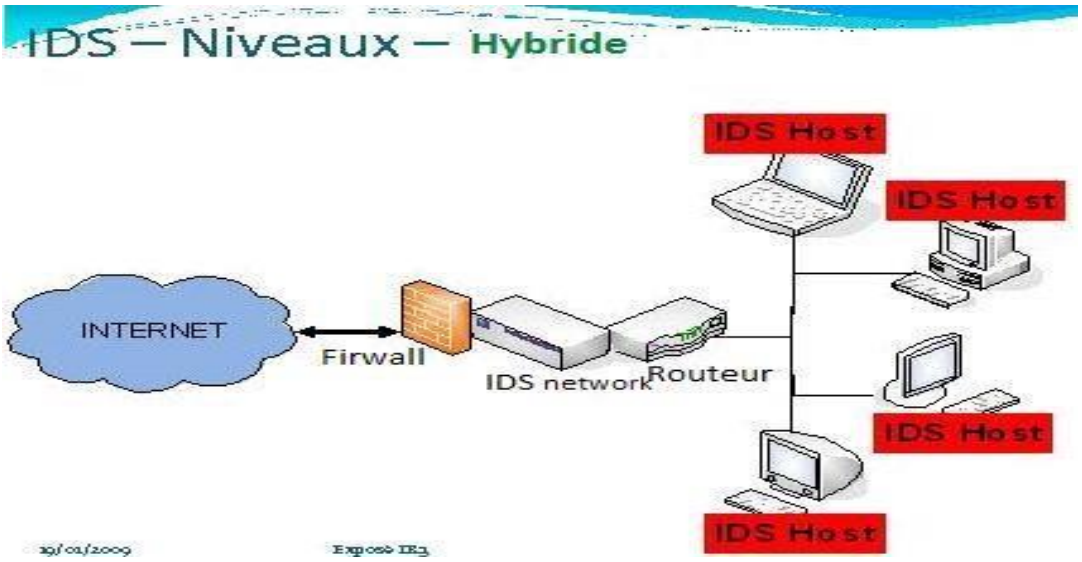


Figure 1. 12 : IDS hybride

Les IDS hybrides rassemblent les caractéristiques des IDS basés sur le réseau et des IDS basés sur l'hôte. Le système de détection hybride est flexible et augmente le niveau de sécurité. Il

combine les emplacements des capteurs IDS et signale que les attaques visent des segments particuliers ou l'ensemble du réseau [18].

- **IDS centralisé**

En fonction de la localité et de la manière dont les composants des systèmes de détection d'intrusion sont distribués, l'IDS peut être classé comme : I**DS distribué ou décentralisé** et **IDS centralisé.**

Le système de détection d'intrusion centralisé est une approche dans laquelle l'analyse des données se fait à un nombre défini d'emplacements, indépendamment du nombre d'hôtes surveillés [18]. Ce système est composé de deux éléments : le système expert et les IDS (Systèmes de Détection d'Intrusion, qu'ils soient HIDS ou NIDS). Lorsqu'un IDS détecte une anomalie sur son réseau local ou sur son hôte, il envoie cette anomalie au système expert, qui évalue s'il s'agit d'une attaque globale contre différents systèmes ou non. Cette approche présente des avantages avec un taux de détection élevé, cependant, elle comporte deux inconvénients majeurs. Tout d'abord, le système centralisé possède un point de défaillance unique, ce qui signifie que si le système expert tombe en panne, l'ensemble du système de détection devient inutilisable. Deuxièmement, en cas d'attaque majeure, certaines alertes peuvent être ignorées en raison de leur grand nombre, ou elles peuvent être traitées ultérieurement, potentiellement après l'attaque elle-même [19].

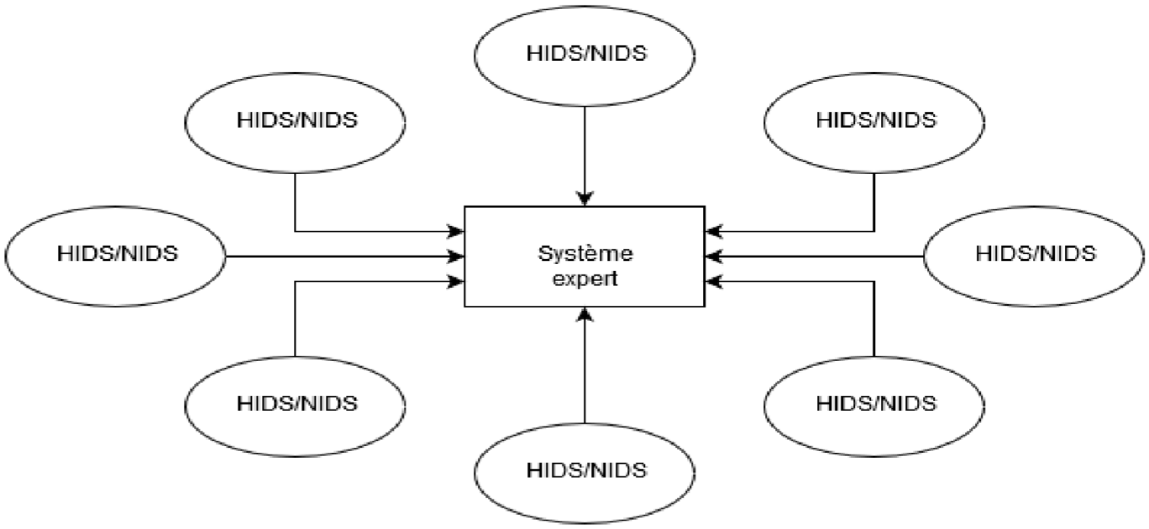


Figure 1. 13 : IDS centralisé

❖ **IDS distribué**

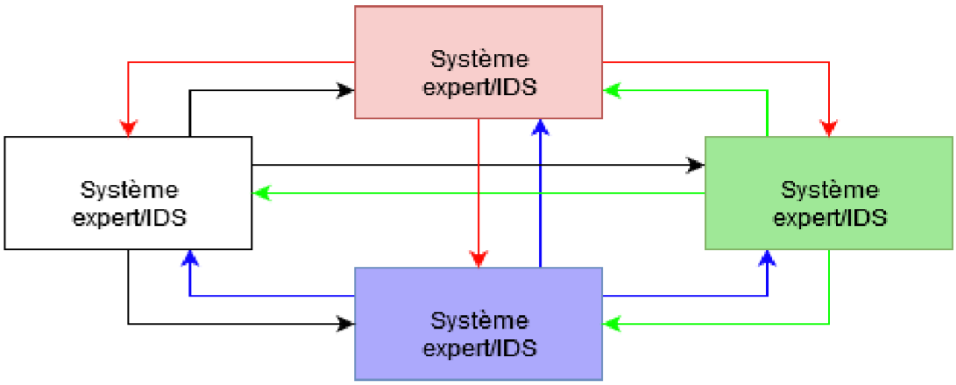


Figure 1. 14 : IDS distribué

Un système de détection d'intrusion distribué est un système dans lequel l'analyse des données est effectuée dans un nombre d'emplacements proportionnel au nombre d'hôtes surveillés. Cette approche permet d'éviter d'avoir des points de défaillance, qui pourrait mettre à mal tout le système de détection [18]. Pour cela, chaque nœud est collecteur d'information ainsi qu'analyseur. Ces nœuds détectent localement les attaques et sont capables de corréler les informations des nœuds voisins pour détecter les attaques globales [19].

1.7.7 Système de Prévention D'Intrusion IPS

Un système de prévention des intrusions également appelé système de détection et de prévention des intrusions IDPS est un outil qui est similaire à un système de détection d'intrusion, permettant de prendre des mesures afin de diminuer les impacts d'une attaque. Une Intrusion Prevention System (IPS) est conçue pour identifier les attaques potentielles et exécuter de façon autonome une contre-mesure pour les empêcher, sans affecter le système d'exploitation normal. L'objectif principal d'un IPS est d'anticiper et de stopper les attaques. Contrairement au système de détection d'intrusion qui découvre les attaques après qu'elles ont eu lieu, un système de prévention tente de mettre en garde de telles attaques. Et pendant que l'IDS tente de donner l'alerte, l'IPS bloque le trafic jugé dangereux. Les IPS sont toujours en ligne sur le réseau pour superviser le trafic et intervenir activement en limitant ou en supprimant le trafic jugé hostile en interrompant les sessions suspectes ou en prenant d'autres mesures de réaction à une attaque ou une intrusion [14]. Les systèmes de prévention des intrusions analysent les contextes de connexion, automatisent l'analyse des logs et suspendent les connexions suspectes. Alors qu'un IDS n'a aucun moyen efficace de bloquer une intrusion (si ce n'est via l'utilisation de réponses actives), un IPS pourra, de par son positionnement en coupure, bloquer une intrusion en temps réel. Le positionnement en coupure, tel un firewall ou

un proxy, est le seul mode permettant d'analyser à la volée les données entrantes ou sortantes et de détruire dynamiquement les paquets intrusifs avant qu'ils n'atteignent leur destination. Un IPS peut également corriger les erreurs de contrôle de redondance cyclique (CRC), défragmenter les flux de paquets, atténuer les problèmes de séquencement TCP et nettoyer les options de transport et de couche réseau indésirables [18]. L'IPS permet les fonctionnalités comme : la supervision du comportement de l'application, créer des alertes pour l'application, la compréhension des réseaux IP, la maitrise des sondes réseaux et l'analyse des logs etc…

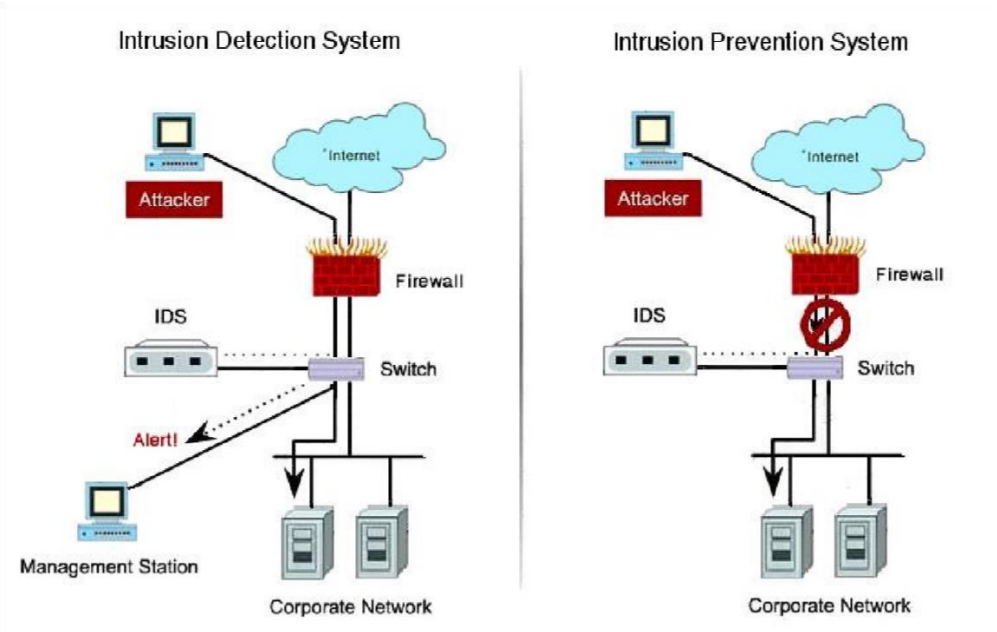


Figure 1. 15 : Système de détection et de prévention d'intrusion

Les systèmes de prévention des intrusions peuvent être classés en 4 catégories différentes :

- **Le système de prévention d'intrusion basée sur l'hôte HIPS**

Contrairement au système HIDS, ce système est destiné à bloquer les trafics anormaux avant qu'une mise à jour de la détection spécifique ne soit effectuée en surveillant le comportement du code. La plupart des solutions HIPS (Host Intrusion Prevention System) surveillent le code au moment de son exécution et interviennent si le code est jugé suspect ou malveillant [14].

- **Le système de prévention d'intrusion basé sur le réseau NIPS**

Même principe que le NIDS, à la grande différence que le NIPS (Network Intrusion Prevention System) peut bloquer des flux suspects. L'aspect sensible de ce genre de dispositif de prévention est qu'en cas de faux positif, c'est le trafic du système qui est directement affecté. Les erreurs doivent donc d'être les moins nombreux possibles car elles ont un impact direct sur la disponibilité ou la sécurité des systèmes [17]. Il existe

de nombreuses approches existantes pour IDPS (Intrusion Detection and Prevention System), dans lesquelles SNORT qui est un système de prévention des intrusions open source capable d'analyser le trafic en temps réel et d'enregistrer les paquets. SNORT a trois utilisations principales : un simple renifleur de paquets comme tcpdump, un enregistreur de paquets (utile pour le débogage du trafic réseau, etc.) ou un système complet de prévention des intrusions sur le réseau [18].

- **Analyse du comportement du réseau NBA**

C'est un moyen de renforcer la sécurité d'un réseau en surveillant le trafic et en notant les écarts inhabituels par rapport au fonctionnement normal. Le NBA (Network Behavior Analysis) surveille ce qui se passe à l'intérieur du réseau, regroupant les données de plusieurs points pour ainsi prendre en charge l'analyse hors ligne. Après avoir établi une référence pour le trafic normal, le programme NBA surveille passivement l'activité du réseau et signale des modèles inconnus, nouveaux ou inhabituels qui pourraient indiquer la présence d'une menace [18]. L'analyse du comportement réseau est très utile pour repérer les nouveaux logiciels malveillants.

- **Système de prévention des intrusions sans fil WIPS : Wireless Intrusion Prevention System**

Ce système est un dispositif réseau qui surveille le spectre radio pour détecter la présence de points d'accès non autorisés et peut automatiquement prendre des contre-mesures [18]. L'objectif principal de ce système consiste à empêcher l'accès réseau non autorisée aux réseaux locaux par des dispositifs sans fil.

Conclusion

L'internet des Objets et ses technologies constituent un avantage non négligeable pour les usagers. L'IoT ouvre des perspectives nouvelles pour les sciences de l'information et de la communication. Cependant, il faut rester prudent à la sensibilité des données collectées qui transitent et qui sont très vulnérables aux attaques. Assurer la protection des réseaux IoT est un défi majeur car ceux-ci sont caractérisés par une forte hétérogénéité des microcontrôleurs, des topologies réseaux mais aussi des ressources contraintes en termes de mémoires, capacités de traitement et débits d'émission-réception. De plus, comme la technologie IoT devient de plus en plus importante dans notre vie, l'élaboration des normes technologiques pour l'IoT devient cruciale.

Chapitre 2

Machine Learning pour la classification

2.1 Introduction

La machine Learning ML (apprentissage automatique) est une discipline de l'intelligence artificielle (IA) qui se concentre sur le développement de techniques permettant aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances sans être explicitement programmés. La classification des données est l'une des tâches les plus courantes en machine Learning, et elle consiste à attribuer une étiquette ou une classe à une nouvelle observation en se basant sur les caractéristiques des données. Le Deep Learning (DL) qui repose sur le modèle des réseaux de neurones est une branche du ML qui peut atteindre des performances exceptionnelles car il peut surmonter certaines des limites du ML et est plus efficace pour traiter le Big Data. Il a la capacité de faciliter une analyse approfondie des données du réseau et une détection rapide des anomalies.

2.2 L'Intelligence Artificielle

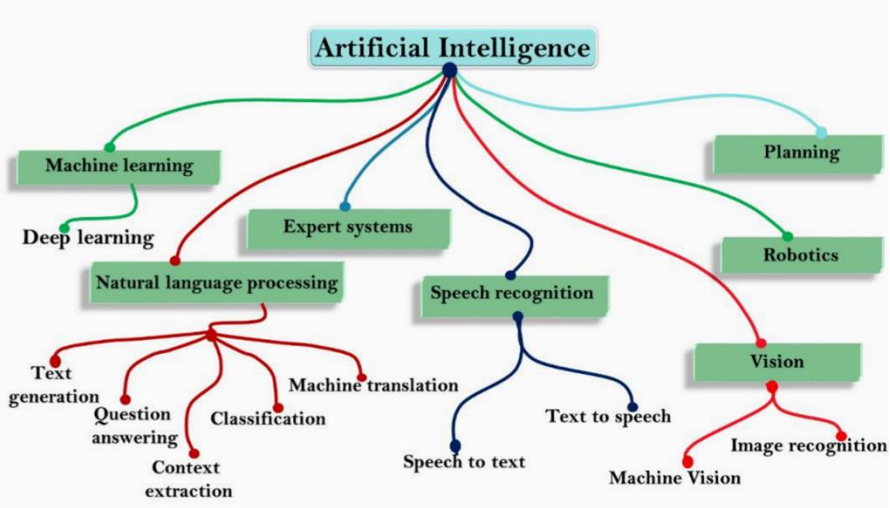


Figure 2. 1: Les Branches de L'intelligence artificielle

Le terme « intelligence artificielle » a été créé en 1965 par le célèbre chercheur en informatique John McCarthy qui a simplement définit L'IA comme "la science et l'ingénierie de fabrication de machines intelligentes". L'intelligence artificielle peut être définie comme étant un processus d'apprentissage des machines qui identifient des modèles et des systèmes provenant d'une quantité de données et d'une capacité de traitement élevées. Son but principal est de développer l'intelligence humaine dans les machines grâce à des algorithmes d'apprentissage

qui tentent de se rapprocher le plus possible du comportement humain. L'intelligence artificielle avec ses branches représentent l'avenir des processus décisionnels complexes.

2.3 Le Machine Learning (ML)

La machine Learning désigne l'étude scientifique des algorithmes et des modèles statistiques que les systèmes informatiques utilisent pour effectuer une tâche spécifique sans être explicitement programmés [20]. Autrement dit, les algorithmes de la machine Learning apprennent de manière autonome à effectuer une tâche à partir de données de formation spécifiques à une tâche. Son utilité est de permettre aux machines de gérer plus efficacement les données. Le développement d'un modèle de machine Learning repose sur quatre phases fondamentales [21] :

- **Phase d'entraînement**

Dans cette étape, un ensemble de données d'entraînement est sélectionné et soigneusement préparé, nettoyé et organisé. Ces données serviront à former le modèle de machine Learning.

- **Phase d'exécution**

À cette étape, un algorithme est choisi et exécuté sur l'ensemble des données d'entraînement. Le choix de l'algorithme dépend du volume de données d'entraînement et des objectifs du modèle.

- **Phase d'entraînement de l'algorithme**

L'algorithme est exécuté avec différentes variables, et les résultats produits sont comparés avec ceux attendus. Ce processus itératif continue jusqu'à ce que l'algorithme parvienne à produire les résultats corrects. L'algorithme ainsi entraîné devient le modèle de machine Learning.

- **Phase d'utilisation et d'amélioration du modèle**

Une fois le modèle créé, il est utilisé pour effectuer des tâches sur de nouvelles données. Si nécessaire, le modèle peut être amélioré en fonction des performances et des besoins spécifiques, assurant ainsi une utilisation optimale du modèle de machine learning dans des scénarios réels.

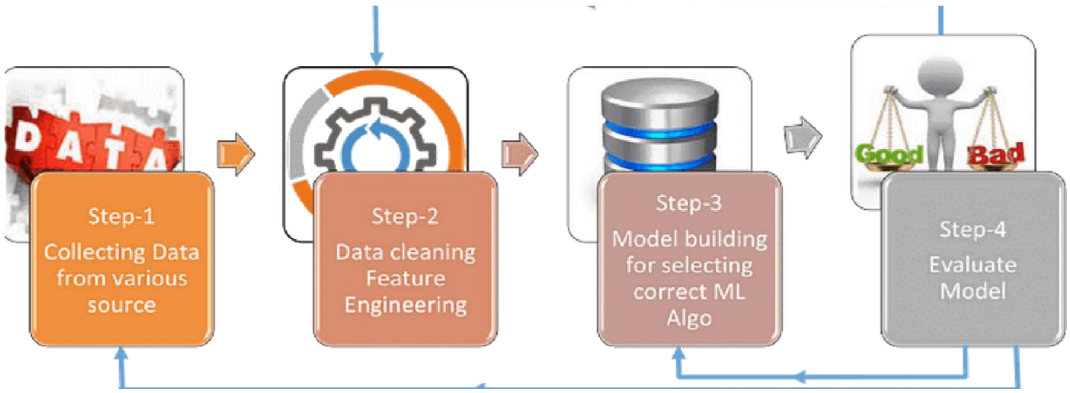


Figure 2. 2 : développement d'un modèle de machine Learning

2.3.1 Quelques Modèles d'apprentissage du ML

Il existe 3 principaux types d'apprentissage automatique : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement

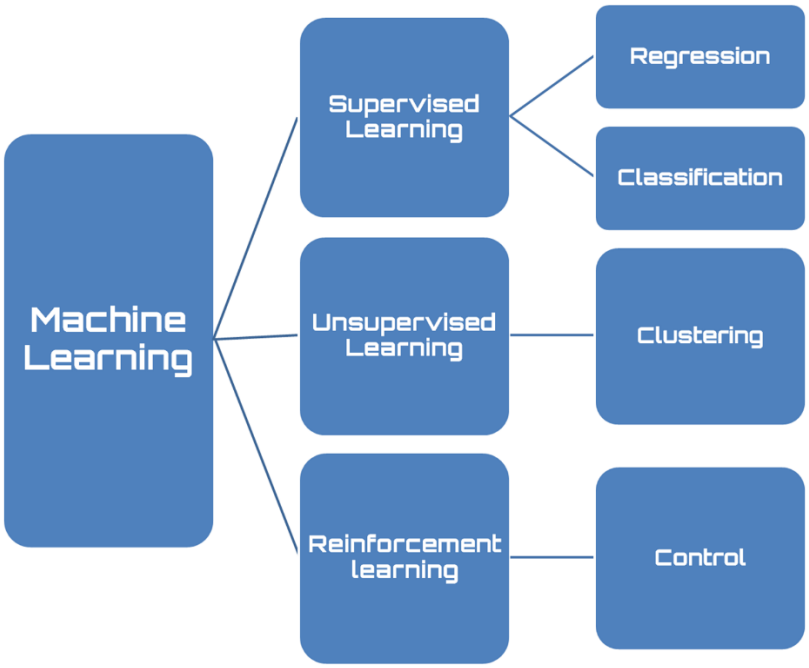


Figure 2. 3: Algorithmes machine Learning

2.3.2 L'apprentissage supervisé

L'apprentissage supervisé repose sur un ensemble de données d'entrainement qui a des instances étiquetées pour la classe normale ainsi que pour la classe d'anomalies. La classification est la tâche la plus courante dans l'apprentissage supervisé. L'ensemble de données d'entrée est divisé en ensemble de données d'apprentissage (ou d'entrainement) et de test [20]. Les algorithmes d'apprentissage supervisé, ont été largement utilisés pour la sécurité IoT.

❖ **Arbre de décision**

Elle est l'un des premiers algorithmes de machine Learning utilisé pour représenter visuellement les choix et les résultats pour des tâches de classification et de régression.

❖ **Machine à vecteur de support (SVM)**

Ce modèle analyse les données utilisées pour la classification et l'analyse de régression. Les SVM sont une généralisation des classificateurs linéaires. Mais, ils peuvent effectuer efficacement une classification non linéaire en utilisant ce qu'on appelle l'astuce du noyau [20]. L'idée est de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, où un classificateur linéaire peut être utilisé et obtenir de bonnes performances.

❖ **Naïve Bayes**

La classification naïve bayésienne est basée sur le théorème de Bayes. En termes simples, un classificateur Naïve Bayes suppose que la présence d'une caractéristique particulière dans une classe n'est pas liée à la présence de toute autre caractéristique [20].

2.3.3 Apprentissage non supervisée

Contrairement à l'apprentissage supervisé, ici les données ne sont pas étiquetées. Les algorithmes sont laissés à eux-mêmes pour découvrir et présenter la structure intéressante des données. Les algorithmes d'apprentissage non supervisé apprennent peu de fonctionnalités à partir des données. Il est principalement utilisé pour le clustering et la réduction des fonctionnalités.

❖ **K-Means**

Désigne l'un des algorithmes d'apprentissage non supervisé les plus simples car il résout le problème de clustering. Le clustering est une technique d'apprentissage du ML qui permet de regrouper des chaînes de données par distance ou par similarité. Étant donnés des points et un entier k, l'algorithme vise à diviser les points en k groupes, appelés clusters, homogènes et compacts.

2.3.4 Apprentissage par renforcement

Cette approche d'apprentissage, connue sous le nom de "Reinforcement Learning" (apprentissage par renforcement), vise à apprendre par le biais d'expériences itératives en utilisant une approche essai-erreur. En d'autres termes, elle consiste à effectuer différentes tentatives jusqu'à ce que le succès soit atteint ou jusqu'à ce que le testeur décide d'arrêter. L'objectif est de permettre à une entité virtuelle telle qu'un robot ou un programme d'interagir avec un environnement interactif et de sélectionner des actions qui maximisent les récompenses

quantitatives. L'environnement fournit un retour d'information sous la forme de récompenses positives ou négatives en fonction des actions effectuées par l'entité virtuelle. Cette dernière utilise ces récompenses pour améliorer sa stratégie d'action et ainsi optimiser ses performances dans cet environnement interactif.

2.4 Les Réseaux de neurones

Les réseaux de neurones artificiels (ANN), également connus sous le nom de réseaux de neurones font partie intégrante du domaine de l'apprentissage automatique, où ils permettent aux machines d'apprendre à traiter les données en imitant le fonctionnement du cerveau humain. Leur conception s'inspire directement du comportement de signalisation des neurones présents dans les réseaux de neurones biologiques**.** Les réseaux de neurones artificiels se structurent en couches interconnectées par des axones. Ils comprennent une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie. Il est à noter qu'il n'existe pas de connexion entre les neurones d'une même couche, mais uniquement avec les neurones de la couche suivante. Généralement, chaque neurone d'une couche est relié à tous les neurones situés en aval, limitant ainsi les connexions exclusivement à cette configuration.

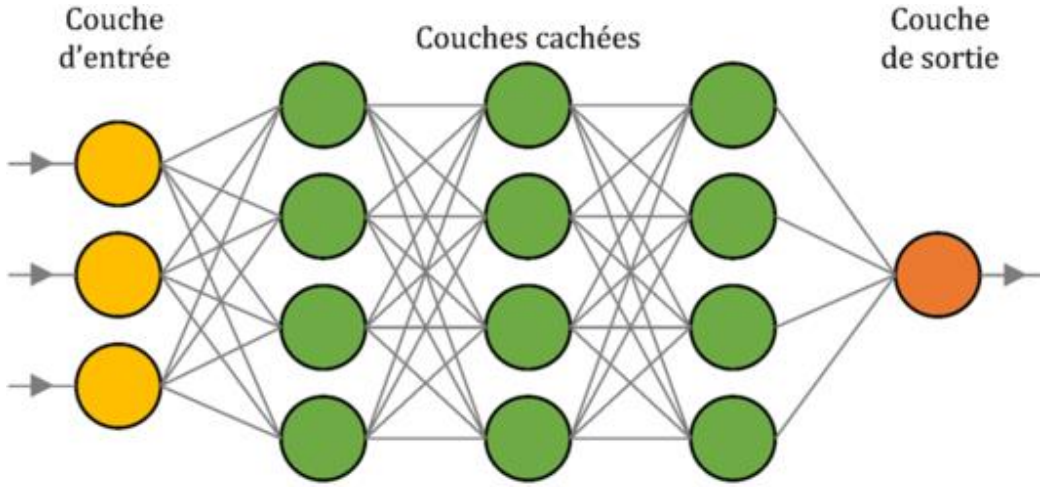


Figure 2. 4 : Structure d'un réseau de neurone artificiel ou perceptron multicouche

Afin d'optimiser les performances du réseau, des ajustements sont effectués sur les valeurs numériques, en particulier sur les poids et les biais. Les poids sont utilisés pour moduler les entrées de chaque neurone, tandis que les biais sont ajoutés après l'application des poids aux valeurs d'entrée. Ces mises à jour permettent d'améliorer le fonctionnement global du réseau de neurones [22]. La fonction d'activation est une formule mathématique appliquée au signal de sortie d'un neurone artificiel. Son rôle est de décider si le neurone doit être activé ou non. Les fonctions d'activation introduisent la non-linéarité dans les réseaux de neurones, nécessaire

pour résoudre des problèmes complexes [22]. Les neurones des couches adjacentes interagissent en utilisant des fonctions d'activation, qui transforment la somme des entrées d'un neurone en une sortie. Cette sortie prend la valeur 1 lorsque la somme pondérée dépasse un seuil donné, sinon elle est égale à 0. Ainsi, les fonctions d'activation jouent un rôle crucial dans la communication entre les neurones, permettant de déterminer leur activité en fonction des signaux reçus.

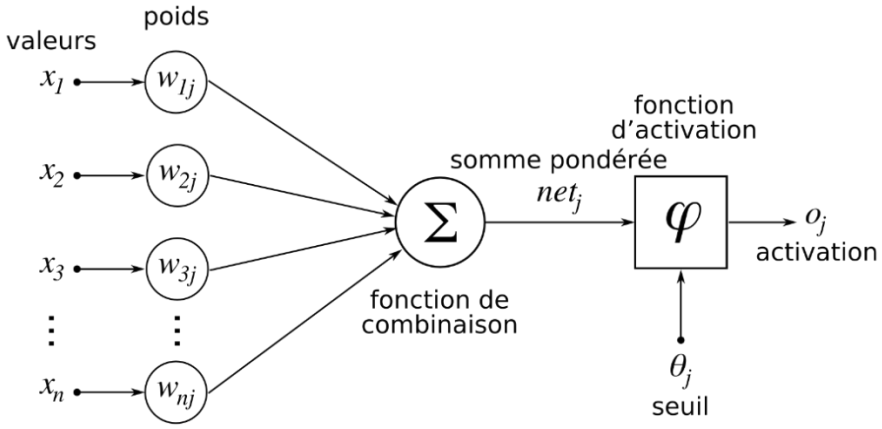


Figure 2. 5 : notion de fonction d'activation

Il existe aussi plusieurs catégories de réseaux de neurones chacun ayant ses propres caractéristiques et applications spécifiques. Les principales catégories de réseaux de neurones : les réseaux de neurones convolutifs (CNN), les réseaux de neurones récurrents (RNN), les réseaux de neurones auto encodeurs (AE), les réseaux de neurones adverses génératifs (GAN) que nous allons voir sur la partie de l'apprentissage profond (Deep Learning).

2.5 Le Deep Learning

Le Deep Learning (apprentissage profond) est un concept d'apprentissage automatique basé sur des réseaux de neurones artificiels. IL met l'accent sur l'exploitation de réseaux de neurones comprenant plusieurs couches, d'où son appellation "profond". Contrairement aux approches traditionnelles d'apprentissage machine, qui nécessitent souvent une extraction manuelle de caractéristiques à partir des données, le Deep Learning vise à automatiser le processus d'apprentissage de caractéristiques hiérarchiques directement à partir des données brutes. Les réseaux de neurones profonds sont constitués d'une superposition de couches cachées, organisées en structures d'architecture profondément emboîtées. De plus, contrairement aux réseaux de neurones artificiels simples, ils intègrent des neurones sophistiqués. En d'autres termes, ils peuvent incorporer des opérations avancées, telles que des convolutions, ou utiliser plusieurs activations dans un seul neurone plutôt que de se limiter à une fonction d'activation

simple [23]. L'apprentissage profond a gagné en popularité grâce à ses performances remarquables dans une multitude de domaines tels que la vision par ordinateur, le traitement du langage naturel, les jeux, et bien d'autres. La remarquable capacité des réseaux de neurones profonds à saisir des modèles complexes au sein des données a entraîné des avancées significatives dans de nombreux aspects de l'intelligence artificielle. Les modèles du Deep Learning offrent diverses approches pour la reconnaissance de classes [24] :

- ✓ **Le Deep Learning pour l'apprentissage supervisé**

Il s'agit des modèles discriminatifs tels que le DNN, le RNN et le CNN qui utilisent un apprentissage supervisé en se basant sur des étiquettes pour reconnaître les classes.

- ✓ **Le Deep Learning pour l'apprentissage non supervisé**

Les modèles génératifs tels que l'Auto-encodeur, le DBN permettent un apprentissage non supervisé sans dépendre d'étiquettes pour identifier les classes.

- ✓ **L'Approche hybride**

Comme les modèles GAN qui combinent les aspects discriminatifs et génératifs pour améliorer la performance et la qualité des résultats.

2.5.1 Quelques Méthodes de Deep Learning

Ils existent diverses approches en Deep Learning, parmi lesquelles nous examinerons quelques-unes.

2.5.2 Les réseaux de neurones convolutifs CNN

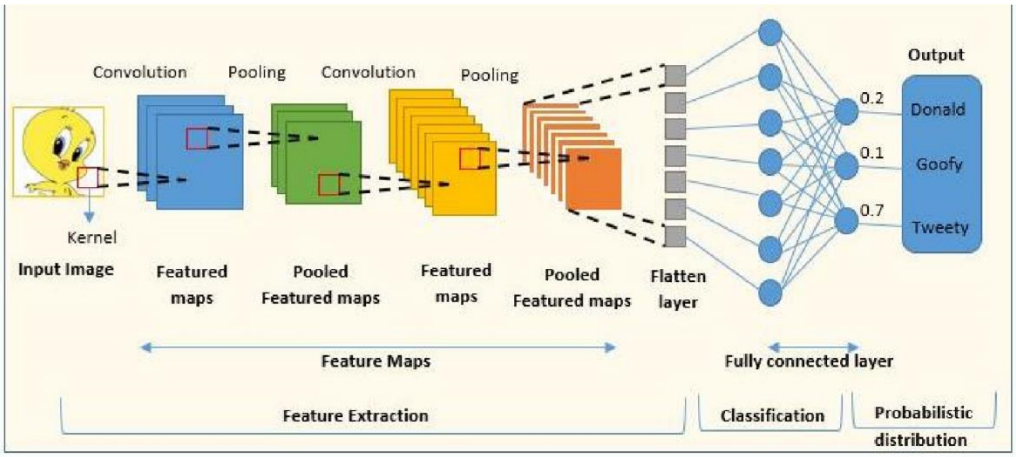


Figure 2. 6 : Architecture typique d'un CNN

Les réseaux neuronaux convolutifs (CNN) également connu sous le nom de ConvNets se distinguent comme l'algorithme prééminent et le plus répandu dans le domaine des réseaux neuronaux, conçus pour traiter efficacement des données structurées en grille. Les réseaux de neurones convolutifs sont structurés en multiples couches et trouvent leur utilité principale dans le domaine du traitement d'images et de la détection d'objets. Les modèles sont très efficaces

pour l'analyse sémantique, la modélisation de phrase, la classification ou la traduction C'est en 1988 que Yann LeCun pionna le premier CNN, baptisé LeNet, qui s'avéra révolutionnaire pour la reconnaissance de caractères tels que les codes postaux et les chiffres [25]. L'idée fondamentale derrière est d'imiter le processus visuel humain en reconnaissant les motifs et les caractéristiques visuelles dans les données. À la différence d'un modèle MLP (Perceptron Multicouches) traditionnel qui se compose uniquement d'une phase de classification, la structure intègre en amont une étape de convolution. De ce fait, elle se divise en deux segments clairement différenciés [26] :

❖ **Étape de Convolution**

Son but ultime réside dans l'extraction de caractéristiques intrinsèques propres à chaque image, tout en les comprimant pour diminuer leur taille d'origine. En résumé, l'image d'entrée traverse une série de filtres engendrant ainsi la création de nouvelles représentations visuelles désignées sous le nom de cartes de convolutions ou cartes de caractéristiques. Enfin, ces cartes de convolutions résultantes sont fusionnées en un vecteur caractéristique appelé code CNN.

❖ **Étape de Classification**

Le code CNN résultant de l'étape de convolution est acheminé en entrée dans une deuxième partie composé de couches entièrement connectées nommées perceptron multicouche (MLP pour Multi Layers Perceptron). La fonction fondamentale de cette étape consiste à fusionner les caractéristiques du code CNN afin de classer l'image.

Un CNN est généralement constitué de plusieurs couches, notamment :

- **Couches de convolution**

Au cœur d'un CNN se trouve la couche de convolution, qui agit comme l'élément fondamental de traitement où la majeure partie des calculs s'opère. Son fonctionnement repose sur plusieurs éléments essentiels, à savoir les données d'entrée, un filtre et une carte de caractéristique ou carte d'activation appelée feature map. Si on suppose que la donnée d'entrée est une image, le filtre se déplacera sur les champs réceptifs de l'image pour vérifier si la caractéristique est présente. Ce processus est appelé convolution. Le filtre ou noyau est une matrice bidimensionnelle, qui représente une partie de l'image. Lorsque le filtre est appliqué sur une zone de l'image, un produit scalaire est engendré entre les pixels d'entrée et le filtre. Ce produit scalaire est ensuite intégré dans une matrice de sortie. Par la suite, le filtre se déplace de manière successive pour itérer ce processus jusqu'à ce que l'ensemble de l'image ait été scanné. La sortie finale de cette série de produits scalaires entre l'entrée et le filtre est couramment appelée la carte caractéristique. À la suite de chaque opération de convolution, un CNN met en œuvre une

transformation ReLU (Rectified Linear Unit) sur la carte d'activation instaurant ainsi une composante non linéaire au sein du modèle [27].

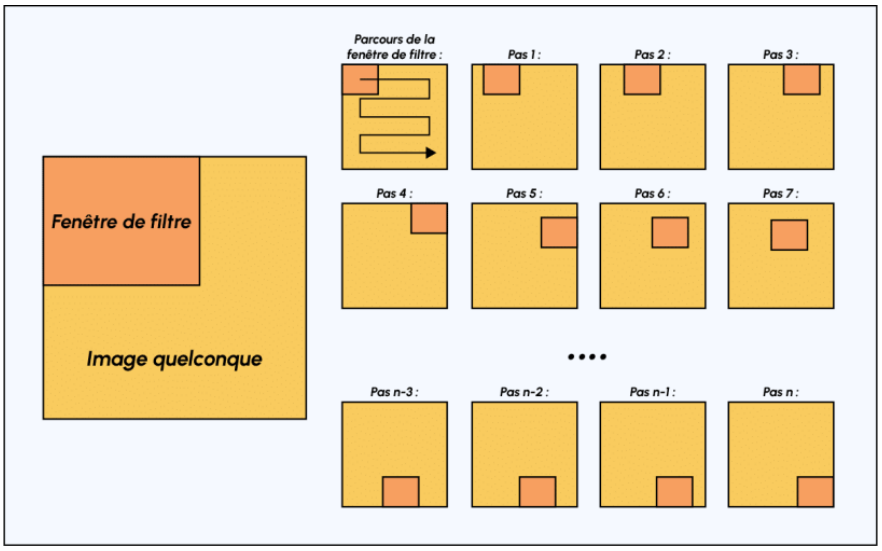


Figure 2. 7 : Schéma du parcours de la fenêtre de filtre sur l'image

- **Couche de pooling**

Les couches de pooling, parfois désignées sous le nom de couches de sous-échantillonnage se situent généralement entre deux étapes de convolution. Elles prennent en entrée les cartes de caractéristiques engendrées à la sortie de l'étape de convolution et servent à diminuer la dimensionnalité, c'est-à-dire à restreindre le nombre de paramètres en entrée tout en préservant leurs caractéristiques les plus essentielles. Tout comme avec la couche de convolution, l'action de pooling implique le déplacement d'un filtre sur l'intégralité de l'entrée. Il est important de noter que ce filtre est dépourvu de poids. Au lieu de cela, le filtre met en œuvre une opération d'agrégation sur les valeurs du champ récepteur remplissant ainsi la matrice de sortie. La couche de pooling contribue à réduire la complexité, à accroître l'efficacité et à atténuer le risque de sur ajustement.

Deux formes majeures de pooling sont distinguées :

- ✓ **Le Max pooling**

À mesure que le filtre parcourt l'entrée, il choisit le pixel ayant la valeur maximale pour la transférer à la matrice de sortie. Elle est plus efficace que la moyenne, car elle maximise le poids des activations fortes

- ✓ **Average pooling**

Au fur et à mesure que le filtre se déplace à travers l'entrée il calcule la valeur moyenne dans le champ réceptif, puis transmet cette moyenne à la matrice de sortie.

Prenons l'exemple d'une matrice initiale de dimensions 4×4 représentant notre entrée, ainsi qu'un filtre de fenêtre 2×2 que nous appliquerons à cette entrée. À chaque déplacement du filtre sur les différentes zones, le max-pooling retiendra la valeur maximale résultant en la formation d'une nouvelle matrice de sortie. Dans cette matrice, chaque élément sera le maximum des valeurs rencontrées dans chaque zone balayée par le filtre [26].

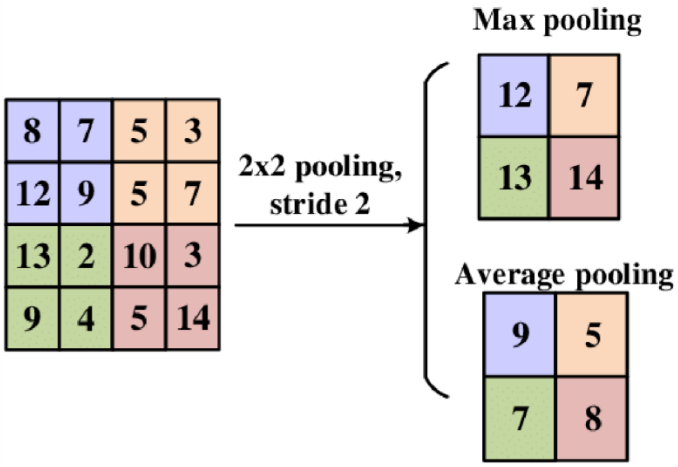


Figure 2. 8 : Exemple de moyen et Max Pooling avec un filtre 2 x 2 en pas de 2

- **La couche d'activation ReLU** (Rectified Linear Units)

Cette couche transforme les valeurs négatives en zéro pour toutes les entrées reçues. L'utilité de ces couches d'activation réside dans leur capacité à conférer au modèle une nature non linéaire, ce qui augmente sa complexité [26].

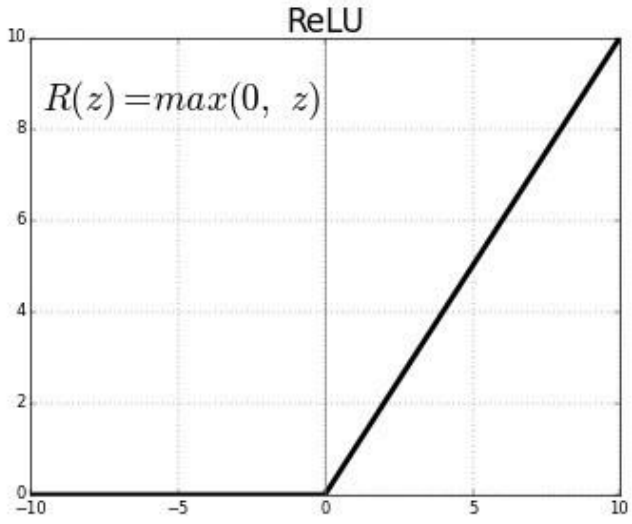


Figure 2. 9 : Fonction d'activation ReLU

- **Couche Fully Connected**

Ces couches sont positionnées à la fin de la structure du CNN et sont entièrement connectées à l'ensemble des neurones de sortie d'où le terme fully-connected. Cette couche réalise la

classification en se basant sur les caractéristiques déduites des couches antérieures ainsi que de leurs divers filtres. Tandis que les couches de convolution et de pooling tendent à adopter les fonctions ReLu, les couches entièrement connectées recourent généralement à une fonction d'activation comme Softmax pour réaliser une classification adéquate des entrées, engendrant une probabilité comprise entre 0 et 1 [27].

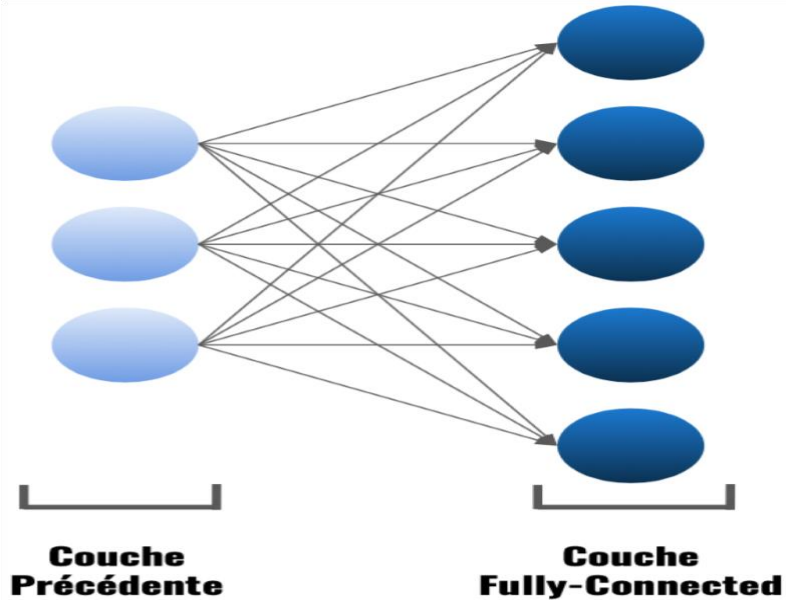


Figure 2. 10 : Architecture d'une couche Fully-Connected

2.5.3 Les Réseaux de neurones récurrents RNN (Recurrent Neural Networks)

Les RNN sont spécifiquement conçus en vue de traiter des formats de données séquentielles, tels que les séries chronologiques, les suites d'événements et le langage naturel. Tout comme les réseaux de neurones convolutifs à propagation avant, les réseaux de neurones récurrents se basent sur des données d'entraînement pour leur apprentissage. Ce qui les démarque, c'est leur aptitude à intégrer une forme de "mémoire" : ils intègrent des informations des entrées précédentes afin d'influer sur les entrées et sorties actuelles. En d'autres termes, La caractéristique fondamentale des réseaux de neurones récurrents réside dans l'intégration de boucles récurrentes qui permettent à l'information de se propager. À chaque étape de la séquence, que ce soit pour chaque mot dans une phrase par exemple, l'analyse tient compte des données antérieures pour une compréhension contextuelle. Cependant, Les architectures RNN simples présentent des limitations, car ils sont sujets à des problèmes de gradients qui disparaissent ou explosent. En effet, les RNN classiques sont en mesure de gérer uniquement des séquences de longueurs limitées.

Puisque les réseaux de neurones récurrents sont caractérisés par un grand nombre de connexions, il est courant de recourir à des représentations simplifiées où une seule flèche symbolise une matrice de poids complète, notée W. La figure ci-dessous illustre un réseau classique comprenant une couche récurrente qui est ensuite suivie par une couche dense classique [28].

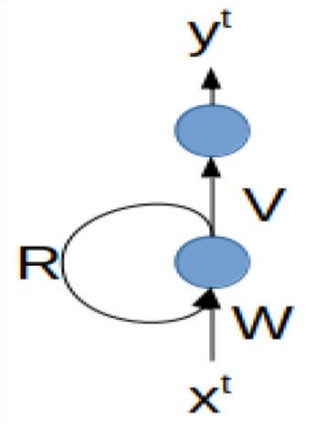


Figure 2. 11 : Illustration simplifiée d'un RNN

Il s'avère compliquer de faire apparaitre la dimension temporelle sur la figure particulièrement pour les connexions récurrentes qui intègrent l'information du temps précédent. Pour remédier à cette situation, une méthode courante consiste à adopter une vision "dépliée dans le temps" du réseau, qui permet de mettre en évidence cette dimension de manière explicite [28].

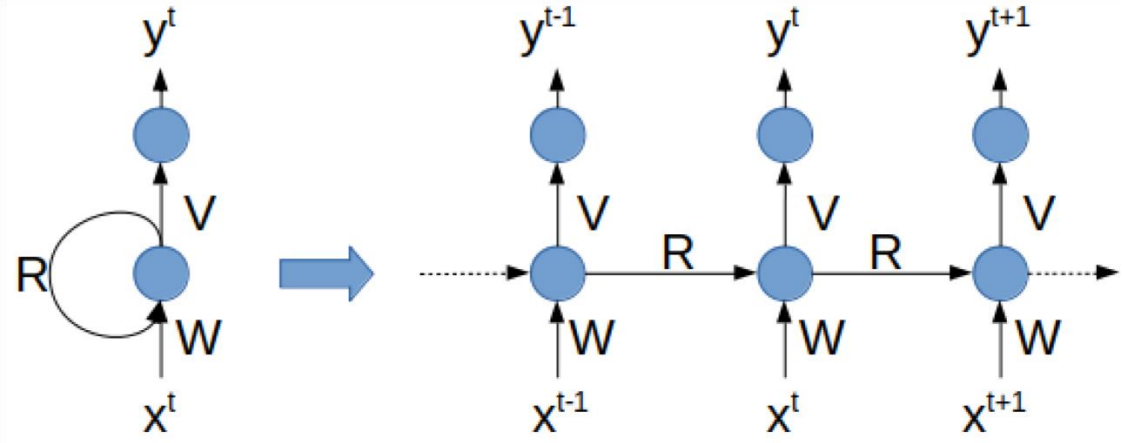


Figure 2. 12 : version récurrente et version dépliée [28]

À chaque instant temporel t, chaque neurone récurrent reçoit simultanément le vecteur d'entrée x(t) ainsi que le vecteur de sortie provenant de l'instant temporel précédent, y(t-1). Chaque neurone récurrent dispose de deux catégories de poids :

- Les poids W qui relient l'entrée à la sortie, similairement à un réseau de neurones classique.

- Les poids R qui agissent comme des connexions récurrentes, reliant la sortie à l'entrée de la couche.

Cependant, les RNN classiques ont des limitations, notamment en ce qui concerne leur capacité à capturer des dépendances à long terme dans les séquences. C'est pourquoi plusieurs variantes de RNN ont été développées pour surmonter ces problèmes. Parmi elles, on peut citer :

❖ **LSTM (Long Short-Term Memory)**

C'est une structure RNN largement adoptée, ayant été développée par Sepp Hochreiter et Juergen Schmidhuber comme une solution au problème de disparition du gradient. Leur travail se concentre sur la résolution des dépendances à long terme. C'est-à-dire que si l'état précédent qui influence la prévision en cours n'est pas dans un passé récent, le modèle RNN peut ne pas être en mesure de prévoir avec précision l'état en cours. Au cas contraire cela rend difficile, voire impossible, pour le RNN de connecter les informations. Afin de résoudre ce problème, les LSTM intègrent des "cellules" au sein des couches cachées du réseau neuronal. Ces cellules sont dotées de trois portes distinctes : une porte d'entrée, une porte de sortie et une porte d'oubli. Ces portes contrôlent le flux d'informations nécessaires pour anticiper la sortie au sein du réseau et gère une mémoire adaptative (désignée par C) qui évolue en accord avec la séquence temporelle des données [27].

La porte oublie représente une opération qui confère la faculté d'oublier une information (ou de considérablement réduire son importance) qui s'avérait pertinente au moment t-1, mais qui ne l'est plus à l'instant t. la porte d'entrée est une opération qui permet à la cellule d'acquérir la capacité de stocker une nouvelle information au temps t, alors que cette même information était inexistante ou peu significative (présentant un poids très faible) au moment t-1. La porte de sortie dirige l'information qui sera relayée au moment t+1 en fonction de la mémoire C et de fonction d'activation. En se basant sur ce vecteur mémoire C, la cellule LSTM mémorise les valeurs sur des plages de temps arbitraires, tandis que les trois portes contrôlent le flux des informations entrantes et sortantes de la cellule. En raison de sa structure plus complexe, la phase d'apprentissage des LSTM demande davantage de temps par rapport à celle d'un réseau neuronal classique ou d'un RNN. Cependant, les performances offertes par un LSTM sont nettement supérieures [29].

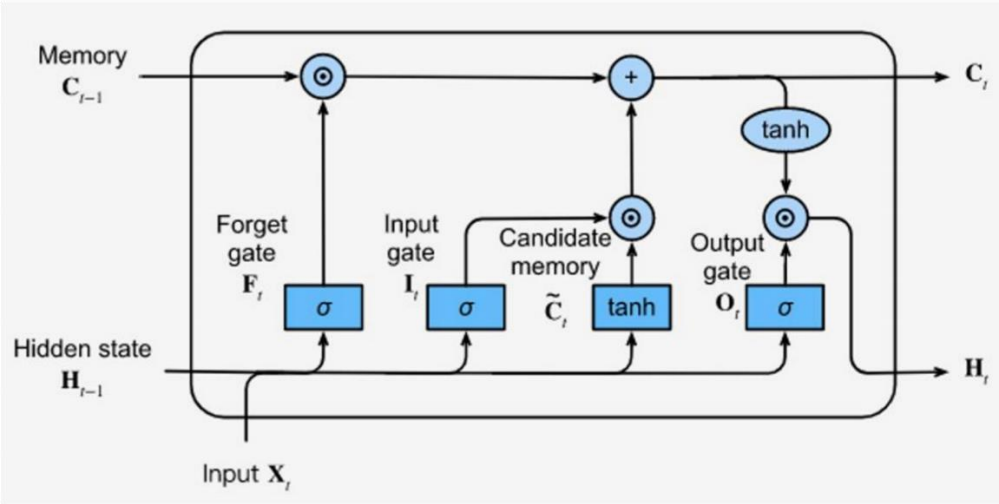


Figure 2. 13 : Architecture LSTM

C_{t-1} : mémoire à l'instant t-1

C_t : mémoire à l'instant t

H_{t-1} : entrée de la couche cachée à l'instant t-1

H_t : entrée de la couche cachée à l'instant t

X_t : entrée de la cellule

La fonction sigmoïde **σ** et **tanh** sont des fonctions d'activation

❖ **Unités récurrentes fermées (GRU)**

Cette version de RNN présente des similitudes avec les LSTM, car elle aborde également la problématique de la mémoire à court terme propre aux modèles RNN. Plutôt que d'employer les informations régulées par les « états de cellule », elle tire parti des états cachés. De plus, elle se distingue par la présence de deux portes plutôt que trois, à savoir une porte de réinitialisation et une porte de mise à jour. Ces portes, tout comme celles des LSTM, contrôlent la quantité et le type d'informations à conserver [27].

Le rôle principal attribué à la porte de mise à jour réside dans sa capacité à évaluer avec précision la quantité optimale d'informations antérieures revêtant de l'importance pour les événements à venir. Cette fonction détient une signification prépondérante en raison de sa capacité à permettre au modèle de reproduire chaque élément du passé pour éliminer le problème du gradient d'évanouissement.

La pertinence fondamentale de la porte de réinitialisation se révèle dans sa capacité à définir quelle part d'informations mérite d'être omise. Une analogie appropriée pourrait être établie entre la porte de réinitialisation et la porte d'oubli dans le contexte des LSTM, car elle accomplit une opération similaire de triage des données non pertinentes, permettant au modèle de les négliger et de poursuivre son processus. Cependant, il convient de noter que les GRU (Gated

Recurrent Unit) ne sont pas aussi complexes que les LSTM et leur calcul ne prend pas trop de temps

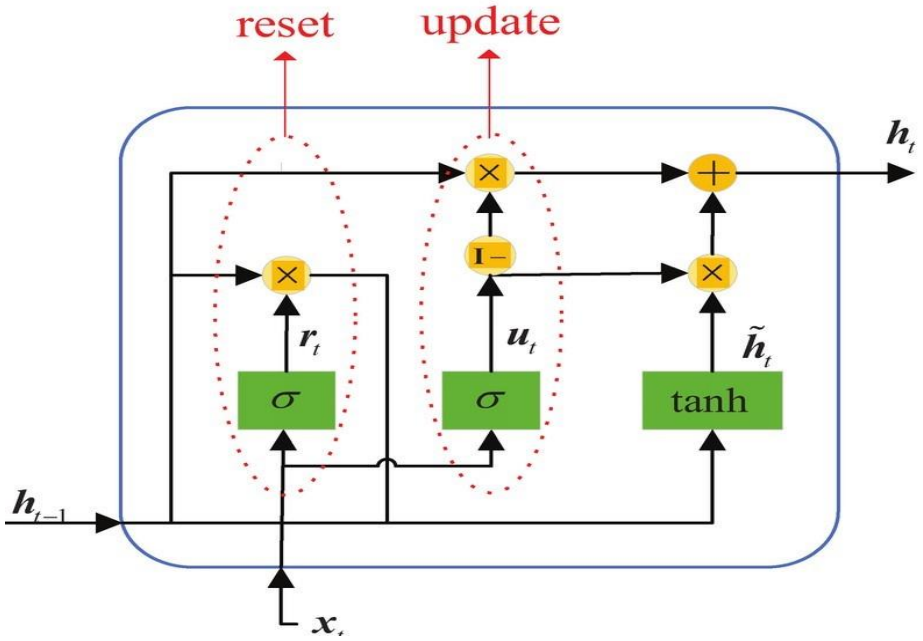


Figure 2. 14 : Architecture GRU

2.5.4 Auto encodeurs AE

Une auto encodeur représente un algorithme d'extraction de caractéristiques qui se fonde sur un réseau de neurones non supervisé. Le but d'un auto-encodeur est d'avoir une sortie la plus proche possible de l'entrée. Cette démarche d'apprentissage s'inscrit dans une logique auto supervisée, étant donné que l'erreur à réduire correspond au coût de reconstruction entre la sortie générée et l'entrée initiale. Compte tenu que les données agissent comme leurs propres références, il n'est pas nécessaire de les étiqueter, ce qui confère à ce modèle son caractère non supervisé. Néanmoins, du fait de leur capacité à quantifier les erreurs de reconstruction, censés être notablement accrus pour les échantillons anormaux que pour les instances régulières,
Ils peuvent aussi être employés pour repérer des anomalies, comme les opérations frauduleuses sur les marchés financiers [30].
Typiquement, Un auto encodeur peut être subdivisée en trois parties :

- **Encodeur**

Vise à capturer les caractéristiques essentielles des données d'entrées tout en réduisant leur dimension. En d'autres termes, l'encodeur comprime les informations de l'entrée en une forme plus concise.

- **Goulot d'étranglement (ou bottleneck)**

Dans cette portion du réseau réside la version condensée de l'entrée, qui sera ensuite insérée dans le décodeur**.** Cette couche cachée a normalement une dimension plus petite que celle de l'entrée**.**

- **Décodeur**

Cherche à reconstruire les données originales à partir de la représentation compressée. Il joue un rôle crucial dans le processus de reconstruction de l'auto-encodeur. Il a la même dimension que la couche d'entrée (encodeur). L'entrée passe par l'encodeur où elle est compressée et stockée dans le goulot d'étranglement, puis le décodeur décompresse ces données afin de retrouver l'entrée originale du réseau [31].

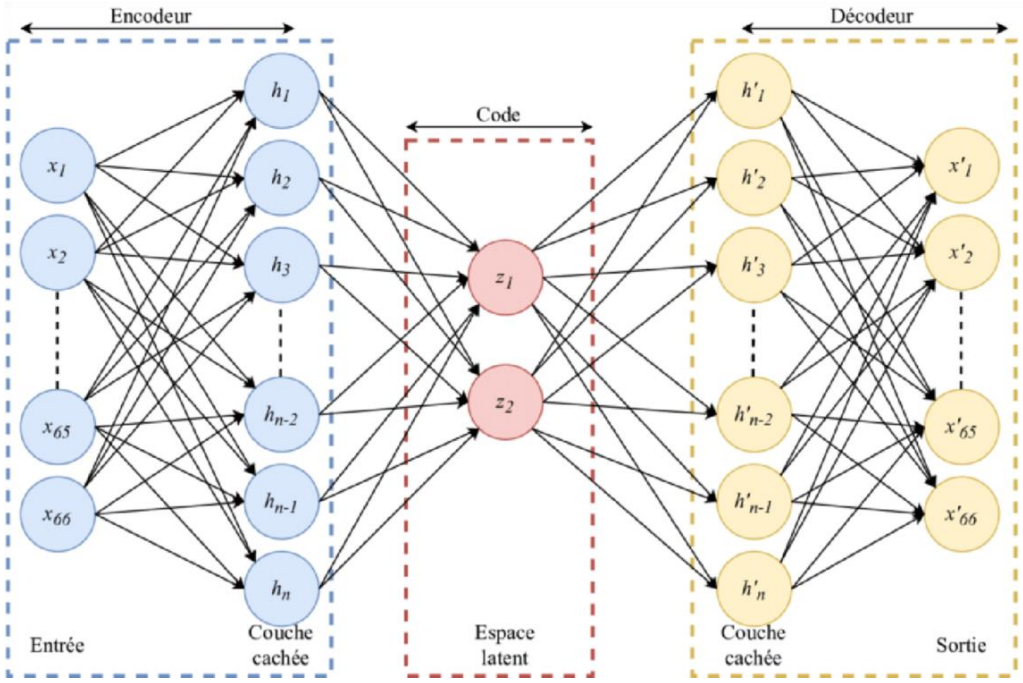


Figure 2. 15 : Structure d'une auto encodeur [31]

2.5.5 Exemples d'Auto encodeurs

Les Auto-encodeurs de dé-bruitage DAE (Denoising AutoEncoder)

Ce genre de méthode introduit un certain degré de perturbation dans l'image d'entrée, et s'entraîne ensuite à la supprimer. Cela évite de simplement copier l'entrée vers la sortie sans obtenir de connaissances plus approfondies sur les données**.** Pendant l'apprentissage, les auto-encodeurs travaillent avec une entrée partiellement corrompue pour récupérer l'entrée non corrompue. Le modèle développe un vecteur de champs afin de relier les données d'entrée à une dimension réduite qui capture les caractéristiques intrinsèques des données et qui annule le bruit introduit. Ainsi, l'encodeur extrait les éléments essentiels et acquiert une représentation plus

robuste des données. Ce type d'approche est utilisé pour améliorer la qualité de données bruitées ou pour restaurer des images détériorées.

- **Les Auto-encodeurs profonds**

Les auto-encodeurs profonds sont des variantes d'auto-encodeurs qui comportent plusieurs couches cachées entre la couche d'entrée et la couche de sortie. Contrairement aux auto-encodeurs classiques qui ont une seule couche cachée, les auto-encodeurs profonds en ont plusieurs, ce qui permet de capturer des niveaux de représentation de plus en plus abstraits et complexes à mesure que l'on traverse les couches. Chaque couche cache une représentation intermédiaire de l'entrée et ces représentations sont utilisées pour reconstruire progressivement l'entrée jusqu'à la sortie. Cette profondeur accrue peut réduire les coûts de calcul et la quantité de données d'entraînement requises, ainsi que produire des degrés de précision plus élevés [32].

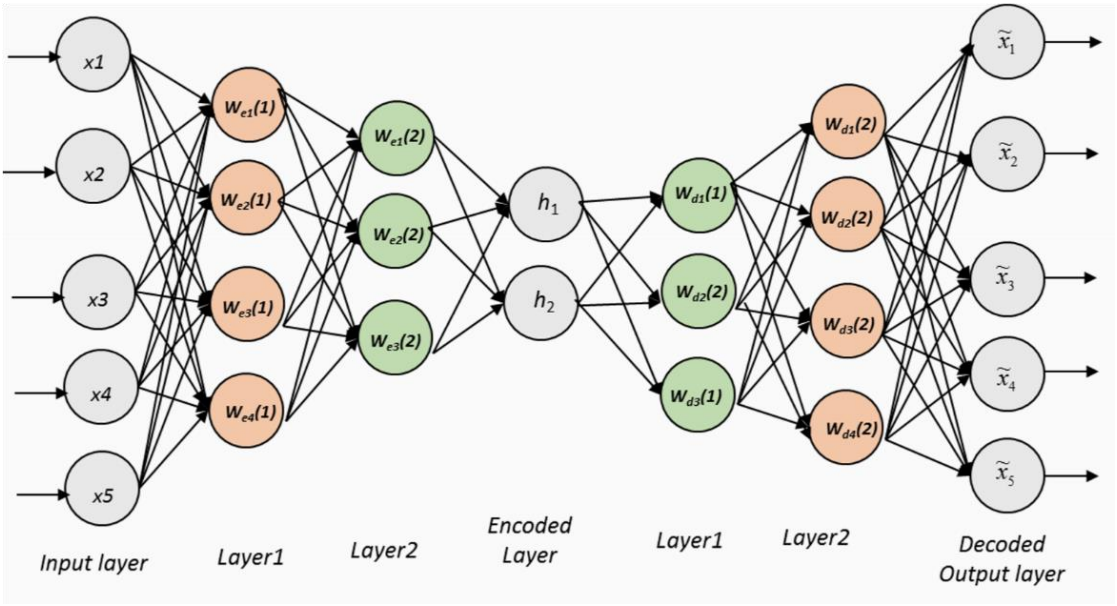


Figure 2. 16 : Structure d'une auto encodeur profond

Conclusion

Le Machine Learning (ML) est une discipline essentielle de l'intelligence artificielle qui permet de trier et de catégoriser des données en fonction de modèles appris à partir d'exemples. Grâce à des algorithmes variés tels que les forêts aléatoires, les machines à vecteurs de support (SVM), les réseaux de neurones artificiels et les méthodes bayésiennes, il est possible de traiter une multitude de types de données et de résoudre divers problèmes de classification.

Chapitre 3

Classification des données de trafic réseau dans l'Internet des Objets (IoT)

3.1 Introduction

La classification des données de trafic réseau dans l'Internet des objets (IoT) est une étape importante pour gérer et analyser efficacement les flux de données générés par les dispositifs IoT connectés. Elle permet de mieux comprendre comment les dispositifs interagissent et quelles sont les données qui transitent à travers le réseau, ce qui peut être utile pour la surveillance, la maintenance et la prise de décisions informées. Une classification binaire consiste à attribuer une étiquette à chaque observation, indiquant si elle est normale ou anormale et une classification multi-classe comme son nom l'indique permet de classer les données en plusieurs classes exemple la classification d'images (par exemple, la classification d'objets dans des images). Dans le cas d'une classification binaire, un modèle d'apprentissage automatique est formé sur un ensemble de données contenant à la fois des exemples normaux et des exemples anormaux. Le modèle apprend ainsi à distinguer les caractéristiques des données normales de celles des données anormales. Une fois le modèle formé, il peut être utilisé pour classer de nouvelles observations en tant que normales ou anormales en fonction de leurs caractéristiques. La classification des données à l'aide de techniques de Deep Learning est devenue très populaire en raison de la capacité des réseaux de neurones profonds à extraire automatiquement des caractéristiques pertinentes à partir de données brutes, sans nécessiter une extraction manuelle de caractéristiques. Le processus de classification peut être réalisé de différentes manières, en fonction de la nature des données et des objectifs de classification. Dans ce mémoire nous allons faire une classification multi-classe qui implique la répartition des données parmi plusieurs catégories distinctes afin de mieux comprendre la nature des données et de gérer efficacement le trafic réseau.

3.2 Présentation des Outils utilisés

3.2.1 Python

Nous avons opté pour Python comme langage de programmation pour notre travail en raison de son caractère interprété, de haut niveau et de sa polyvalence, largement utilisé dans divers domaines allant du développement web à l'analyse de données et à l'intelligence artificielle. Créé par Guido van Rossum et publié pour la première fois en 1991, Python est apprécié pour sa syntaxe claire et lisible, ce qui facilite l'apprentissage et la collaboration entre les

développeurs. Python est un langage interprété, ce qui signifie que le code est exécuté ligne par ligne, facilitant le débogage et l'expérimentation en temps réel.

3.2.2 Anaconda

Anaconda est une distribution libre et open source de Python et R, conçue pour simplifier la gestion des packages et l'environnement de développement. Anaconda inclut plusieurs outils populaires pour la science des données, tels que Jupiter Notebook, JupyterLab, Spyder et RStudio, qui offrent des interfaces interactives pour le développement et l'expérimentation. Avec son vaste écosystème de bibliothèques et d'outils pour la manipulation des données, l'analyse, la visualisation et le machine learning, Anaconda est une solution complète pour les projets de science des données.

3.2.3 Jupyter Notebook

Jupyter Notebook est un environnement interactif open source qui permet de créer et de partager des documents contenant du code exécutable, des équations, des visualisations et du texte narratif. Initialement développé dans le cadre du projet IPython, Jupyter s'est étendu pour supporter plusieurs langages de programmation, dont Python, R, et Julia, d'où son nom (Ju(lia), Python), and R). Jupyter Notebook permet des tests efficaces en offrant la possibilité d'exécuter les cellules individuellement, ce qui évite de devoir exécuter l'ensemble du programme à chaque modification. Grâce à des bibliothèques comme Matplotlib, Seaborn, et Plotly, Jupyter Notebook permet de créer des visualisations interactives et statiques directement dans le document.

3.2.4 Environnement de développement

- **Tensorflow**

Tensorflow est une bibliothèque open source développée par Google Brain Team pour le machine learning et l'apprentissage profond. Lancée en 2015, Tensorflow permet de construire, entraîner et déployer des modèles de machine learning de manière efficace et évolutive. L'une des principales caractéristiques de Tensorflow est que les opérations courantes sur les réseaux de neurones sont effectuées à l'aide de tableaux de données multidimensionnels appelés tenseurs (Tensors), d'où son nom. Ces tenseurs peuvent représenter divers types de données, y compris des scalaires, des vecteurs et des matrices de n dimensions.

- **Keras**

Keras est une bibliothèque open source de haut niveau pour le développement et l'entraînement de modèles de réseaux de neurones. Créée par François Chollet en 2015, Keras est conçue pour être simple, modulaire et extensible. Elle s'appuie sur des bibliothèques de calcul bas niveau comme TensorFlow, Theano pour effectuer les opérations de calcul. Keras offre une interface

utilisateur intuitive qui permet de créer des modèles de réseaux de neurones complexes avec peu de lignes de code, facilitant ainsi le prototypage rapide. Grâce à sa simplicité, sa modularité et sa compatibilité avec plusieurs backends, Keras est devenu un outil incontournable pour les chercheurs et les développeurs travaillant dans le domaine de l'apprentissage profond.

3.2.5 Les librairies

Dans Anaconda, nous avons accès à une gamme étendue de bibliothèques qui simplifient le traitement des données et la mise en œuvre de modèles d'intelligence artificielle. Dans notre cas, nous utilisons les bibliothèques suivantes **:**

- **NumPy et Pandas**

NumPy est une bibliothèque essentielle pour le calcul numérique en Python. Elle offre une prise en charge pour les tableaux et matrices multidimensionnels ainsi que des fonctions mathématiques de haut niveau pour effectuer des opérations sur ces tableaux. Pandas est une bibliothèque Python construite sur NumPy qui offre des structures de données et des outils de manipulation de données puissants et faciles à utiliser. Pandas est largement utilisé pour nettoyer, transformer, analyser et visualiser des données dans divers domaines.

- **la bibliothèque Scikit-learn**

Scikit-learn est une bibliothèque open source de machine learning pour Python. Elle offre des outils simples et efficaces pour l'apprentissage automatique supervisé et non supervisé, le prétraitement des données, la sélection de modèle et l'évaluation des performances. Scikit-learn s'intègre bien avec d'autres bibliothèques Python comme NumPy, Pandas et Matplotlib, ce qui facilite la manipulation et la visualisation des données.

- **la bibliothèque matplolib**

Matplotlib est une bibliothèque de visualisation de données en 2D. Elle offre une grande flexibilité pour créer une grande variété de graphiques, de tracés et de visualisations à partir de données.

3.3 Présentation de notre Dataset

Canadian Institute for Cybersecurity (**CIC) -IoT-Dataset2022**, conçu par Sajjad Dadkhah, Hassan Mahdikhani, Priscilla Kyei Danso, Alireza Zohourian, Kevin Anh Truong et Ali A. Ghorbani, représente une ressource essentielle pour l'étude des objets connectés. Ce dataset vise à faciliter le profilage, l'analyse comportementale et les tests de vulnérabilité des dispositifs IoT, couvrant des protocoles tels qu'IEEE 802.11, Zigbee et Z-Wave.

La mise en réseau du laboratoire, équipé de divers dispositifs IoT, est réalisée à l'aide de deux cartes d'interface réseau, connectées à une passerelle et à un commutateur réseau non

administré. Des expériences variées ont été menées, capturant le trafic réseau dans des conditions telles que l'activation individuelle des appareils, l'inactivité nocturne, les interactions humaines et des scénarios de simulation d'une maison intelligente. La collecte de données s'est effectuée via Wireshark et dumpcap, couvrant des expériences actives, des périodes d'inactivité, des interactions spécifiques et des scénarios de simulation. Des attaques par inondation et par force brute RTSP ont également été réalisées, capturant le trafic réseau d'attaque sur certains appareils. Au total, 20 expériences ont été menées, impliquant trois catégories d'appareils : Audio, Caméra et Domotique. Ces expériences ont généré un ensemble de données comprenant 48 fonctionnalités. Le modèle formé a été déployé dans un autre laboratoire pour des tests visant à prédire la classe de chaque appareil en se basant sur des caractéristiques spécifiques [33]

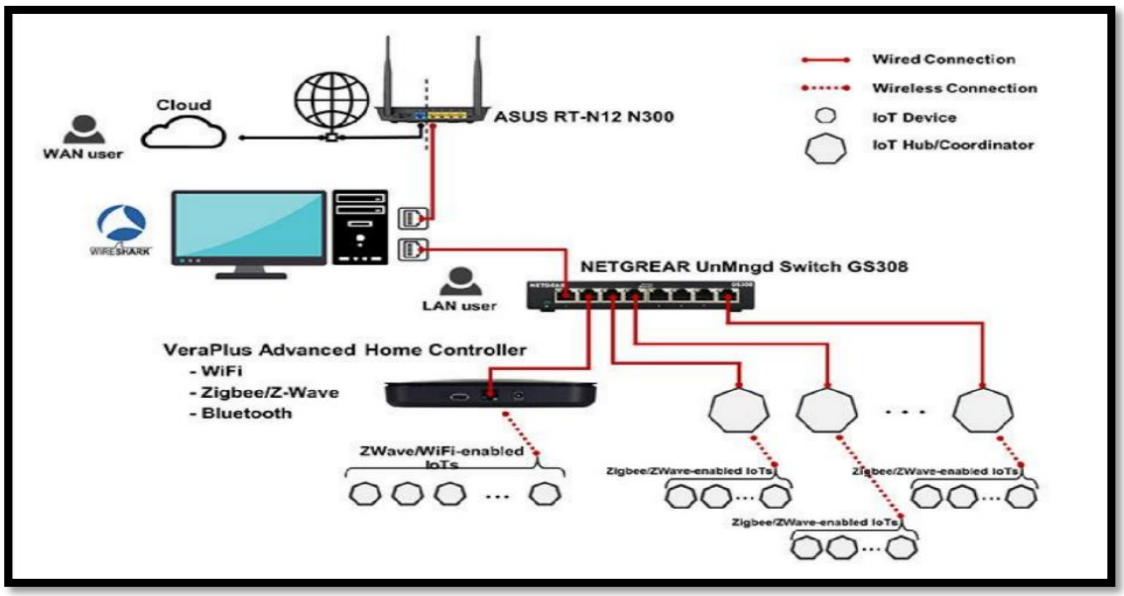


Figure 3.1 : Laboratoire intelligente [33]

Afin de créer un système de classification intelligent pour notre laboratoire, nous allons exploiter les données de trafic réseau de celle-ci pour élaborer des méthodes basées sur l'intelligence artificielle. Pour parvenir à cet objectif, nous avons recouru à diverses ressources destinées au traitement et à l'analyse des données réseau, en plus de mettre en œuvre plusieurs modèles d'intelligence artificielle. Les prochains paragraphes détailleront ces outils ainsi que la mise en place de ces modèles.

3.4 Analyse Exploratoire des Données (EDA)

Elle vise à comprendre et à explorer un jeu de données afin d'obtenir des informations préliminaires sur ses caractéristiques principales. L'objectif est de comprendre au maximum les

données dont on dispose pour définir une stratégie de modélisation. L'EDA (Exploratory Data Analysis) peut être divisé en 2 parties :

- ✓ **L'analyse de la forme** : Ici, on identifie la variable de sortie (la variable Target), le nombre de lignes et de colonnes de nos données, les types de variables et les valeurs manquantes.
- ✓ **L'analyse du fond** : visualisation de la variable Target, compréhension des différentes variables, visualisation des relations features-target, identification des outliers.

3.4.1. Analyse de la forme

Nous avons commencé par repérer la « variable target », spécifiée ici sous le nom de « global_category ». Cette variable constitue notre sortie principale, ce qui signifie que son comportement doit être modélisé et anticipé en fonction des autres variables contenues dans notre jeu de données. Les données sont réparties sur un total de 229506 lignes et 52 colonnes, chacune de ces colonnes correspondant aux caractéristiques du trafic réseau. Chaque caractéristique est associée à une variable en fonction du numéro de flux. L'examen de la composition de nos données révèle la présence de deux catégories de variables : 8 variables qualitatives et 44 variables quantitatives dans notre jeu de données. Ces variables sont réparties comme suit : 16 variables de type float, 28 variables de type int, 4 variables de type bool et 4 variables de type catégory. Pour terminer avec l'analyse de forme nous avons cherché à savoir s'il existe des données manquantes avec la commande **df.isna ()** et les résultats de la figure (3.2) visualisés à l'aide de la commande **sns.heatmap(df.isna())** nous ont montré qu'il n'existe aucune donnée manquante.

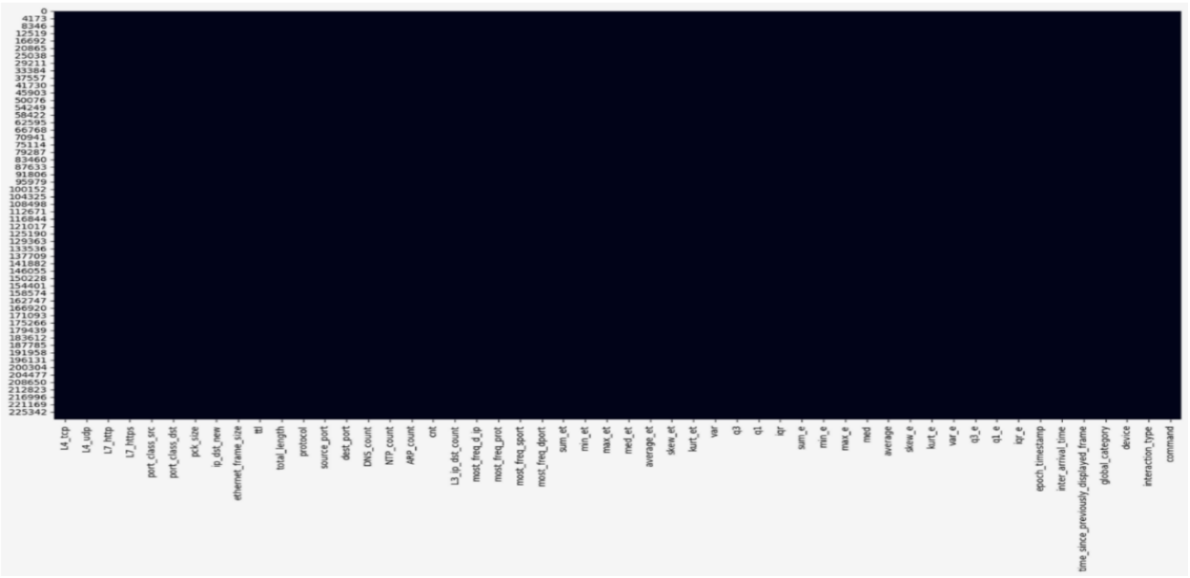


Figure 3.2 : Répartition de la totalité des données

3.4.2 Analyse du fond :

Pour l'analyse de fond nous avons débuté par visualiser la "variable Target" (« Global_category») et les résultats nous montrent qu'il est constitué de **191175** éléments de type CAMERA, **19557** éléments de type HOME AUTOMATION et **18774** éléments de type AUDIO. La catégorie CAMERA occupe 83% du trafic réseau ce qui montre que les classes sont très déséquilibrées.

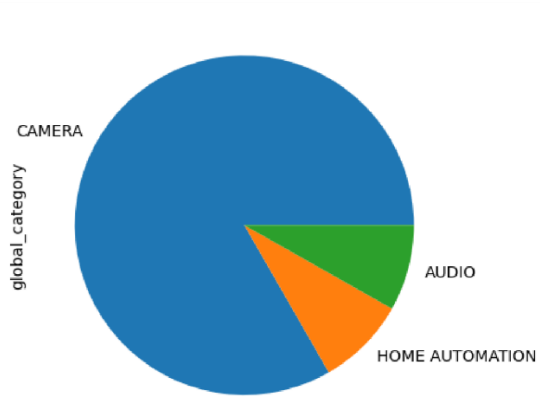


Figure 3.3 : Répartition des données de la variable Target.

Avoir des classes déséquilibrées dans un ensemble de données peut poser plusieurs inconvénients pour l'entraînement et l'évaluation des modèles :

- Les modèles peuvent devenir biaisés en faveur de la classe majoritaire, car ils sont "exposés" à plus d'exemples de cette classe pendant l'entraînement. Cela peut conduire à une précision élevée pour la classe majoritaire, mais une précision très faible pour les classes minoritaires.
- Les mesures globales de performance telles que l'exactitude (Accuracy) peuvent être trompeuses. Par exemple, si 95 % des étiquettes sont de la classe A et 5 % de la classe B, un modèle qui prédit toujours A aura une précision de 95 %, mais ce modèle ne sera pas utile pour identifier la classe B.

Il existe plusieurs stratégies pour gérer les classes déséquilibrées. Dans notre cas, nous avons utilisé la technique du Sous-échantillonnage qui consiste à réduire le nombre d'échantillons de la classe majoritaire pour équilibrer la distribution des classes. Nous avons aussi supprimé les colonnes 'device', 'command' et 'interaction type' que nous avons jugés inutiles car n'ayant aucune influence sur notre sortie. Ainsi, notre nouveau dataset est composé de **58506** lignes et **49** colonnes. La classe CAMERA constitue maintenant **20175** éléments (occupant **34%** des données). Ceci nous permet d'avoir des classes équilibrées.

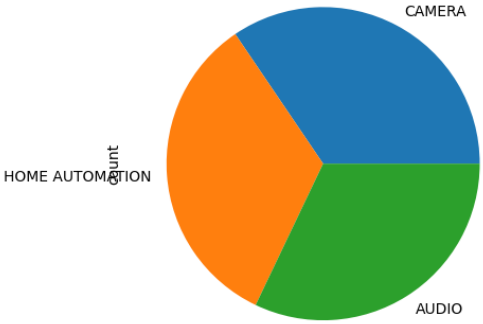


Figure 3.4 : Nouvelle répartition des données de la variable Target

Les libellés des caractéristiques et les données associées suggèrent une possible corrélation entre elles. Afin de vérifier cette hypothèse, nous avons effectué des calculs de corrélation entre les caractéristiques et avons ensuite représenté ces résultats sous forme de carte thermique en utilisant la bibliothèque Seaborn. Cette analyse montre que les features qui ont une grande corrélation (supérieur à 0.8) ne sont pas nombreux et la plupart sont moyennement corrélés. La matrice de corrélation est couramment utilisée pour réduire la dimensionnalité, prendre des décisions en matière de sélection de fonctionnalités.

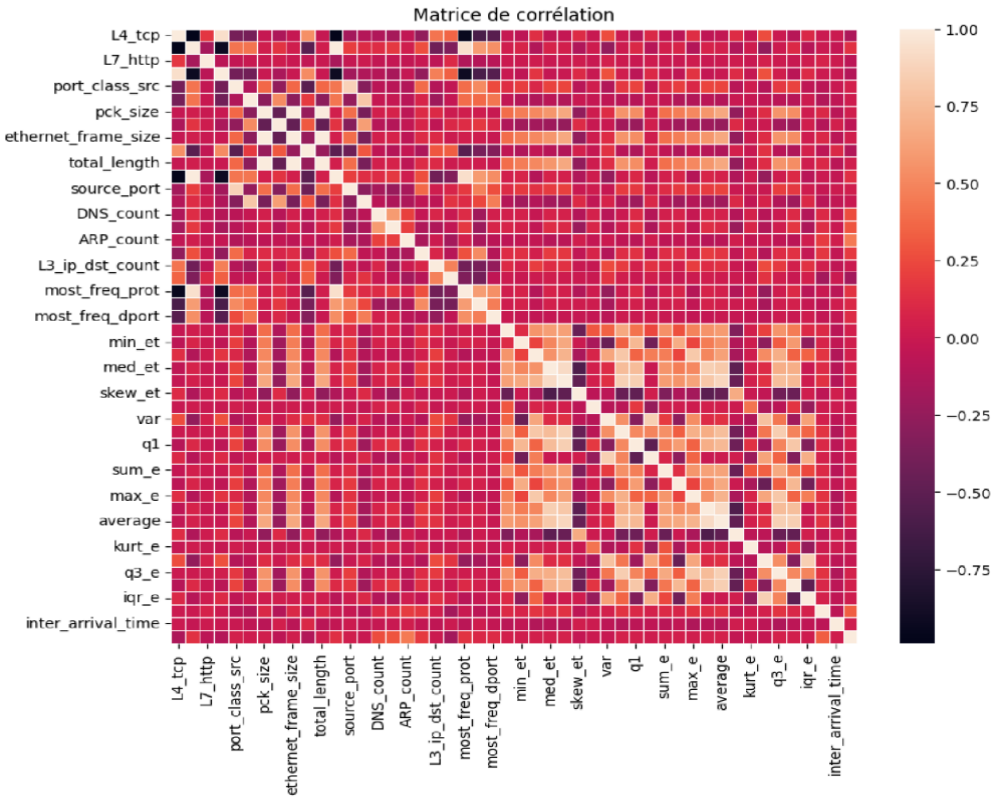


Figure 3.5 : Matrice de corrélation

3.5 Prétraitement des données

C'est l'étape qui consiste à préparer nos données avant de le donner à la machine pour son apprentissage. Le but est de mettre les données dans un format propice au ML et d'améliorer la performance du modèle.

Après avoir accompli la phase de compréhension et d'extraction des caractéristiques nous disposons désormais d'un nouveau Dataset avec 48 features (voir tableau 1) sur lequel nous allons faire un prétraitement des données. Le prétraitement englobe l'étape cruciale de la répartition des données en un ensemble d'entraînement et un ensemble de test. Cette opération est réalisée à l'aide de la fonction train_test_split du module Scikit-Learn. 70% des données sont allouées à l'ensemble d'entraînement, tandis que les 30% restants sont réservés à l'ensemble de test. Cette séparation est essentielle pour faciliter la convergence du modèle. Par conséquent, nous obtenons deux ensembles distincts : x_train et y_train destinés à l'apprentissage, et x_test et y_test réservés aux évaluations de performance. La dimension de ces données est représentée dans le tableau 2. Par la suite, nous avons encodé les étiquettes à l'aide de la fonction Label Encoder qui attribue la valeur 0 à la classe "AUDIO", la valeur 1 à la classe "CAMERA" et la valeur 2 à la classe HOME AUTOMATION. À ce stade, les données sont prêtes à être utilisées pour l'évaluation des modèles d'apprentissage automatique.

Tableau 1 : Nouveaux Caractéristiques du trafic réseau

		DESCRIPTION	
1	**L4_tcp**	Le protocole de contrôle de transmission est utilisé pour connecter les périphériques réseau à Internet (ConnectionOriented).	
2	**L4_udp**	Le protocole de datagramme utilisateur est utilisé pour établir des connexions à faible latence et tolérant les pertes (sans connexion)	
3	**L7_http**	Le protocole de transfert hypertexte est utilisé pour transmettre des données sur le World Wide Web (WWW).	
4	**L7_https**	Version sécurisée de HTTP	
5	**port_class_src**	Classe de numéro de port, aucun port = 0, port bien connu (0,1023) = 1, port enregistré (1024,49151)=2, port dynamique (49152,65535) =3	

6	**port_class_dst**	Classe de numéro de port, aucun port = 0, port bien connu (0,1023) = 1, port enregistré (1024,49151) =2, port dynamique (49152,65535) =3	
7	**pck_size**	Taille totale du paquet en octets, y compris les en-têtes de protocole et les données intégrés à l'en-tête IP. Champ de 16 bits	
8	**ip_dst_new**	Adresse IP de destination de l'appareil IoT	
9	**ethernet_frame_size**	Type de trame Ethernet utilisée par l'appareil	
10	**ttl**	La durée de vie (TTL) ou limite de sauts est la durée de vie ou la durée de vie d'un paquet	
11	**total_length**	Taille du paquet, en-tête compris	
12	**protocol**	Protocoles utilisés dans les couches 3, 4 et 5	
13	**source_port**	Port source de l'appareil	
14	**dest_port**	Port de destination de l'appareil	
15	**DNS_count**	Nombre de requêtes DNS	
16	**NTP_count**	Nombre de requêtes NTP	
17	**ARP_count**	Nombre de requêtes ARP	
18	**cnt**	Nombre de protocoles utilisés par l'appareil	
19	**L3_ip_dst_count**	Nombre d'adresses IP de destination accessibles par l'appareil IoT	
20	**most_freq_d_ip**	Adresse IP de destination la plus fréquemment visitée	
21	**most_freq_prot**	Le protocole le plus fréquemment utilisé	
22	**most_freq_sport**	numéro de port source utilisé plus fréquemment	
23	**most_freq_dport**	Numéro de port auquel le périphérique IoT accède le plus fréquemment	

24	**max_et**	Taille maximale du paquet IP en 20 paquets	
25	**med_et**	La taille médiane des paquets IP en 20 paquets	
26	**average_et**	Longueur moyenne du paquet IP en 20 paquets	
27	**pck_rawdata**	Données qui n'ont pas été traitées pour utilisation	
28	**var**	La mesure de l'écart entre les nombres dans 20 paquets	
29	**q3**	Le point central qui se situe entre la médiane et le plus grand nombre de distribution	
30	**q1**	Le nombre du milieu qui se situe entre la plus petite valeur de l'ensemble de données et la médiane.	
31	**iqr**	La quantité de propagation au milieu 50 % de 20 paquets	
32	**sum_e**	Agrégation des données	
33	**min_e**	Taille Ethernet minimale en 20 paquets	
34	**max_e**	Taille Ethernet maximale en 20 paquets	
35	**med**	La taille médiane d'Ethernet en 20 paquets	
36	**average**	Longueur moyenne de la trame Ethernet en 20 paquets	
37	sum_et	Agrégation des données	
38	min_et	Taille minimale du paquet IP en 20 paquets	
39	**var_e**	La mesure de l'écart entre les nombres dans 20 paquets	
40	**q3_e**	Le point central qui se situe entre le nombre médian et le nombre de distribution le plus élevé.	
41	**q1_e**	Le nombre du milieu qui se situe entre la plus petite valeur de l'ensemble de données et la médiane.	

42	**iqr_e**	la quantité de propagation au milieu 50% de 20 paquets	
43	**epoch_timestamp**	L'horodatage auquel le paquet spécifique a été enregistré	
44	**inter_arrival_time**	Le temps entre le paquet actuel et le paquet précéden	
45	**time_since_previously_displayed_frame**	Temps écoulé entre le paquet actuel et le premier paquet capturé.	
46	**ip_ralert**	Utilisé pour informer un routeur que le paquet IP actuel présente certaines particularités qui doivent être étudiées avant d'être transmis	
47	**ethernet_frame_type**	La taille de trame d'une trame Ethernet standard est la somme de l'en-tête Ethernet, de la charge utile et du Champ Séquence de vérification de trame (FCS).	
48	ip_padding	Le remplissage est essentiellement utilisé pour garantir que l'en-tête du paquet IP a une longueur multiple de 32.	

Le tableau suivant illustre la répartition des données de notre variable de sortie, après avoir été divisées en ensembles d'entraînement et de test.

global_category	**Données d'entrainement**	**Données de test**
AUDIO	**13107**	**5667**
CAMERA	**14176**	**5999**
HOME AUTOMATION	**13671**	**5886**

Tableau 2 : Répartition des données de la variable Target après division

3.6 Résultats

3.6.1 Métriques d'évaluation

Les métriques d'évaluation sont des mesures qui permettent d'évaluer la performance d'un modèle par rapport à un ensemble de données. Elles servent à quantifier à quel point les prédictions d'un modèle correspondent aux valeurs réelles, ce qui est essentiel pour évaluer la qualité et l'efficacité. Au cours des expériences de classification du trafic, nous avons fait appel

à des métriques telles que la Précision, le Rappel, et le F1-Score (comme défini dans les équations a,b,c et d) , la matrice de confusion et le rapport de classification.

- **Vrai Positif (TP**) : Pour une classe donnée, c'est le nombre d'échantillons correctement prédits comme appartenant à cette classe.
- **Faux Positif (FP) :** Pour une classe donnée, c'est le nombre d'échantillons incorrectement prédits comme appartenant à cette classe.
- **Faux Négatif (FN)** : Pour une classe donnée, c'est le nombre d'échantillons appartenant à cette classe qui ont été incorrectement prédits comme appartenant à une autre classe.
- **Vrai Négatif (TN)** : Pour une classe donnée, c'est le nombre d'échantillons qui n'appartiennent pas à cette classe et qui ont été correctement prédits comme n'appartenant pas à cette classe

$$Efficacité = \frac{TP + TN}{TP + TN + FP + FN} \quad (a)$$

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

Le taux de vrais positifs **T P R** fait référence à la sensibilité

$$TPR = \frac{TP}{TP + FN} \quad (c)$$

$$F1 = \frac{2\,TP}{2\,TP + FP + FN} \quad (d)$$

F1 représente la moyenne harmonique de la précision et de la sensibilité

3.6.2 Algorithmes du Machine Learning

Nous avons appliqué des algorithmes d'apprentissage automatique de classification pour modéliser nos données. Dans un premier temps, nous avons utilisé un modèle de base, à savoir l'Arbre de Décision afin d'ajuster notre prétraitement. Nous avons évalué d'autres modèles de classification, notamment le Random Forest et l'AdaBoost

3.6.2.1 Evaluation de Decision-Tree

Le tableau ci-dessous présente les résultats obtenus à l'aide de notre modèle initial, l'Arbre de Décision. Les résultats semblent indiquer une prédiction parfaite, avec une précision et une justesse de 100%. Néanmoins, il est impératif de vérifier que le modèle n'a pas sur-appris et d'évaluer sa performance sur de nouvelles données qui n'ont pas été incluses lors de la phase

d'apprentissage. Pour ce faire, nous avons également effectué l'entraînement en utilisant des données de validation.

	Precision	Recall	f1_score	Support	Matrice de confusion
0	1.00	1.00	1.00	5667	
1	1.00	1.00	1.00	5999	
2	1.00	1.00	1.00	5886	
accuracy			1.00	17552	[[5656 6 5] [2 5994 3] [0 2 5884]]
Macro avg	1.00	1.00	1.00	17552	
Weighted avg	1.00	1.00	1.00	17552	

Tableau 3: Résultats de la modélisation avec Decision Tree

Le graphique suivant (figure 3.6) illustre l'évolution des scores tant pour les données d'entraînement que pour les données de validation.



Figure 3.6 : Résultats de la modélisation avec Random Forest

Un écart notable entre les deux indique que le modèle peut présenter des difficultés à prédire efficacement de nouvelles données. Notre modèle a un score de 100% sur les données d'entrainements par contre il est incapable de généraliser sur de nouvelles données qu'il n'a jamais vu. Par conséquent, afin d'améliorer ces résultats, il est envisageable d'apporter davantage d'ajustements à notre prétraitement.

On peut procéder à une sélection de variable qui consiste à choisir un sous-ensemble pertinent de caractéristiques (variables) à partir de l'ensemble complet de caractéristiques disponibles dans un ensemble de données. Cela vise à améliorer la performance d'un modèle en éliminant les variables inutiles, redondantes ou bruitées.

Dans la figure 3.7 ci-dessous on peut voir les features les plus importantes utilisés par DecisionTree.

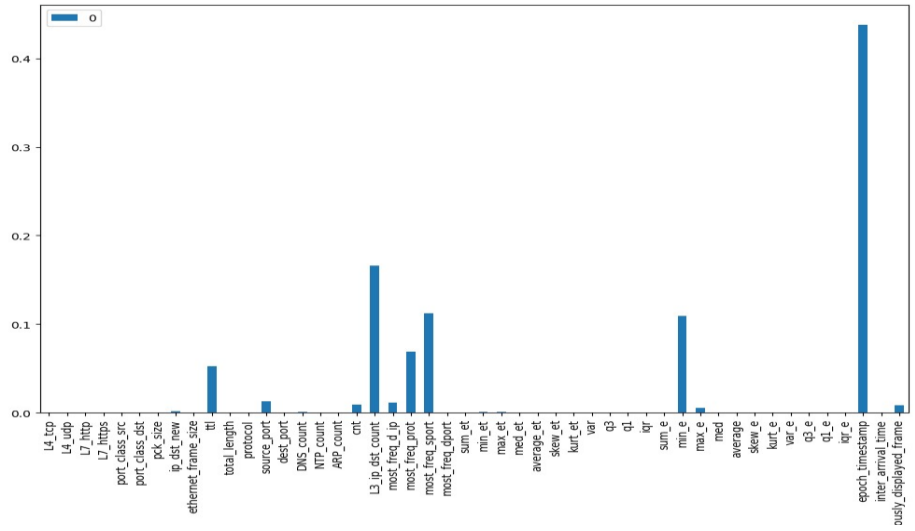


Figure 3.7 : Histogramme d'évaluation des caractéristiques importantes utilisées par Decision-Tree

Cet histogramme met en évidence les caractéristiques les plus significatives telles que "epoch_timestamp","L3_ip_dst_count","most_freq_sport ","most_freq_sport".

Par conséquent, il est envisageable d'optimiser davantage le modèle en éliminant les caractéristiques moins importantes, tout en évaluant les performances avec les caractéristiques essentielles, c'est-à-dire celles qui ont le plus d'impact.

3.6.2.2 Evaluation du Modèle Random Forest

Le Random Forest est un modèle qui lutte contre le surapprentissage (overfitting). En combinant plusieurs arbres, il réduit le risque d'overfitting par rapport à un seul arbre de décision. Comme pour l'Arbre de Décision, Les résultats semblent indiquer une prédiction parfaite, avec une précision et une justesse de 100%. Evaluons maintenant la performance du modèle avec des données de validations et d'entrainements.

	Precision	Recall	f1_score	support	Matrice de confusion
0	1.00	1.00	1.00	5667	
1	1.00	1.00	1.00	5999	
2	1.00	1.00	1.00	5886	[[5662 0 5] [5 5991 3] [0 1 5885]]
accuracy			1.00	17552	
Macro avg	1.00	1.00	1.00	17552	
Weighted avg	1.00	1.00	1.00	17552	

Tableau 4 : Résultats de la modélisation avec Random Forest

L'écart entre les deux courbes montrent que le RandomForest peut aussi avoir des difficultés à prédire efficacement de nouvelles données. Par contre, Le modèle RandomForest a réussi à diminuer le surapprentissage avec un f1_validation_score qui est légèrement supérieur au f1_validation_score du modèle de l'arbre de décision.

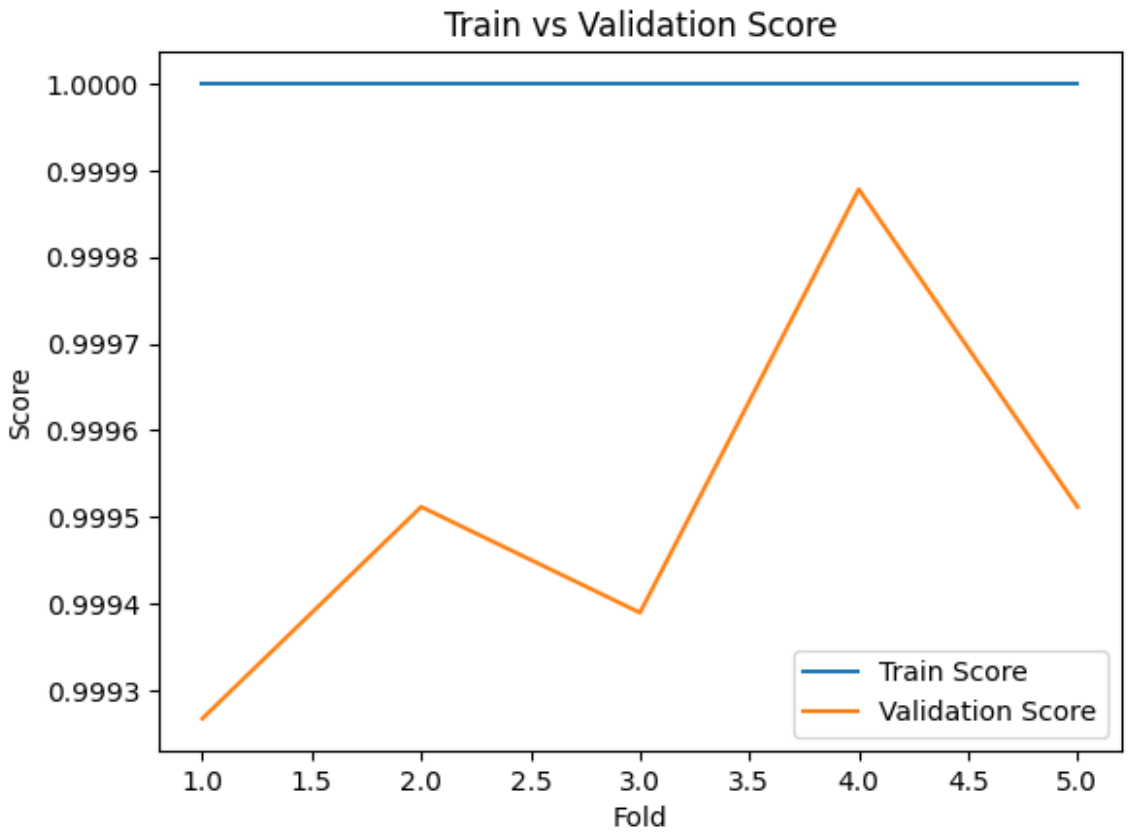


Figure 3.8 : Résultats de la modélisation avec RandomForest

3.6.2.3 Evaluation du modèle d'Adaboost

	Precision	Recall	f1_score	support	Matrice de confusion
0	0.94	0.87	0.90	5667	
1	0.78	0.91	0.84	5999	
2	0.97	0.87	0.92	5886	[[4910 757 0]
					[341 5477 181]
accuracy			0.89	17552	[0 758 5128]]
Macro avg	0.89	0.88	0.89	17552	
Weighted avg	0.99	0.88	0.89	17552	

Tableau 5 : Résultats de la modélisation avec AdaBoost

Selon les résultats observés dans les tableaux ci-dessus le Radom Forest offre plus de performance avec un f1_score de 100 % pour toutes les classes par rapport au modèle AdaBoost.

Néanmoins, nous avons analysé leur adaptation à de nouvelles données en examinant les scores générés en utilisant l'ensemble d'entraînement comme référence. Le fait qu'AdaBoost n'obtienne pas un score parfait sur les données d'entraînement indique qu'il ne sur-apprend pas autant que les modèles de DecisionTree et de RandomForest. Au lieu de chercher à obtenir une précision parfaite sur l'ensemble d'entraînement, AdaBoost met l'accent sur l'amélioration progressive de la performance, ce qui conduit souvent à une meilleure généralisation sur les données de test.

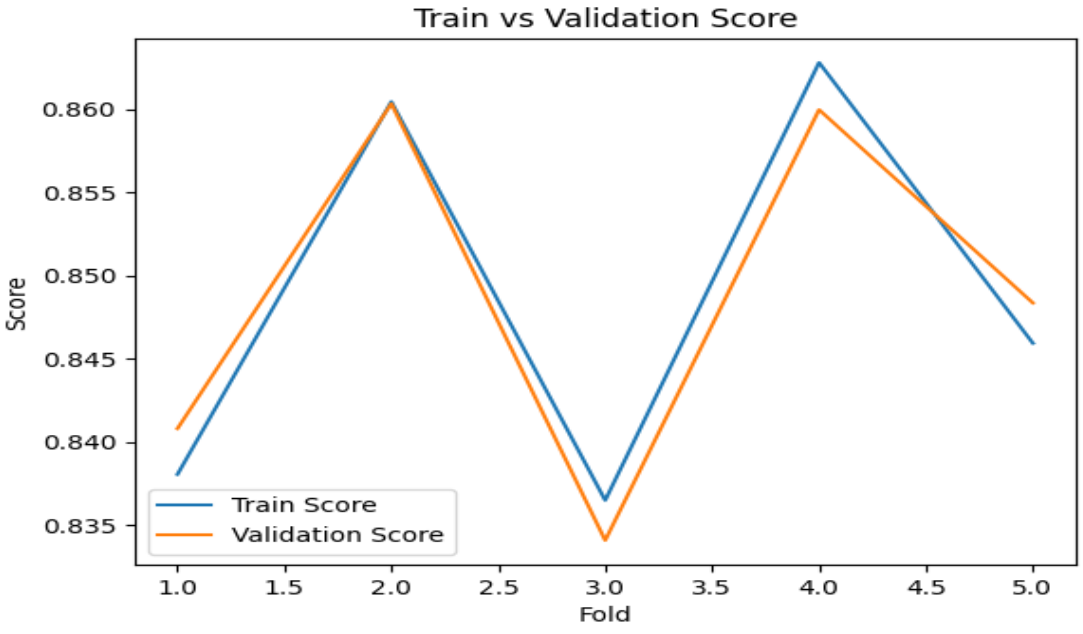


Figure 3.9 : Résultats de la modélisation avec AdaBoost

3.6.3 Evaluation avec un modèle de Réseaux de neurones à 4 couches

Le réseau de neurone que nous avons conçu est un modèle de classification composé de quatre couches, incluant deux couches cachées, une couche d'entrée avec 48 caractéristiques et une couche de sortie. Pour la couche de sortie, nous avons utilisé la fonction d'activation softmax, tandis que les autres couches utilisent la fonction ReLU. Après chaque couche nous avons ajouté une couche de Dropout pour prévenir l'overfitting ou l'underfitting.

Le tableau ci-dessous résume les différentes couches du réseau.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                3136

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 32)                2080

 dropout_1 (Dropout)         (None, 32)                0

 dense_2 (Dense)             (None, 3)                 99

=================================================================
Total params: 5315 (20.76 KB)
Trainable params: 5315 (20.76 KB)
Non-trainable params: 0 (0.00 Byte)
```

Nous avons entraîné le modèle sur 50 épochs et avons visualisé à la fois l'erreur et la précision tant sur les données d'entraînement que sur les données de validation. Les résultats obtenues montre un test accuracy (précision sur l'ensemble de test) de 97.98% ce qui signifie que notre modèle de classification a correctement classé 97.98% des exemples dans l'ensemble de test. Cela indique que notre modèle a une très haute précision lorsqu'il est évalué sur des données qu'il n'a pas vues pendant l'entraînement.

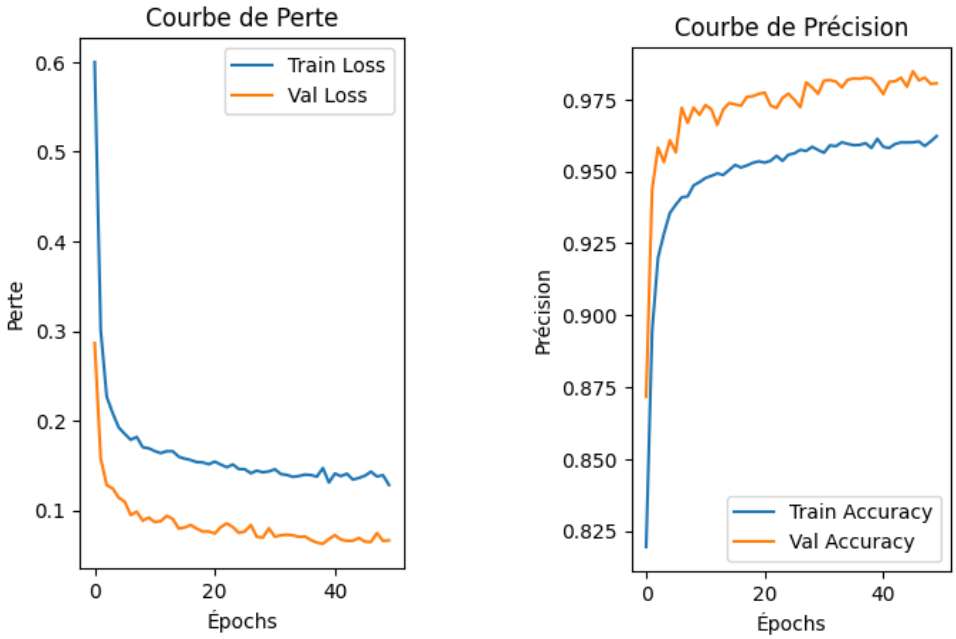


Figure 3.10 : Performance du réseau de neurone par rapport aux données d'entrainement et de validation

3.6.4 Evaluation avec le Modèle de Réseaux de Neurones Récurrents (RNN)

Nous avons évalué un modèle RNN simple avec une couche récurrente, des couches denses et des couches de dropout pour éviter le sur-apprentissage**.**

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn_9 (SimpleRNN)    (None, 32)                2592

 dropout_18 (Dropout)        (None, 32)                0

 dense_18 (Dense)            (None, 16)                528

 dropout_19 (Dropout)        (None, 16)                0

 dense_19 (Dense)            (None, 3)                 51

=================================================================
Total params: 3171 (12.39 KB)
Trainable params: 3171 (12.39 KB)
Non-trainable params: 0 (0.00 Byte)
```

Nous avons entrainé le modèle sur 50 epochs avec **Adam** comme optimisateur.

Le modèle de RNN développé a atteint une précision (Accuracy) de 98.65% sur l'ensemble de test, démontrant sa capacité à prédire avec succès les classes pour la majorité des exemples. Les courbes d'apprentissage révèlent une tendance où la validation_accuracy est légèrement

supérieure au train_accuracy, suggérant une bonne généralisation du modèle aux données de validation sans sur- apprentissage significatif.

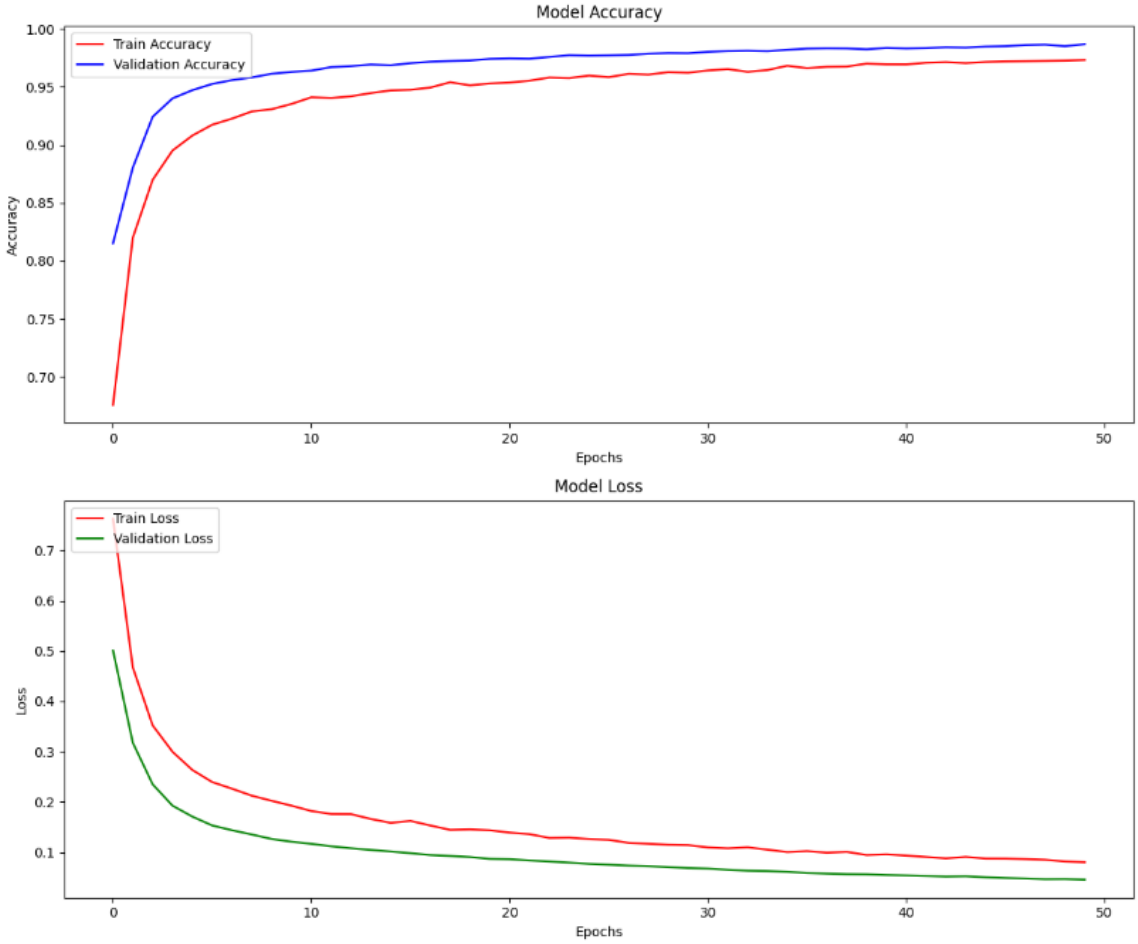


Figure 3.11 : Performance du RNN par rapport aux données d'entrainement et de validation

3.7 Discussion

Dans cette étude, nous avons évalué plusieurs modèles de Machine Learning traditionnels ainsi que des réseaux de neurones pour la classification des données. Les résultats obtenus offrent un aperçu intéressant des capacités et des limitations de chaque approche. Les résultats montrent que DecisionTree et RandomForest ont tendance à sur-apprendre aux données d'entraînement, capturant non seulement les véritables patterns mais aussi le bruit spécifique à ces données. Ce sur-apprentissage se traduit par des performances exceptionnelles sur les données d'entraînement mais une généralisation médiocre sur de nouvelles données. AdaBoost, en revanche, en adoptant une approche de correction des erreurs à chaque itération, parvient à éviter ce piège et montre une meilleure performance sur les données de test.

Le modèle de réseau de neurones à 4 couches, a montré une précision de 97.98%. Le RNN simple a montré une précision légèrement supérieure de 98%. Comme pour le modèle à 4 couches, les courbes de validation_accuracy légèrement supérieures à celles de train_accuracy indiquent une bonne capacité de généralisation. Cependant, ces précisions élevées peuvent être dues à la complexité du modèle. La complexité d'un modèle est souvent déterminée par deux principaux facteurs : le nombre de paramètres ajustables et la structure de son architecture. Plus un modèle a de paramètres, plus il peut mémoriser les détails des données d'entraînement, ce qui peut conduire à une meilleure performance sur ces données. La complexité d'un modèle joue un rôle crucial dans sa capacité à traiter des tâches complexes et variées. Il est important de trouver le bon équilibre entre la complexité du modèle et sa capacité à généraliser efficacement pour obtenir des résultats optimaux dans différentes applications de machine Learning.

Conclusion :

Nous avons proposé dans ce chapitre 3 modèles de Machine Learning (Arbre de Décision, RandomForest et AdaBoost), un modèle de réseau de neurone à 4 couches et un modèle de réseau de neurone récurrent RNN. Nous avons exploré la partie traitée de l'ensemble de données **CIC-IoT-dataset2022**. Cela nous a aidés dans l'étude à bien exploiter les données, en prétraitant l'ensemble de données pour former nos modèles. Cette étude met en lumière les forces et les limitations des différents modèles évalués. Bien que les modèles traditionnels comme AdaBoost présentent une bonne capacité à généraliser, les réseaux de neurones offrent également des performances prometteuses, en particulier pour des tâches complexes nécessitant une compréhension contextuelle plus profonde. Le choix du modèle devrait être guidé par la nature spécifique des données, les exigences de la tâche et la nécessité de généraliser efficacement à de nouvelles données.

Conclusion Générale

L'Internet des Objets (IoT) est au cœur de la transformation numérique actuelle, interconnectant une multitude d'appareils intelligents pour créer un écosystème dynamique et informatisé. Cette révolution technologique offre des avantages considérables en termes de commodité, d'efficacité et de productivité, mais elle présente également des défis majeurs, notamment en matière de gestion du trafic réseau. La classification des données de trafic réseau se révèle donc être un enjeu crucial, nécessitant des solutions novatrices pour garantir la sécurité, la performance et la fiabilité des réseaux. Dans ce contexte, les techniques de Machine Learning (ML) émergent comme des outils puissants pour aborder ces problématiques complexes. Ce mémoire a exploré les aspects fondamentaux de la classification des données de trafic réseau dans l'IoT en se concentrant sur l'intégration de ces techniques avancées. En combinant l'intelligence artificielle avec les défis spécifiques du trafic IoT, nous avons cherché à concevoir des modèles adaptatifs capables de comprendre et de catégoriser efficacement les données de trafic, contribuant ainsi à une gestion plus efficace et sécurisée des réseaux. Au cours de notre travail, nous avons d'abord présenté les fondamentaux sur l'IoT, incluant ses concepts de base, ses architectures, ses applications, ainsi que les enjeux auxquels il est exposé. Nous avons ensuite abordé les techniques de Machine Learning en détaillant les méthodes allant de l'apprentissage automatique à l'apprentissage profond, en passant par les réseaux de neurones. Enfin, nous avons présenté notre apport personnel en proposant une classification intelligente des données de trafic réseau, en utilisant des modèles avancés de Machine Learning. Nous avons également décrit les outils et le dataset utilisés pour nos expérimentations. Les résultats obtenus démontrent que les modèles de réseaux de neurones, en particulier, offrent une précision remarquable dans la classification des données de trafic réseau, surpassant les méthodes traditionnelles de Machine Learning. Il est important de reconnaître les limitations des approches traditionnelles de Machine Learning. Ces méthodes, bien que puissantes, peuvent se heurter à des difficultés lorsqu'il s'agit de traiter des volumes de données extrêmement larges et complexes. Pour des recherches futures on peut utiliser un jeu de donnée avec une grande quantité de données ou même faire la technique du Sur-échantillonnage avec notre jeu de donnée qui consiste à augmenter le nombre d’échantillons des classes minoritaires pour évaluer des modèles des modèles de réseaux de neurones profonds pour la classification des données. Les réseaux de neurones profonds, surmontent ces limitations. Ils capturent des caractéristiques complexes et des relations non linéaires dans les données, améliorant ainsi les performances en classification du trafic réseau.

Bibliographie

[1] Sethi Pallavi and Sarangi Smruti. “ Internet of things: architectures, protocols, and applications”, Journal of Electrical and Computer Engineering, 2017.

[2] Les Fondamentaux de l’IoT [en ligne]. Consulté le 12/02/2023. Disponible sur : https://www.itu.int/en/ITU-D/Regional-Presence/Africa/Documents/PRIDA%202020%20%20ONLINE%20Capacity%20building%20%26%20digital%20services/FR_Workshop_Slid es.pdf

[3] Cloud Computing et IoT [en ligne]. Consulté le 16/02/2023. Disponible sur : https://iotindustriel.com/iot-iiot/cloud-computing-iot-le-duoincontournable/#:~:text=Diff%C3%A9rences%20de%20l'IoT%20et%20le%20Cloud%20Computing,D'une%20part&text=Dans%20l'IoT%2C%20les%20donn%C3%A9es,des%20Data%20Cente rs%20via%20Internet.

[4] Said Omar and Masud Mehedi. “ Towards internet of things: Survey and future vision”. International Journal of Computer Networks. 2013, Volume 5, numéro 1, p 1-17.

[5] Architecture IoT l’essentiel à savoir [en ligne]. Consulté le 22/02/2023. Disponible sur : https://iotindustriel.com/iot-iiot/architecture-iot-lessentiel-a-savoir/

[6] Hassija Vikas, Chamola Vinay, Saxena Vikas, Jain Divyansh, Goyal Pranav and Sikdar Biplab. “ A survey on IoT security: application areas, security threats, and solution architectures”. IEEE Access. 2019, volume 7, p 82721-82743.

[7] APWG. Phishing Activity Trends Report. Accessed: Feb. 12, 2019. [Online]. Available: https://www.google.com/search?q=APWG.+Phishing+Activity+Trends+Report.+Accessed%3A+Feb.+12%2C+2019.+%5BOnline%5D.+Available%3A+https%3A%2F%2Fdocs.apwg.org%2Freports%2Fapwg_trends_+report_q4_2017.pdf&rlz=1C1XXVF_frSN961SN961&oq=APWG.+Phishing+Activity+Trends+Report.+Accessed%3A+Feb.+12%2C+2019.+%5BOnline%5D.+Available%3A+https%3A%2F%2Fdocs.apwg.org%2Freports%2Fapwg_trends_+report_q4_2017.pdf&aqs=chrome..69i57.469j0j4&sourceid=chrome&ie=UTF-8

[8] Alhammadi Nafea Ali Majeed and Zaboon Khalid Hameed. “ A Review of IoT Applications, Attacks and Its Recent Defense Methods”. Journal of Global Scientific Research. 2022, volume 7, numéro 3, p 2128-2134

[9] Khalaf Bashar Ahmed, Mostafa S. A., Mustapha Aida, Ismaila Azizan, Mahmoud M. A., Jubaira Mohammed Ahmed and Hassan M. H. “ A simulation study of syn flood attack in cloud computing environment ”. AUS journal. 2019, volume 26, numéro 1, p 188-197.

[10] Yan Zheng, Zhang Peng and Vasilakos Athanasios V. “ A survey on trust management for Internet of Things”. Journal of network and computer applications. 2014, volume 42, p 120-134.

[11] Shifa Amna, Asghar Mamoona N. and Fleury Martin. “ Multimedia security perspectives in IoT ”. Sixth International Conference on Innovative Computing Technology (INTECH). IEEE. 2016, p 550-555.

[12] Hindy Hanan, Brosset David, Bayne Ethan, Seeam Amar Kumar, Tachtatzis Christos, Atkinson Robert and Bellekens Xavier. “ A taxonomy of network threats and the effect of current datasets on intrusion detection systems ”. IEEE Access. 2020, volume 8, p 104650–104675.

[13] Neelankavil Shilvi Wilson and S Minla K. “ A Survey of Intrusion Detection Systems in Wireless Sensor Networks ”. International Journal of Engineering Research & Technology. 2020/03/31, volume 8, numéro 4.

[14] Kumar B. Santos, Raju T. Chandra Sekhara Phani, Baba Sk Dawood, Ratnakar M. and Sudhakar N. “ Intrusion detection system-types and prevention ”. International Journal of Computer Science and Information Technologies. 2013, volume 4, numéro 1, p 77-82

[15] Das Niva and Sarkar Tanmoy. “ Survey on host and network based intrusion detection system ”. International Journal of Advanced Networking and Applications. 2014, volume 6, numéro 14, p 2266-2269.

[16] Les sondes de sécurité IDS/IPS [en ligne]. Consulté le 15/05/2023. Disponible sur : http://igm.univ-mlv.fr/~dr/XPOSE2009/Sonde_de_securite_IDS_IPS/diff.html

[17] Kumar Sailesh. “ Survey of current network intrusion detection techniques”. Washington, Univ. in St. Louis. 2007, p 1-18

[18] Bijone Manu. “A survey on secure network: intrusion detection & prevention approaches”.American Journal of Information Systems. 2016, volume 4, numéro 3, p 69-88.
[19] Riquet Damien. “ Discus: Une architecture de détection d'intrusions réseau distribuée basée sur un langage dédié ”. 2015.

[20] Mahesh Batta. “ Machine learning algorithms-a review”. International Journal of Science and Research (IJSR). [Internet]. 2020, volume 9, numéro 1, p 381–386.

[21] Machine Learning : Définition, fonctionnement, utilisations [en ligne]. Consulté le 05/08/2023. Disponible sur : https://datascientest.com/machine-learning-tout-savoir

[22] Comprendre les réseaux de neurones [en ligne]. Consulté le 05/08/2023. Disponible sur: https://moncoachdata.com/blog/comprendre-les-reseaux-de-neurones/

[23] Janiesch Christian, Zschech Patrick and Heinrich Kai. “ Machine learning and deep learning”. Electronic Markets. 2021, volume 31, numéro 3, p 685–695

[24] Aldweesh Arwa, Derhab Abdelouahid and Emam Ahmed Z. “ Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues”. Knowledge-Based Systems. 2020, volume 189, p 105124.

[25] Biswal Avijeet. “ Top 10 deep learning algorithms You should Know in 2021”. https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learningalgorithm, accessed. 2021, volume 5, p 26

[26] Convolutional Neural Network : Tout ce qu'il y a à savoir [en ligne]. Consulté 07/08/2023. Disponible sur : https://datascientest.com/convolutional-neural-network

[27] Qu'est-ce qu'un réseau de neurones convolutifs ? [En ligne]. Consulté le 07/08/2023 Disponible sur : https://www.ibm.com/fr-fr/topics/convolutional-neural-networks

[28] Découvrez le fonctionnement des réseaux de neurones récurrents. [En ligne]. Consulté le 07/08/2023. Disponible sur : https://openclassrooms.com/fr/courses/5801891-initiez-vous-au deep-learning/5814651-decouvrez-le-fonctionnement-des-reseaux-de-neurones-recurrents

[29] Les réseaux de neurones récurrents pour les séries temporelles. [En ligne]. Consulté le 09/08/2023/. Disponible sur https://metalblog.ctif.com/2021/09/06/les-reseaux-de-neuronesrecurrents-pour-les-series-temporelles/

[30] Janiesch Christian, Zschech Patrick and Heinrich Kai. “ Machine learning and deep learning”. Electronic Markets. 2021, Volume 31, numéro 3, p 685–695

[31] Auto-encodeurs en Deep Learning : tout savoir. [En ligne]. Consulté le 10/08/2023. Disponible sur : https://blent.ai/blog/a/auto-encodeurs-deep-learning

[32] Shone Nathan, Ngoc Tran. Nguyen, Phai Vu. Dinh and Shi Qi. “ A deep learning approach to network intrusion detection”. IEEE transactions on emerging topics in computational intelligence. 2018, Volume 2, Numéro 1, p 41–50.

[33] Sajjad Dadkhah, Hassan Mahdikhani, Priscilla Kyei Danso, Alireza Zohourian, Kevin Anh Truong, Ali A. Ghorbani, “Towards the development of a realistic multidimensional IoT profiling dataset”, Submitted to: The 19th Annual International Conference on Privacy, Security & Trust (PST2022) August 22-24, 2022, Fredericton, Canada.