REPUBLIQUE DU SENEGAL



Un Peuple-Un But-Une Foi

MINISTERE DE L'ENSEIGNEMENT SUPERIEURE DE LA RECHERCHE ET

DE L'INNOVATION

UNIVERSITE GASTON BERGER DE SAINT-LOUIS



U.F.R DES SCIENCES APPLIQUÉES ET DE TECHNOLOGIE

DÉPARTEMENT DE PHYSIQUE APPLIQUÉE

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

MASTER INGÉNIERIE EN ÉLECTRONIQUE ET TÉLÉCOMMUNICATION (MIETEL)

THÈME :

DÉTECTION PRÉCOCE DES INTRUSIONS DANS LES RÉSEAUX IoT PAR UNE APPROCHE D'INTELLIGENCE ARTIFICIELLE CENTRALISÉE

Landing DIATTTA — Présentateur

Soutenance le 11/02/2026 devant le jury composé de :

Professeur Adama SY — Président du jury

Professeur Papa Alioune FALL et Docteur Abdoul Karim MBODJI — Examinateurs

Professeur Chérif DIALLO — Encadrant

ANNEE UNIVERSITAIRE 2023/2024

Remerciement

Je commence par rendre grâce à Allah, le Tout-Puissant, pour m'avoir accordé la santé, la patience et la détermination nécessaires à l'achèvement de ce mémoire.

Je tiens à exprimer ma profonde reconnaissance à mon encadrant, Professeur Cherif DIALLO pour son accompagnement, ses conseils précieux et sa disponibilité tout au long de ce travail.

Je remercie également les membres du jury pour l'attention portée à ce mémoire et pour le temps consacré à son évaluation.

Un remerciement tout particulier va à mon petit père, Boubacar Diatta, pour son soutien indéfectible, sa confiance et ses encouragements constants, qui ont été une source de motivation tout au long de ce parcours.

Je souhaite également remercier chaleureusement mes amis, Ousmane Diémé, Mamadou Diarra, Mouhamed Diallo, mes camarades de classe et tous ceux qui m'ont soutenu de près ou de loin. Leur présence, leurs conseils, et les moments partagés ont été d'un grand réconfort et m'ont permis de rester motivé et concentré.

Je tiens à remercier chaleureusement mes camarades de chambre, Babacar Oubiran Sané**,** Simon et Kalidou Tall, pour leur soutien, leur convivialité et tous les moments de partage qui ont rendu cette expérience plus agréable sur le plan social et humain.

Je remercie également les imams du village K, dont les prières et conseils spirituels m'ont apporté force et sérénité.

Enfin, je remercie toutes les personnes et institutions qui ont contribué, directement ou indirectement, à la réalisation de ce mémoire, et qui ont enrichi cette expérience tant sur le plan académique que personnel.

***Dédicace** :*

Je dédie ce mémoire à ma chère mère, Mariama DIATTA, pour son amour, ses sacrifices et son soutien indéfectible.

À mon père, Lamine DIATTA, pour ses encouragements constants et sa confiance en moi.

À mon oncle, Boubacar DIATTA, pour ses conseils avisés et sa présence bienveillante.

À mes frères et sœurs, pour leur affection et leur soutien tout au long de mon parcours.

Et à toute ma famille, pour leur encouragement, leur patience et leur inspiration quotidienne.

Résumé :

L'Internet des Objets (IoT) occupe une place centrale dans de nombreux domaines critiques, mais son expansion rapide s'accompagne de défis majeurs en matière de sécurité, liés à l'hétérogénéité des dispositifs, aux ressources limitées et à l'ouverture des architectures réseau. Face à des attaques de plus en plus sophistiquées, les mécanismes de sécurité traditionnels se révèlent insuffisants, rendant nécessaire le recours à des solutions intelligentes et adaptatives.

Ce mémoire propose une approche de détection précoce des intrusions basée sur une intelligence artificielle centralisée dans les réseaux IoT. Plusieurs modèles de Machine Learning et de Deep Learning ont été implémentés et évalués à l'aide du dataset CICIoT2023, en s'appuyant sur des métriques de performance standards telles que l'exactitude, la précision, le rappel et le score F1.

Les résultats obtenus mettent en évidence que les modèles de Machine Learning, et en particulier Random Forest, offrent d'excellentes performances en matière de détection des intrusions, tout en garantissant une rapidité d'exécution et une simplicité d'implémentation appréciables. À l'inverse, les modèles de Deep Learning, bien que puissants d'un point de vue théorique, ne parviennent pas à surpasser les modèles de Machine Learning dans ce contexte, notamment en raison de leurs besoins plus élevés en ressources computationnelles et en temps de traitement.

Ces travaux confirment la pertinence des approches basées sur le Machine Learning pour renforcer la sécurité des réseaux IoT et ouvrent des perspectives vers des solutions de détection plus adaptatives, distribuées et mieux adaptées aux contraintes spécifiques des environnements IoT.

Mots-clés : Internet des Objets (IoT), Sécurité des réseaux, Détection d'intrusions, Intelligence artificielle, Machine Learning, Deep Learning, CICIoT2023.

Abstract :

The Internet of Things (IoT) plays a central role in many critical application domains; however, its rapid expansion introduces major security challenges related to device heterogeneity, limited resources, and open network architectures. As cyberattacks become increasingly sophisticated, traditional security mechanisms prove insufficient, highlighting the need for intelligent and adaptive intrusion detection solutions.

This thesis proposes an early intrusion detection approach based on centralized artificial intelligence in IoT networks. Several Machine Learning and Deep Learning models are implemented and evaluated using the CICIoT2023 dataset, relying on standard performance metrics such as accuracy, precision, recall, and F1-score.

The experimental results demonstrate that Machine Learning models, particularly Random Forest, achieve excellent intrusion detection performance while ensuring fast execution and ease of implementation. In contrast, Deep Learning models, although theoretically powerful, do not outperform Machine Learning approaches in this context due to their higher computational and time requirements.

These findings confirm that Machine Learning–based approaches currently represent the most suitable solution for real-world IoT deployments constrained by limited resources and open perspectives toward more adaptive and distributed intrusion detection systems.

Keywords : Internet of Things (IoT), Network Security, Intrusion Detection, Artificial Intelligence, Machine Learning, Deep Learning, CICIoT2023.

Table des matières

..... 2

Remerciement 2

Dédicace : 3

Résumé : 4

Abstract : 5

Table des matières 6

Listes des acronymes et abréviations 9

Liste des figures 11

Listes des tableaux 13

INTRODUCTION GENERALE 14

CHAPITRE 1 : DEFINITIONS ET CONCEPTS DE BASE 15

1.1. INTRODUCTION 15

1.2. LES RESEAUX IoT 15

1.2.1. Définition, architecture, composants 15

1.2.2. Protocoles IoT 19

1.2.3. Contraintes des réseaux IoT 23

1.2.4. Défis de l'internet des objets 25

1.2.5. Domaines d'applications de l'IoT 28

1.3. SYSTÈME DE DÉTECTION D'INTRUSION 34

1.3.1. Définition Système de Détection d'Intrusion 34

1.3.2. Fonctionnement et rôle d'un Système de Détection d'Intrusion 35

1.3.3. Méthodes de détection dans les systèmes IDS 35

1.3.4. Les types de système détection d'intrusion 37

1.4. INTELLIGENCE ARTIFICIELLE, MACHINE LEARNING ET DEEP LEARNING 38

1.4.1. Intelligence artificielle 38

1.4.2. Machine learning 39

1.4.3. Deep learning 44

1.5. CONCLUSION 46

CHAPITRE 2 : ÉTAT DE L'ART SUR LES APPROCHES MACHINE LEARNING ET DEEP LEARNING POUR LA DETECTION D'INTRUSION DANS LES RESEAUX IoT 47

2.1. INTRODUCTION 47

2.2. APPROCHE MACHINE LEARNING POUR LA DETECTION D'INTRUSION 47

2.2.1. Réseaux de neurones artificiels 48

2.2.2. Machine à vecteurs de support (SVM) 48
2.2.3. K plus proches voisins (KNN) 49
2.2.4. Régression logistique (LR) 49
2.2.5. Arbres de décision 49
2.2.6. Regroupement 50
2.2.7. Forêt aléatoire (Random Forest – RF) 50
2.3. APPROCHE DEEP LEARNING POUR LA DETECTION D'INTRUSION ... 50
2.3.1. Réseaux de neurones profonds (DNN) 51
2.3.2. Réseaux de neurones convolutifs (CNN) 51
2.3.3. Réseaux de neurones récurrents (RNN) 52
2.4. ANALYSE 52
2.5. CONCLUSION 55
CHAPITRE 3 : IMPLEMENTATION, TESTS ET EVALUATION DES PERFORMANCES . 56
3.1. INTRODUCTION 56
3.2. OUTILS ET ENVIRONNEMENT DE DEVELOPPEMENT 56
3.2.1. Google colab 56
3.2.2. Python 56
3.2.3. SKLearn 57
3.2.4. TensorFlow 57
3.2.5. Keras 58
3.3. PRESENTATION DU DATASET 58
3.4. PRETRAITEMENT DES DONNEES 59
3.5. CHOIX DE L'ALGORITHME 60
3.6. ARCHITECTURE GLOBALE DU SYSTEME 60
3.7. METRIQUES D'EVALUATIONS 62
3.7.1. Exactitude (Accuracy) 63
3.7.2. Précision (Precision) 63
3.7.3. Rappel (Recall) 63
3.7.4. Score F1 (F1-Score) 64
3.7.5. Matrice de confusion 64
3.8. RESULTATS DES MODELES D'APPRENTISSAGES AUTOMATIQUES ET PROFOND 64
3.8.1. Résultats des modèles d'apprentissage automatique (ML) 65
3.8.2. Résultats des modèles d'apprentissage profond (DL) 69
3.9. Analyse des performances des modèles d'apprentissage automatique (ML) et profond (DL) 71

3.9.1. Analyse des performances des modèles d'apprentissage automatique (ML)..... 71
3.9.2. Analyse des performances des modèles d'apprentissage profond (DL)..... 72
3.9.3. Comparaison des modèles ML et DL..... 73
3.10. CONCLUSION 74
CONCLUSION GENERALE..... 75
BIBLIOGRAPHIE 76

Listes des acronymes et abréviations

AE : Auto-encoder

ANN : Artificial Neural Network

CNN : Convolutional Neural Network

CoAP : Constrained Application Protocol

DL : Deep Learning

DDS : Data Distribution Service

DNN : Deep Neural Network

DT : Decision Tree

FN : False Negatives

FP : False Positives

GPS : Global Positioning System

HIDS : Host-based Intrusion Detection System

HTTP : Hypertext Transfer Protocol

IA : Intelligence Artificielle

IDS : Intrusion Detection System

IETF : Internet Engineering Task Force

IoT : Internet of Things

IoTM : Internet of Things Magazine

KNN : K-Nearest Neighbor

LR : Régression logistique

ML : Machine Learning

MQTT : Message Queuing Telemetry Transport

NIDS : Network-based Intrusion Detection System

OSI : Open Systems Interconnection

QoS : qualité de service

REST : Representational State Transfer

RFID : radio frequency identification

RNN : réseaux neuronaux récurrents

SVM : Support Vector Machine

TCP/IP : Transmission Control Protocol/Internet Protocol

TLS **:** Transport Layer Security

TN : True Negatives

TP : True Positives

VAE : AutoEncoder variationnel

WSN : Wireless Sensor Networks

XMPP : eXtensible Messaging and Presence Protocol

Liste des figures

Figure 1.1 : L'architecture de l'Internet des objets**……...17**

Figure 1.2 : Modèle d'interaction REST http**……19**

Figure 1.3 : Fonctionnalités CoAP **……...20**

Figure 1.4 : Communication dans XMPP**……21**

Figure 1.5 : Architecture conceptuelle du DDS **……...22**

Figure 1.6 : L'architecture du protocole MQTT**……...23**

Figure 1.7 : Différents types d'applications agricoles pour l'IoT**……..31**

Figure 1.8 : Industrie 4.0**……...34**

Figure 1.9 : Système de détection basée sur les signatures**……36**

Figure 1.10 : Système de détection d'anomalie**……37**

Figure1.11 :L'apprentissage supervisé**……40**

Figure 1.12 : L'apprentissage non supervisé**……40**

Figure 1.13 : L'apprentissage par renforcement**……41**

Figure 3.1 : Flux de traitement pour la détection précoce d'intrusions IoT à l'aide du dataset CICIoT2023**……..62**

Figure 3.2 : Matrice de confusion Logistic Regression**……65**

Figure 3.3 : Matrice de confusion SVM**……66**

Figure 3.4 : Matrice de confusion ANN**……67**

Figure 3.5 : Matrice de confusion Arbre de décision**……68**

Figure 3.6 : Matrice de confusion Random Forest……**69**

Figure 3.7 : Matrice de confusion DNN……**70**

Figure 3.8 : Matrice de confusion CNN……**71**

Figure 3.9 : Comparaison des performances des modèles ML……..**72**

Figure 3.10 : Comparaison des performances des modèles DL……**73**

Figure 3.11 : Comparaison des performances des modèles ML et DL……..**74**

Listes des tableaux

Tableau 2.1 : Récapitulatif des approches ML et DL pour la détection d'intrusions dans les réseaux IoT…**……..……..…...53**

Tableau 3.1 : Performance du modèle régression logistique**……65**

Tableau 3.2 : Performance du modèle SVM**……66**

Tableau 3.3 : Performance du modèle ANN**……67**

Tableau 3.4 : Performance du modèle Arbre de décision**……68**

Tableau 3.5 : Performance du modèle Random Forest**……68**

Tableau 3.6 : Performance du modèle Réseaux de neurones profonds (DNN**)……69**

Tableau 3.7 : Performance du modèle Réseaux de neurones convolutifs (CNN**)……..70**

INTRODUCTION GENERALE

L'Internet des Objets (IoT) s'est imposé comme un élément central des systèmes numériques modernes, en permettant l'interconnexion de milliards de dispositifs physiques capables de collecter, transmettre et traiter des données en temps réel. Ces dispositifs sont aujourd'hui largement déployés dans des domaines stratégiques tels que les villes intelligentes, l'industrie 4.0, la santé connectée, l'agriculture intelligente ou encore les systèmes de transport. Cette évolution favorise l'automatisation, l'optimisation des processus et la prise de décision intelligente, mais elle s'accompagne également de nouveaux défis techniques et sécuritaires.

En raison de leur nature distribuée, de leur forte hétérogénéité et des limitations en ressources des objets connectés (énergie, mémoire, puissance de calcul), les réseaux IoT présentent une surface d'attaque particulièrement étendue. Ils sont ainsi vulnérables à de nombreuses menaces, notamment les attaques par déni de service (DDoS), les intrusions, les malwares et les manipulations de données. Dans ce contexte, la détection précoce des comportements anormaux constitue un enjeu majeur afin de préserver la confidentialité, l'intégrité et la disponibilité des systèmes IoT. Les approches traditionnelles de sécurité, basées sur des règles statiques ou des signatures prédéfinies, montrent rapidement leurs limites face à des environnements IoT dynamiques et évolutifs. Face à ces insuffisances, l'intelligence artificielle (IA), et plus particulièrement le Machine Learning et le Deep Learning, s'imposent comme des solutions prometteuses pour la détection automatique et proactive des intrusions. Ces techniques permettent de modéliser le comportement normal du réseau et d'identifier efficacement les anomalies, y compris les attaques inconnues.

Ce mémoire s'inscrit dans cette dynamique et vise à concevoir et évaluer un système de détection précoce d'intrusions basé sur une intelligence artificielle centralisée dans les réseaux IoT. L'objectif est d'analyser les performances de différentes approches d'apprentissage automatique et profond, tout en tenant compte des contraintes spécifiques de l'IoT. Le travail est structuré en trois chapitres : un premier chapitre consacré aux concepts fondamentaux, un second dédié à l'état de l'art, et un troisième axé sur l'expérimentation et l'évaluation des performances du système proposé.

CHAPITRE 1 : DEFINITIONS ET CONCEPTS DE BASE

1.1.INTRODUCTION

L'Internet des Objets repose sur un ensemble de technologies interconnectées dont la compréhension est indispensable pour appréhender les problématiques de sécurité qui lui sont associées. Avant d'explorer les méthodes avancées de détection d'intrusion basées sur l'intelligence artificielle, il est nécessaire de définir les concepts fondamentaux qui constituent le socle de ce travail. Ce chapitre a pour objectif de présenter les notions essentielles relatives aux réseaux IoT, en détaillant leur définition, leur architecture, leurs composants et les principaux protocoles de communication utilisés. Il met également en évidence les contraintes spécifiques à ces réseaux, telles que la limitation des ressources, l'hétérogénéité des dispositifs, les problématiques de routage, de qualité de service et de sécurité. Ces contraintes expliquent en grande partie la complexité de la mise en œuvre de mécanismes de protection efficaces dans les environnements IoT. Par ailleurs, ce chapitre introduit les systèmes de détection d'intrusion (IDS), en exposant leurs principes de fonctionnement, leurs méthodes de détection ainsi que leurs différentes catégories. Enfin, les concepts fondamentaux de l'intelligence artificielle, du Machine Learning et du Deep Learning sont présentés, afin de poser les bases théoriques nécessaires à la compréhension des approches de détection étudiées dans les chapitres suivants.

1.2. LES RESEAUX IoT

1.2.1. Définition, architecture, composants

1.2.1.1.Définition

Bien que la définition précise de l'Internet des objets (IoT) soit encore en débat, il est généralement admis qu'il s'agit de l'interconnexion d'objets physiques intégrant des capteurs et des logiciels [1]. Ces « objets » peuvent échanger des données et communiquer entre eux**,** créant ainsi un environnement intelligent et interconnecté [2].

Il existe plusieurs façons d'appréhender l'IoT. Certains le considèrent comme un réseau d'objets physiques disposant d'identifiants uniques, capables de communiquer et de collecter des données. D'autres le perçoivent comme un système étendu, englobant des dispositifs**,** des machines et même des personnes**,** tous interagissant et partageant des informations [3].

De manière générale, l'IoT repose sur des objets intelligents interconnectés qui échangent des données via Internet. Ces données peuvent être analysées pour améliorer le développement de produits, la fourniture de services et les processus de maintenance**.** Il s'agit fondamentalement d'un monde dans lequel les objets interagissent entre eux et avec leur environnement pour atteindre des objectifs spécifiques [4].

Les éléments clés de l'IoT incluent les objets intelligents, la connectivité Internet, le partage de données et l'interconnexion. Cette technologie permet une connectivité « à tout moment, en tout lieu, pour toute personne », transformant profondément notre manière de vivre et d'interagir avec le monde [5].

Au-delà de la collecte de données, l'IoT permet le contrôle à distance des objets connectés, estompant la frontière entre le monde physique et le monde numérique**.** Il s'agit d'un réseau qui relie non seulement des objets, mais aussi des personnes, leur permettant de partager des informations et des données comme jamais auparavant [6].

La définition la plus complète envisage l'IoT comme une infrastructure mondiale intégrant les objets physiques au monde virtuel**.** Ces objets deviennent intelligents et capables de prendre des décisions en fonction des données collectées [7]. Cela va de pair avec le développement de l'informatique en nuage (cloud computing) et l'expansion de la capacité d'adressage de l'Internet [8].

En résumé, cette compréhension élargie met en évidence l'interdépendance entre les objets du quotidien, les individus et les dispositifs intelligents, tous œuvrant à la création d'un monde plus intelligent et interconnecté [9].

1.2.1.2.Architecture

L'architecture de l'Internet des objets repose généralement sur les modèles classiques de communication de l'Internet, notamment le modèle TCP/IP et le modèle OSI.[10] Elle se structure essentiellement autour de trois couches principales, chacune remplissant une fonction spécifique dans le processus de collecte, de traitement et d'utilisation des données.[11]

- La couche de perception constitue la base de l'architecture IoT. Elle est composée principalement d'objets connectés tels que les étiquettes RFID, les dispositifs de géolocalisation GPS, les capteurs ou encore divers terminaux intelligents. Cette couche

est chargée de collecter les données issues de l'environnement réel et de les transmettre à la couche supérieure sous forme de signaux numériques.

- La couche réseau, aussi appelée couche de transport et de traitement, a pour rôle d'assurer la transmission et le traitement des informations issues de la couche de perception. Grâce à des technologies de communication telles que le Wi-Fi, le Bluetooth, ou encore des protocoles spécifiques comme MQTT, cette couche permet de transporter de manière fiable les données vers la couche application.
- La couche application représente l'interface avec l'utilisateur final. Elle est chargée d'analyser les informations reçues afin de déclencher des actions ou des services selon les besoins. Cela peut inclure, par exemple, l'envoi de courriels**,** l'activation d'une alarme, ou encore le pilotage à distance d'un appareil (allumer, éteindre, etc.).

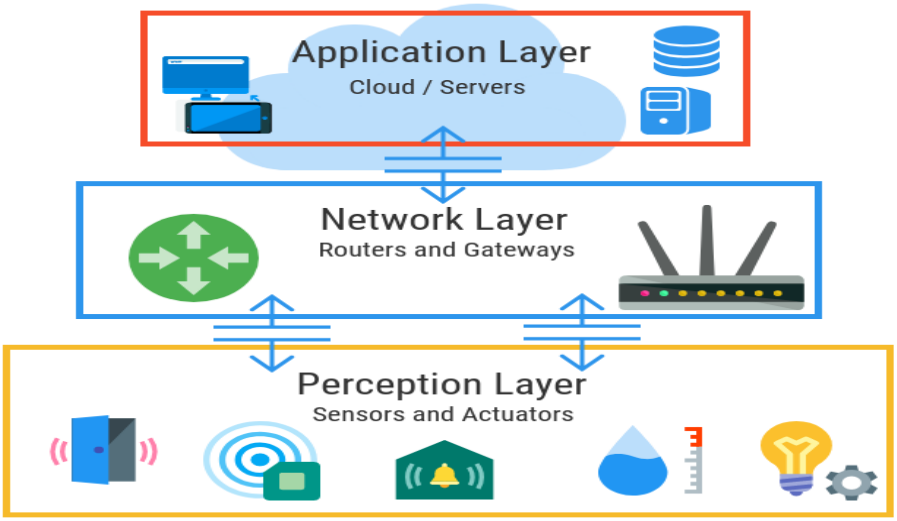


Figure 1.1 : L'architecture de l'Internet des objets

1.2.1.3.Composants

L'écosystème de l'Internet des Objets (IoT) repose sur un ensemble coordonné de composants technologiques, allant des objets physiques aux plateformes cloud, en passant par les couches de communication et de traitement intermédiaires. Ces composants interagissent pour assurer la collecte, la transmission, le traitement et **l'**exploitation intelligente des données issues de l'environnement réel.

- **Objets connectés (nœuds physiques ou terminaux)**

Les objets connectés constituent les éléments terminaux de l'architecture IoT. Ils sont conçus pour capturer des données à partir de leur environnement immédiat. Ces objets peuvent être nativement connectés (par exemple, une montre intelligente) ou rendus connectés grâce à l'ajout de modules spécifiques.

- **Capteurs et actionneurs**

Les capteurs sont les éléments de base permettant à l'objet connecté d'interagir avec le monde physique. Ils mesurent des grandeurs environnementales (température, pression, gaz, lumière) et les transforment en données exploitables. Les actionneurs, quant à eux, permettent de traduire les décisions prises en actions physiques (mouvements, ouverture de vannes, alertes sonores, etc.).

- **Modules de communication**

Les modules de communication sont responsables de la transmission des données collectées vers d'autres entités (passerelles, cloud). Ces modules peuvent être filaires ou sans fil, et le choix du protocole dépend fortement du contexte d'utilisation, des contraintes énergétiques, de la bande passante et de la portée.

- **Passerelles (Gateways)**

Les passerelles sont des nœuds intermédiaires qui facilitent la communication entre les objets connectés et les serveurs distants (cloud ou fog). Elles peuvent aussi effectuer un prétraitement local des données, convertir des protocoles hétérogènes, et assurer la sécurité des échanges.

- **Plateformes (Cloud, Edge et Fog)**

Les plateformes de traitement jouent un rôle central dans l'IoT moderne. Le cloud computing assure le stockage massif des données et l'exécution d'algorithmes complexes (notamment d'apprentissage automatique). Le fog computing introduit un niveau intermédiaire de traitement distribué, et le edge computing pousse les capacités de calcul jusqu'aux objets ou passerelles.

1.2.2. Protocoles IoT

1.2.2.1.HTTP (HyperText Transfer Protocol)

Le protocole HTTP constitue le pilier du modèle de communication client–serveur utilisé sur le Web. Il bénéficie d'une compatibilité particulièrement élevée avec les infrastructures réseau existantes, ce qui facilite son intégration dans de nombreux systèmes connectés. À ce jour, la version la plus largement déployée demeure HTTP/1.1. Dans le contexte de l'IoT, HTTP est souvent utilisé dans des architectures reposant sur le paradigme REST (Representational State Transfer). La figure 1.2 illustre un exemple d'interaction REST, où les échanges entre clients et ressources se font à travers des requêtes HTTP standardisées.

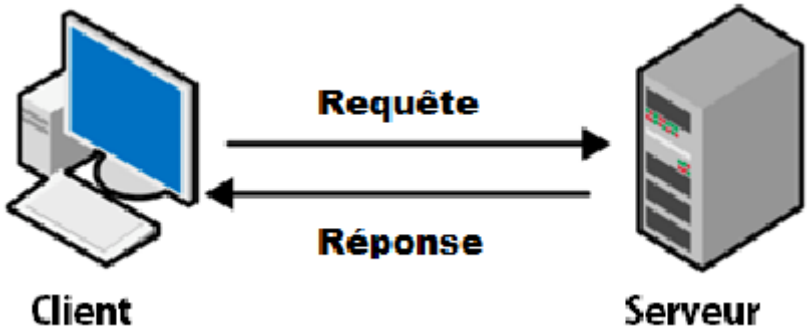


Figure 1.2 : Modèle d'interaction REST http

HTTPS représente la version sécurisée du protocole HTTP, intégrant un chiffrement via TLS (Transport Layer Security). L'un des inconvénients majeurs de l'utilisation de HTTP dans le contexte de l'IoT réside dans le fait que les dispositifs contraints transmettent généralement de petites quantités de données et ce, de façon sporadique. Or, l'établissement d'une connexion TCP, nécessaire pour HTTP, engendre un temps de latence ainsi que des frais généraux superflus, peu adaptés à ce type de communication intermittente.[12]

1.2.2.2.CoAP (Constrained Application Protocol)

Le Constrained Application Protocol (CoAP) est un protocole développé par l'IETF (Internet Engineering Task Force) afin de répondre aux besoins spécifiques des dispositifs contraints en ressources au sein des environnements IoT. Il s'agit d'un protocole Web reposant sur une architecture client/serveur, à l'image de HTTP. Toutefois, contrairement à ce dernier, CoAP s'appuie sur la pile UDP/IPv6/6LoWPAN, ce qui permet des échanges de messages allégés et mieux adaptés aux contraintes des réseaux à faible bande passante et des objets à capacité

limitée. À l'instar de HTTP, CoAP a été conçu selon le modèle REST**,** ce qui facilite son intégration avec les architectures Web existantes tout en réduisant la complexité des communications.

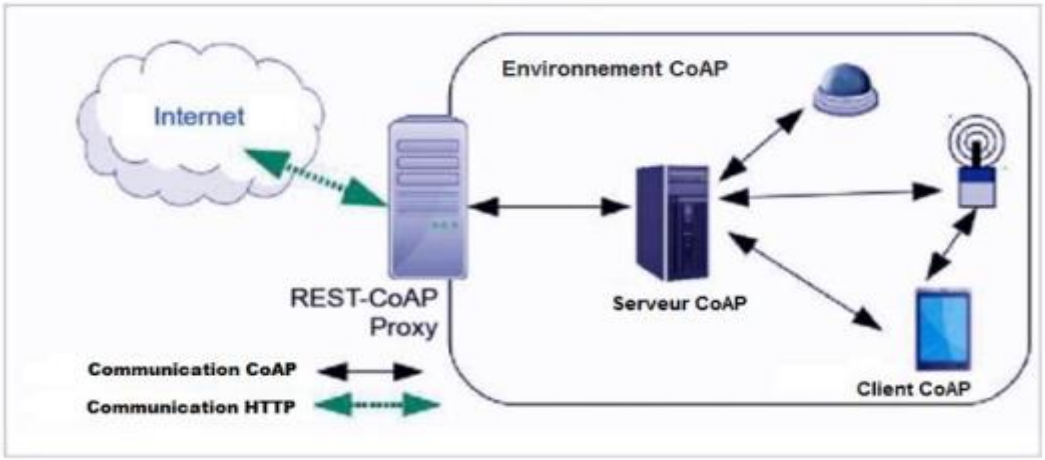


Figure 1.3 : Fonctionnalités CoAP [13]

CoAP intègre son propre mécanisme de fiabilité, compensant l'absence de garanties inhérentes au protocole UDP sur lequel il repose. Pour cela, il définit quatre types de messages distincts : les messages confirmables (nécessitant un accusé de réception), les non-confirmables, les accusés de réception, ainsi que les messages de réinitialisation (reset).
Sur le plan de la sécurité, CoAP s'appuie sur DTLS (Datagram Transport Layer Security), une adaptation du protocole TLS spécifiquement optimisée pour les environnements contraints comme ceux de l'IoT. Cette version allégée permet de garantir la confidentialité, l'authenticité et l**'**intégrité des communications, tout en respectant les limites des objets connectés à faibles ressources [14].

1.2.2.3.XMPP (eXtensible Messaging and Presence Protocol)

Le XMPP (eXtensible Messaging and Presence Protocol) est un protocole standard ouvert de messagerie, normalisé par l'IETF. Conçu initialement pour la messagerie instantanée et l'échange de données entre applications, il repose sur un format textuel structuré en XML. XMPP offre une communication flexible, prenant en charge aussi bien le modèle client–serveur qu'un mécanisme de publication/abonnement, et il s'exécute au-dessus du protocole TCP. Dans le cadre de l'Internet des objets, XMPP s'avère particulièrement adapté pour l'échange de messages en temps réel. Il fournit également une gestion efficace de la présence, permettant de

suivre l'état de disponibilité ou d'activité des dispositifs connectés, ce qui constitue un atout important pour de nombreuses applications IoT.

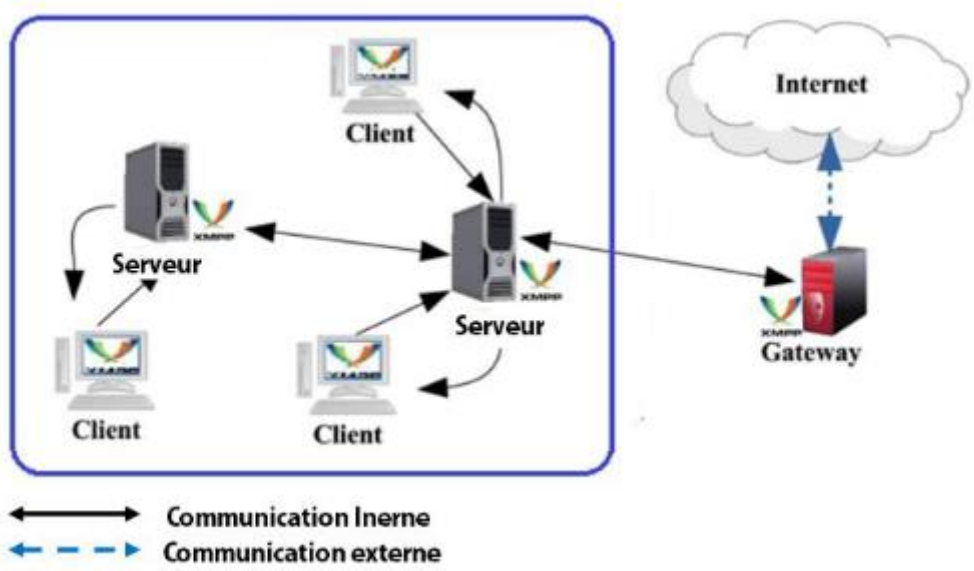


Figure 1.4 : Communication dans XMPP [15]

Une version spécialisée du protocole, appelée XMPP-IoT, a été développée afin de répondre aux exigences spécifiques de l'Internet des objets. Elle dispose de nombreuses implémentations dans divers langages de programmation, accessibles via des API dédiées. Les extensions récentes des spécifications XMPP intègrent par ailleurs des fonctionnalités avancées en matière de sécurité, d'authentification, de confidentialité et de contrôle d'accès. En plus du chiffrement TLS, XMPP prend également en charge le mécanisme SASL (Simple Authentication and Security Layer), qui assure notamment l'authentification du serveur. Grâce à cet ensemble de mécanismes robustes, XMPP est aujourd'hui considéré comme l'un des protocoles de messagerie les plus sûrs dans les environnements IoT[16].

1.2.2.4.DDS (Data Distribution Service)

Le Data Distribution Service (DDS) est une norme dédiée à l'interopérabilité des données en temps réel, reposant sur un modèle d'interaction de type publication/abonnement**.** Contrairement à plusieurs protocoles orientés messagerie, DDS adopte une architecture décentralisée, favorisant une communication peer-to-peer centrée sur l'échange direct de données plutôt que sur l'envoi de messages. Par défaut, DDS s'appuie sur le protocole UDP,

mais il peut également fonctionner sur TCP lorsque les exigences applicatives l'imposent. L'une de ses principales forces réside dans sa grande évolutivité, rendue possible notamment grâce à la découverte dynamique des participants et des ressources au sein du réseau. Sur le plan sécuritaire, DDS intègre plusieurs mécanismes, incluant l'utilisation de SSL/TLS, afin de garantir la confidentialité, l'intégrité et l'authentification des échanges. Par ailleurs, la qualité de service (QoS) est assurée par un ensemble de politiques permettant de contrôler finement le comportement du système, aussi bien lors de la réception des données que lors de leur disponibilité.[16].

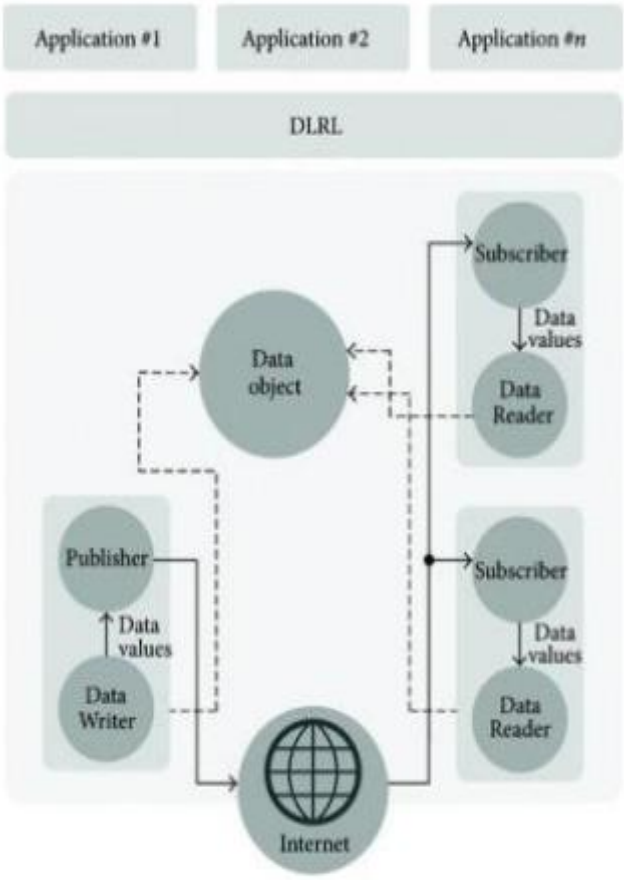


Figure 1.5 : Architecture conceptuelle du DDS [16]

1.2.2.5.MQTT (Message Queuing Telemetry Transport)

Le protocole MQTT, développé à l'origine par IBM, est aujourd'hui largement utilisé dans l'écosystème IoT en raison de sa simplicité et de la taille extrêmement réduite de ses en-têtes, nettement plus compacte que celle d'autres protocoles de messagerie. Il est ainsi fréquemment recommandé comme solution privilégiée pour assurer la communication au sein des réseaux

IoT. MQTT repose sur un modèle de communication publish/subscribe, dans lequel un courtier (broker) gère la distribution des messages entre éditeurs et abonnés. Son fonctionnement s'appuie sur la pile TCP/IP, ce qui lui confère une fiabilité de transport tout en maintenant une grande légèreté. Cette légèreté se manifeste par un nombre restreint de types de messages et par la petite taille des trames, incluant notamment un en-tête fixe de seulement 2 octets. Le protocole offre également trois niveaux de qualité de service (QoS 0, **1** et 2)**,** permettant d'ajuster la fiabilité de la transmission en fonction des exigences applicatives. Enfin, la sécurité des communications peut être renforcée grâce à l'intégration du chiffrement TLS**,** qui assure la confidentialité, l'intégrité et l'authentification des échanges [17].

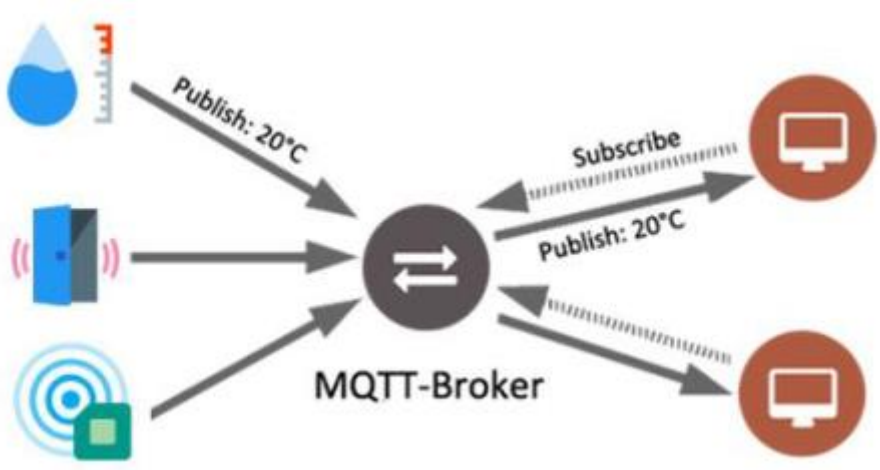


Figure 1.6 : L'architecture du protocole MQTT

1.2.3. Contraintes des réseaux IoT

À l'ère de l'hyperconnectivité, les réseaux IoT connaissent une croissance fulgurante, avec des milliards de dispositifs appelés à y être interconnectés dans un avenir proche. Cette prolifération massive des objets connectés engendrera une augmentation considérable du trafic réseau, accompagnée d'une génération exponentielle de données [18]. Cette surcharge aura un impact direct sur la performance globale du réseau, en affectant des aspects critiques tels que le routage, la consommation d'énergie, la qualité de service, la sécurité ou encore la fiabilité des communications. Face à ces enjeux, il est impératif de concevoir des solutions innovantes et efficaces permettant d'optimiser les performances des réseaux IoT. Parmi les principales contraintes identifiées, on peut citer [19]**:**

1.2.3.1.Le routage des données

Le routage dans un environnement IoT est un processus crucial visant à acheminer les données du point de collecte vers la destination appropriée. Ce processus repose sur la sélection de chemins optimaux, en tenant compte de plusieurs critères : la durée du trajet, la consommation énergétique, la longueur du chemin, le niveau de sécurité, ou encore une combinaison de ces facteurs. Un mauvais choix de routage peut non seulement entraîner des pertes d'informations, mais aussi épuiser inutilement les ressources des nœuds du réseau.

1.2.3.2.La gestion de l'énergie

La majorité des dispositifs IoT fonctionnent sur batterie et sont déployés dans des environnements difficiles d'accès. Pour garantir leur longévité, des stratégies d'économie d'énergie s'imposent. Cela passe notamment par l'implémentation de mécanismes de mise en veille programmée (sommeil-éveil), de protocoles de communication à faible consommation, et d'algorithmes d'optimisation énergétique au niveau des capteurs et des passerelles.

1.2.3.3.Le contrôle de la congestion

L'augmentation du nombre de dispositifs connectés génère une forte pression sur l'infrastructure réseau, provoquant des phénomènes de congestion. Ces derniers se traduisent par une dégradation des performances : retards de transmission, augmentation du taux de perte de paquets, détérioration de la qualité des communications et vulnérabilités en matière de sécurité. Il est donc essentiel de mettre en œuvre des mécanismes dynamiques de gestion et de contrôle de la congestion, adaptés aux caractéristiques des réseaux IoT.

1.2.3.4.L'hétérogénéité

Les réseaux IoT sont composés d'un large éventail de dispositifs – capteurs, actionneurs, objets intelligents – aux caractéristiques variées (protocoles, puissances, types de données, formats, etc.). Cette diversité impose la mise en place de solutions d'interopérabilité et de gestion des données capables d'unifier les différents flux pour un traitement cohérent et efficace, malgré les différences de nature et de structure.

1.2.3.5.L'évolutivité

L'une des caractéristiques fondamentales d'un réseau IoT est sa capacité à évoluer, c'est-à-dire à intégrer facilement de nouveaux objets sans compromettre sa stabilité ni sa performance. Cette évolutivité représente un défi technique majeur, car elle exige une architecture flexible, des protocoles adaptatifs, ainsi qu'une allocation intelligente des ressources réseau.

1.2.3.6.La sécurité

La sécurité constitue une préoccupation majeure dans les réseaux IoT. Les données sensibles transmises à travers ces réseaux doivent être protégées contre toute forme d'interception, de falsification ou de perte. Cela nécessite la mise en œuvre de mécanismes robustes de chiffrement, d'authentification, de contrôle d'accès et de détection des intrusions, en tenant compte des ressources limitées des dispositifs impliqués.

1.2.3.7.La qualité de service (QoS)

La QoS dans les réseaux IoT repose sur plusieurs indicateurs de performance : bande passante, latence, gigue, taux de perte de paquets, etc. Maintenir un niveau de qualité satisfaisant est essentiel, surtout pour les applications critiques (santé, industrie, transport). Il convient donc de développer des algorithmes intelligents de gestion des ressources pour garantir une QoS adéquate, même en cas de conditions réseau fluctuantes.

1.2.4. Défis de l'internet des objets

L**'**Internet des objets (IoT) repose sur l'interconnexion à grande échelle de dispositifs intelligents capables de collecter, échanger et traiter continuellement d'importantes quantités de données. Lorsqu'elles sont analysées et exploitées de manière efficace, ces données constituent une source d'information stratégique permettant la mise en œuvre d'applications contextuelles, c'est-à-dire capables d'adapter dynamiquement leurs services en fonction des situations, des conditions environnementales et des besoins des utilisateurs. Cependant, malgré cette perspective prometteuse visant à rendre les environnements plus intelligents, réactifs et efficaces, le déploiement généralisé de l'IoT s'accompagne de nombreux défis techniques, opérationnels et sécuritaires. Comme toute technologie émergente, l'IoT doit surmonter ces obstacles afin de garantir un fonctionnement fiable**,** évolutif et sécurisé des systèmes connectés. La section suivante présente les principaux défis inhérents à l'écosystème IoT**.**

1.2.4.1.L'hétérogénéité des systèmes IoT

Les dispositifs, plateformes et applications IoT se caractérisent par une forte hétérogénéité**,** aussi bien sur le plan matériel que logiciel. À ce jour, il n'existe pas de norme universelle garantissant une interopérabilité complète entre les différents systèmes IoT, ce qui complique considérablement les processus d**'**intégration, de maintenance et d'évolution des solutions déployées. Chaque type d'objet connecté repose sur des protocoles de communication distincts et présente des capacités variables en termes de puissance de calcul, de stockage, de bande passante ou encore de consommation énergétique. De plus, ces dispositifs évoluent souvent dans des environnements hétérogènes, avec des exigences spécifiques relatives au débit de communication, à la latence ou à la durabilité énergétique. Dans ce contexte, la diversité technologique constitue un obstacle majeur à la mise en place d'un écosystème IoT homogène. Il devient donc indispensable de promouvoir l'adoption de standards et de protocoles communs, afin d'assurer une interopérabilité fluide, une cohabitation harmonieuse et une communication efficace entre les différents objets connectés.

1.2.4.2.Les limitations en ressources

La majorité des objets connectés sont conçus pour être légers**,** autonomes et économes en énergie, ce qui implique des limitations significatives en matière de ressources**.** Ces dispositifs disposent généralement d'une capacité de calcul restreinte**,** d'une mémoire réduite, d'une autonomie énergétique limitée**,** ainsi que d'une bande passante étroite**.** Ces contraintes exigent la conception et le déploiement de protocoles de communication optimisés, à la fois peu énergivores et faibles en surcharge computationnelle, afin de minimiser la consommation des ressources tout en assurant un fonctionnement fiable et une qualité de service acceptable pour l'utilisateur final. Le développement de solutions adaptées à ces restrictions représente un enjeu majeur pour l'essor de l'IoT, en particulier dans les domaines à fortes contraintes tels que la santé connectée, l'agriculture intelligente ou les systèmes embarqués, où la durabilité et la performance doivent coexister dans des environnements à ressources limitées.

1.2.4.3.La fiabilité

Les systèmes IoT sont exposés à de multiples vulnérabilités pouvant compromettre leur fiabilité. Ces défaillances techniques, qu'elles soient matérielles ou logicielles, peuvent entraîner des dysfonctionnements majeurs. Cette problématique devient particulièrement

critique lorsqu'il s'agit d'applications sensibles, telles que les systèmes de soins de santé, de production industrielle ou de transport, où un incident peut engendrer des conséquences graves, voire irréversibles. Il est donc indispensable de développer des solutions fiables, capables de fonctionner correctement en toutes circonstances, et de résister aux pannes, perturbations et conditions extrêmes.

1.2.4.4.Le nombre croissant d'objets connectés

L'Internet des Objets est caractérisé par une croissance exponentielle du nombre de dispositifs connectés à Internet. Depuis 2010, le nombre d'objets connectés a dépassé celui des êtres humains. Cette prolifération massive soulève de nouveaux défis d'évolutivité auxquels les protocoles et architectures IoT doivent s'adapter. Il devient impératif de concevoir des infrastructures capables de gérer efficacement un très grand nombre d'entités connectées, tout en maintenant la qualité de service et la réactivité du système. Parmi les aspects concernés figurent la gestion des ressources hétérogènes, la représentation numérique des objets physiques (digital twins)**,** ainsi que l'interconnexion transparente d'un grand volume de dispositifs dans des environnements complexes et dynamiques.

1.2.4.5.L'interopérabilité

L'interopérabilité est une condition essentielle à la réussite de l'IoT. Elle désigne la capacité des systèmes et dispositifs à communiquer entre eux, malgré des technologies, protocoles ou plateformes différentes. Or, dans la pratique, chaque industrie tend à développer ses propres solutions techniques, souvent propriétaires, rendant difficile la compatibilité entre systèmes. Cette fragmentation limite la synergie entre les applications et freine le potentiel d'innovation intersectorielle. Il devient alors crucial de normaliser les interfaces, les protocoles et les modèles de données afin de permettre une communication transparente entre systèmes hétérogènes. Cela est particulièrement important pour les applications traversant plusieurs domaines (santé, transport, logistique, etc.), où la coopération entre organisations et systèmes divers est indispensable.

1.2.4.6.La sécurité et la confidentialité

La sécurité et la protection de la vie privée constituent aujourd'hui des défis techniques et éthiques majeurs dans le domaine de l'Internet des objets. Les dispositifs IoT sont fréquemment

déployés dans des environnements ouverts, peu contrôlés ou difficilement accessibles, ce qui accroît leur exposition aux attaques, qu'elles soient physiques ou logiques. Dans ce contexte, il devient indispensable de mettre en place des mécanismes de sécurité avancés permettant d'empêcher les accès non autorisés, de préserver la confidentialité des données et d'assurer la fiabilité des communications entre les différentes entités du système. Au cours des dernières années, la sécurisation des réseaux IoT s'est imposée comme un axe prioritaire de recherche. Les principaux enjeux concernent notamment l'authentification des objets, l'intégrité des messages échangés et la protection des données personnelles. Ces exigences varient également selon les domaines d'application. Dans les environnements commerciaux, par exemple, les dispositifs IoT doivent être protégés contre l'espionnage industriel et les fuites d'informations sensibles ou stratégiques.

Les défis spécifiques à la sécurité et à la confidentialité dans l'IoT peuvent être regroupés comme suit :

- Sécurisation de l'architecture IoT : le développement rapide et la diversité des objets connectés rendent complexe la tâche des administrateurs pour garantir une gestion sécurisée des dispositifs. Cela incite les chercheurs à proposer de nouvelles approches de sécurisation dès la phase de conception et tout au long du cycle de vie des objets.
- Sécurisation du réseau : il s'agit ici de détecter proactivement les anomalies de trafic, d'identifier les attaques (DDoS, Man-in-the-Middle, malwares, etc.) et de protéger l'ensemble de l'écosystème IoT contre ces menaces. La résilience des réseaux face aux attaques est une condition indispensable à la fiabilité des systèmes.
- Protection de la vie privée des utilisateurs : cela concerne notamment le contrôle des emplacements physiques, des données personnelles collectées, et la nécessité d'implémenter des mécanismes de respect de la vie privée, soutenus par des réglementations appropriées, des normes internationales**,** ainsi que des technologies de gestion sécurisée de l'identité des objets et des utilisateurs.

1.2.5. Domaines d'applications de l'IoT

Étant donné que l'Internet des objets (IoT) se définit par des appareils ou objets interconnectés à un réseau, ses applications sont extrêmement variées. Les responsables de diverses entreprises et organisations reconnaissent le potentiel transformateur de cette technologie et investissent de plus en plus dans ces dispositifs afin d'en tirer des bénéfices stratégiques et opérationnels.[20]

1.2.5.1.Santé

L'Internet des objets médicaux (IoTM) constitue un sous-domaine émergent qui transforme profondément la manière dont les soins de santé sont dispensés grâce à l'intégration de technologies IoT (Joyia). L'adoption de l'IoT dans le secteur médical a connu une progression notable et demeure un champ à fort potentiel de croissance. Des innovations marquantes, telles que le moniteur cardiaque AliveCor reposant sur des capteurs IoT, démontrent clairement l'impact de ces technologies, notamment dans des situations critiques où elles contribuent directement à sauver des vies. Historiquement, les avancées technologiques ont toujours joué un rôle structurant dans le domaine médical, et les dispositifs IoT y occupent aujourd'hui une place de plus en plus centrale. L'une de leurs applications majeures réside dans la surveillance à distance des patients, permettant un suivi continu de l'état de santé à domicile tout en limitant la nécessité d'hospitalisations prolongées. Les données recueillies par ces dispositifs peuvent être analysées afin de prédire rapidement l'apparition de pathologies et d'améliorer la prise de décision clinique. Durant la pandémie de COVID-19, les capteurs IoT se sont révélés particulièrement précieux en facilitant la surveillance en temps réel des paramètres physiologiques critiques, permettant aux professionnels de santé d'intervenir rapidement en cas de variation significative. L'utilisation de capteurs représente un élément fondamental dans la prestation des services de santé. Dans les dispositifs médicaux, ces capteurs constituent une interface essentielle entre le monde physique et les systèmes d'information, en assurant la collecte d'un large ensemble de données cliniques. Ils offrent aux professionnels de santé la possibilité de surveiller en continu des paramètres physiologiques clés, indispensables pour évaluer l'état du patient et prendre les mesures adéquates. Intégrés dans des dispositifs connectés, les capteurs médicaux permettent de mesurer des informations variées telles que la température corporelle, la fréquence respiratoire, le rythme cardiaque, le poids, la conductance cutanée, la réponse galvanique, le flux sanguin ou la saturation en oxygène (SpO_2), la glycémie, l'activité musculaire ou encore les mouvements corporels. Reliés à des réseaux de capteurs sans fil (Wireless Sensor Networks WSN), ils transmettent ces données à divers acteurs du secteur de la santé patients, personnels médicaux, établissements hospitaliers, assureurs, entre autres. Le champ d'application des capteurs médicaux est particulièrement vaste, couvrant des dispositifs critiques tels que les moniteurs ECG, les systèmes de mesure du taux de glucose ou les capteurs d'oxygénation sanguine. De manière générale, toutes les technologies intégrées dans le domaine de la santé visent à améliorer les résultats cliniques, et les dispositifs IoT constituent aujourd'hui un levier majeur dans cette dynamique. Les avancées récentes dans les

technologies IoT continueront d'influencer profondément les systèmes de santé, favorisant la création d'infrastructures plus robustes, plus réactives et davantage orientées vers la prévention. À mesure que ces technologies se perfectionnent, la dépendance du secteur sanitaire aux dispositifs connectés et aux données qu'ils génèrent ne cessera de croître.[21]

1.2.5.2.Agriculture

Avec la croissance rapide de la population mondiale, la mise en place de systèmes de distribution alimentaire plus efficaces est devenue un enjeu majeur, stimulant de manière significative les avancées dans le domaine de l'agriculture intelligente. À cette pression démographique s'ajoutent des facteurs tels que le changement climatique et la raréfaction des ressources en eau, qui renforcent la nécessité de développer des solutions agricoles plus durables et performantes. Une grande partie des technologies IoT appliquées à l'agriculture vise précisément à relever ces défis en réduisant le gaspillage des ressources, comme l'illustre la Figure. L'intégration de l'IoT dans les environnements agricoles s'impose désormais comme un levier stratégique pour optimiser la gestion des opérations et constitue l'un des domaines d'application les plus dynamiques de cette technologie. Les chaînes d'approvisionnement alimentaires, essentielles pour garantir la qualité et la quantité de la production, bénéficient particulièrement des systèmes IoT, qui en améliorent l'efficacité, la traçabilité et la résilience. Ainsi, la quête de solutions plus performantes pour la distribution alimentaire a largement favorisé l'adoption de l'IoT dans le secteur agricole, les acteurs ayant constaté les bénéfices tangibles offerts par ces technologies. Parmi les applications les plus prometteuses de l'IoT en agriculture figure l'automatisation, qui permet aux dispositifs connectés de réagir automatiquement à diverses conditions environnementales sans intervention humaine. Cette capacité améliore considérablement l'efficacité des opérations. Les réseaux de capteurs sans fil (Wireless Sensor Networks – WSN) jouent un rôle central dans ce processus, en fournissant en continu les données nécessaires pour permettre aux dispositifs IoT de prendre des décisions adaptées en temps réel. Ce principe est particulièrement pertinent dans les exploitations agricoles de grande envergure, où l'étendue des surfaces à gérer exige des mécanismes optimisés pour maximiser les rendements. Par exemple, des capteurs d'humidité du sol peuvent être déployés sur les parcelles afin de mesurer en continu la teneur en eau et ainsi alimenter des systèmes d'irrigation intelligents. Lorsque le niveau d'humidité atteint un seuil critique, les dispositifs IoT peuvent émettre une alerte et activer automatiquement le système d'arrosage, garantissant une gestion rationnelle de l'eau tout en préservant la santé des cultures. Ainsi, la

surveillance en temps réel et les réponses automatisées constituent des éléments clés pour comprendre la manière dont l'IoT modernise et optimise les pratiques agricoles, en améliorant à la fois la gestion des ressources et la productivité globale.[21]

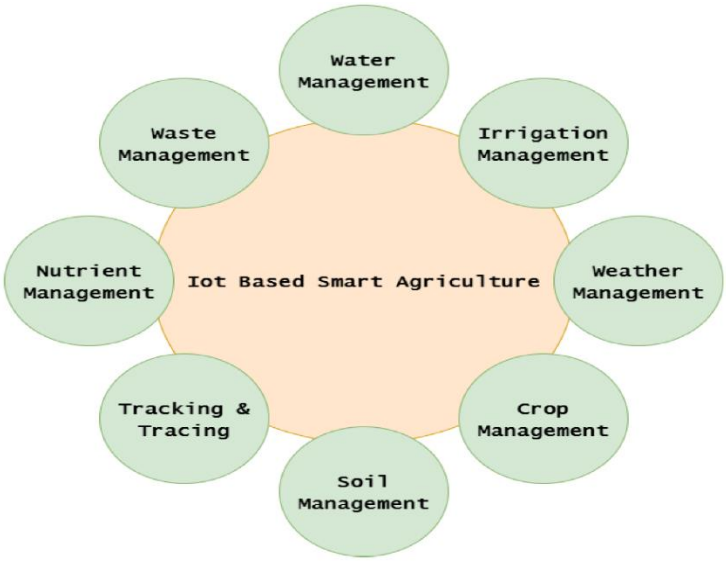


Figure 1.7 : Différents types d'applications agricoles pour l'IoT

1.2.5.3. Maison intelligente

Les applications de maison connectée constituent aujourd'hui l'un des domaines d'usage les plus prometteurs de l'IoT, offrant aux utilisateurs un ensemble croissant de services intelligents. L'émergence de ces systèmes s'inscrit dans une évolution amorcée dès les années 1970 avec l'apparition du protocole X10, première technologie permettant une communication entre des appareils domestiques automatisés. Depuis, la domotique a connu une transformation majeure, portée par l'intégration progressive de dispositifs IoT capables de mesurer l'environnement domestique, de gérer les appareils électroménagers ou encore de contrôler l'accès au domicile. Dans ce contexte, la maison connectée représente un espace privilégié où la technologie IoT est largement déployée. De nombreux appareils électroménagers peuvent désormais être commandés à distance ou automatisés grâce à des capteurs et des modules de communication intégrés, améliorant à la fois l'efficacité énergétique, le confort et la sécurité. Ainsi, les avantages offerts par ces systèmes dépassent largement la simple notion de commodité. Par exemple, l'intégration de capteurs IoT peut faciliter le quotidien des personnes âgées ou à mobilité réduite, notamment en permettant l'activation d'appareils difficiles d'accès ou en assurant la détection automatique des chutes à l'aide de capteurs au sol ou de caméras

intelligentes. La croissance du marché des maisons connectées est également soutenue par la popularité grandissante d'appareils intelligents grand public, tels que les enceintes interactives proposées par Amazon ou Google, qui servent d'interfaces de contrôle centralisées pour de nombreux équipements IoT. Diverses technologies contribuent à rendre ces environnements intelligents plus performants, parmi lesquelles les systèmes d'identification par radiofréquence (RFID) jouent un rôle particulièrement important. La RFID constitue une technologie clé permettant d'identifier des objets, d'enregistrer des données ou encore de contrôler des éléments spécifiques à l'aide d'ondes radio. Les dispositifs RFID présentent une grande polyvalence et trouvent des applications dans des contextes variés. Dans le secteur de l'éducation, par exemple, ils peuvent être intégrés aux cartes d'identité étudiantes pour faciliter l'accès aux bâtiments ou assurer le suivi des présences. Dans le domaine de l'assistance aux personnes âgées, les systèmes RFID sont également utilisés pour analyser les déplacements en intérieur, offrant ainsi des informations précieuses pour comprendre l'état de santé de personnes vivant seules et détecter précocement des situations anormales.[21]

1.2.5.4.Ville intelligente

Les dispositifs de l'Internet des objets jouent un rôle central dans le développement des villes intelligentes, en offrant une large gamme d'applications destinées à améliorer la gestion urbaine. Une ville intelligente peut être définie comme un environnement urbain doté de technologies avancées notamment des réseaux de capteurs sans fil et des actionneurs capables de collecter et de transmettre des données essentielles à la prise de décision dans les opérations municipales. Ces systèmes se caractérisent par une grande complexité, liée au nombre élevé d'appareils déployés, à la diversité des technologies de liaison utilisées et à la multiplicité des services qu'ils supportent. Le concept de ville intelligente repose sur l'intégration de réseaux de détection, d'infrastructures hétérogènes et de systèmes de traitement de l'information qui fonctionnent conjointement pour améliorer différents aspects de la vie urbaine. L'adoption de technologies IoT dans ce contexte vise principalement à renforcer la qualité de vie des citoyens, en optimisant les services municipaux tout en tenant compte de la consommation des ressources naturelles. L'objectif fondamental des villes intelligentes est ainsi d'exploiter les données collectées par les dispositifs IoT afin d'améliorer l'efficacité des services urbains, de faciliter la prise de décision et de promouvoir une gestion durable des ressources. Parmi les applications les plus importantes de l'IoT dans les villes intelligentes, la gestion du trafic occupe une place stratégique. Les zones métropolitaines étant souvent marquées par une forte densité de

population, la congestion routière constitue un défi majeur. Les villes intelligentes s'appuient donc sur les technologies de l'information et de la communication pour mettre en œuvre une gestion dynamique du flux de circulation. Les systèmes de transport intelligents (STS) reposent sur la collecte en temps réel de données, nécessitant l'intervention d'appareils IoT capables de capter rapidement les informations relatives au trafic et de les transmettre pour traitement. Dans ces systèmes, différents types de capteurs peuvent être intégrés directement dans les routes afin de détecter le passage des véhicules tous les 500 ou 1 000 mètres. Parallèlement, des caméras équipées de techniques avancées de traitement d'images numériques appliquent des algorithmes permettant d'estimer la densité du trafic ; ces informations sont ensuite utilisées pour adapter les politiques de gestion de circulation. Outre l'optimisation du trafic, les systèmes intelligents de gestion routière contribuent également à l'amélioration de la qualité de l'air et à la sécurité des usagers, y compris les personnes âgées, en réduisant les risques liés à la congestion et à l'exposition prolongée aux polluants.[21]

1.2.5.5.L'industrie 4.0

Le secteur manufacturier connaît une transformation profonde avec l'avènement de l'Industrie 4.0, laquelle inaugure une ère de systèmes intelligents, autonomes et interconnectés. L'une des tendances majeures de cette révolution industrielle est l'adoption rapide des dispositifs et capteurs de l'Internet des objets industriels (Industrial Internet of Things IIoT), comme l'illustre la Figure. Ces dispositifs intégrés permettent aux machines, aux équipements et aux produits de collecter et de transmettre en continu des données opérationnelles essentielles. Les données générées par les dispositifs IIoT jouent un rôle déterminant dans la mise en place de stratégies de maintenance prédictive. En analysant les informations issues des équipements en fonctionnement, les fabricants sont en mesure d'anticiper les pannes potentielles, de planifier les interventions techniques et ainsi de réduire significativement les temps d'arrêt non planifiés. Cette approche améliore non seulement l'efficacité opérationnelle, mais contribue aussi à prolonger la durée de vie des machines et à optimiser l'utilisation des ressources. Parallèlement, la cybersécurité est devenue un enjeu prioritaire dans le contexte de l'Industrie 4.0. L'interconnexion croissante des usines, alliée à la dépendance accrue envers les technologies numériques, expose les environnements industriels à un nombre croissant de cybermenaces. Pour y faire face, les industriels investissent massivement dans des mécanismes de sécurité avancés visant à protéger leurs infrastructures critiques et leurs données sensibles. Ces mesures incluent l'adoption de techniques de chiffrement robustes, de protocoles d'authentification

renforcés et de systèmes de détection d'intrusion capables d'identifier rapidement les activités suspectes. L'objectif est d'assurer la confidentialité, l'intégrité et la disponibilité des systèmes industriels, piliers essentiels de la continuité opérationnelle. L'intelligence artificielle (IA) et l'apprentissage automatique (Machine Learning ML) constituent également des leviers fondamentaux dans l'expansion de l'Industrie 4.0. Étant donné le volume considérable de données générées par les dispositifs IIoT, les algorithmes d'IA et de ML offrent des capacités avancées d'analyse permettant d'extraire des informations pertinentes de ces ensembles massifs. Les fabricants exploitent ces technologies pour optimiser les processus de production, renforcer le contrôle qualité et soutenir la prise de décision. Grâce à la détection en temps réel de schémas, d'anomalies et de comportements inhabituels, les systèmes intelligents peuvent ajuster automatiquement les paramètres de fabrication, identifier des défauts ou recommander des améliorations. Cela se traduit par une productivité accrue, une meilleure qualité des produits et une réduction notable des coûts opérationnels.[21]

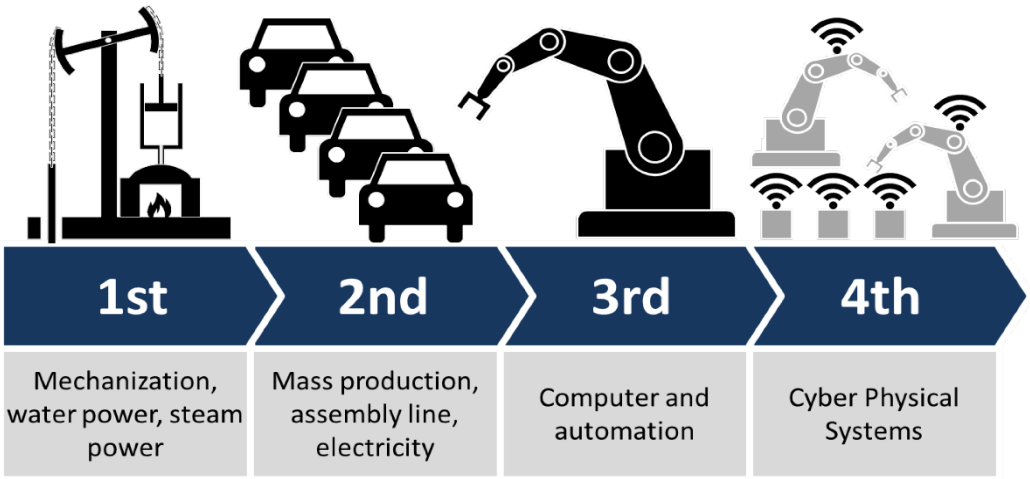


Figure 1.8 : Industrie 4.0

1.3.SYSTÈME DE DÉTECTION D'INTRUSION

1.3.1. Définition Système de Détection d'Intrusion

Un système de détection d'intrusion **(**Intrusion Detection System – IDS) est un ensemble de composants matériels et logiciels interconnectés, dont l'objectif principal est d'assurer une surveillance continue**,** en temps réel et de manière transparente, du trafic réseau ou de l'activité d'un système informatique. Il vise à identifier toute action suspecte ou non autorisée susceptible de compromettre la confidentialité (protection et accès exclusif aux données), l'intégrité

(prévention des modifications illicites) ou la disponibilité (maintien permanent de l'accès aux services) des ressources informatiques.

1.3.2. Fonctionnement et rôle d'un Système de Détection d'Intrusion

La collecte d'informations à partir de points critiques d'un réseau constitue l'un des principes fondamentaux de la détection d'intrusions. Ces données sont ensuite analysées selon des règles prédéfinies afin d'identifier la présence éventuelle d'activités malveillantes et de classifier les attaques en cours. Ce processus est communément appelé système de détection d'intrusion (Intrusion Detection System, IDS). [22]

Les sources de données présentes au sein des réseaux renferment généralement des informations essentielles, telles que les modifications apportées à des fichiers sensibles, l'exécution de programmes inhabituels ou l'observation du trafic réseau. Une fois collectées, ces données sont soumises à une phase de traitement et d'analyse, souvent appelée analyse de l'information, dont l'objectif est d'identifier de potentielles intrusions ou comportements anormaux. Pour cela, plusieurs approches peuvent être mobilisées, parmi lesquelles les techniques d'exploration de données, l'appariement de motifs, ainsi que différentes méthodes d'apprentissage automatique intégrées, permettant d'identifier des schémas anormaux ou suspects au sein du système. La réponse à une intrusion détectée dépend des résultats de cette analyse et peut inclure diverses actions de post-traitement, telles que l'archivage des données pour une utilisation ultérieure**,** la reconfiguration des routeurs**,** ou encore d'autres mesures correctives équivalentes. Positionné en aval du pare-feu, le système de détection d'intrusion (IDS) joue un rôle complémentaire essentiel. Alors que le pare-feu filtre le trafic entrant selon des règles statiques, l'IDS surveille en temps réel les flux de données afin de détecter tout comportement anormal. Lorsqu'une activité suspecte est identifiée, il peut agir conjointement avec le pare-feu pour bloquer la connexion concernée, contribuant ainsi à renforcer la sécurité globale du système.[22]

1.3.3. Méthodes de détection dans les systèmes IDS

Il existe plusieurs approches permettant de détecter les intrusions au sein d'un réseau. Parmi les plus courantes figurent les méthodes basées sur les signatures et celles basées sur les anomalies.[23]

1.3.3.1.La détection basée sur signatures

La détection par signatures consiste à reconnaître des modèles d'attaques déjà connus, pour lesquels un ensemble de signatures a été préalablement défini dans une base de données ou une bibliothèque. Lorsqu'un événement réseau correspond à l'un de ces modèles, une alerte est automatiquement déclenchée. Cette technique se révèle particulièrement efficace pour identifier les attaques récurrentes d'un même type et présente généralement un faible taux de faux positifs face. Cependant, sa principale limite réside dans son incapacité à détecter les attaques dites zero-day, c'est-à-dire nouvelles ou inédites, qui ne figurent pas encore dans la base de signatures. Elle dépend donc fortement de l'historique des menaces connues et demeure, de ce fait, vulnérable à des techniques d'attaque encore non répertoriées.

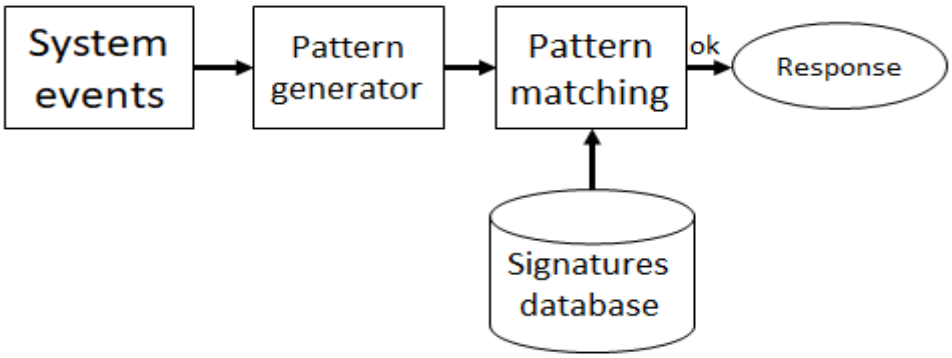


Figure 1.9 : Système de détection basée sur les signatures

1.3.3.2.Détection basée sur les anomalies

La détection des anomalies repose sur l'analyse comportementale globale du système. Elle consiste à définir un modèle représentant le fonctionnement normal en observant les flux de données, qu'ils proviennent de sources internes ou externes au réseau. Ce modèle sert de référence, et tout écart significatif par rapport au comportement attendu est considéré comme une anomalie, pouvant indiquer une tentative d'intrusion. Cette méthode se révèle particulièrement efficace pour identifier des attaques inconnues, y compris les attaques zero-day. Cependant, elle présente certaines limites : la modélisation précise du comportement normal dans des environnements complexes ou dynamiques peut s'avérer difficile, entraînant souvent un taux élevé de faux positifs. Par ailleurs, l'impossibilité d'analyser des données chiffrées laisse la porte ouverte à certaines attaques furtives.

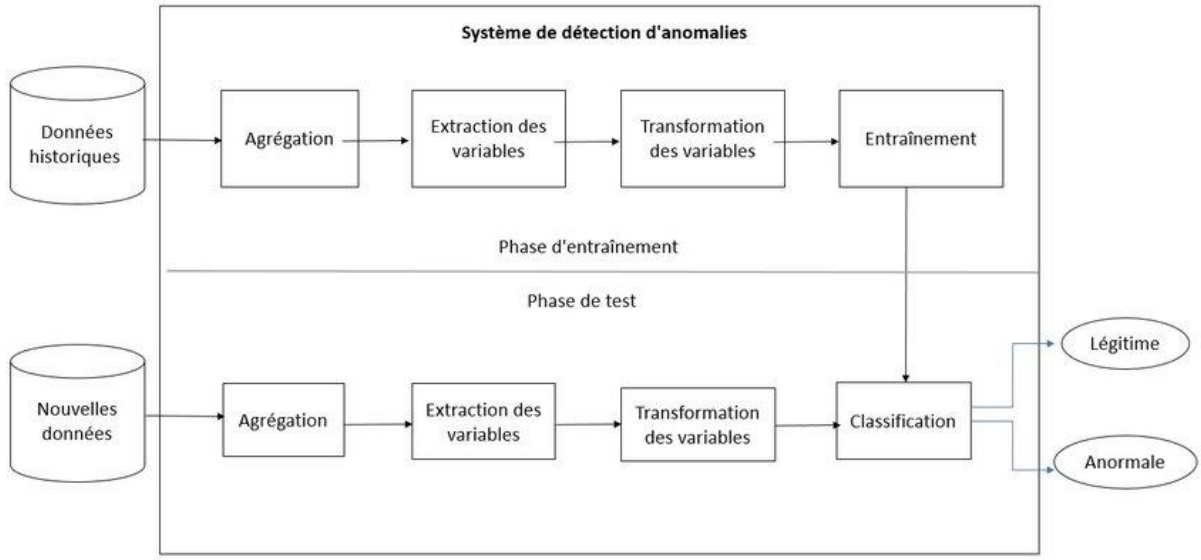


Figure 1.10 : Système de détection d'anomalie

1.3.4. Les types de système détection d'intrusion

Il existe plusieurs types de systèmes de détection d'intrusion (IDS)**,** chacun étant adapté à un environnement spécifique et à des besoins de sécurité particuliers. Les deux principales catégories sont les systèmes de détection basés sur l'hôte (HIDS) et les systèmes de détection basés sur le réseau (NIDS).[24]

1.3.4.1.Système de détection d'intrusion basé sur l'hôte (HIDS)

Le HIDS (Host-based Intrusion Detection System) est un système installé directement sur un appareil, tel qu'un serveur ou un ordinateur, afin de surveiller les activités internes de cette machine. Il analyse les fichiers journaux (logs)**,** les événements système, les zones mémoire ainsi que les actions des utilisateurs, dans le but de repérer tout comportement suspect ou anormal. Grâce à sa capacité d'observation fine des activités locales, le HIDS se montre particulièrement précis dans la détection d'attaques ciblant un hôte spécifique, ce qui en fait un outil adapté aux environnements critiques**.** Cependant, ce type de système présente plusieurs limites. Il est notamment vulnérable aux attaques de falsification (tampering), ce qui nécessite un renforcement de sa propre sécurité. De plus, sa consommation importante de ressources peut dégrader les performances globales de l'appareil hôte. Enfin, bien que le HIDS puisse intégrer certaines fonctions de prévention, celles-ci demeurent limitées, car la majorité des menaces ne sont détectées qu'après leur exécution sur la machine.[24]

1.3.4.2.Système de détection d'intrusions basé sur le réseau (NIDS)

Le NIDS (Network-based Intrusion Detection System) est un système de détection d'intrusions déployé au niveau du réseau, généralement sur des points de contrôle tels que les passerelles ou les routeurs, afin de surveiller le trafic entrant et sortant entre les différents hôtes. Il procède à une inspection approfondie des paquets circulant sur le réseau et les compare à une base de signatures d'attaques connues ou à des modèles comportementaux, dans le but de détecter toute anomalie dans le fonctionnement des applications. Le principal avantage d'un NIDS réside dans sa capacité à offrir une vision d'ensemble du trafic réseau tout en préservant les performances des hôtes surveillés, puisqu'il opère de manière passive. De plus, son fonctionnement discret rend son déploiement difficilement détectable par un attaquant. Cependant, ce type de système présente certaines limites. En période de trafic élevé, le NIDS peut rencontrer des difficultés à analyser l'intégralité des paquets en temps réel, ce qui augmente le risque de non-détection d'attaques furtives. Par ailleurs, il se révèle inefficace face aux données chiffrées ou encapsulées, comme celles transitant via des tunnels VPN, ce qui réduit sa capacité à identifier certaines menaces sophistiquées exploitant le chiffrement.

1.4.INTELLIGENCE ARTIFICIELLE, MACHINE LEARNING ET DEEP LEARNING

1.4.1. Intelligence artificielle

L'intelligence artificielle (IA) désigne un programme informatique conçu pour reproduire les mécanismes de l'intelligence humaine à l'aide d'algorithmes de calcul. Elle permet aux machines d'exécuter des opérations complexes et de simuler des raisonnements similaires à ceux d'un être humain.

Selon John McCarthy, l'un des fondateurs du domaine, l'IA est définie comme :

« La science et l'ingénierie de la conception de machines intelligentes. »

Ce champ de l'informatique vise à développer des systèmes capables d'accomplir des tâches qui, jusqu'à présent, nécessitaient des capacités cognitives humaines. Toutefois, l'intelligence artificielle reste un concept vaste et multidimensionnel, difficile à cerner précisément en raison de son évolution constante et de la diversité des technologies qu'elle englobe.[25]

1.4.2. Machine learning

L'apprentissage automatique (machine learning) constitue un sous-domaine de l'informatique et représente l'une des branches fondamentales de **l'**intelligence artificielle. Il regroupe un ensemble de techniques permettant aux programmes informatiques d'apprendre automatiquement à partir de données, en détectant des motifs ou structures cachées sans avoir été explicitement programmés à cet effet. Les algorithmes de machine learning sont conçus pour analyser des ensembles de données, en en tirer des enseignements et, ainsi, formuler des prédictions ou prendre des décisions. Avant de pouvoir effectuer de telles prédictions, un algorithme doit passer par une phase d'apprentissage supervisé, au cours de laquelle il est exposé à un grand nombre d'exemples annotés, c'est-à-dire des données d'entrée accompagnées de la réponse attendue. Cette étape permet à l'algorithme de modéliser la relation entre les variables d'entrée et les sorties correspondantes. Une fois cette phase terminée, le modèle est capable de généraliser ses connaissances à de nouvelles données non vues précédemment. Par exemple, dans le domaine médical, un algorithme de machine learning peut être entraîné à partir de séries temporelles représentant la fréquence cardiaque de patients. En apprenant la relation entre l'heure de la journée et la fréquence cardiaque, le système peut ensuite prédire la valeur normale attendue à un instant donné. En comparant cette prédiction avec la fréquence cardiaque réelle mesurée, il devient possible de détecter des anomalies physiologiques, contribuant ainsi à une surveillance proactive de l'état de santé du patient**.**[23]

1.4.2.1.Types des systèmes de l'apprentissage automatique

1.4.2.1.1. L'apprentissage supervisé

L'apprentissage supervisé est une forme d**'**apprentissage automatique dans laquelle un modèle apprend à partir de données étiquetées, c'est-à-dire que chaque entrée est associée à une sortie correcte. Le modèle effectue des prédictions puis les compare aux résultats réels, en ajustant progressivement ses paramètres afin de réduire les erreurs et améliorer sa précision au fil du temps. L'objectif principal de cette approche est de permettre au modèle de généraliser ses connaissances pour formuler des prédictions précises sur de nouvelles données jamais rencontrées auparavant. Par exemple, un modèle entraîné sur un ensemble d'images de chiffres manuscrits peut, une fois entraîné, reconnaître correctement de nouveaux chiffres qu'il n'a jamais vus durant la phase d'apprentissage.[26]

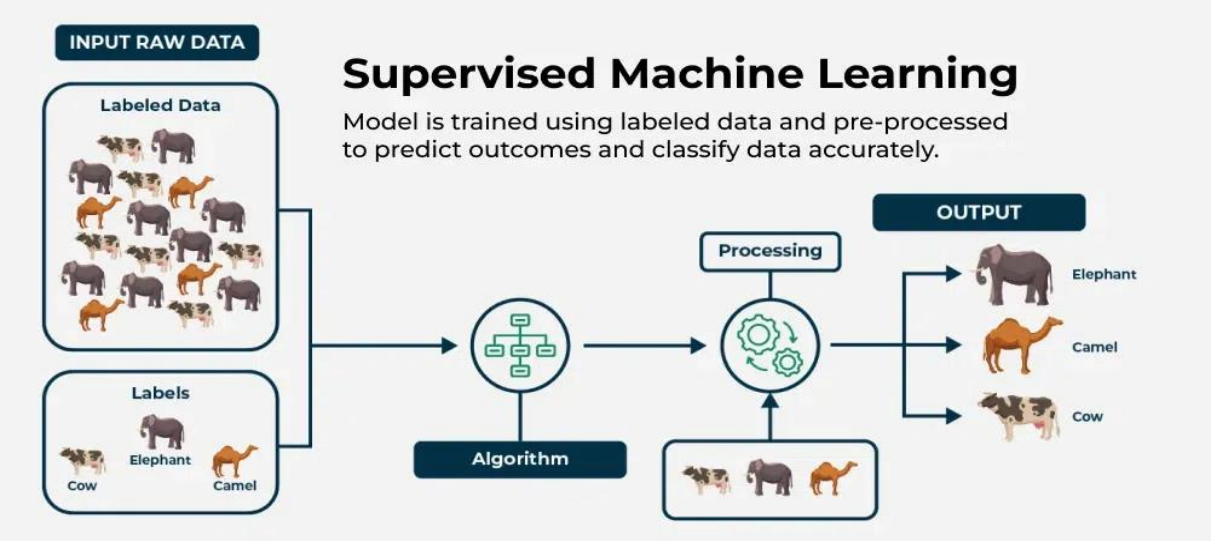


Figure 1.11 : L'apprentissage supervisé

1.4.2.1.2. L'apprentissage non supervisé

L'apprentissage non supervisé est une forme d'apprentissage automatique qui vise à analyser et modéliser des données dépourvues de réponses étiquetées ou de catégories prédéfinies**.** Contrairement à l'apprentissage supervisé, où l'algorithme apprend à partir de paires entrée-sortie, les méthodes non supervisées ne disposent que des données d'entrée et cherchent à découvrir, de manière autonome, des modèles, des structures ou des relations cachées au sein de l'ensemble de données. Cette approche permet donc à l'algorithme d'identifier des regroupements naturels (clustering) ou des corrélations implicites sans aucune intervention humaine ni connaissance préalable de la signification des données traitées.[26]

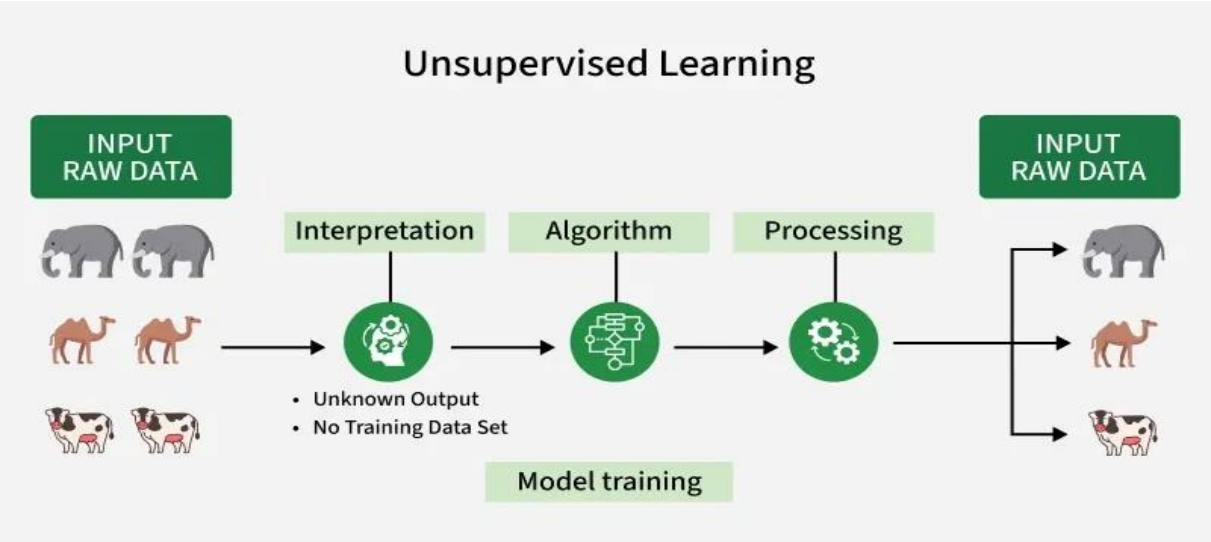


Figure 1.12 : L'apprentissage non supervisé

1.4.2.1.3. L’apprentissage par renforcement

L’apprentissage par renforcement (Reinforcement Learning, RL) est une branche de l’apprentissage automatique qui s’intéresse à la manière dont un agent peut apprendre à prendre des décisions par essais et erreurs**,** dans le but de maximiser une récompense cumulative. Dans ce paradigme, l’agent interagit continuellement avec un environnement, observe les conséquences de ses actions et ajuste son comportement en fonction des retours reçus. Ces retours prennent la forme de récompenses ou de pénalités, qui guident progressivement l’agent vers la stratégie optimale lui permettant d’atteindre ses objectifs de performance.[26]

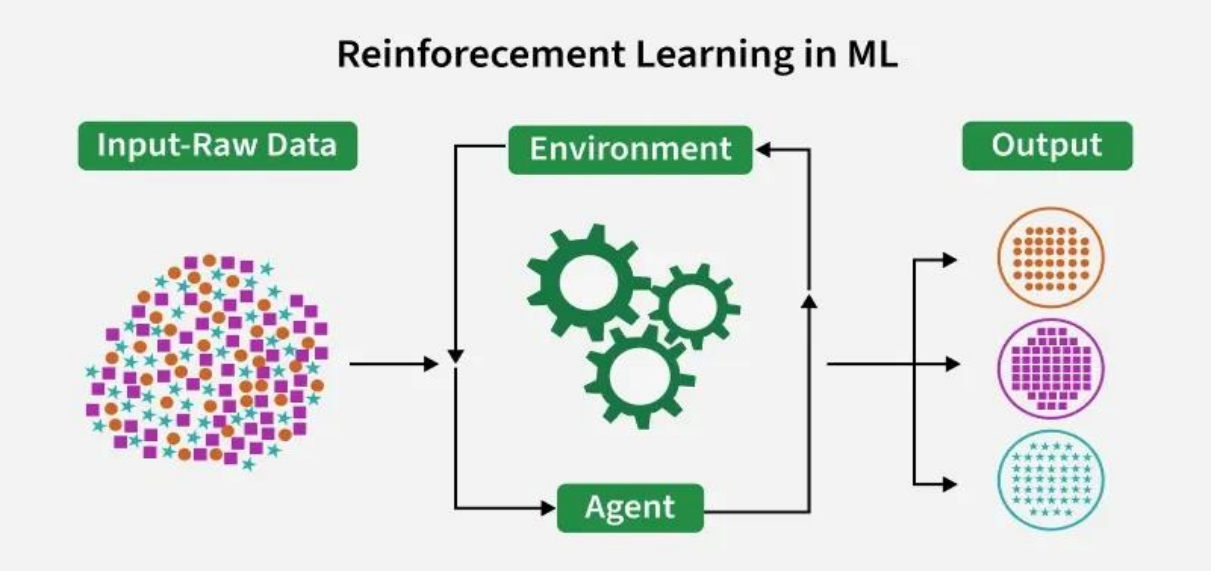


Figure 1.13 : L’apprentissage par renforcement

1.4.2.2.Les algorithmes d’apprentissage automatique

Les algorithmes de Machine Learning (ML)**,** également appelés apprentissage peu profond (Shallow Learning), les plus couramment utilisés dans les systèmes de détection d’intrusion (IDS) incluent notamment l’arbre de décision, le K-Nearest Neighbor (KNN)**,** le réseau neuronal artificiel (ANN)**,** la machine à vecteurs de support (SVM)**,** le clustering K-Means, le réseau d’apprentissage rapide, ainsi que diverses méthodes d’ensemble (ensemble methods). Ces approches constituent les fondations classiques de la détection automatisée d’anomalies dans les réseaux informatiques et IoT.

1.4.2.2.1. Arbre de décision (Decision Tree (DT))

Le Decision Tree (DT) est l'un des algorithmes fondamentaux de l'apprentissage supervisé**.** Il est couramment utilisé pour des tâches de classification et de régression à partir d'un ensemble de données donné, en appliquant une série de décisions hiérarchiques basées sur des règles. Le modèle adopte une structure arborescente classique composée de nœuds, de branches et de feuilles. Chaque nœud représente un attribut ou une caractéristique du jeu de données, chaque branche correspond à une décision ou une règle, et chaque feuille indique un résultat final ou une étiquette de classe. L'algorithme DT sélectionne automatiquement les caractéristiques les plus pertinentes pour la construction de l'arbre, puis applique une opération d'élagage (pruning) afin de supprimer les branches non significatives et d'éviter le surapprentissage (overfitting).[27]

1.4.2.2.2. K-Nearest Neighbor (KNN)

Le K-Nearest Neighbor (KNN) est l'un des algorithmes d'apprentissage supervisé les plus simples et les plus intuitifs du Machine Learning. Il repose sur le principe de la similarité des caractéristiques pour prédire la classe d'un nouvel échantillon de données. L'algorithme identifie la classe d'un échantillon en se basant sur ses k voisins les plus proches, déterminés par le calcul d'une distance (souvent euclidienne) entre les points. Le paramètre k joue un rôle essentiel dans les performances du modèle : une valeur de k trop faible peut conduire à un surapprentissage (overfitting), tandis qu'une valeur trop élevée risque d'entraîner une mauvaise classification en incluant des voisins appartenant à d'autres classes.[27]

1.4.2.2.3. Machine à Vecteurs de Support

La Machine à Vecteurs de Support (SVM--Support Vector Machine) est un algorithme d'apprentissage supervisé fondé sur le principe de la séparation optimale des classes à l'aide d'un hyperplan de marge maximale dans un espace de caractéristiques à n dimensions. Cet algorithme peut être appliqué aussi bien à des problèmes linéaires qu'à des problèmes non linéaires. Dans le cas non linéaire, des fonctions noyau **(**kernel functions) sont utilisées afin de projeter les données d'entrée d'un espace de faible dimension vers un espace de caractéristiques de dimension supérieure**.** Dans cet espace transformé, le SVM détermine un hyperplan optimal à marge maximale**,** qui sert de frontière de décision entre les différentes classes, à l'aide d'un

ensemble restreint de vecteurs de support représentant les points les plus déterminants pour la classification.[27]

1.4.2.2.4. Regroupement de la moyenne k

Le clustering est une approche consistant à regrouper les données en ensembles homogènes, appelés clusters, de manière à ce que les éléments présentant des caractéristiques similaires soient placés dans le même groupe. Parmi les méthodes les plus répandues, le K-Means clustering figure parmi les algorithmes itératifs de Machine Learning non supervisé les plus populaires, reposant sur le principe des centroïdes. Le paramètre K représente le nombre de clusters (ou centroïdes) à former dans un ensemble de données. L'attribution d'un point de données à un cluster se fait généralement en calculant la distance entre ce point et les différents centroïdes, puis en l'associant au plus proche d'entre eux. L'objectif principal de cet algorithme est de minimiser la somme des distances entre les points de données et leurs centroïdes respectifs, afin d'obtenir des groupes compacts et bien séparés.[27]

1.4.2.2.5. Réseau de neurone artificielle

Le réseau neuronal artificiel (Artificial Neural Network – ANN**)** est un algorithme d'apprentissage supervisé inspiré du fonctionnement du système nerveux biologique**.** Il est constitué d'unités de traitement appelées neurones artificiels (ou nœuds)**,** interconnectés entre eux par des poids synaptiques**.** Ces neurones sont généralement organisés en trois types de couches : une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie, formant ainsi une architecture hiérarchique permettant le traitement progressif de l'information. L'algorithme de rétropropagation de l'erreur (backpropagation) est couramment utilisé comme méthode d'apprentissage dans les ANN. Il ajuste les poids des connexions afin de minimiser l'erreur entre la sortie prédite et la sortie réelle, améliorant ainsi les performances du modèle au fil des itérations. L'un des principaux avantages des réseaux neuronaux artificiels réside dans leur capacité à modéliser des relations non linéaires complexes et à généraliser à partir de grands ensembles de données**.** Toutefois, leur principal inconvénient est le temps d'entraînement élevé, dû à leur complexité computationnelle et au grand nombre de paramètres à optimiser, ce qui peut parfois conduire à des solutions sous-optimales ou à un surapprentissage (overfitting) si le modèle n'est pas correctement régularisé.[27]

1.4.3. Deep learning

Le Deep Learning (DL) constitue un sous-ensemble du Machine Learning (ML), caractérisé par l'utilisation de réseaux neuronaux profonds comportant de multiples couches cachées. Ces couches permettent au modèle d'extraire et de représenter progressivement des caractéristiques hiérarchiques et abstraites à partir des données brutes. Grâce à cette profondeur architecturale, les techniques de Deep Learning surpassent généralement les approches de Machine Learning traditionnelles en termes de performance et de capacité de généralisation. En effet, elles peuvent apprendre automatiquement les caractéristiques pertinentes des données sans nécessiter une ingénierie manuelle des variables, et produire ainsi des résultats plus précis et plus robustes.

1.4.3.1.Méthodes d'apprentissage profond

1.4.3.1.1. Réseaux de Neurones Récurrents (RNN)

Les réseaux neuronaux récurrents (RNN) représentent une extension des réseaux neuronaux traditionnels, spécifiquement conçus pour traiter et modéliser des données séquentielles**.** Leur architecture repose sur des unités d'entrée, cachées et de sortie, dans lesquelles les unités cachées jouent le rôle de mémoire, en conservant l'information issue des étapes précédentes du traitement. Ainsi, la décision prise à un instant donné dépend à la fois de l'entrée courante et de l'état précédent du réseau, ce qui confère aux RNN la capacité d'exploiter les dépendances temporelles ou contextuelles au sein des données. Les RNN sont largement utilisés dans divers domaines tels que le traitement automatique de la parole, la reconnaissance d'activités humaines**,** la prédiction de séquences manuscrites, ou encore la compréhension du langage naturel**,** grâce à leur aptitude à capturer la dynamique temporelle et la continuité des informations.[27]

1.4.3.1.2. Les auto-encodeurs

L'Auto-encoder (AE) est une technique d'apprentissage profond (Deep Learning) largement utilisée, appartenant à la famille des réseaux neuronaux non supervisés. Son principe repose sur la reconstruction de l'entrée à partir de la sortie**,** en apprenant automatiquement les caractéristiques les plus représentatives des données. L'architecture d'un AutoEncoder est composée d'une couche d'entrée et d'une couche de sortie de dimensions identiques, tandis que les couches cachées possèdent généralement une dimension inférieure, afin de forcer le réseau à comprimer l'information et à en extraire les caractéristiques essentielles. L'AE adopte une

structure symétrique de type encodeur-décodeur : la partie encodeur apprend à représenter les données dans un espace latent réduit, tandis que la partie décodeur tente de reconstruire les données originales à partir de cette représentation comprimée. Plusieurs variantes de cette architecture existent, notamment l'AutoEncoder empilé (Stacked AE**)**, l'AutoEncoder clairsemé (Sparse AE) et l'AutoEncoder variationnel (VAE)**,** chacune adaptée à des besoins spécifiques en termes de généralisation, de régularisation ou de génération de données.[27]

1.4.3.1.3. Réseau de neurones profonds

Le Deep Neural Network (DNN) représente l'une des structures fondamentales de l'apprentissage profond (Deep Learning)**.** Il se compose d'une couche d'entrée, d'une couche de sortie et de plusieurs couches cachées interconnectées, permettant au modèle d'apprendre à différents niveaux de représentation. Le DNN est particulièrement efficace pour modéliser des relations non linéaires complexes entre les variables d'un ensemble de données. L'ajout de couches cachées supplémentaires accroît la profondeur du réseau et permet au modèle d'extraire des caractéristiques de plus en plus abstraites, renforçant ainsi sa capacité à traiter des données complexes et à effectuer des prédictions précises.[27]

1.4.3.1.4. Réseaux de neurones convolutifs (CNN)

Le réseau neuronal convolutif (Convolutional Neural Network, CNN) est une architecture d'apprentissage profond (DL) particulièrement adaptée à l'analyse de données structurées sous forme de tableaux, telles que les images, les signaux ou encore certaines représentations de trafic réseau. Un CNN est généralement composé d'une couche d'entrée, suivie de plusieurs couches convolutives et de couches de regroupement (pooling) destinées à l'extraction automatique des caractéristiques. Ces couches sont ensuite connectées à une ou plusieurs couches entièrement connectées (fully connected)**,** aboutissant à un classificateur softmax chargé de produire la sortie finale. Grâce à sa capacité à extraire et hiérarchiser les caractéristiques locales et spatiales des données, le CNN s'est imposé comme une architecture de référence dans de nombreux domaines, notamment la vision par ordinateur, la reconnaissance de formes et, plus récemment, la détection d'intrusions dans les réseaux IoT**.**[27]

1.5. CONCLUSION

Ce chapitre a présenté les concepts fondamentaux liés aux réseaux IoT et aux problématiques de sécurité qui leur sont associées. Il a permis de définir l'architecture, les composants et les protocoles de communication des systèmes IoT, tout en mettant en évidence les contraintes spécifiques telles que l'hétérogénéité, la limitation des ressources, l'évolutivité et les exigences de qualité de service. Les principes des systèmes de détection d'intrusion ont également été introduits, ainsi que leurs principales méthodes de détection. Cette analyse a montré les limites des approches traditionnelles face aux environnements IoT dynamiques et complexes. Enfin, l'intelligence artificielle, le Machine Learning et le Deep Learning ont été présentés comme des solutions prometteuses pour améliorer l'efficacité et l'adaptabilité des mécanismes de détection d'intrusion. Ces bases théoriques constituent le socle nécessaire à l'étude des approches avancées de détection présentées dans le chapitre suivant, consacré à l'état de l'art des méthodes Machine Learning et Deep Learning appliquées aux réseaux IoT.

CHAPITRE 2 : ÉTAT DE L'ART SUR LES APPROCHES MACHINE LEARNING ET DEEP LEARNING POUR LA DETECTION D'INTRUSION DANS LES RESEAUX IoT

2.1. INTRODUCTION

Face à l'augmentation continue des cybermenaces ciblant les réseaux IoT, de nombreuses recherches ont été menées afin de proposer des solutions de détection d'intrusion plus intelligentes, adaptatives et efficaces. Les approches basées sur le Machine Learning et le Deep Learning occupent aujourd'hui une place centrale dans la littérature scientifique, en raison de leur capacité à analyser de grands volumes de données et à identifier des schémas complexes de comportements anormaux. Le Machine Learning permet d'exploiter des techniques supervisées, non supervisées ou hybrides pour classifier le trafic réseau et détecter les intrusions. Ces méthodes incluent notamment les réseaux de neurones artificiels, les machines à vecteurs de support, les arbres de décision, les forêts aléatoires ou encore les algorithmes de regroupement. En complément, le Deep Learning, à travers des architectures profondes telles que les réseaux neuronaux profonds, convolutifs et récurrents, offre des performances accrues en capturant des relations non linéaires et des dépendances temporelles complexes. Ce chapitre propose une revue critique des travaux de recherche récents portant sur l'utilisation du Machine Learning et du Deep Learning pour la détection d'intrusion dans les réseaux IoT. Il analyse les principaux algorithmes utilisés, leurs avantages, leurs limites et leurs domaines d'application. Cette analyse permet d'identifier les tendances actuelles et les lacunes existantes, servant ainsi de base à la conception du système de détection centralisé proposé dans ce mémoire.

2.2. APPROCHE MACHINE LEARNING POUR LA DETECTION D'INTRUSION

Le Machine Learning (ML) constitue une approche particulièrement efficace pour la détection d'intrusions dans les réseaux IoT, grâce à sa capacité à apprendre automatiquement à partir des données et à reconnaître des schémas anormaux. Plusieurs algorithmes sont ainsi mobilisés pour classer le trafic ou les comportements réseau en deux grandes catégories : normal ou malveillant.

Les principaux algorithmes de machine learning couramment utilisés dans les systèmes de détection d'intrusions (IDS) :

2.2.1. Réseaux de neurones artificiels

Les ANN sont entraînées à reconnaître des signatures d'attaque spécifiques ainsi que des comportements anormaux en analysant les caractéristiques du trafic réseau. Selon les recherches de Sohaib et al.[28], cette méthode a démontré une précision moyenne de 84 % ainsi qu'un faible taux de faux positifs, inférieur à 8 %**,** dans le cadre d'une validation croisée répétée appliquée à la détection d'intrusions dans les systèmes IoT. L. Jamal et al.[29], ont également utilisé les ANN pour détecter des comportements malveillants sur un ensemble de données IoT comprenant 461 043 enregistrements, dont 300 000 bénins et 161 043 malveillants. Avec la méthodologie proposée, les logiciels malveillants ont été détectés avec une précision de 94,17 **%** et classés avec une exactitude de 97,08 %**.** Ces résultats mettent en évidence la capacité des réseaux neuronaux à traiter efficacement de grands ensembles de données hétérogènes. Toutefois, un défi majeur subsiste concernant la distinction fine entre les paquets malveillants et légitimes, en particulier lorsque l'ANN doit gérer des volumes de données très importants.

2.2.2. Machine à vecteurs de support (SVM)

Les machines à vecteurs de support (SVM) sont largement employées pour la détection d'intrusions dans les environnements IoT. Goeschel et al.[30], ont démontré l'efficacité des SVM en atteignant une précision de 99,62 % tout en maintenant un faible taux de faux positifs de 1,57 %**,** ce qui confirme la robustesse de cette méthode pour la détection d'intrusions dans les réseaux IoT. Jing et al.[31] ont utilisé des SVM pour la détection d'intrusion sur l'ensemble de données UNSW-NB15, en intégrant une méthode de mise à l'échelle non linéaire qui a permis d'améliorer les performances du modèle. Cette approche a atteint une précision de 85,99 % pour la classification binaire et 75,77 % pour la classification multiclasse. Ces résultats confirment le potentiel des SVM en tant qu'outils robustes et performants pour la détection d'anomalies dans les environnements IoT, renforçant leur pertinence dans les solutions de sécurité réseau. Cependant, les SVM présentent certaines limites lorsqu'il s'agit de traiter des données IoT très variées, dynamiques et en constante évolution. La construction et l'optimisation de modèles SVM deviennent particulièrement coûteuses en calcul lorsque les ensembles de données sont volumineux ou comportent de nombreuses caractéristiques. De plus, leur capacité de généralisation peut diminuer face à des environnements inconnus ou non stationnaires. Ainsi, des travaux de recherche supplémentaires restent nécessaires pour améliorer leur efficacité et leur adaptabilité dans des contextes IoT réels.

2.2.3. K plus proches voisins (KNN)

K-Nearest Neighbor (KNN) est une méthode de classification non paramétrique particulièrement adaptée aux tâches de détection d'intrusion dans les systèmes IoT. Abdaljabar et al.[32], ont proposé une approche hybride combinant les algorithmes KNN et Decision Tree, qui a atteint une précision de 100 % après la normalisation des données, démontrant une efficacité remarquable pour la détection des attaques IoT. Une autre étude d'Aref et al.[33] ont introduit une approche semi-supervisée utilisant un KNN hyper-paramétré sur l'ensemble de données NSL-KDD, atteignant une précision de 95 %**.** Ces études mettent en évidence l'efficacité de KNN pour la détection d'intrusions, tout en soulignant certaines limites, notamment sa sensibilité au choix du paramètre « k » et sa vulnérabilité au bruit présent dans les données**.**

2.2.4. Régression logistique (LR)

La régression linéaire a été employée avec succès pour identifier des menaces spécifiques au sein des environnements IoT. Swathi Sambangi et al.[34] ont exploré l'utilisation de la régression linéaire multiple pour détecter les attaques DDoS dans les environnements cloud, en s'appuyant sur l'ensemble de données CIC-IDS 2017**.** Une technique de sélection de caractéristiques a été appliquée afin d'identifier les attributs les plus pertinents pour la prédiction de ces attaques. Le modèle obtenu a atteint une précision de 73,79 % en utilisant les 16 attributs sélectionnés**,** démontrant ainsi sa capacité à détecter de manière robuste les attaques DDoS dans les environnements cloud.

2.2.5. Arbres de décision

Les arbres de décision se révèlent particulièrement efficaces pour la classification et la détection des comportements anormaux dans les environnements IoT. Qaddoura et al.[35] ont utilisé des arbres de décision pour analyser le flux de données réseau, identifier les sources de trafic suspectes et contribuer à la détection des attaques de type déni de service distribué (DDoS)**.** Kajal Rai et al.[36] ont mis en œuvre l'algorithme DT C4.5 pour la détection d'intrusions dans les systèmes IoT, en mettant l'accent sur la sélection et l'optimisation des fonctionnalités. En utilisant l'ensemble de données NSL-KDD, ils ont obtenu des performances satisfaisantes en termes de vitesse de construction du modèle, de taux de faux positifs**,** de taux de vrais positifs et de précision. Toutefois, ces performances se sont avérées sensibles à la taille de l'ensemble

de données ainsi qu'au nombre de fonctionnalités utilisées. Ces travaux démontrent l'adaptabilité et l'efficacité des arbres de décision dans des applications complexes. Néanmoins, certains défis subsistent, tels que la difficulté à identifier de nouvelles formes d'attaques et la dépendance à une sélection appropriée des caractéristiques**,** ce qui peut limiter leur précision dans des environnements de cybersécurité dynamiques.

2.2.6. Regroupement

Le clustering constitue une technique essentielle de l'apprentissage automatique non supervisé. Il consiste à regrouper des données de manière à ce que les éléments appartenant à un même groupe (ou *cluster*) présentent une forte similarité entre eux, tout en étant nettement distincts des éléments appartenant à d'autres groupes. Cette approche permet ainsi de révéler des structures intrinsèques dans les données sans recourir à des étiquettes prédéfinies. Muniyandi et al.[37], ont proposé une méthode de détection d'anomalies combinant l'algorithme des K-moyennes et l'arbre de décision C4.5. Toutefois, leurs résultats ont montré que les performances du clustering K-means demeurent inférieures à celles des approches supervisées, en particulier pour l'identification des attaques déjà connues.

2.2.7. Forêt aléatoire (Random Forest – RF)

La forêt aléatoire (Random Forest) est un algorithme d'ensemble reposant sur la combinaison de multiples arbres de décision, ce qui lui confère une grande robustesse et une précision élevée. Dans[38], les auteurs ont proposé un mécanisme de détection fondé sur cet algorithme afin d'identifier les attaques ciblant les dispositifs IoT caractérisés par des ressources limitées. Les résultats obtenus démontrent que la Random Forest parvient à détecter efficacement les anomalies tout en maintenant une consommation de ressources relativement faible, ce qui le rend adapté aux systèmes IoT à ressources limitées.

2.3. APPROCHE DEEP LEARNING POUR LA DETECTION D'INTRUSION

Le deep learning occupe aujourd'hui une place centrale dans la détection d'intrusions, notamment au sein des environnements IoT caractérisés par leur complexité et leur hétérogénéité. Grâce à leurs architectures constituées de multiples couches cachées, les modèles d'apprentissage profond offrent une capacité avancée d'analyse et d'extraction automatique de caractéristiques, ce qui les rend particulièrement efficaces pour identifier des schémas subtils

et complexes associés aux activités malveillantes. Plusieurs modèles d'apprentissage profond ont été utilisés pour détecter de manière efficace une large gamme d'attaques au sein des environnements IoT :

2.3.1. Réseaux de neurones profonds (DNN)

Les Deep Neural Networks (DNN) ont montré des améliorations notables dans la détection d'intrusions au sein des architectures IoT, grâce à leur capacité à apprendre directement à partir de données brutes et à extraire automatiquement des représentations pertinentes. Ahmad et al.[39] ont démontré qu'en utilisant l'ensemble de données IoT-Botnet 2020, les DNN ont permis d'améliorer la précision du modèle de 0,57 à 2,6 % et de réduire le taux de fausses alertes de 0,23 à 7,98 % par rapport à d'autres méthodes, confirmant ainsi leur supériorité pour la détection d'intrusions dans les réseaux IoT. Jin Kim et al.[40], ont également étudié un système de détection d'intrusion basé sur un DNN utilisant l'ensemble de données KDD Cup 99, révélant une précision élevée pour la détection des intrusions avec un faible taux de fausses alertes. Le modèle a atteint un taux de détection de 99,95 **%** pour les flux anormaux, bien qu'une légère diminution (3,87 à 10,99 **%**) ait été observée pour la détection des flux bénins par rapport à d'autres algorithmes. Malgré cela, le DNN a obtenu un score global de 96,085 **%**, surpassant les méthodes comparées. Ces résultats confirment l'efficacité des DNN pour la détection d'intrusions tout en mettant en évidence la nécessité d'une optimisation continue pour améliorer la détection des flux bénins. Les principaux avantages de cette approche incluent une grande précision et une adaptabilité aux attaques évolutives, tandis que ses limites concernent la complexité du modèle et le besoin de grands volumes de données pour l'entraînement.

2.3.2. Réseaux de neurones convolutifs (CNN)

Dans le domaine de l'IoT, les CNN ont été exploités pour détecter les logiciels malveillants ciblant les systèmes Android. Des recherches [41], ont démontré que les CNN sont capables d'apprendre automatiquement des caractéristiques pertinentes pour la détection de logiciels malveillants à partir de données brutes, supprimant ainsi le besoin de sélection manuelle des fonctionnalités. Cette approche constitue une avancée par rapport aux méthodes traditionnelles d'apprentissage automatique, en permettant une modélisation complète de bout en bout. Chen et al.[42], ont développé un système de détection d'intrusion réseau basé sur un CNN, entraîné à la fois sur des caractéristiques extraites et sur le trafic réseau brut, montrant une précision supérieure par rapport aux modèles utilisant uniquement les caractéristiques extraites. Pour

leurs expériences, ils ont utilisé des ensembles de données standard, tels que NSL-KDD. Le principal atout de cette approche réside dans la capacité des CNN à traiter directement les données brutes sans nécessiter un prétraitement intensif, ce qui améliore l'efficacité de la détection. Néanmoins, la complexité du modèle et les exigences de calcul élevées constituent des limitations importantes.

2.3.3. Réseaux de neurones récurrents (RNN)

Les RNN se révèlent particulièrement efficaces pour les applications de détection de menaces**,** où les modèles doivent prendre en compte la dimension temporelle des données. Shin Park et al. ont étudié une méthode de détection d'intrusion basée sur le LSTM, mettant en évidence sa supériorité par rapport à d'autres techniques en termes d'erreur quadratique moyenne (MSE) et d'erreur absolue moyenne (MAE)**.**[43]. Tang et al. [44], ont proposé une méthode de détection d'intrusion basée sur un réseau neuronal récurrent profond (GRU-RNN) pour les SDN, atteignant une précision de 89 % en utilisant seulement six caractéristiques brutes. Ce modèle s'est montré efficace pour la détection en temps réel, tout en n'impactant pas de manière significative les performances globales du réseau. Torres et al. [45], ont démontré l'efficacité des RNN dans l'analyse du comportement du trafic réseau pour identifier les activités malveillantes, confirmant ainsi le rôle crucial de ces modèles dans la classification précise du trafic réseau**.**

2.4. ANALYSE

La détection d'intrusion dans les environnements IoT évolue rapidement sous l'effet des avancées en intelligence artificielle. Les techniques modernes notamment l'apprentissage profond, l'apprentissage non supervisé, améliorent significativement la précision et la rapidité des systèmes IDS, tout en renforçant la confidentialité et l'adaptabilité face à la diversité des menaces IoT. L'optimisation des performances des modèles repose de plus en plus sur **l'**équilibrage des classes**,** la sélection optimale des caractéristiques et l'ajustement précis des hyperparamètres**.** L'application du Machine Learning dans l'IoT impose néanmoins des contraintes fortes : limitations des ressources, hétérogénéité des dispositifs et besoin d'analyse instantanée. Une comparaison des principaux modèles met en évidence leurs spécificités :

- **SVM** : très précis mais coûteux en calcul et sensibles aux paramètres.
- **Arbres de décision** : rapides et interprétables, mais sujets au sur-ajustement.

- **KNN** : simple et efficace pour les anomalies, mais très coûteux sur grands volumes.
- **DNN** : performants et adaptables, nécessitent beaucoup de données et de calcul.
- **Autoencodeurs** : efficaces en non supervisé mais peuvent perdre des informations.
- **RNN** : adaptés aux données temporelles, mais lourds en calcul et sujets au gradient nul.
- **Random forest** : une grande robustesse et une précision élevée

Tableau 2.1 : Récapitulatif des approches ML et DL pour la détection d'intrusions dans les réseaux IoT.

Méthode	**Principe / Description**	**Résultats clés (selon études)**	**Avantages**	**Limites**
ANN (Réseaux de neurones artificiels)	Apprennent les motifs d'attaque à partir des caractéristiques du trafic.	Sohaib et al. : précision 84 %, <8 % FPR. Jamal et al. : 94,17 % précision, 97,08 % exactitude.	Très efficaces sur grands datasets ; bonne détection des comportements malveillants.	Confusion entre paquets légitimes/malveillants pour gros volumes ; besoin important en données.
SVM (Support Vector Machine)	Maximisation des marges pour séparer classes normales et malveillantes.	Goeschel : 99,62 % précision, 1,57 % FPR. Jing : 85,99 % (binaire), 75,77 % (multiclasse).	Très robuste ; bonnes performances sur données structurées.	Coûts computationnels élevés ; faible adaptabilité aux environnements IoT dynamiques.

Méthode	Principe/ Description	Résultats clés (selon études)	Avantages	Limites
KNN (K-Nearest Neighbors)	Classification basée sur la similarité des données proches.	Hybride KNN+DT:100 % précision. Semi-supervisé : 95 %.	Simple, efficace, bonne performance en anomalies.	Sensible au choix de *k* ; coûts élevés sur grands volumes ; sensible au bruit.
Régression Logistique (LR)	Modélise la probabilité qu'un événement soit malveillant.	Sambangi : 73,79 % précision (CIC-IDS 2017).	Bonne interprétabilité ; efficace sur données linéaires	Performances limitées sur données complexes et attaques variées.
Arbres de décision (DT)	Partitionnement récursif des données pour classifier les attaques.	NSL-KDD : bons taux FP/TP, modèle rapide.	Interprétables, rapides, efficaces pour sélection de caractéristiques.	Surapprentissage ; difficulté à détecter nouvelles attaques.
Clustering (K-means)	Regroupe données similaires sans étiquettes.	Performance < approches supervisées.	Utile en absence d'étiquettes ; détecte anomalies inconnues.	Moins efficace sur attaques connues ; dépend du nombre de clusters.
Random Forest (RF)	Ensemble d'arbres pour classification robuste.	Bonne détection d'anomalies avec faible consommation IoT.	Robuste, précis, faible coût pour IoT limité en ressources.	Moins efficace pour motifs très complexes ; taille du modèle élevée.

Méthodes	Principe/ Description	Résultats clés (selon études)	Avantages	Limites
DNN (Deep Neural Networks)	Réseaux profonds apprenant automatiquement des représentations complexes.	Amélioration précision de 0,57 à 2,6 % ; réduction FPR de 0,23 à 7,98 %.	Haute précision ; très adaptatif.	Très coûteux en calcul ; nécessite grands datasets.
CNN (Convolutional Neural Networks)	Extraction automatique de caractéristiques (end-to-end).	Meilleure précision que modèles ML traditionnels.	Traite données brutes ; évite feature engineering.	Forte complexité ; besoins computationnels élevés.
RNN / LSTM / GRU	Capturent dépendances temporelles du trafic réseau.	GRU-RNN : 89 % précision ; LSTM meilleur MSE/MAE.	Excellents pour données séquentielles ; efficaces en temps réel.	Problèmes de gradient ; besoin en ressources ; moins efficace sur données non temporelles.

2.5. CONCLUSION

Ce chapitre a présenté l'état de l'art des approches basées sur le Machine Learning et le Deep Learning pour la détection d'intrusions dans les réseaux IoT. L'analyse des travaux existants a mis en évidence l'efficacité de ces méthodes pour identifier des comportements anormaux et faire face à la diversité et au volume des données générées par les environnements IoT. Les techniques de Machine Learning offrent des solutions efficaces pour la classification du trafic et la détection d'attaques connues, tandis que les approches de Deep Learning permettent une meilleure modélisation des schémas complexes et des dépendances temporelles. Toutefois, ces méthodes présentent des défis liés à la complexité, à la consommation de ressources et à la latence. En conclusion, le choix d'une approche de détection repose sur un compromis entre performance et contraintes opérationnelles. Les enseignements tirés de cet état de l'art ont guidé les choix méthodologiques adoptés pour l'implémentation et l'évaluation expérimentale du système proposé, présentées dans le chapitre suivant.

CHAPITRE 3 : IMPLEMENTATION, TESTS ET EVALUATION DES PERFORMANCES

3.1. INTRODUCTION

Ce chapitre présente l'implémentation et l'évaluation expérimentale du système de détection précoce d'intrusions proposé pour les réseaux IoT. Il décrit l'environnement de développement, le jeu de données utilisé, les étapes de prétraitement ainsi que les modèles de Machine Learning et de Deep Learning mis en œuvre. Les performances des différents modèles sont évaluées à l'aide de métriques standard afin d'analyser leur efficacité dans la détection des attaques et des comportements anormaux.

3.2. OUTILS ET ENVIRONNEMENT DE DEVELOPPEMENT

3.2.1. Google colab

Google Colab (abréviation de Google Colaboratory) est une plateforme gratuite proposée par Google qui permet d'écrire et d'exécuter du code Python directement depuis un navigateur, sans nécessiter l'installation préalable de logiciels ni de configurations matérielles spécifiques. Hébergé sur le cloud, Colab offre un environnement prêt à l'emploi pour manipuler des notebooks Jupyter, largement utilisés en science des données, en programmation et en apprentissage automatique (machine learning)**.** Cette plateforme met à disposition des ressources de calcul puissantes, notamment des GPU (processeurs graphiques) et des TPU (unités de traitement de tenseurs), permettant d'accélérer significativement l'entraînement de modèles de deep learning. En facilitant l'accès à ces ressources et aux principales bibliothèques d'apprentissage automatique, Google Colab constitue ainsi un outil essentiel pour l'expérimentation, le prototypage et le développement collaboratif de projets en intelligence artificielle.[46]

3.2.2. Python

Python est un langage de programmation interprété**,** orienté objet et de haut niveau, caractérisé par une sémantique dynamique. Grâce à ses structures de données intégrées et à ses mécanismes de typage et de liaison dynamiques, il constitue un outil particulièrement adapté au développement rapide d'applications, mais aussi à une utilisation en tant que langage de script ou de liaison entre différents composants logiciels. Sa syntaxe claire et facile

à apprendre favorise la lisibilité du code et contribue à réduire les coûts de maintenance des programmes. Par ailleurs, Python prend en charge les modules et les packages, ce qui encourage la modularité, la réutilisation du code et une structuration efficace des projets. Enfin, l'interpréteur Python et sa vaste bibliothèque standard sont librement disponibles, tant en version source qu'en version binaire, pour la plupart des plateformes majeures, ce qui facilite grandement leur diffusion et leur accessibilité.[47]

3.2.3. SKLearn



Scikit-learn (ou Sklearn**)** est l'une des bibliothèques les plus puissantes et les plus largement utilisées pour **l'**apprentissage automatique en Python. Elle met à disposition un ensemble d'outils performants pour la modélisation statistique et le machine learning, couvrant des tâches telles que la classification, la régression, le regroupement (clustering) et la réduction de dimensionnalité, le tout à travers une interface cohérente et intuitive. Principalement écrite en Python, Scikit-learn repose sur des bibliothèques fondamentales telles que NumPy, SciPy et Matplotlib. Initialement appelée scikits.learn, elle a été développée par David Cournapeau dans le cadre du programme Google Summer of Code en 2007. Par la suite, en 2010**,** Fabian Pedregosa**,** Gaël Varoquaux**,** Alexandre Gramfort et Vincent Michel, chercheurs à **l**'INRIA (Institut national de recherche en informatique et en automatique, France), ont considérablement amélioré le projet et publié la première version publique (v0.1 beta) le 1er février 2010**.**

3.2.4. TensorFlow

TensorFlow est une bibliothèque open-source développée par Google pour le machine learning et le deep learning. Initialement créée en 2011 sous le nom DistBelief, elle a été rendue publique en 2015. TensorFlow offre un ensemble d'outils permettant de concevoir, d'entraîner et de déployer des modèles d'apprentissage automatique, en particulier des réseaux de neurones profonds. Son fonctionnement repose sur une architecture fondée sur des graphes de flux de données (dataflow graphs). Dans cette approche, les nœuds du graphe représentent des opérations mathématiques, tandis que les arcs transportent des tenseurs, c'est-à-dire des matrices ou vecteurs multidimensionnels. Cette représentation confère à TensorFlow une grande flexibilité et permet l'exécution efficace des modèles sur divers matériels, notamment les CPU, GPU et TPU.



3.2.5. Keras

Keras est une bibliothèque open-source écrite en Python, conçue pour faciliter le développement et le prototypage rapide de modèles de deep learning. Créée en 2015 par François Chollet dans le cadre du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), Keras se présente comme une API de haut niveau qui repose sur des frameworks d'apprentissage automatique tels que TensorFlow, JAX ou PyTorch. Grâce à son interface intuitive et modulaire, Keras permet de construire et d'entraîner aisément différents types de réseaux de neurones, notamment les réseaux convolutifs (CNN), les réseaux récurrents (RNN, LSTM, GRU) ou encore les réseaux entièrement connectés (DNN). Cette simplicité d'utilisation en fait un outil privilégié pour le prototypage rapide, l'expérimentation et l'apprentissage du deep learning, tout en offrant suffisamment de flexibilité pour des projets avancés.

3.3. PRESENTATION DU DATASET

Le jeu de données CICIoT2023 constitue une ressource moderne et complète développée pour la recherche en sécurité des réseaux de l'Internet des objets (IoT), en particulier pour l'évaluation des systèmes de détection d'intrusion (IDS) et de détection d'anomalies. Publié par l'Institut canadien sur la cybersécurité (CIC), ce jeu de données reproduit fidèlement le trafic réseau IoT réel ainsi que divers scénarios d'attaques, offrant ainsi une base solide pour l'expérimentation en apprentissage automatique appliqué à la cybersécurité. Le CICIoT2023 a été généré à partir d'un banc d'essai réaliste simulant la communication entre plusieurs dispositifs IoT courants tels que les téléviseurs intelligents, webcams, thermostats connectés et appareils portables. Il comprend à la fois du trafic bénin et une large gamme de trafics malveillants, couvrant des attaques telles que le déni de service (DoS), le déni de service distribué (DDoS), les attaques par force brute, les botnets, la reconnaissance ainsi que d'autres menaces avancées.

Parmi ses principales caractéristiques, le jeu de données :

- Contient un mélange de trafic réseau IoT normal et malveillant ;
- Comprend 34 types d'attaques distincts, représentant des scénarios de cybermenaces modernes et variés ;

- Fournit des données étiquetées, adaptées à l'entraînement de modèles de machine learning supervisé ;
- Met à disposition des caractéristiques de flux réseau extraites (telles que la taille des paquets, la durée, les indicateurs ou les mesures statistiques) permettant la classification du trafic et la détection d'anomalies ;
- Soutient la recherche en détection d'intrusion, analyse comportementale et élaboration de stratégies de sécurité IoT.

Ainsi, le CICIoT2023 comble le fossé entre les ensembles de données de sécurité réseau traditionnels et les nouveaux modèles de communication propres aux environnements IoT, en constituant un point de référence pour l'évaluation et la validation des solutions de sécurité basées sur l'intelligence artificielle.[48]

3.4. PRETRAITEMENT DES DONNEES

Avant d'entraîner les modèles d'apprentissage, les données issues du dataset sont soumises à une série d'opérations de prétraitement indispensables pour assurer la qualité des entrées et optimiser les performances des algorithmes. Ces étapes se déroulent comme suit :

- **Chargement et fusion des données :** Les différents fichiers CSV (provenant de capteurs, journaux systèmes ou trafic réseau) sont intégrés et combinés afin de constituer une base de données unique et cohérente.
- **Traitement des valeurs manquantes :** Les observations comportant des valeurs absentes (NaN) sont soit supprimées, soit imputées à l'aide de méthodes statistiques telles que la moyenne ou la médiane, selon la nature des attributs concernés.

- **Encodage des variables catégorielles :** Les caractéristiques non numériques, comme les types de protocoles ou de services, sont converties en valeurs exploitables par les modèles grâce à des techniques d'encodage telles que le One-Hot Encoding ou le Label Encoding.
- **Normalisation des variables numériques :** Les attributs continus (par exemple, la bande passante ou l'utilisation CPU) sont mis à l'échelle à l'aide de méthodes telles que la normalisation min-max ou la standardisation Z-score, garantissant une homogénéité des variations entre les différentes caractéristiques.

- **Séparation des données :** Les données prétraitées sont ensuite réparties en ensembles d'entraînement et de test, permettant d'une part l'apprentissage des modèles et, d'autre part, leur évaluation sur des données jamais vues auparavant.

3.5. CHOIX DE L'ALGORITHME

Le choix de l'algorithme d'apprentissage à utiliser dans le cadre de notre étude et sur le jeu de données CICIoT2023 constitue une étape cruciale, mais souvent complexe. En effet, il n'existe pas de méthode universelle permettant d'identifier à l'avance l'algorithme le plus performant pour un jeu de données donné. Dans la majorité des cas, une approche empirique s'impose : elle consiste à expérimenter et évaluer plusieurs modèles d'apprentissage afin de déterminer celui qui offre les meilleurs résultats selon les critères de performance retenus. Ainsi, nous avons choisi de tester et comparer plusieurs algorithmes d'apprentissage automatique afin d'identifier le plus adapté à notre problématique de détection d'intrusions dans les réseaux IoT.

Ci-dessous, nous présentons la liste des algorithmes d'apprentissage que nous avons retenus pour les phases de test et de comparaison. L'objectif est d'identifier celui offrant les meilleures performances sur le jeu de données CICIoT2023 et le plus adapté à la détection d'intrusions dans les réseaux IoT. Ces algorithmes sont :

- Régression Logistique
- Machine à vecteurs de support (SVM)
- Réseau de neurones artificielle (ANN)
- Arbre de décision
- Random Forest
- Réseau de neurones profonds (DNN)
- Réseau de neurones convolutifs (CNN)

3.6. ARCHITECTURE GLOBALE DU SYSTEME

Cette section présente l'architecture générale du système de détection d'intrusions conçu pour un environnement IoT, basée sur l'exploitation des données issues du dataset CICIoT2023. Le processus comprend plusieurs étapes, allant de l'acquisition des données jusqu'à leur classification à l'aide d'algorithmes de machine learning (ML) et de deep learning (DL)**.**

Afin de faciliter l'analyse, l'architecture est structurée en cinq modules fonctionnels, décrits en détail ci-après :

- **Collecte des données :** Les données utilisées proviennent du dataset CICIoT2023, lequel simule un environnement IoT réel comportant des sources de données variées.
- **Prétraitement des données :** Cette étape inclut le nettoyage et la normalisation des données, la sélection des caractéristiques pertinentes**,** ainsi que la répartition des données en ensembles d'entraînement et de test.
- **Extraction des caractéristiques :** Les attributs discriminants sont identifiés et extraits afin de représenter de manière efficace le comportement du système**.**
- **Phase de classification :** Différents modèles sont mis en œuvre pour classifier les données :

  Apprentissage automatique (ML) : Random Forest, Arbres de décision, etc. **Apprentissage profond (DL)** : Réseaux de neurones profonds (DNN), RNN, etc.

- **Évaluation et détection :** Les performances des modèles sont mesurées à l'aide de métriques standard telles que l'accuracy, la précision, le rappel, le F1-score et la matrice de confusion, afin de permettre une détection des intrusions fiable et robuste.

Modélisation :

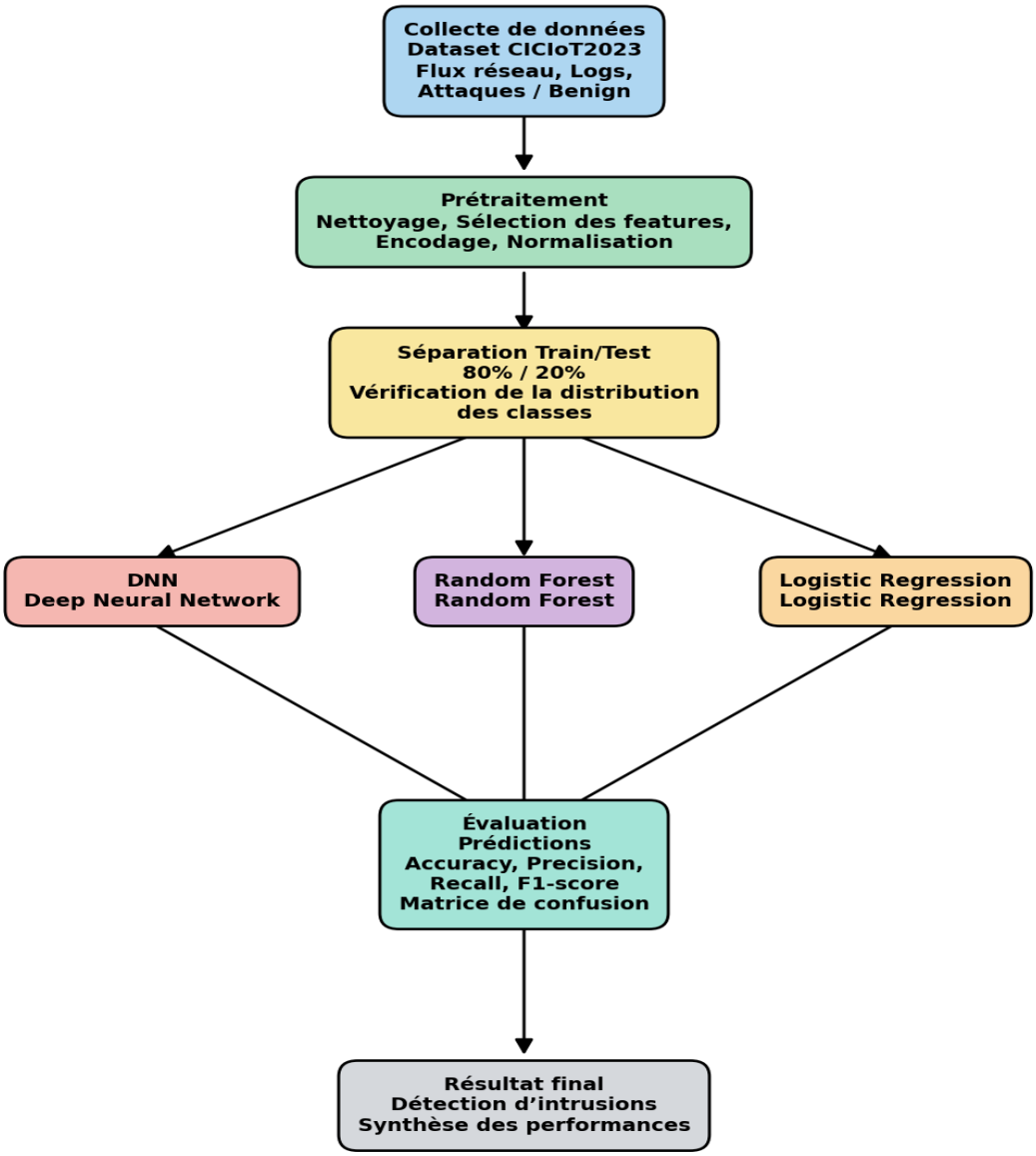


Figure 3.1 : Flux de traitement pour la détection précoce d'intrusions IoT à l'aide du dataset CICIoT2023

3.7. METRIQUES D'EVALUATIONS

L'objectif principal de tout algorithme de classification est de s'appuyer sur des données étiquetées afin de construire un modèle capable d'identifier avec la plus grande précision possible la classe d'appartenance de nouvelles données non étiquetées. Pour évaluer la performance et la fiabilité de ces modèles, plusieurs métriques sont utilisées. Parmi les plus couramment employées figurent l'Accuracy (exactitude**)**, la Precision (précision), le Recall

(rappel) et le F1-score, qui permettent d'obtenir une vision complète de l'efficacité du modèle en termes de détection correcte des classes.[48]

3.7.1. Exactitude (Accuracy)

L'exactitude représente la proportion totale de prédictions correctes, qu'elles soient positives ou négatives, par rapport à l'ensemble des échantillons observés. Elle permet donc d'évaluer la performance globale d'un modèle de classification.

Elle est définie par la formule suivante :

$$\mathbf{Accuracy} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

Où :

- **TP (True Positives)** : nombre de cas positifs correctement prédits,
- **TN (True Negatives)** : nombre de cas négatifs correctement prédits,
- **FP (False Positives)** : nombre de cas négatifs incorrectement prédits comme positifs,
- **FN (False Negatives)** : nombre de cas positifs incorrectement prédits comme négatifs.

3.7.2. Précision (Precision)

La précision indique la proportion de prédictions correctes parmi l'ensemble des instances que le modèle a identifiées comme appartenant à une classe donnée (par exemple, les attaques). Elle mesure donc la capacité du modèle à éviter les fausses alertes. Il répond à la question suivante : « lorsque le modèle prédit VRAI, combien de fois a-t-il eu raison ? Elle est définie par la relation suivante :

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}}$$

3.7.3. Rappel (Recall)

Le rappel mesure la capacité du modèle à identifier correctement toutes les instances positives, c'est-à-dire la proportion d'attaques détectées parmi l'ensemble réel des attaques présentes dans le jeu de données. Il répond à la question suivante : « lorsque la classe est VRAI, combien de

fois le classificateur s'est trompé ? Il permet d'évaluer le taux de détection du modèle et se calcule selon la formule suivante :

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}}$$

3.7.4. Score F1 (F1-Score)

Le score F1 correspond à la moyenne harmonique entre la précision et le rappel. Il permet de trouver un équilibre entre ces deux métriques, notamment lorsque les classes sont déséquilibrées. Un score F1 élevé indique que le modèle maintient à la fois une bonne précision et un bon rappel. Il est calculé selon la formule suivante :

$$\mathbf{F1} = \mathbf{2} \times \frac{\mathbf{Precision} \times \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}}$$

3.7.5. Matrice de confusion

Elle est une représentation tabulaire couramment utilisée pour évaluer la performance d'un algorithme d'apprentissage automatique appliqué à une tâche de classification. Elle permet de visualiser, pour chaque classe, le nombre de prédictions correctes et incorrectes, offrant ainsi une compréhension fine des erreurs commises par le modèle.

3.8. RESULTATS DES MODELES D'APPRENTISSAGES AUTOMATIQUES ET PROFOND

Dans cette section, nous présentons les performances des modèles d'apprentissage automatique (Machine Learning, ML) et d'apprentissage profond (Deep Learning, DL) appliqués à la détection d'intrusions dans les environnements IoT. L'évaluation repose sur plusieurs métriques classiques telles que l'Accuracy, la Precision, le Recall et le F1-Score. Ces indicateurs permettent d'apprécier la capacité des modèles à identifier les comportements anormaux tout en tenant compte des contraintes computationnelles propres aux dispositifs IoT.

3.8.1. Résultats des modèles d'apprentissage automatique (ML)

3.8.1.1. Régression Logistique (LR)

Tableau 3.1 : Performance du modèle régression logistique

Modèle	Accuracy	Precision	Recall	F1-score
LR	0,9769469288379	0,9999938677260	0,9764023559722	0,9880573102679

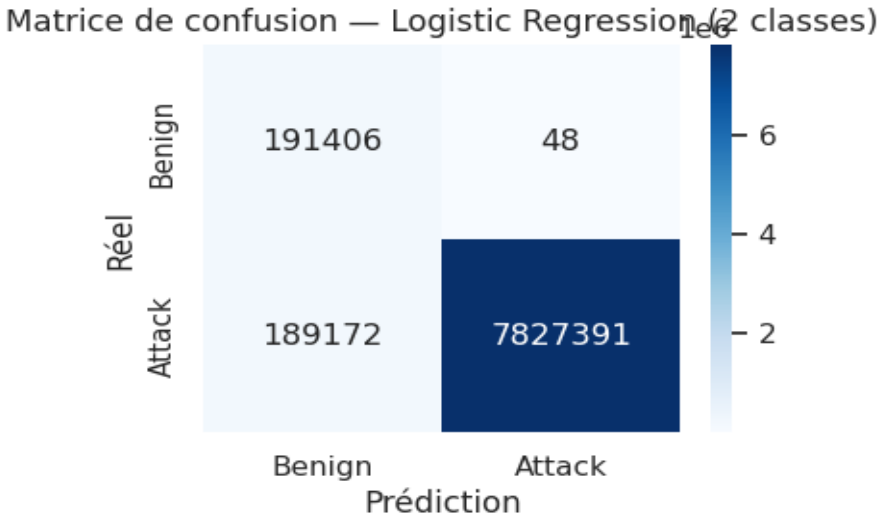


Figure 3.2 : Matrice de confusion Logistic Regression

Interprétation : La régression logistique (LR) obtient une accuracy de 97,69 %**,** avec une precision quasi parfaite (≈ 100 %)**,** indiquant un très faible taux de fausses alertes. En revanche, son recall de 97,64 % révèle une capacité légèrement réduite à détecter l'ensemble des attaques, ce qui se traduit par un F1-score de 98,81 **%**. Ce comportement suggère un modèle conservateur, privilégiant la réduction des faux positifs au détriment de la détection complète des intrusions.

3.8.1.2. Machine à vecteurs de support (SVM)

Tableau 3.2 : Performance du modèle SVM

Modèle	Accuracy	Precision	Recall	F1-score
SVM	0,9767067977563	0,9999975720122	0,9761528724966	0,9879313643572

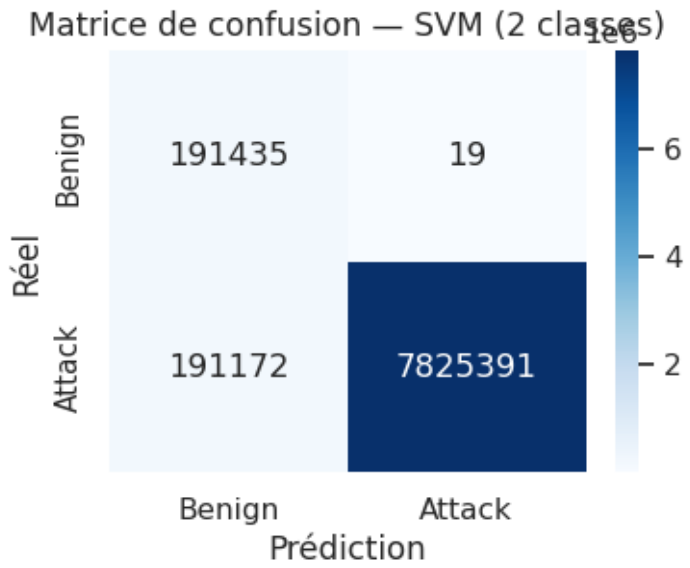


Figure 3.3 : Matrice de confusion SVM

Interprétation : Le SVM présente des performances très proches de celles de la régression logistique, avec une accuracy de 97,67 % et une precision également proche de 100 %**.** Son recall (97,62 %) et son F1-score (98,79 %) confirment une bonne capacité de classification, bien que légèrement inférieure à celle des modèles plus complexes. Ces résultats indiquent une forte fiabilité, mais une sensibilité modérée face à certaines attaques.

3.8.1.3. Réseau de neurone artificielle (ANN)

Tableau 3.3 : Performance du modèle ANN

Modèle	Accuracy	Precision	Recall	F1-score
ANN	0,9902291869034	0,9957884899385	0,9942006318667	0,9949939274084

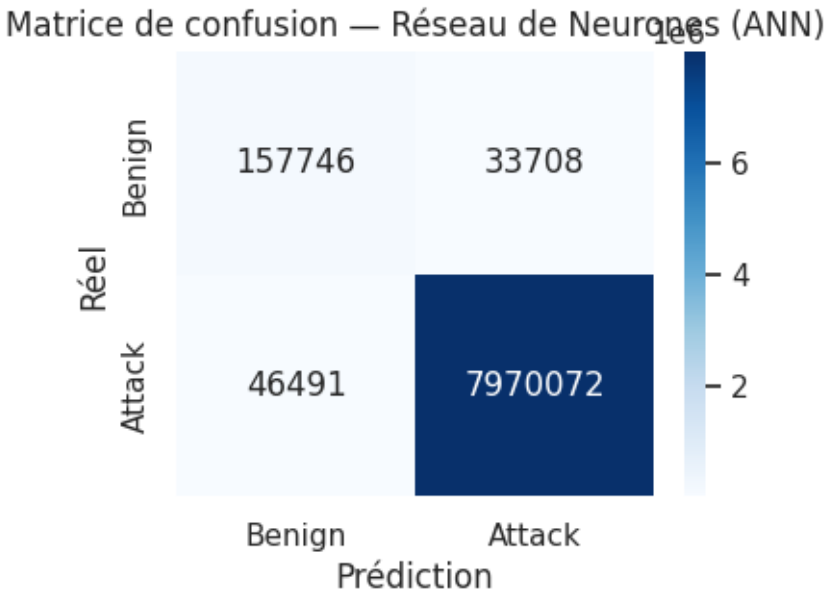


Figure 3.4 : Matrice de confusion ANN

Interprétation : Le réseau de neurones artificiel (ANN) affiche une nette amélioration, atteignant une accuracy de 99,02 %**.** Les valeurs élevées de precision (99,58 %) et de recall (99,42 %) traduisent une excellente capacité à distinguer les comportements normaux des activités malveillantes. Le F1-score de 99,50 % met en évidence un très bon équilibre entre détection et réduction des fausses alarmes.

3.8.1.4. Arbre de décision (Decision Tree -DT)

Tableau 3.4 : Performance du modèle Arbre de décision

Modèle	Accuracy	Precision	Recall	F1-score
DT	0,9885557010907	0,9985760056063	0,9896937128791	0,9941150190965

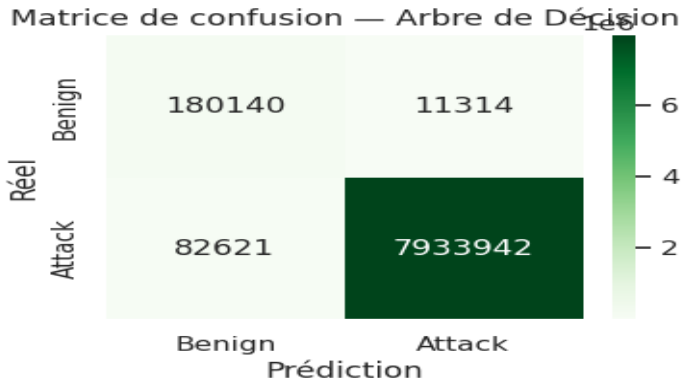


Figure 3.5 : Matrice de confusion Arbre de décision

Interprétation : L'arbre de décision obtient également des résultats solides avec une accuracy de 98,86 %. Sa precision élevée (99,86 %) et son recall de 98,97 % montrent une bonne efficacité dans la détection des intrusions, tandis que le F1-score de 99,41 % confirme sa pertinence. Toutefois, ce modèle peut rester sensible au surapprentissage, en particulier dans des environnements IoT hétérogènes.

3.8.1.5. Random Forest

Tableau 3.5 : Performance du modèle Random Forest

Modèle	Accuracy	Precision	Recall	F1-score
Random Forest	0,9920448507843	0,9958731221096	0,995982193366	0,9959276547517

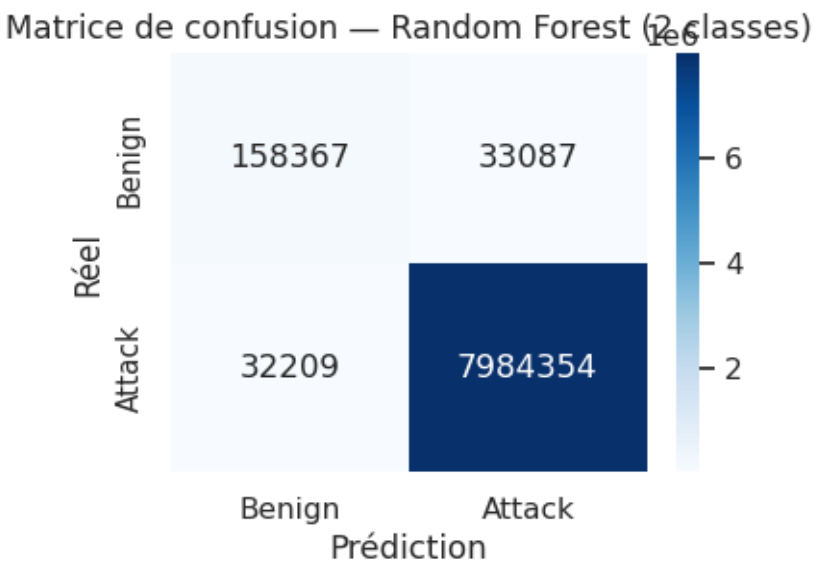


Figure 3.6 : Matrice de confusion Random Forest

Interprétation : Le Random Forest se distingue comme le modèle le plus performant. Il atteint une accuracy de 99,20 %, accompagnée de valeurs élevées de precision (99,59 %) et de recall (99,60 %). Son F1-score de 99,59 % reflète un excellent compromis entre sensibilité et spécificité. Grâce à sa structure en ensemble d'arbres, le Random Forest offre une meilleure robustesse, une bonne généralisation et une résistance accrue au bruit, ce qui le rend particulièrement adapté à la détection d'intrusions dans les environnements IoT.

3.8.2. Résultats des modèles d'apprentissage profond (DL)

3.8.2.1. Réseaux de neurones profonds (DNN)

Tableau 3.6 : Performance du modèle Réseaux de neurones profonds (DNN)

Modèle	**Accuracy**	**Precision**	**Recall**	**F1-score**
DNN	0,9914761385118	0,9960346436541	0,9952347408733	0,995634531601

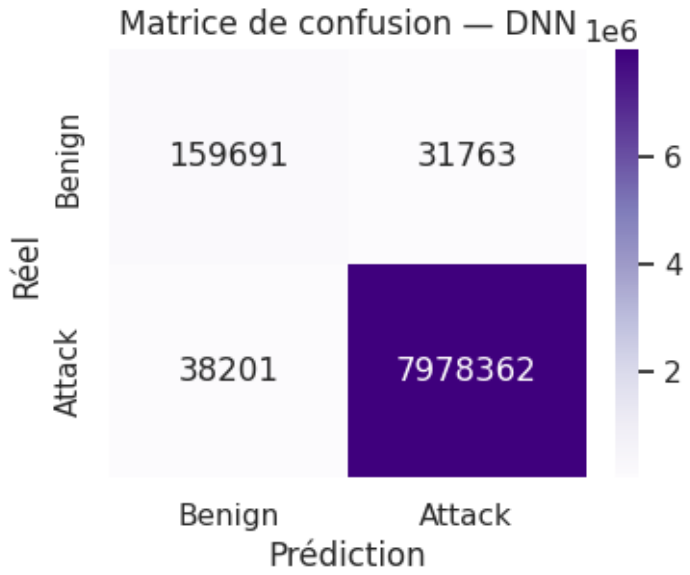


Figure 3.7 : Matrice de confusion DNN

Interprétation : Le Deep Neural Network (DNN) atteint une accuracy de 99,15 %, avec une precision de 99,60 % et un recall de 99,52 %. Son F1-score de 99,56 **%** traduit un excellent équilibre entre la capacité à détecter les attaques et la limitation des fausses alertes. Ces résultats confirment l'efficacité des DNN à apprendre des représentations complexes directement à partir des données réseau, ce qui est particulièrement pertinent dans des environnements IoT caractérisés par une forte hétérogénéité du trafic.

3.8.2.2. Réseaux de neurones convolutifs (CNN)

Tableau 3.7 : Performance du modèle Réseaux de neurones convolutifs (CNN)

Modèle	**Accuracy**	**Precision**	**Recall**	**F1-score**
CNN	0,9915503342646	0,9965969595339	0,9947452542941	0,9956702459840

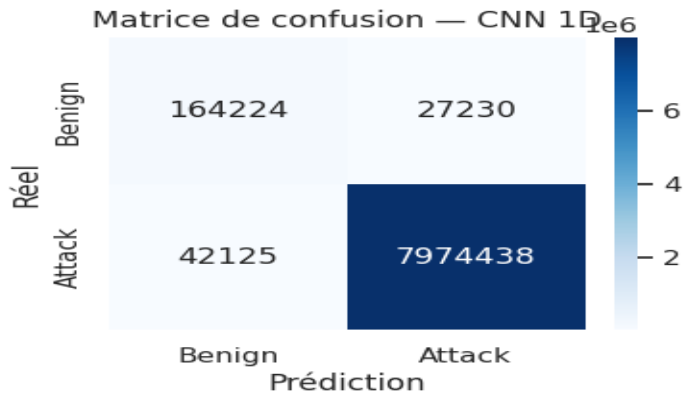


Figure 3.8 : Matrice de confusion CNN

Interprétation : Le Convolutional Neural Network (CNN) obtient des performances très proches, avec une accuracy légèrement supérieure de 99,16 %. Il se distingue par une précision élevée de 99,66 %, indiquant une excellente fiabilité dans la classification des flux malveillants. Son recall de 99,47 % reste très élevé, bien qu'un léger compromis apparaisse par rapport à la précision. Le F1-score de 99,57 % confirme la robustesse globale du modèle. La capacité des CNN à exploiter automatiquement des motifs locaux dans les données contribue à une détection fine des attaques, sans nécessiter une ingénierie manuelle approfondie des caractéristiques.

3.9. Analyse des performances des modèles d'apprentissage automatique (ML) et profond (DL)

3.9.1. Analyse des performances des modèles d'apprentissage automatique (ML)

La figure 3.9 illustre une comparaison des performances des modèles d'apprentissage profond selon quatre métriques clés : l'Accuracy, la Precision, le Recall et le F1-Score. Chaque barre

représente le score atteint par un modèle spécifique pour une métrique donnée.

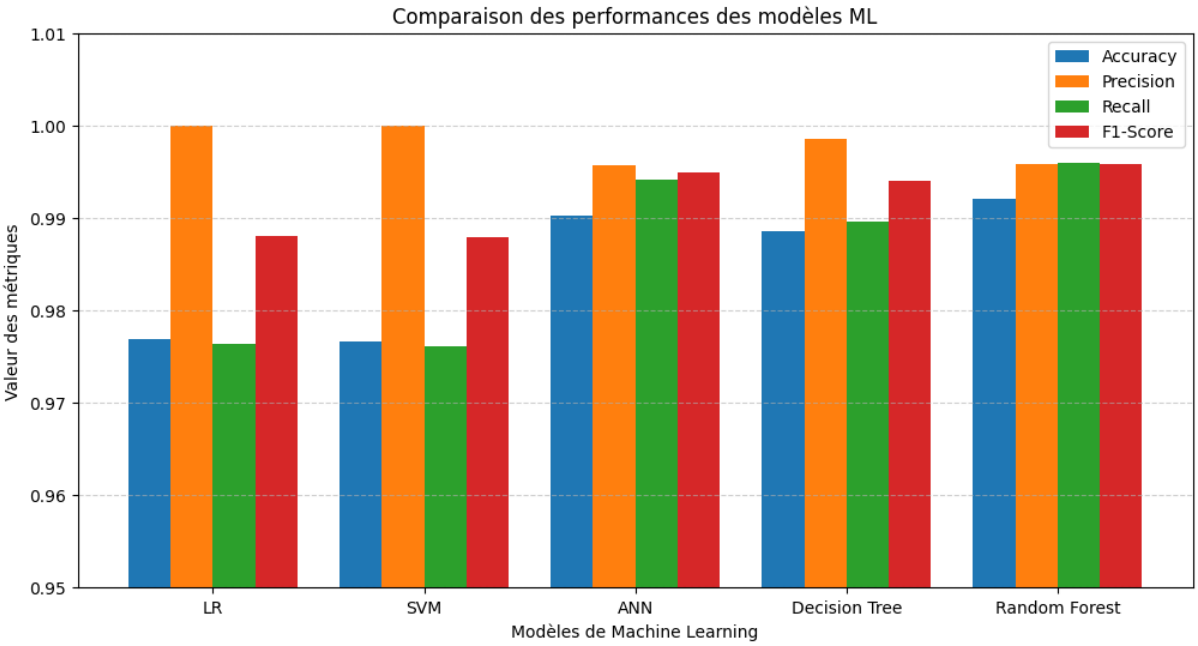


Figure 3.9 : Comparaison des performances des modèles ML

Nous observons que les modèles Random forest, Réseau de neurones artificielle, Décision Tree ont globalement de bonnes performances. Par contre les modèles Logistique régression et SVM ont des précisions bonnes mais le F1-score un peu au-dessous, ces deux modèles ont des problèmes à équilibrer la précision et F1-score.

3.9.2. Analyse des performances des modèles d'apprentissage profond (DL)

La figure 3.10 illustre une analyse comparative des performances des différents modèles d'apprentissage profond à travers quatre métriques clés : l'Accuracy, la Precision, le Recall et le F1-Score. Les barres associées à chaque modèle représentent les valeurs obtenues pour chacune de ces métriques.

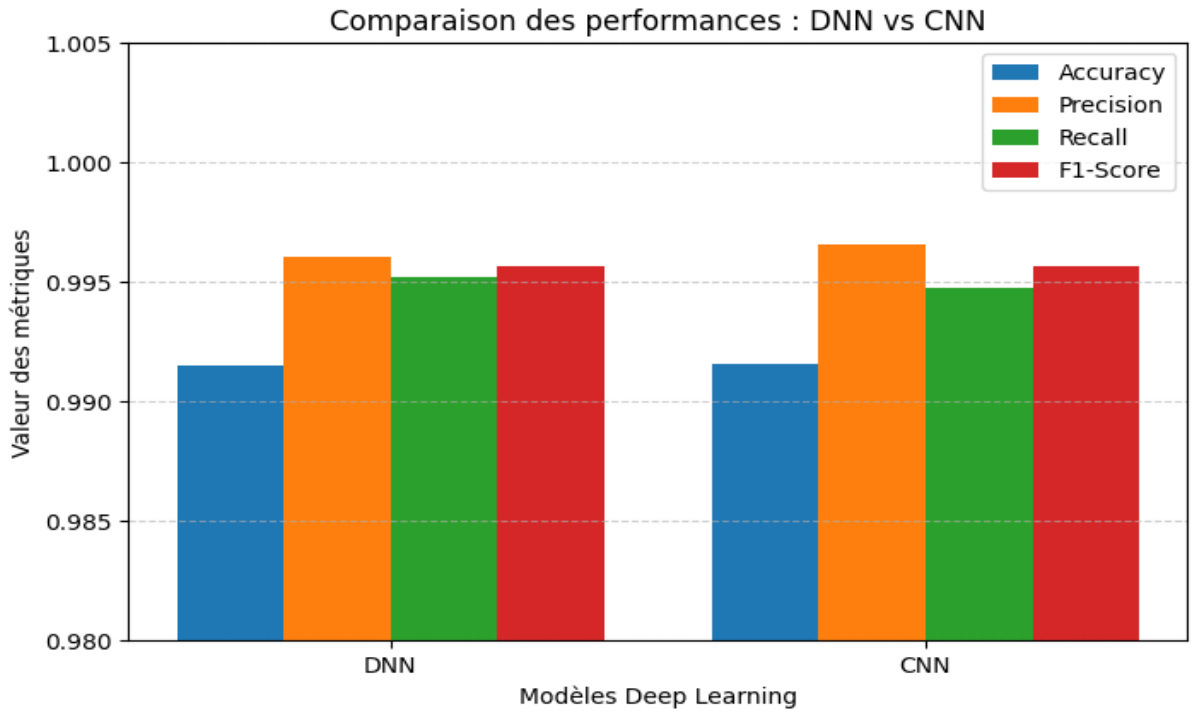


Figure 3.10 : Comparaison des performances des modèles DL

Nous observons ici que les deux algorithmes ont quasiment les mêmes performances à par une légère différence notée sur précision, celle de CNN surpasse un celle de DNN. Ils ont de bonne performance, ce qui montre leur capacité à bien détecter les comportements anormaux.

3.9.3. Comparaison des modèles ML et DL

Les résultats mettent en évidence une supériorité globale des modèles d'apprentissage automatique, en particulier le Random Forest et l'arbre de décision, qui surpassent nettement les modèles d'apprentissage profond en termes de performances et d'adaptabilité aux contraintes des environnements IoT. Les résultats obtenus indiquent que, dans le contexte spécifique de la détection d'intrusions au sein des environnements IoT, les modèles d'apprentissage automatique (ML) se révèlent globalement plus performants et mieux adaptés. Ils présentent un compromis équilibré entre précision de détection, rapidité d'exécution et simplicité de mise en œuvre. Bien que les modèles d'apprentissage profond (DL) affichent des performances prometteuses, leurs exigences élevées en ressources computationnelles limitent encore leur pertinence pour des déploiements en temps réel dans des environnements IoT contraints, sauf en cas d'optimisation avancée.

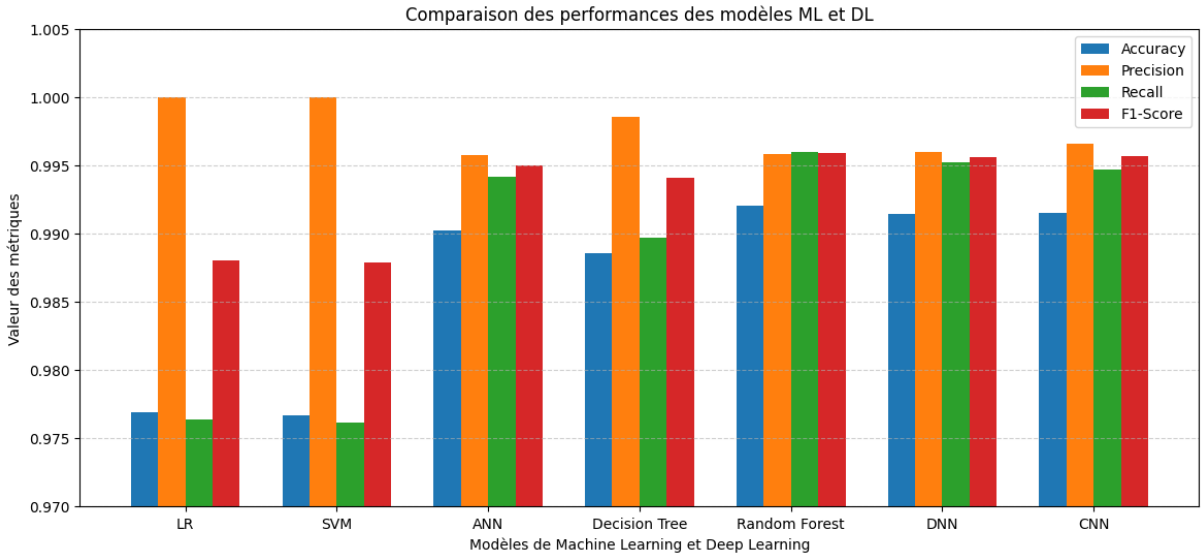


Figure 3.11 : Comparaison des performances des modèles ML et DL

3.10. CONCLUSION

Ce chapitre a permis d'évaluer en détail la performance de plusieurs modèles d'apprentissage automatique et profond pour la détection d'intrusions dans les environnements IoT. Les modèles d'apprentissage automatique, en particulier Forest se distingue par leurs performances globales élevées, leur rapidité d'exécution et leur faible complexité. À l'inverse, les modèles d'apprentissage profond, bien que prometteurs sur le plan théorique, n'ont pas atteint les mêmes niveaux de performance, en raison de leur complexité, de leur sensibilité au surapprentissage et de leur coût computationnel important. Ces observations suggèrent une préférence pour l'utilisation des modèles ML dans les contextes IoT, où légèreté, rapidité et précision constituent des contraintes majeures. Toutefois, les avancées récentes en deep learning offrent des perspectives intéressantes pour de futurs travaux dans ce domaine.

CONCLUSION GENERALE

L'Internet des Objets (IoT) occupe aujourd'hui une place stratégique dans la transformation numérique, touchant de nombreux secteurs d'activité. Cependant, la généralisation de la connectivité s'accompagne d'une augmentation significative des risques en matière de cybersécurité, rendant les systèmes de détection d'intrusion (Intrusion Detection Systems – IDS) indispensables pour la protection des réseaux IoT.

Ce mémoire a porté sur la conception et l'évaluation d'un système de détection précoce des intrusions dans les réseaux IoT, fondé sur une approche d'intelligence artificielle centralisée. L'étude a été structurée en trois chapitres : une présentation des notions fondamentales liées aux réseaux IoT et aux IDS, un état de l'art des approches basées sur le Machine Learning et le Deep Learning, puis une phase d'implémentation, de tests et d'évaluation comparative des modèles étudiés.

Les résultats obtenus montrent que les modèles de Machine Learning, en particulier Random Forest, offrent d'excellentes performances en matière de détection des intrusions, tout en garantissant une rapidité d'exécution et une simplicité d'implémentation. À l'inverse, les modèles de Deep Learning, bien que puissants sur le plan théorique, ne parviennent pas à surpasser les approches ML dans ce contexte, en raison de leurs besoins plus élevés en ressources computationnelles.

Ces résultats soulignent l'importance de tenir compte des contraintes spécifiques des environnements IoT lors du choix des solutions de détection. En conclusion, ce travail met en évidence que les approches basées sur le Machine Learning constituent actuellement la solution la plus adaptée pour des déploiements réels dans des environnements IoT contraints. En perspective, des améliorations peuvent être envisagées, notamment l'optimisation des modèles pour des architectures distribuées, l'intégration de mécanismes adaptatifs et l'extension de l'approche proposée à des scénarios critiques tels que la santé connectée ou l'Industrie 4.0.

BIBLIOGRAPHIE

[1] S. Rangasamy, K. Rajamohan, V. S. Lavan, C. Mayur, et M. F. Lalitha, « Evolutionized industry with the internet of things », in *Handbook of Research on Machine Learning-Enabled IoT for Smart Applications Across Industries*, IGI Global, 2023, p. 407-434. Consulté le: 8 juillet 2025. [En ligne]. Disponible sur: https://www.igi-global.com/chapter/evolutionized-industry-with-the-internet-of-things/326008

[2] J. E. Ibarra-Esquer, F. F. González-Navarro, B. L. Flores-Rios, L. Burtseva, et M. A. Astorga-Vargas, « Tracking the evolution of the internet of things concept across different application domains », *Sensors*, vol. 17, n° 6, p. 1379, 2017.

[3] A. S. Gillis, « What is internet of things (IoT) », *IoT Agenda*, vol. 17, p. 2024, 2021.

[4] J. Fruhlinger, « What is IoT? The internet of things explained », *Netw. World*, 2020.

[5] P. Jain, F. Adrangi, et M. Venkatachalam, « Cellular IoT network architecture », 28 janvier 2020 Consulté le: 8 juillet 2025. [En ligne]. Disponible sur: https://patents.google.com/patent/US10548000B2/en

[6] A. Ghosh, D. Chakraborty, et A. Law, « Artificial intelligence in Internet of things », *CAAI Trans. Intell. Technol.*, vol. 3, n° 4, p. 208-218, déc. 2018, doi: 10.1049/trit.2018.1008.

[7] I.-T. S. Group, « New ITU standards define the Internet of Things and provide the blueprints for its development ». ITU, 2012.

[8] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, et B. Stiller, « Landscape of IoT security », *Comput. Sci. Rev.*, vol. 44, p. 100467, 2022.

[9] M. Humayun, « Role of emerging IoT big data and cloud computing for real time application », *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, n° 4, 2020, Consulté le: 8 juillet 2025. [En ligne]. Disponible sur: https://search.proquest.com/openview/26adfa3bc7e3508cb73c995ffcf9f21a/1?pq-origsite=gscholar&cbl=5444811

[10] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, et H.-Y. Du, « Research on the architecture of Internet of Things », in *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, IEEE, 2010, p. V5-484. Consulté le: 28 juin 2025. [En ligne]. Disponible sur: https://ieeexplore.ieee.org/abstract/document/5579493/

[11] K. Ashton, « I could be wrong, but I'm fairly sure the phrase "Internet of Things" started life as the title of a presentation I made at Procter & Gamble (P&G) in 1999 », *RFID J.*, vol. 22, p. 1-10, 2009.

[12] N. Naik, « Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP », in *2017 IEEE international systems engineering symposium (ISSE)*, IEEE, 2017, p. 1-7. Consulté le: 28 juin 2025. [En ligne]. Disponible sur: https://ieeexplore.ieee.org/abstract/document/8088251/

[13] I. Malloug et H. B. Rabeh, « Gateway IoT de pilotage et de surveillance des capteurs domestiques via le protocole MQTT », in *Colloque sur les Objets et systèmes Connectés 2022*, 2022. Consulté le: 28 juin 2025. [En ligne]. Disponible sur: https://hal.science/hal-04397658/document

[14] F. Merabet, « Solutions de sécurité pour l'internet des objets dans le cadre de l'assistance à l'autonomie à domicile », PhD Thesis, Université de Limoges; Université Mouloud Mammeri (Tizi-Ouzou, Algérie), 2021. Consulté le: 28 juin 2025. [En ligne]. Disponible sur: https://theses.hal.science/tel-03326960/

[15] M. I. Malik, I. N. McAteer, P. Hannay, S. N. Firdous, et Z. Baig, « XMPP architecture and security challenges in an IoT ecosystem », 2018, Consulté le: 28 juin 2025. [En ligne]. Disponible sur: https://ro.ecu.edu.au/ism/219/

[16] K. GUEYE, « ETUDE DE LA CONVERGENCE DES RESEAUX LTE-A/IOT ET SES APPLICATIONS EN E-SANTE, E-FORMATION ET E-AGRICULTURE », PhD Thesis, Université Cheikh Anta Diop, 2020.

[17] J. Clément, « Objets connectés: stratégies et technologies pour une interaction réussie », 2018, Consulté le: 28 juin 2025. [En ligne]. Disponible sur: https://www.unifr.ch/inf/softeng/en/assets/public/files/research/students_projects/bachelor/Bachelor_Clement_Julien.pdf

[18] N. N. Srinidhi, S. D. Kumar, et K. R. Venugopal, « Network optimizations in the Internet of Things: A review », *Eng. Sci. Technol. Int. J.*, vol. 22, n° 1, p. 1-21, 2019.

[19] V. Alcácer et V. Cruz-Machado, « Scanning the industry 4.0: A literature review on technologies for manufacturing systems », *Eng. Sci. Technol. Int. J.*, vol. 22, n° 3, p. 899-919, 2019.

[20] « Une solution intelligente de gestion des déchets destinée aux citoyens ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://www.mdpi.com/1424-8220/20/8/2380

[21] R. Chataut, A. Phoummalayvane, et R. Akl, « Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0 », *Sensors*, vol. 23, n° 16, p. 7194, janv. 2023, doi: 10.3390/s23167194.

[22] H. Liao *et al.*, « A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things », *IEEE Access*, vol. 12, p. 4745-4761, 2024, doi: 10.1109/ACCESS.2023.3349287.

[23] M. T. A. Tashfeen, « Intrusion Detection System Using AI and Machine Learning Algorithm », in *Cyber Security for Next-Generation Computing Technologies*, 1re éd., Boca Raton: CRC Press, 2023, p. 120-140. doi: 10.1201/9781003404361-7.

[24] S. Agrawal *et al.*, « Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions », 16 juin 2021, *arXiv*: arXiv:2106.09527. doi: 10.48550/arXiv.2106.09527.

[25] J. Robert, « Intelligence Artificielle : Définition, histoire, enjeux », DataScientest. Consulté le: 7 novembre 2025. [En ligne]. Disponible sur: https://datascientest.com/intelligence-artificielle-definition

[26] « Machine Learning Tutorial », GeeksforGeeks. Consulté le: 8 novembre 2025. [En ligne]. Disponible sur: https://www.geeksforgeeks.org/machine-learning/machine-learning/

[27] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, et F. Ahmad, « Network intrusion detection system: A systematic study of machine learning and deep learning approaches », *Trans. Emerg. Telecommun. Technol.*, vol. 32, nº 1, p. e4150, 2021, doi: 10.1002/ett.4150.

[28] « Hanif : Détection d'intrusion dans l'IoT à l'aide de... - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Intrusion+Detection+In+IoT+Using+Artificial+Neural+Networks+On+UNSW-15+Dataset&conference=Proceedings+of+the+2019+IEEE+16th+International+Conference+on+Smart+Cities:+Improving+Quality+of+Life+Using+ICT+&+IoT+and+AI+(HONET-ICT)&author=Hanif,+S.&author=Ilyas,+T.&author=Zeeshan,+M.&publication_year=2019&pages=152%E2%80%93156&doi=10.1109/HONET.2019.8908122

[29] « Jamal : Détection et classification des logiciels malveillants dans l'IoT... - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Malware+Detection+and+Classification+in+IoT+Network+using+ANN&author=Jamal,+A.&author=Faisal,+H.M.&author=Nasir,+M.&publication_year=2022&journal=Mehran+Univ.+Res.+J.+Eng.+Technol.&volume=41&pages=80%E2%80%9391&doi=10.22581/muet1982.2201.08

[30] « Goeschel : Réduire les faux positifs dans la détection d'intrusion... - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Reducing+false+positives+in+intrusion+detection+systems+using+data-mining+techniques+utilizing+support+vector+machines,+decision+trees,+and+naive+Bayes+for+off-line+analysis&conference=Proceedings+of+the+SoutheastCon+2016&author=Goeschel,+K.&publication_year=2016&pages=1%E2%80%936&doi=10.1109/SECON.2016.7506774

[31] « Jing : Détection d'intrusion réseau basée sur SVM pour... - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=SVM+Based+Network+Intrusion+Detection+for+the+UNSW-NB15+Dataset&conference=Proceedings+of+the+2019+IEEE+13th+International+Conference+on+ASIC+(ASICON)&author=Jing,+D.&author=Chen,+H.B.&publication_year=2019&pages=1%E2%80%934&doi=10.1109/ASICON47005.2019.8983598

[32] « Abdaljabar : Un système de détection d'intrusion pour l'IoT... - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=An+Intrusion+Detection+System+for+IoT+Using+KNN+and+Decision-Tree+Based+Classification&conference=Proceedings+of+the+2021+International+Conference+of+Modern+Trends+in+Information+and+Communication+Technology+Industry+(MTICTI)&author=Abdaljabar,+Z.H.&author=Ucan,+O.N.&author=Alheeti,+K.M.A.&publication_year=2021&pages=1%E2%80%935&doi=10.1109/MTICTI53925.2021.9664772

[33] « Aref : Apprentissage par renforcement multi-agents... - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Multi-Agent+Reinforcement+Learning+Based+Cognitive+Anti-Jamming&conference=Proceedings+of+the+2017+IEEE+Wireless+Communications+and+Networking+Conference+(WCNC)&author=Aref,+M.A.&author=Jayaweera,+S.K.&author=Machuzak,+S.&publication_year=2017&pages=1%E2%80%936&doi=10.1109/WCNC.2017.7925694

[34] « Sambangi : une approche d'apprentissage automatique pour les DDoS (distribution... - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=+A+Machine+Learning+Approach+for+DDoS+(Distributed+Denial+of+Service)+Attack+Detection+Using+Multiple+Linear+Regression&author=Sambangi,+S.&author=Gondi,+L.&publication_year=2020&journal=Proceedings&volume=63&pages=51&doi=10.3390/proceedings2020063051

[35] « Qaddoura : une approche de classification en plusieurs étapes... - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=A+Multi-Stage+Classification+Approach+for+IoT+Intrusion+Detection+Based+on+Clustering+with+Oversampling&author=Qaddoura,+R.&author=Al-Zoubi,+A.M.&author=Almomani,+I.&author=Faris,+H.&publication_year=2021&journal=Appl.+Sci.&volume=11&pages=3022&doi=10.3390/app11073022

[36] « Rai : Algorithme basé sur un arbre de décision pour la détection d'intrusion - Google Scholar ». Consulté le: 15 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Decision+Tree+Based+Algorithm+for+Intrusion+Detection&author=Rai,+K.&author=Devi,+M.S.&author=Guleria,+A.&publication_year=2016&journal=Adv.+Netw.+Appl.&volume=7&pages=2828%E2%80%932834

[37] « Muniyandi : Détection d'anomalies réseau par cascade... - Google Scholar ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Network+Anomaly+Detection+by+Cascading+K-Means+Clustering+and+C4.5+Decision+Tree+Algorithm&author=Muniyandi,+A.P.&author=Rajeswari,+R.&author=Rajaram,+R.&publication_year=2012&journal=Procedia+Eng.&volume=30&pages=174%E2%80%93182&doi=10.1016/j.proeng.2012.01.849

[38] Y. Meidan *et al.*, « ProfilIoT: a machine learning approach for IoT device identification based on network traffic analysis », in *Proceedings of the Symposium on Applied Computing*, Marrakech Morocco: ACM, avr. 2017, p. 506-509. doi: 10.1145/3019612.3019878.

[39] « Ahmad: Anomaly detection using deep neural network... - Google Scholar ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Anomaly+Detection+Using+Deep+Neu

ral+Network+for+IoT+Architecture&author=Ahmad,+Z.&author=Khan,+A.S.&author=Nisar,+K.&author=Haider,+I.&author=Hassan,+R.&author=Haque,+M.R.&author=Tarmizi,+S.&author=Rodrigues,+J.J.P.C.&publication_year=2021&journal=Appl.+Sci.&volume=11&pages=7050&doi=10.3390/app11157050

[40] « Kim: Method of intrusion detection using deep neural network - Google Scholar ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Method+of+intrusion+detection+using+deep+neural+network&conference=Proceedings+of+the+2017+IEEE+International+Conference+on+Big+Data+and+Smart+Computing+(BigComp)&author=Kim,+J.&author=Shin,+N.&author=Jo,+S.Y.&author=Kim,+S.H.&publication_year=2017&pages=313%E2%80%93316&doi=10.1109/BIGCOMP.2017.7881684

[41] « Zhu : Algorithme de détection de valeurs aberrantes approximatives basé sur KNN... - Google Scholar ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=KNN-Based+Approximate+Outlier+Detection+Algorithm+over+IoT+Streaming+Data&author=Zhu,+R.&author=Ji,+X.&author=Yu,+D.&author=Tan,+Z.&author=Zhao,+L.&author=Li,+J.&author=Xia,+X.&publication_year=2020&journal=IEEE+Access&volume=8&pages=42749%E2%80%9342759&doi=10.1109/ACCESS.2020.2977114

[42] « Chen : Un nouveau système de détection d'intrusion réseau... - Google Scholar ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=A+Novel+Network+Intrusion+Detection+System+Based+on+CNN&conference=Proceedings+of+the+2020+Eighth+International+Conference+on+Advanced+Cloud+and+Big+Data+(CBD)&author=Chen,+L.&author=Kuang,+X.&author=Xu,+A.&author=Suo,+S.&author=Yang,+Y.&publication_year=2020&pages=243%E2%80%93247&doi=10.1109/CBD51900.2020.00051

[43] « Park : prédiction basée sur RNN pour la détection d'intrusion dans le réseau - Google Scholar ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=RNN-Based+Prediction+for+Network+Intrusion+Detection&conference=Proceedings+of+the+2020+International+Conference+on+Artificial+Intelligence+in+Information+and+Communication+(ICAIIC)&author=Park,+S.H.&author=Park,+H.J.&author=Choi,+Y.J.&publication_year=2020&pages=572%E2%80%93574&doi=10.1109/ICAIIC48513.2020.9065249

[44] « Tang : Réseau de neurones récurrents profonds pour l'intrusion... - Google Scholar ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=Deep+Recurrent+Neural+Network+for+Intrusion+Detection+in+SDN-Based+Networks&conference=Proceedings+of+the+2018+4th+IEEE+Conference+on+Network+Softwarization+and+Workshops+(NetSoft)&author=Tang,+T.A.&author=Mhamdi,+L.&author=McLernon,+D.&author=Zaidi,+S.A.R.&author=Ghogho,+M.&publication_year=2018&pages=202%E2%80%93206&doi=10.1109/NETSOFT.2018.8460090

[45] « Torres : Une analyse des réseaux neuronaux récurrents... - Google Scholar ». Consulté le: 16 novembre 2025. [En ligne]. Disponible sur: https://scholar.google.com/scholar_lookup?title=An+analysis+of+Recurrent+Neural+Networks+for+Botnet+detection+behavior&conference=Proceedings+of+the+2016+IEEE+Biennial+Congress+of+Argentina+(ARGENCON)&author=Torres,+P.&author=Catania,+C.&author=Garcia,+S.&author=Garino,+C.G.&publication_year=2016&pages=1%E2%80%936&doi=10.1109/ARGENCON.2016.7585247

[46] A. Jaillet, « Google Colab : la force du cloud pour l'apprentissage automatique », DataScientest. Consulté le: 10 novembre 2025. [En ligne]. Disponible sur: https://datascientest.com/google-colab-tout-savoir

[47] « 2015- Guido van Rossum Tutorialde Python3-3.pdf ». Consulté le: 11 novembre 2025. [En ligne]. Disponible sur: https://bida.uclv.edu.cu/bitstream/handle/123456789/17626/2015-%20Guido%20van%20Rossum%20Tutorialde%20Python3-3.pdf?sequence=1&isAllowed=y

[48] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, et A. A. Ghorbani, « CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment », *Sensors*, vol. 23, n° 13, p. 5941, janv. 2023, doi: 10.3390/s23135941.