Université Assane SECK de Ziguinchor

UFR Sciences et Technologies

Département Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master

Mention : Informatique

Spécialité : Réseaux et Systèmes

Sujet : L'Intelligence artificielle pour la détection d'intrusions dans l'Internet des Objets : le cas du Protocole MQTT

Présenté par :

Mlle Amy THIANDOUM

Sous la direction de

PR Youssou FAYE

Mémoire soutenu le 14 Décembre 2024 devant le jury composé de :

Pr Khadim	DRAME	Maître de Conférences CAMES	Président
Pr Youssou	FAYE	Maître de Conférences CAMES	Encadrant
Pr Ousmane	DIALLO	Maître de Conférences CAMES	Rapporteur
Pr Ibrahima	DIOP	Maître de Conférences. CAMES	Examinateur

Année 2023/2024

Résumé

L'Internet des Objets (IoT) s'intègre progressivement dans presque tous les aspects de notre vie quotidienne et professionnelle, facilitant la collecte, l'échange et l'analyse de données. La prolifération des objets connectés s'accompagne de nouveaux défis pour la sécurité : Les capteurs sur les objets connectés très limités en ressource, l'incompatibilité des technologies IoT, la sécurité non prise en compte au départ pour certains protocoles etc…. Aujourd'hui, de nombreux travaux de recherche se développent autour de l'IoT pour répondre aux enjeux de sécurité. Dans ce mémoire, après avoir présenté les bases théoriques des technologies IoT, telles que les réseaux de capteurs sans fil, la RFID et l'architecture IoT, nous avons abordé les défis spécifiques notamment les vulnérabilités des protocoles de communication comme MQTT et CoAP, exposés à des cybermenaces en raison de leur simplicité.

Même si des solutions cryptographiques existent, l'intelligence artificielle, notamment l'apprentissage automatique, se révèle très prometteuse en offrant des méthodes avancées pour détecter les anomalies, classer les menaces et prévenir les attaques pour renforcer la sécurité de l'IoT. Ainsi, une analyse des méthodes de surveillance, de détection et de classification des menaces, nous a permis de démontrer l'efficacité de l'apprentissage automatique pour prévenir les intrusions et améliorer la sécurité dans IoT en particulier du protocole MQTT. Une étude comparative de plusieurs modèles de l'apprentissage automatique fait ressortir l'efficacité XGBoost et CNN-LSTM quant à la détection d'intrus dans le protocole MQTT.

Une implémentation des deux modèles sur des données de test et de validation d'un ensemble de données provenant de Kaggle[5] nous a permis de montrer que le modèle XGBoost bien qu'offrant des performances légèrement inférieures à CNN-LSTM pour la détection de séquences complexes, s'est révélé plus adapté aux environnements IoT en raison de sa rapidité et de sa faible consommation de ressources. Cette solution optimise non seulement la sécurité des communications via MQTT, mais elle est également viable pour les dispositifs IoT à capacité limitée.

Mots clés : Internet des objets, sécurité, détection d'intrus, Intelligence artificielle, Machine Learning, XGBoost, CNN-LSTM, MQTT

Abstract

L'Internet des Objets (IoT) s'intègre progressivement dans presque tous les aspects de notre vie quotidienne et professionnelle, facilitant la collecte, l'échange et l'analyse de données. La prolifération des objets connectés s'accompagne de nouveaux défis pour la sécurité : Les capteurs sur les objets connectés très limités en ressource, l'incompatibilité des technologies IoT, la sécurité non prise en compte au départ pour certains protocoles etc…. Aujourd'hui, de nombreux travaux de recherche se développent autour de l'IoT pour répondre aux enjeux de sécurité. Dans ce mémoire, après avoir présenté les bases théoriques des technologies IoT, telles que les réseaux de capteurs sans fil, la RFID et l'architecture IoT, nous avons abordé les défis spécifiques notamment les vulnérabilités des protocoles de communication comme MQTT et CoAP, exposés à des cybermenaces en raison de leur simplicité.

Même si des solutions cryptographiques existent, l'intelligence artificielle, notamment l'apprentissage automatique, se révèle très prometteuse en offrant des méthodes avancées pour détecter les anomalies, classer les menaces et prévenir les attaques pour renforcer la sécurité de l'IoT. Ainsi, une analyse des méthodes de surveillance, de détection et de classification des menaces, nous a permis de démontrer l'efficacité de l'apprentissage automatique pour prévenir les intrusions et améliorer la sécurité dans IoT en particulier du protocole MQTT. Une étude comparative de plusieurs modèles de l'apprentissage automatique fait ressortir l'efficacité XGBoost et CNN-LSTM quant à la détection d'intrus dans le protocole MQTT.

Une implémentation des deux modèles sur des données de test et de validation d'un ensemble de données provenant de Kaggle[5] nous a permis de montrer que le modèle XGBoost bien qu'offrant des performances légèrement inférieures à CNN-LSTM pour la détection de séquences complexes, s'est révélé plus adapté aux environnements IoT en raison de sa rapidité et de sa faible consommation de ressources. Cette solution optimise non seulement la sécurité des communications via MQTT, mais elle est également viable pour les dispositifs IoT à capacité limitée.

Key words: Internet of Things, Artificial intelligence, Supervised learning, Deep learning, XGBoost, CNN-LSTM, MQTT

REMERCIEMENT

Tout d'abord, je remercie DIEU, pour m'avoir accordé la patience, la santé et la détermination nécessaires à l'accomplissement de ce travail.

Ensuite, j'adresse mes remerciements les plus sincères à mon encadreur, le **Pr Youssou FAYE**, pour l'orientation, la confiance, le temps qu'il m'a accordé, la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené. Merci infiniment pour tout le savoir que vous m'avez transmis et pour votre bienveillance tout au long de ce parcours.

Je suis particulièrement reconnaissant à :

- Monsieur Khadim DRAME, Maître de conférences à l'Université Assane Seck de Ziguinchor, pour avoir consacré du temps à l'évaluation de ce travail ;
- Monsieur Ibrahima DIOP, Maître de conférences à l'Université Assane Seck de Ziguinchor, pour son implication et ses observations ;
- Monsieur Ousmane DIALLO, Maître de conférences à l'Université Assane Seck de Ziguinchor, pour son attention et ses remarques constructives.

Je souhaite également remercier l'administration de l'UFR ST et le corps professoral du département informatique pour avoir contribué à ma formation.

Je ne saurais clore ces remerciements sans adresser une profonde reconnaissance à mes parents, dont le soutien indéfectible et l'amour ont été ma plus grande force tout au long de ce travail. Que Dieu les protège et leur accorde une longue vie, pleine de santé et de bonheur, afin qu'ils puissent être témoins de notre succès.

À Mon Oncle Zalle Faye et sa famille de m'avoir accueilli et soutenu.

À mes frères et sœurs pour leur soutien sans faille et leurs encouragements constants, qui m'ont accompagné tout au long de ce parcours.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours soutenue et encouragé au cours de la réalisation de ce mémoire.

Merci à tous.

DEDICACE

Je dédie cet humble travail :

À mon cher père et à ma chère mère.

À mes Frères et sœurs qui je souhaite un avenir radieux plein de réussite À toute ma famille.

À mon oncle Zalle FAYE

À mes Amis qui me sont chers,

À mes professeurs et surtout mon encadreur Mr Youssou FAYE,

À tous les gens qui ont contribué à ma réussite de près ou de loin.

TABLE DES MATIERES

LISTE DES FIGURES

LISTE DES TABLEAUX

GLOSSAIRE

RFID: (Radio frequency Identification)

WSN: (Wireless Sensor Network)

WPAN: (Wireless Personal Area Network)

LoraWan: (Long Range Wide-area network)

IoT: (Internet of things)

LPWAN: (Wireless local Area network)

NB-IoT: (Narrow Band IOT)

LTE-M: (Long-Term Evolution for Machines)

6LOWPAN: (IPv6 Low power Wireless Personal Area Networks)

TCP: (Transmission Control Protocol)

OSI: ((International Standardisation Organisation)

UIT: (Union internationale des télécommunications)

DSL: (Digital Subscriber Line)

GSM: (Global System for Mobile communication)

M2M: (Machine to Machine)

UMTS: (Universal Mobile Telecommunications System)

MiTM: (Man in The Middle)

DDOS: (Distributed Denial of Service)

MQTT: (Message Queuing Telemetry Transport)

COAP: (Constrained Application Protocol)

AMQP: (Advanced Message Queueing Protocol)

DDS: (Doctor of Dental Surgery)

XMPP: (eXtensible Messaging and Presence Protocol)

mDNS : (multicast Domain Name System)

SSDP: (Simple Service Discovery Protocol)

TLS: (Transport layer system)

DTLS: (Datagram Transport Layer Security)

UDP : (User Datagram Protocol)

IA : (Intelligence Artificielle)

SVM: (Support Vector Machine)

ML: (Machine learning)

DBSCAN: (density-based spatial clustering of applications with noise)

ANN: (Artificiel Neural Network)

KNN: (K-nearest neighbors)

RL: (Reinforcement Learning)

DT: (Decision Tree)

CNN-LSTM: (Convolutional Neural Networks-Long Short-Term Memory)

XGBoost : (Extreme Gradient Boosting)

IDS : (Détection d'Intrusions)

Introduction Générale

L'Internet des objets ou Internet of Things en anglais (IoT) est une révolution technologique majeure qui transforme radicalement la manière dont nous interagissons avec le monde. Loin d'être une simple tendance, l'IoT s'intègre progressivement dans presque tous les aspects de notre vie quotidienne et professionnelle, facilitant la collecte, l'échange et l'analyse de données en temps réel. Grâce aux capteurs, aux réseaux sans fil et aux technologies de communication avancées, les objets connectés sont désormais capables d'interagir et de collaborer entre eux pour fournir des services intelligents dans des domaines variés comme l'agriculture, la santé, les villes intelligentes et l'industrie.

Cependant, cette prolifération des objets connectés s'accompagne de nouveaux défis, notamment en matière de sécurité : les capteurs sur les objets connectés très limités en ressource, l'incompatibilité des technologies IoT, la sécurité non prise en compte au départ pour certains protocoles etc… Les protocoles de communication IoT, tels que Message Queuing Telemetry Transport (MQTT) n'ont pas intégré la sécurité au départ, ce qui ouvre la voie à des vulnérabilités qui nécessitent des solutions robustes pour protéger les données. Aujourd'hui, de nombreux travaux de recherche se développent autour de l'IoT pour répondre aux enjeux de sécurité[1], [2], [3]. Récemment, l'intelligence artificielle (IA), et plus particulièrement le machine learning, joue un rôle crucial pour renforcer la sécurité de l'IoT en offrant des méthodes avancées pour détecter les anomalies, classer les menaces, et prévenir les attaques. Dans la littérature scientifique, plusieurs études exploitent des techniques d'apprentissage supervisé, non supervisé, et même d'apprentissage profond en particulier celles basées sur le modèle XGBoost [4], [5], et les modèles hybrides CNN-LSTM pour répondre au besoin de la sécurité [6], [7].

C'est dans ce contexte que s'inscrit notre mémoire, dont l'objectif principal est d'étudier et d'analyser les mécanismes de détection d'intrus sur les protocoles de communication de l'IoT pour ensuite mesurer la performance des méthodes et algorithmes de l'IA. C'est ainsi que nous nous sommes particulièrement intéressés à la détection d'intrus dans le protocole MQTT. En comparant les modèles CNN-LSTM et XGBoost afin de proposer la meilleure solution pour sa sécurité.

Pour ce faire, nous avons en premier lieu passé en revu les architectures et technologies de communication des Réseaux de Capteurs sans fil (RCSF) et des Radio

Frequency Identication (RFID) qui sous-tendent l'internet des objets. Une vue d'ensemble des couches et technologies de l'IoT nous permet de mieux situer les protocoles de communication.

Ensuite, pour mieux analyser les défis en matière de sécurité de manière générale, et en particulier ceux des protocoles de communication de la couche application dans l'IoT, nous avons orienté notre étude sur une architecture d'une application domotique; étude qui nous a permis d'appréhender au mieux la protocole MQTT (Message Queuing Telemetry Transport), qui, en plus de fournir presque les mêmes services de sécurité que les autres protocoles, est plus économe en ressource et plus évolutif. Ainsi, nous mettons en évidence les vulnérabilités du protocoles MQTT notamment la détection d'intrus qui nous intéresse particulièrement. Et pour renforcer sa sécurité en matière de détection d'intrus, l'intelligence artificielle, et plus particulièrement le machine learning se révèle très prometteuse pour surveiller et détecter les anomalies, mais aussi de classifier les menaces.

Enfin, nous avons mené une étude comparative approfondie sur les algorithmes d'apprentissage automatique (Machine Learning) qui a permis de mettre en évidence et de justifier l'efficacité de XGBoost CNN-LSTM quant à la détection d'intrus. Nous avons analysé et évalué leur performance non seulement pour atteindre un degré de détection élevé mais aussi pour justifier leur choix selon la nature de l'application. Une implémentation avec la distribution Python d'Anaconda sur des données de test et de validation d'un ensemble de données provenant de Kaggle[5] a permis de montrer que le modèle XGBoost constitue la meilleure option pour les environnements IoT en raison de sa rapidité et de sa faible consommation de ressources, le rendant particulièrement adapté aux applications de type 'temps réel".

La suite de ce mémoire est organisée comme suit.

Le chapitre 1 : L'internet des objets : Technologies, Architectures et Protocoles ; dans ce chapitre, nous passons en revue les caractéristiques des Réseaux de Capteurs Sans Fil (RCSF) et des Radio Frequency Identification (RFID), pour mieux comprendre les concepts, les technologies et architectures de l'IoT.

Le chapitre 2 : Etat de l'art sur la sécurité dans l'IoT ; nous présentons les solutions de sécurité proposées pour l'IoT afin de mieux mettre en évidence les vulnérabilités, les attaques et les défis des protocoles de communication notamment ceux du protocole MQTT.

Le chapitre 3 : Les méthodes d'IA pour la détection d'intrusions ; ce chapitre examine les contributions de l'apprentissage automatique à la sécurité de l'IoT, en mettant l'accent sur les solutions de détection d'intrusions et les différents algorithmes de l'apprentissage automatique associés.

Le chapitre 4 : Contribution à la performance de la détection d'intrus dans MQTT ; notre dernier chapitre présente des solutions issues de la littérature basées sur l'IA pour renforcer la sécurité du protocole MQT. Nous contribuons à mettre en œuvre les performances de XGBoost CNN-LSTM.

Chapitre 1 : L'internet des objets : Technologies, Architectures et Protocoles

Dans les années 90, de nombreux concepts ont été élaborés et posent le socle de ce qui deviendra l'IoT, que ce soit Mark Weiser sur l'informatique omniprésente, ou encore Reza Raji sur le concept de spectre IEEE. Il décrit ce dernier comme le déplacement de petits paquets de données vers un grand ensemble de nœuds, dans le but de tout intégrer et automatiser, des appareils électroménagers à des usines entières.

Entre 1993 et 1997, plusieurs entreprises ont proposé des solutions comme Microsoft at Work ou Novell's NEST. Le domaine a pris de l'ampleur lorsque Bill Joy a envisagé la communication de dispositif à dispositif dans le cadre de son projet "Six Webs", présenté au Forum économique mondial de Davos en 99.[9]. Cette même année, on attribue l'expression "Internet des objets" à Kevin Ashton, ingénieur britannique de Procter & Gamble qui l'utilise lors d'un discours. À ce moment-là, il considérait l'identification par radiofréquence (RFID) comme essentielle à l'Internet des objets, car il permettrait aux ordinateurs de gérer tous les objets individuellement [10].

Au cours des dernières années, l'IoT est devenu l'une des technologies les plus importantes et a impacté tous les secteurs de l'activité humaine : l'habitat, les véhicules, les environnements de travail, les usines, les villes, l'agriculture, le système de santé… De même, tous les niveaux de la société (individus, entreprises, États) sont d'ores et déjà concernés, de l'urbain au rural.

Tout cela a été possible grâce à des objets intelligents qui la constituent et qui sont souvent des capteurs dotés de capacités de mesures (température, pression, vibration, luminosité, humidité, tension, etc.) ou des actionneurs capables d'agir. Ces objets disposent de différentes technologies de communication : RFID (adio Frequency Identification), NFC (Near Field Communication), Bluetooth, Wi-Fi, LoRa, etc. Ils peuvent ainsi s'interconnecter pour anticiper et interagir en temps réel.

Le but de ce chapitre est de passer en revue les caractéristiques des réseaux de capteurs sans fil WSN (Wireless Sensor Network) et des RFID pour mieux comprendre les architectures et protocoles d'IoT.

I. Réseaux de capteurs sans fil (RCSF)

Les capteurs existent depuis plusieurs années dans les domaines de l'industrie tels que l'aéronautique, l'automobile. On observe actuellement une forte recrudescence de ce type d'équipements qui sont interconnectés pour former des réseaux de capteurs. Ces capteurs sont maintenant de plus en plus interconnectés par ondes radios (ZigBee par exemple). En 2003,

selon le magazine Technology Review du MIT, le réseau de capteurs sans fil est l'une des dix nouvelles technologies qui bouleverseront le monde et notre manière de travailler et de vivre. En conjonction avec l'augmentation du taux d'utilisation des réseaux de capteurs, les recherches dans ce domaine visent à optimiser les protocoles et les algorithmes conçus pour ces réseaux pour améliorer leurs performances. On peut considérer un réseau de capteurs sans fil (RCSF en anglais Wirless Sensor Network (WSN)) comme un cas particulier des réseaux ad hoc sans fil, où les nœuds sont des capteurs (ou senseurs), déployés à l'intérieur. Ils peuvent être considérés aussi comme une extrapolation extrême de deux tendances générales en informatique : la miniaturisation (construire des ordinateurs plus petits) et l'interconnexion (réseau).

I.1. Le capteur

Le capteur est un instrument de mesure qui permet de transformer une grandeur physique ou chimique observée (température, humidité, accélération, les vibrations, etc.) en un signal électrique. Ce signal de sortie électrique, dit exploitable peut être de nature analogique, numérique, logique etc. Cette transformation doit être le reflet aussi parfait que possible de ces grandeurs. Pour cela le composant principal d'un capteur est l'élément sensible, également appelé transducteur. Voici quelques exemples de capteurs :

- **Capteur de température** : C'est un appareil qui détecte et mesure la chaleur et la fraîcheur en le convertissant en signal électrique. Il est présent dans de nombreux objets quotidiens comme les fers à repasser électrique, les grille-pains, les détecteurs de mouvements, le thermomètre infrarouge pour la mesure de température.

- **Capteur de pression** : Un dispositif qui détecte de la pression et qui la transforme en signal électrique dont la valeur dépend de la pression appliquée. Ces capteurs de pression sont fréquemment utilisés dans le domaine des industries de l'énergie, de l'agro-alimentaire, de l'eau, de l'environnement, du papier et du carton et aussi sur les réservoirs.

- **Capteur de mouvement** : Un appareil qui permet le déclenchement d'un autre dispositif (une alarme, allumer une lampe, la climatisation etc) en détectant le mouvement dans une zone de couverture. L'utilisation de capteurs de mouvement améliore le confort et la sécurité.

I.2. Architecture d'un capteur

L'infrastructure d'un nœud capteur (figure 1) se compose généralement de quatre principales unités : unité de capteur (d'acquisition), unité de traitement (de calcul), unité de communication et unité d'alimentation (la batterie).

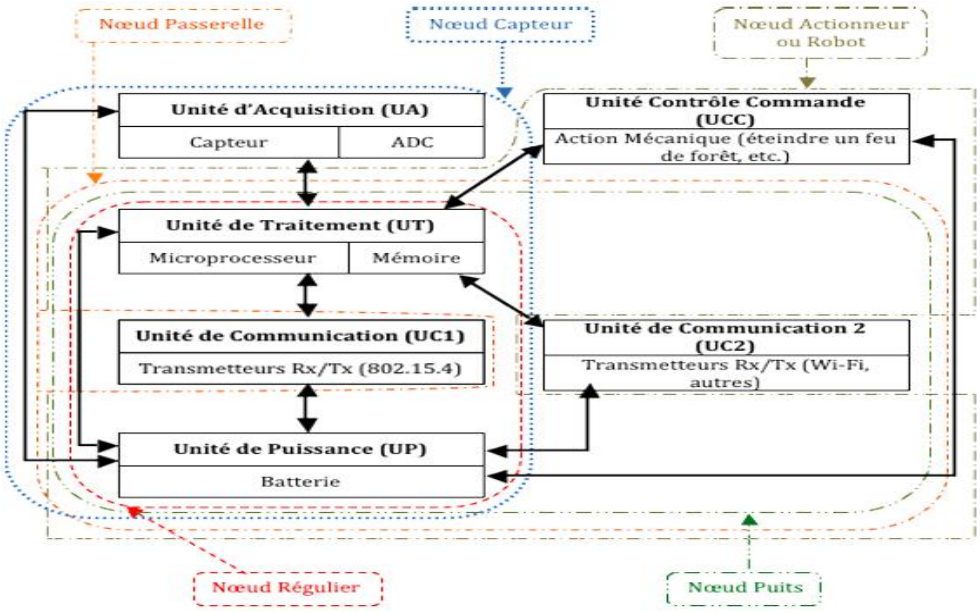


Figure 1: Architecture des différents types de nœuds : régulier, capteur, robot, puits, passerelle [11]

- **Unité de capteur** (d'acquisition*)*

Cette unité est généralement composée de deux sous-unités : un ou des capteurs et de convertisseur Analogique/Numérique (CAN), ou (ADC pour Analog to digital Converter). Les capteurs obtiennent des mesures numériques sur un phénomène physique et les transforment en signaux analogiques. Les ADC convertissent l'analogique des signaux en signaux qui sont ensuite envoyés à l'unité de traitement.

- **Unité de traitement :** Composé d'un microprocesseur avec mémoire et d'un microcontrôleur, elle réceptionne les informations provenant de l'unité d'acquisition pour le stockage ou l'envoie à l'unité de communication et aussi un contrôle intelligent au nœud de capteur.
- **Unité de communication :** Elle est chargée de toutes les transmissions et de réceptions des données par Radiofréquence (RF), par l'laser, l'infrarouge…

- **Unité d'alimentation :** Une source d'énergie (batterie) permettant d'alimenter tous les composants du système.

I.3. Les Réseaux de capteurs

Un réseau de capteurs sans fil est généralement constitué d'un ensemble de petits capteurs répartis dans une zone géographique appelée zone de couverture ou zone d'intérêt d'une manière plus ou moins aléatoire. Le nombre de capteurs déployés dans la zone d'intérêt varie selon le besoin de l'application. Il peut être dans l'ordre de quelques dizaines à de milliers de capteurs. Les nœuds capteurs sont capables de surveiller un phénomène physique sur l'environnement qui les entoure et de collecter des données d'une manière autonome. A l'aide de technologies sans fil telles que le Wi-Fi, le Bluetooth, Zigbee, LoRa, ou d'autres protocoles adaptés aux besoins spécifiques du réseau, les données captées sont acheminées à un nœud considéré comme un "point de collecte", appelé nœud-puits (ou sink en anglais) ou station de base (base station). Le nœud puits transmet ces données à l'utilisateur final du réseau à travers un réseau de communication (internet, satellite…) pour analyser ces données et prendre des décisions. L'utilisateur du réseau peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées en utilisant la station de base comme passerelle.

Les réseaux de capteurs jouent un rôle essentiel dans l'Internet des objets (IoT), où les données collectées par les capteurs peuvent être transmises à des systèmes d'analyse et de gestion pour une prise de décision en temps réel ou des actions automatisées. Leur utilisation est en constante expansion dans divers domaines pour améliorer la surveillance, l'efficacité et la compréhension de l'environnement physique qui nous entoure.

I.4. Architecture des réseaux de capteurs

Comme nous l'avons énuméré dans la section I.2 un réseau de capteurs est conçu pour permettre la collecte, le traitement et le transfert de données entre des nœuds capteurs dispersés dans un environnement donné. Son architecture spécifique dépendra de son application et de ses contraintes spécifiques, telles que la portée requise, le débit de données, les contraintes énergétiques, etc. Dans un RCSF on peut retrouver plusieurs nœuds qui ont des rôles différents (Nœuds passerelle, nœuds actionnaire ou robot, nœuds capteur, nœuds puits, nœuds régulier) mais pour optimiser certains paramètres comme la durée de vie du réseau ou le délai de livraison des données, certains travaux se sont focalisés sur l'architecture (plat, hiérarchique, multi niveaux) des RCSF.

I.4.1. L'architecture en plat

C'est un réseau homogène où tous les nœuds ont les mêmes rôles et fonctionnement et aussi semblable en termes de ressources excepté la station de base qui a le rôle d'une passerelle en transmettant l'information collectée à l'utilisateur final. On peut retrouver la méthode multi-saut pour cette architecture (Voir Figure 2).

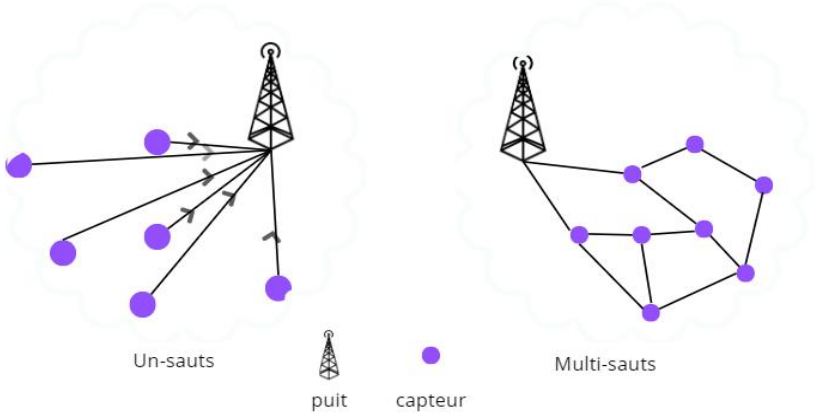


Figure 2: Architecture en plat

I.4.2. L'architecture hiérarchique

Les architectures hiérarchiques (Figure 3) sont utilisées dans les grands réseaux de capteurs. On retrouve des capteurs hétérogènes dont le traitement et le stockage sont centralisés dans un nœud central spécifique. Pour réduire la complexité de la plupart des capteurs de nœuds et leur déploiement, cette architecture a été proposée en divisant les nœuds en plusieurs niveaux de responsabilité. Le clustering est l'une des méthodes les plus courantes utilisées.

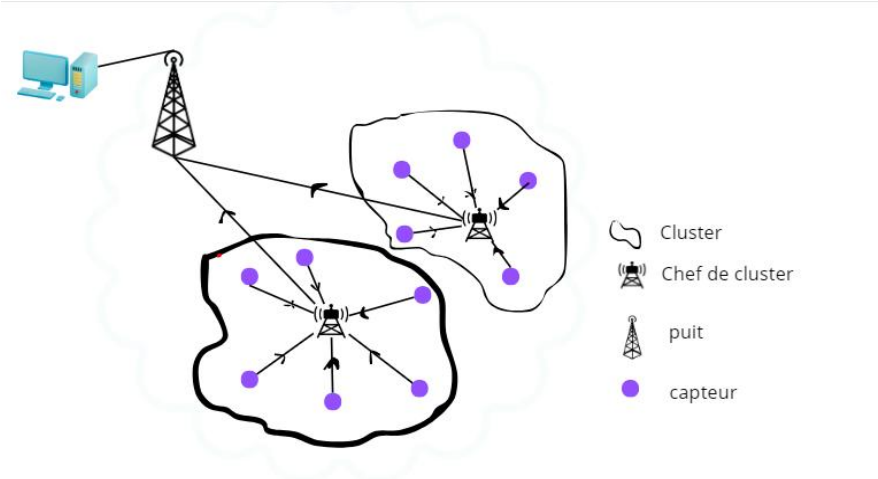


Figure 3: Topologie d'une architecture WSN hiérarchique

I.5.Modèle de communication

Les modèles de communication dans les réseaux de capteurs sans fil (RCSF) décrivent comment les nœuds (capteurs) interagissent les uns avec les autres pour échanger des données recueillies. Pour assurer une communication efficace, économiser de l'énergie et optimiser les performances globales du réseau, ces modèles de communication sont essentiels.

Il existe plusieurs modèles de communication pour les réseaux de capteurs sans fil. Les plus utilisés, comme présenté dans [12], sont décrits dans les sous-sections suivantes :

I.5.1 Modèle de communication à deux étapes (Two-step Model for WSNs)

Le modèle de communication à deux étapes est utilisé dans les réseaux de capteurs sans fil pour économiser l'énergie et prolonger la durée de vie des capteurs. La phase de découverte et la phase de transmission de données composent ce modèle.

- Phase de découverte : Au cours de cette première étape, les nœuds capteurs se mettent en mode d'écoute pour détecter les autres nœuds à proximité. Cela peut être accompli grâce à des méthodes telles que l'écoute régulière des stations de radio ou l'utilisation de signaux de diffusion. L'objectif est de créer un voisinage de communication, c'est-à-dire que chaque nœud capteur identifie les autres nœuds dans sa portée de transmission.
- La deuxième étape : Une fois la phase de découverte terminée, les nœuds capteurs reçoivent des données de leurs voisins. Ils peuvent commencer à transmettre sélectivement leurs données maintenant. Au lieu de dépenser de l'énergie et de l'inutilité en voyant des données à tous les autres nœuds du réseau, les nœuds utilisent les informations recueillies lors de la phase de découverte pour déterminer la meilleure façon d'acheminer leurs données vers la destination devinée.

L'avantage principal de ce modèle est qu'il évite les communications inutiles, et optimisé ainsi la consommation d'énergie. En effet, les nœuds ne transmettent leurs données qu'à un petit groupe de nœuds directement impliqués dans la chaîne de transmission, ce qui limite les opérations de communication et économise l'énergie des capteurs.

I.5.2 Modèle de communication multi-sauts (MCMS for WSN)

Un modèle de communication multi-sauts utilise plusieurs sauts successifs pour transmettre des données entre les nœuds (capteurs) du réseau plutôt qu'une communication directe entre

les nœuds éloignés. Les réseaux de capteurs sans fil (RCSF) utilisent fréquemment ce modèle pour étendre la portée de communication et améliorer l'efficacité énergétique du réseau [13].

Le principe fondamental du modèle de communication multi-sauts est que chaque nœud d'un réseau de capteurs sans fil peut agir en tant que relais pour transmettre les données collectées par d'autres nœuds et en tant que capteur pour collecter des données environnementales. Lorsqu'un nœud capteur souhaite envoyer des données à un nœud distant au-delà de sa portée directe, il peut les transmettre à un nœud voisin qui les relaiera à son tour vers un autre nœud et ainsi de suite jusqu'à ce que les données atteignent leur destination finale.

I.5.3 Modèle de communication basé sur l'emplacement

Ce modèle utilise l'emplacement géographique des capteurs pour faciliter la communication, le routage et la gestion des données. Ce modèle exploite les informations de localisation des capteurs pour optimiser l'efficacité énergétique, la latence, la fiabilité et la sécurité du réseau.

Cependant, il est important de noter que la localisation précise des capteurs peut poser des problèmes, en particulier dans des environnements où le GPS peut ne pas être disponible ou imprécis. De plus, la sécurité des données de localisation et leur gestion doivent être prises en compte avec soin pour éviter toute violation de la vie privée ou attaque potentielle.

I.5.4 Modèle de communication basé sur les clusters

Le modèle de communication basé sur les clusters est une approche populaire pour acheminer des données dans un RSN. Ce modèle regroupe les nœuds capteurs en grappes. Chaque cluster possède un nœud principal qui est chargé de transmettre les données au nœud central [13].

Il a de nombreux avantages. En premier lieu, cela permet d'économiser de l'énergie. Les nœuds capteurs ne doivent pas transmettre directement les données au nœud central. Ils peuvent les envoyer au nœud chef de leur cluster, qui les transfèrent ensuite au nœud central. Cela permet de réduire la consommation d'énergie des nœuds capteurs, qui est l'une des principales préoccupations des RSN. Ensuite la fiabilité de la communication. Si un nœud capteur est défectueux ou s'il perd la communication avec le nœud central, les données qu'il collecte peuvent toujours être acheminées vers le nœud central via le nœud chef de son cluster. Cela permet d'assurer une meilleure fiabilité de la communication, même en cas de panne de nœuds. Et aussi il permet d'améliorer l'évolutivité du réseau. Les nœuds capteurs peuvent être

ajoutés ou supprimés du réseau sans avoir à modifier le nœud central. Cela permet au réseau de s'adapter à l'évolution des besoins de l'application.

Ces modèles de communication dans les réseaux de capteurs sans fil peuvent être utilisés individuellement ou combinés pour répondre aux exigences spécifiques d'une application donnée. Le choix du modèle dépendra de facteurs tels que la topologie du réseau, la consommation d'énergie, les contraintes de latence et les exigences de fiabilité des données.

I.6. Modèle de collection et de livraison

Le mode de collection et de livraison des données dans RCSF dépend du type d'application souhaitée. Ainsi, on peut citer quatre catégories d'applications des réseaux de capteur sans fil.

I.6.1 Application time-driven (pilotée par le temps)

Un réseau time-driven est approprié pour les applications qui nécessitent une collecte de données périodiquement. Ensuite ils envoient ces données captées à la station de base dans des intervalles de temps réguliers. Par exemple, cela peut être bénéfique dans les applications de surveillance (feu, météo) pour générer des rapports réguliers.

I.6.2. Application event-driven (application événementielle)

Dans ce type d'applications les capteurs doivent réagir (événement spécial) immédiatement à des changements soudains des valeurs captées pour l'envoi des données à la station de base. Ce type de réseau peut être appliqué dans différents domaines tels que la surveillance médicale (surveillance de taux de glycémie dans le sang), le contrôle d'édifice (les barrages, les voies des chemins de fer…) la détection de la fumée, la surveillance militaire, etc.

I.6.3. Application request-driven (application pilotée par requête)

Dans les applications orientées requêtes, les capteurs échantillonnent à la demande de la station de base. Lorsqu'un capteur reçoit une requête de la station de base, la collecte de données commence. Ensuite il envoie ces données recueillies à la station de base. Il s'agit d'une catégorie de réseau conçue pour les applications spécifiques à l'utilisateur, telles que les demandes d'informations sur une région spécifique. La topologie et la position des nœuds dans ce type de réseau doivent être connues.

I.6.4. Application hybride

C'est une combinaison entre les trois types d'applications précédemment décrites. Par exemple, dans un réseau conçu pour le suivi d'objets, le réseau peut combiner entre un réseau

de surveillance (time-driven) et un réseau de collecte de données par événements (event-driven) [14].

I.7. Les domaines d'applications des RCSF

La miniaturisation des capteurs, le coût de plus en plus faible, la large gamme des types de capteurs disponibles ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs de se développer dans plusieurs domaines d'application. Ils permettent aussi d'étendre les applications existantes. Les réseaux de capteurs peuvent se révéler très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi les domaines où ces réseaux peuvent offrir les meilleures contributions, nous pouvons citer les domaines : militaire, environnemental, médical, domestique, commercial [12], etc.

II. La RFID (Radio frequency identification)

Parmi les technologies importantes dans le réseau sans fil, on a le réseau de capteur RFID qui appartient au réseau personnel sans fil (Wireless Personal Area Network ou WPAN). Il forme aujourd'hui un nouveau domaine de recherche qui a suscite l'intérêt à la fois des industriels et de la communauté de recherche. Cette nouvelle technologie est le résultat de l'intégration de la technologie d'identification par radiofréquence (Radio Frequency Identification ou RFID) et la technologie réseau de capteurs sans fil (Wireless Sensor Network ou WSN), parce qu'il y a un certain nombre d'avantages par fusionnement de ces deux technologies afin de satisfaire les besoins de certaines applications.

Les technologies d'identification et de collecte automatique des données (AIDC) et les technologies sans fil comprennent la technologie d'identification par radiofréquence (RFID). Ces technologies font référence à des méthodes d'identification automatiques d'objets, la collecte d'informations les concernant et leur entrée dans un système d'acquisition sans intervention humaine. Quelques exemples typiques de ces technologies : les codes à barres ancêtre de la RFID, la biométrie, la reconnaissance optique de caractère, la reconnaissance vocale ou le marquage par bande magnétique.

RFID est la technologie de base de l'IoT et consiste en l'échange d'informations entre un lecteur et une étiquette électronique porteuse d'informations via des ondes de fréquence radio. La RFID est un moyen de capturer des données sur un objet sans recourir à un humain pour lire les données. Tout système RFID comprend un transpondeur ou étiquette qui contient les données de l'élément à identifier, une antenne utilisée pour transmettre le signal (ondes

radiofréquences) entre le lecteur et le transpondeur, un lecteur qui communique avec le transpondeur, via l'antenne (il reçoit le signal émis par le transpondeur et/ou lui transmet des informations) et qui envoie les données à un intergiciel (middleware) chargé du traitement des données. Ces débuts ne datent pas d'aujourd'hui, les premières applications RFID remontent aux années 1930 lorsque les britanniques souhaitaient connaître en temps réel les avions des alliés de ceux de l'ennemi sur les radars. Dans le monde de l'industrie, l'utilisation de la RFID s'est largement répandue depuis les années 80. À l'époque, les constructeurs automobiles ont commencé à l'utiliser pour identifier les carcasses des véhicules prêtes pour la cabine de peinture. Cependant, depuis peu, la convergence de moindre coût, de capacités accrues et la réduction de la taille de l'étiquette à celle d'une tête d'épingle par les sociétés comme Siemens, Texas Instruments, Philips Semi Conductors et Motorola a incité les entreprises à réfléchir sérieusement à ce que la RFID peut faire pour eux. Une avancée majeure est survenue lorsque le géant de la vente au détail Wal-Mart a annoncé de façon spectaculaire qu'il exigerait que ses 100 principaux fournisseurs intègrent des puces RFID (identification par radiofréquence) sur leurs produits d'ici janvier 2005[15]. Ces dernières années, cette technologie est devenue indispensable et est considérée comme un outil incontournable dans le commerce de gros.

II.1. Les lecteurs

Les lecteurs RFID sont des dispositifs destinés à interroger les tags et ainsi récupérer des informations comme le numéro d'identification stocké. Ils sont couplés à un transmetteur RF qui est la source d'onde radios émises pour atteindre l'étiquette passive ou semi-passive et la faire réagir. Cette diffusion d'onde peut être en permanence ou à la demande. Ils utilisent un mécanisme de communication qui permet de reconnaître et de différencier automatiquement les transpondeurs RFID se trouvant dans leur champ de lecture. Après avoir reçu les informations provenant du tag par une antenne qui peut être interne ou externe du lecteur, il l'identifie, et transmet l'identité de l'étiquette ainsi les données reçues vers l'ordinateur central avant que le lecteur agisse. Le lecteur peut également procéder à une écriture d'information dans le tag si celui-ci est en mode lecture-écriture. Nous distinguerons deux types de lecteurs: ***le lecteur fixe*** (figure 4_a), le plus utilisé car il n'a pas besoin d'être en contact direct avec la puce, et le ***lecteur portable*** (figure 4_b) :

- Le lecteur RFID fixe ou statique est fixé et ne peut pas être transporté pour la lecture des puces à distance. Il se présente sous la forme de portique ou de borne, comme dans les caisses des supermarchés.

- Les lecteurs portatifs, qui sont en général utilisés dans les applications de recherche et location de produits dans un entrepôt et dont les antennes intégrées sont incorporées directement dans le dispositif [16].

(a)Lecteur

(b) Lecteur Mobile

Figure 4 : Différents types de lecteur RFID [17]

La technologie RFID fonctionne sur plusieurs bandes de fréquence qui déterminent la portée mais également le type d'applications dédiées (voir tableau 1). En général, il en existe plusieurs types. Mais les fréquences les plus importantes et les plus utilisées sont les basses fréquences (LF, Low frequency) avec une portée d'une dizaine de centimètres au plus, les hautes fréquences (HF, High Frequency) avec une portée plus grande montant (1 m environ), les ultra-hautes fréquences (UHF, Ultra High Frequency).

II.2. Les Tags

Les étiquettes (figure 5), aussi appelées tags (en anglais) contiennent des données de l'objet sur lequel ils sont placés. Elles sont équipées d'un processeur de base, qui est une puce électronique miniaturisée qui forme l'électronique embarquée gérant les données de l'étiquette et une antenne qui assure la transmission de l'information vers le lecteur RFID via une fréquence radio. Lorsque la puce est alimentée par le signal du lecteur, elle exécute des tâches dont : la récupération de l'énergie si c'est une étiquette passive, la réception du signal et l'établissement de la réponse. Pour ce faire, elle démodule le signal en le convertissant en tension pour alimenter les autres parties de la puce (partie numérique). Après activation de la puce (étiquette passive), la mémoire (en centaines de bits à quelques kilo-octets) de la puce contient la clé de la réponse de l'étiquette, qui est ensuite transmise au lecteur via un signal.

L'étiquette RFID (ou tag) génère en premier lieu un code permettant d'identifier l'objet sur lequel il est posé. Les informations qu'ils contiennent peuvent être accessibles comme le prix ou encore les caractéristiques du produit. Le transfert d'informations peut également se faire du lecteur vers le tag. Pour la sécurité, certains tags peuvent effectuer des opérations logiques, d'autres peuvent effectuer de la cryptographie symétrique, des fonctions de hachage ou même de la cryptographie asymétrique. D'une manière générale, les transpondeurs RFID (tag) peuvent être classés en fonction de leur taille, de leur mode d'alimentation (actif, passif ou semi-passif), de leurs propriétés de lecture et/ou d'écriture et de leur fréquence d'utilisation.

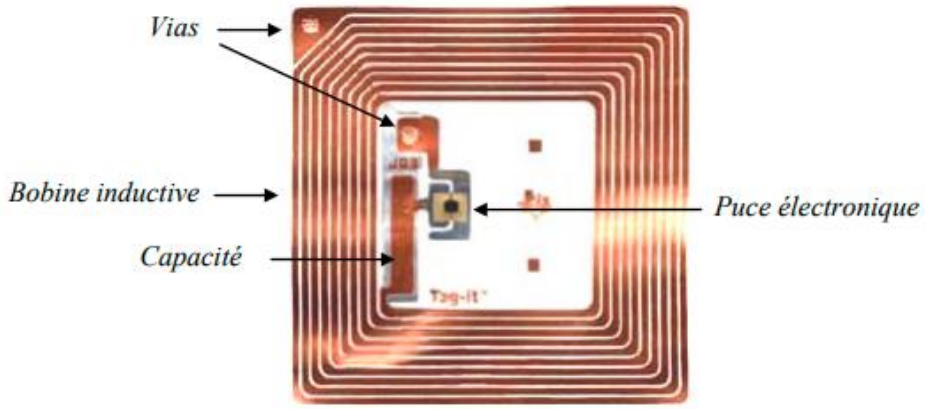


Figure 5 : Tag RFID HF (Tag-it HF de Texas Instrument) [18]

- **Les tags actifs :** Ils utilisent leur propre source d'énergie (une batterie embarquée sur une puce) pour transmettre des informations, ce qui limite leur durée de vie inférieure à 10 ans et enregistre de nouvelles données. Ils sont capables de communiquer sur de plus longues distances (environ 100 m). De ce fait, elles sont les plus chères et les plus complexes à produire, et assurent, outre des fonctions de transmission, des fonctions soit de captage soit de traitement de l'information captée. Elles sont fournies vierges et pourront être écrites plusieurs fois, effacées, modifiées et lues. D'une manière générale, les transpondeurs actifs ont une grande capacité de stockage et peuvent stocker différents types d'informations.
- **Les tags passifs ou semi-passifs :** Ils n'ont pas de source d'énergie interne et utilisent à la place l'énergie générée par la résonance du lecteur. C'est ce courant qui leur permet d'alimenter leurs microcircuits. Leur coût de production n'est pas cher et sont généralement réservés à des productions en volume. Ce sont eux que l'on trouve plus particulièrement dans la logistique et le transport. Ils ont donc une durée de vie

théoriquement illimitée. Pour les transpondeurs semi-passifs, ils sont équipés de capteurs et disposent d'une batterie embarquée pour fournir de l'énergie. Ces tags sont plus robustes et plus rapides en lecture et en transmission que les transpondeurs passifs, mais ils sont aussi plus chers. Les transpondeurs semi-passifs ne sont pas encore largement utilisés dans les applications industrielles.

II.3. Architecture des réseaux RFID

L'architecture typique d'un système RFID se compose principalement de trois composants : des tags, les lecteurs et une base de données communément appelée en anglais le « back-end » ou une application, qui permet de répertorier tous les tags du système et aussi un ensemble d'informations associées à ces tags et lecteur du système (p.ex. les identifiants des tags). Cependant, le back-end peut également fonctionner comme un switch qui ne fait que transférer les informations entre les lecteurs. Dans tous les cas, le back-end ne communique qu'avec les lecteurs. Il peut ne pas être nécessaire dans un système RFID. Par exemple, si un système n'est composé que d'un seul lecteur autonome, alors ce dispositif peut aussi jouer le rôle de back-end. Dans d'autres systèmes, le back-end et les lecteurs sont connectés tous ensemble via un canal sécurisé et peuvent donc être vus comme une seule et unique entité, simplement nommée lecteur [19].

II.4. La communication dans les RFID

La communication consiste en un transfert de données associé à un transfert d'énergie. Cette communication est bidirectionnelle : la communication du lecteur vers le tag est appelée liaison montante et la réponse du tag vers le lecteur est appelée liaison descendante. La communication dans la plupart du temps est initiée par le lecteur RTF : Reader Talks Fisrt), la liaison descendante n'a lieu qu'à la fin de la liaison montante. On distingue deux types de protocole de communication qui sont : le RTF et TTO (Tag Talk Only) et la sélection d'un protocole par rapport à un autre dépend de l'application visée.

II.4.1. Le protocole TTO

Le protocole TTO est un protocole dont le transfert des données se fait dans un seul sens du tag vers le lecteur, il n'existe pas de liaison montante. Un tag utilisant cette procédure pour transmettre ses données de façon régulière après avoir été alimentée en entrant dans le champ d'un lecteur. Tout d'abord le lecteur envoie une invitation au dialogue appelé requête, et le tag (TTF) produit une réponse à cette requête.

II.4.2. Le protocole RTF

Dans le cas du protocole RTF, la communication est initiée par le lecteur RTF. Lorsqu'un tag RTF pénètre dans le champ d'un lecteur, il attend une requête avant de transmettre son identifiant. Ce transfert de données est basé sur plusieurs procédures de communication qui sont composées de deux types :

Transfert continu d'énergie : Ce transfert d'énergie peut être associée à un transfert de données bidirectionnel simultané (FDX, pour Full Duplex) ou alterné (HDX, pour Half Duplex).

- HDX : le transfert de données pendant la liaison descendante est alterné avec le transfert de données de la liaison montante
- FDX : les échanges de données lors des liaisons montantes et descendantes s'effectuent simultanément

L'avantage de ces deux procédures HDX et FDX est que le transfert d'énergie du lecteur vers le tag est continu : même quand le tag est en train de répondre au lecteur, le transfert d'énergie n'est pas stoppé.

Transfert séquentielle d'énergie : Une procédure séquentielle (SEQ) peut également être utilisée pour configurer le transfert de données d'un système RFID, dans lequel le transfert d'énergie se produit pendant une durée limitée et les données du tag vers le lecteur sont transmises lors de pauses du transfert d'énergie.

Ces différentes procédures sont représentées sur la figure 6.

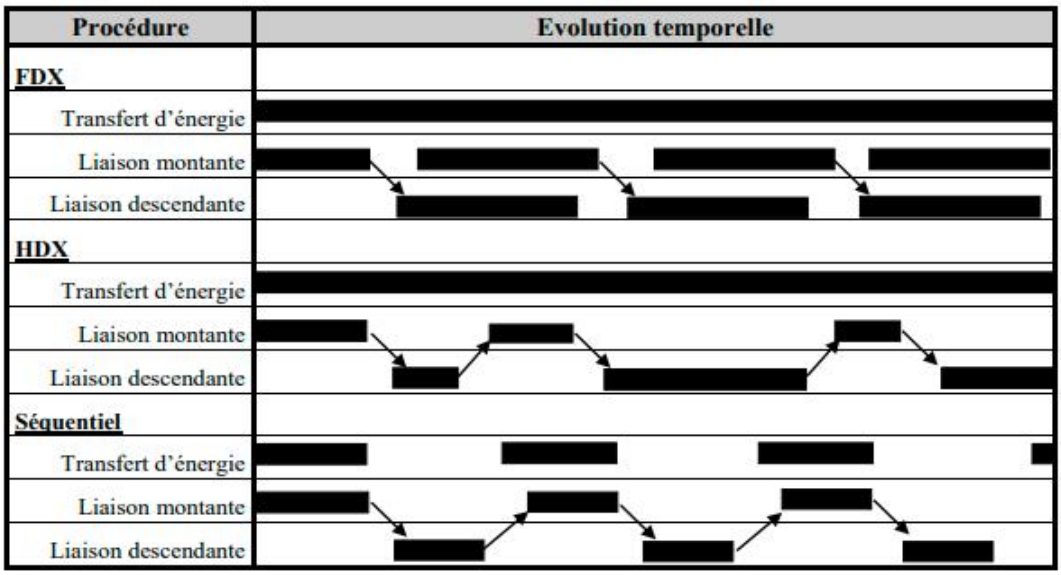


Figure 6: Principaux procédure de communication [20]

La communication (figure 7) dans la liaison descendante repose sur la technique de rétro modulation. Le signal radio issu du lecteur est alors partiellement réfléchi par le tag RFID. Quels que soient les fréquences ou les modes de couplage, le moyen utilisé pour réaliser cette rétro modulation consiste à commuter une charge (impédance) placée en parallèle entre la puce électronique et l'antenne de l'étiquette. Il est bien entendu que ce système de commutation de charge fait partie intégrante de la puce électronique RFID. Le signal réfléchi par l'étiquette vient alors se superposer au signal provenant de l'interrogateur. Dans le cas, très majoritairement rencontré, des étiquettes passives ne possédant pas de source d'énergie embarquée, le rapport entre la puissance du signal émis par l'interrogateur (pour alimenter la puce et transmettre les commandes) et la puissance du signal rétro modulé par l'étiquette ou le tag RFID peut largement dépasser les 60 dB. L'interrogateur doit donc présenter une bonne sensibilité pour détecter et décoder l'information issue de l'étiquette. La difficulté de ces systèmes consiste donc à trouver la meilleure charge permettant de créer de fortes variations de signal réfléchi sans pour autant pénaliser l'alimentation du circuit lui-même.

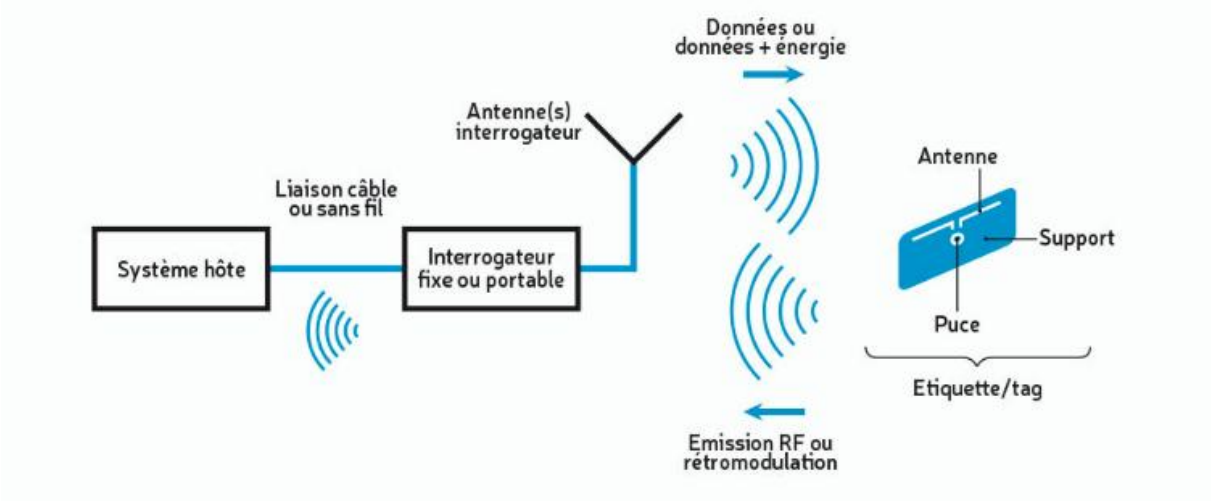


Figure 7: La communication dans un système RFID passif [20]

II.5. Les applications

La technologie RFID apporte des solutions dans plusieurs domaines d'application

II.5.1. Le secteur de retail

Étant certainement le secteur d'activité que la technologie RFID touche le plus. Le marché industriel évolue de plus en plus en raison de l'adaptation de cette architecture pour améliorer la productivité et les contrôles de qualité. Elle est utilisée pour l'identification, le suivi ou le traçage des marchandises, elle facilite ainsi la gestion des produits.

II.5.2. Péage autoroute

La technologie RFID est parfois utilisée pour la perception électronique des péages sur les autoroutes (la carte rapido au Sénégal) et les ponts. Les véhicules peuvent être équipés de tag RFID sur leur pare-brise, qui déduisent automatiquement les péages du compte du conducteur lors de l'approche des barrières (au poste péage) équipé de lecteur.

II.5.3. Gestion d'entrepôt

Dans la gestion des stocks, la technologie RFID permet un contrôle d'inventaire en temps réel, une optimisation des processus d'entreposage (réception, rangement, cueillette et envoi) et une réduction du temps de cycle de distribution.

III. La Technologie IOT

L'Internet des objets est une notion complexe. Il est composé de nombreux éléments complémentaires, y compris les puces RFID, les solutions de nommage ou les middlewares, chacun avec ses propres caractéristiques. Il n'a pas de définition fixe :

On peut dire que, c'est un réseau qui permet, via des systèmes d'identification électronique normalisés unifiés et des dispositifs mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transférer et traiter, sans discontinuité entre les mondes physiques et virtuels, les données s'y rattachant.

Ainsi IoT rend les objets qui nous entourent intelligents en leur offrant la faculté de communiquer entre eux ou avec le nuage (cloud). Plusieurs domaines d'application sont touchés par l'IoT, Parmi ces principaux domaines nous pouvons citer : le domaine de la sécurité, le domaine du transport, l'environnement et l'infrastructure et les services publics,..etc.

Et à l'aide des capteurs appropriés pour chaque domaine, On peut classer ces technologies en fonction de leur portée, de leur débit, et de leur consommation d'énergie.

III.1. Les Technologies de communication de L'IOT

Ils existent différentes catégories de technologies de communication (figure 8)

Figure 8: Les catégories des technologies de communication [21]

III.1.1. Le LPWAN

- Lora Wan : Un protocole réseau basé sur la technologie radio qui consomme peu d'énergie, bas débit (compris entre 0.3 et 50 kbits/s), longue portée (de 2 à 5 Kms en zone rural et jusqu'à 15 Kms en zone urbain) avec une durée de batterie plus de 10 ans.

Il permet d'envoyer et recevoir des données de petite taille d'où leur adaptation aux applications de l'internet des objets

- Sigfox : comme Lora Wan, Sigfox est un protocole de communication basé sur la technologie radio à base consommation, bas débit, longue portée (de 10km en ville et 30 à 50 km) et permet la communication de données de taille réduite entre les appareils connectés sans passer par un téléphone mobile.

III.1.2. Les réseaux cellulaires

Les réseaux cellulaires peuvent être utilisés pour connecter les objets physiques (comme les capteurs) à l'internet en même temps que le smartphone. Ce service est offert par des opérateurs de télécommunication. Il peut transférer des quantités raisonnables de données sur des distances considérables sans épuiser leur ressource d'énergie (un débit théorique de 2 Mbits pour le 2G, jusqu'à 100 Mbits/s pour le 4G). Avec le 5G à l'horizon, l'avenir s'annonce prometteur.

III.1.3. NB-IoT et LTE-M

Nouveaux sur le marché, ils sont spécifiquement basés sur les objets connectés et s'appuient sur l'infrastructure déjà existante de la 4G pour connecter des objets à faible consommation d'énergie au réseau mobile.

III.1.4. Réseaux locaux sans fil (Wireless Local Area Network, WLAN)

C'est un réseau sans fil qui permet de connecter des matériels comme des pc, des équipements électroniques ou informatiques dans le cadre professionnel, immeubles de bureaux, bâtiments industriels ou même des milieux privés ou publics. Les technologies les plus connus sont :

- Wi-Fi : Plus connus des réseaux LAN, c'est un protocole de communication basé sur la norme IEEE 802.11. Il fournit un réseau qui permet de relier plusieurs objets via des ondes radio sur une portée de 20 à 100m avec des débits pouvant atteindre 11 Mbits/s mais consomme un gain énorme d'énergie.
- Z-Wave : C'est un protocole de communication destiné à la domotique. Comme le Wi-Fi, mais créée spécialement pour l'automatisation de la maison avec une portée de 30 à 100m sur une bande de 868 MHz et en consommant peu d'énergie.
- Bluetooth Low Energy : C'est une technologie sans fil qui propose des communications à faible puissance et à courte portée entre les appareils. Il utilise la

même bande de fréquence de 2,4 GHz que le Bluetooth standard mais des canaux différents avec une portée de 60 m et 10 fois moins d'énergie.

- Le 6LoWPAN : C'est un protocole de communication à faible puissance développé pour définir l'adaptation d'IPv6. Du coup, il a la possibilité d'utiliser des infrastructures et technologies IP qui existent déjà et sont approuvées et aussi de se connecter facilement à d'autres réseaux IP sans avoir besoin d'un intermédiaire (passerelles).

III.2. L'architecture de L'IoT

Jusqu'à nos jours l'IoT n'a pas encore une architecture uniforme et pourtant le développement de ces technologies augmentent de jour en jour sur plusieurs domaines tels que la santé, l'industrie, l'agriculture, le transport etc. Toutefois il existe une architecture de base généralement acceptée, basée sur les modèles de base d'Internet (TCP / IP et OSI) (représentée dans la figure 9). Elle est composée de trois couches (Figure 9) : une couche de perception (objets), une couche réseau (transport et traitement) et la couche application (services et applications).

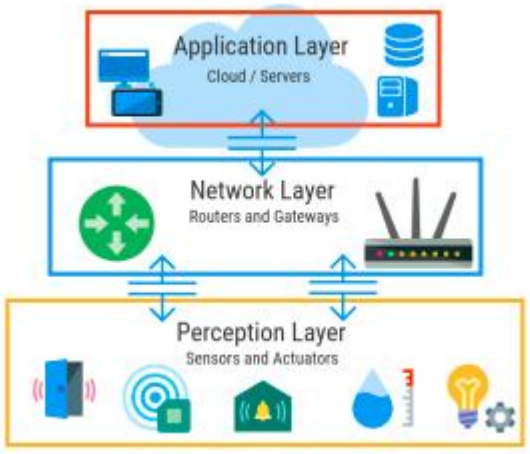


Figure 9: Architecture en 3 couches [22]

Il existe d'autre architectures de références proposées par des organisations de normalisation et de recherche pour faciliter la création de nouvelles applications comme le modèle de Cisco, IIC (Industrial Internet consortium) proposée par le consortium Internet Industriel, UIT proposé par l'Union international des Télécommunications, l'ISO/IEC a proposé à travers la norme ISO/IEC 30141 :2018.

Par la suite nous allons présenter l'architecture UIT. Les années 2000 l'UIT avait proposé une architecture basée sur cinq couches qui est le UIT-T M.3010 mais en 2012 il propose une autre nommée UIT-T Y.2060 compose de quatre couches (Figure 10). Ces quatre couches sont : la couche de détection, la couche réseau, la couche service, et la couche application [23] associées des capacités de gestion et de sécurité. Les deux premières couches reflètent les deux couches de l'architecture de base et les deux derniers représentent la couche application.

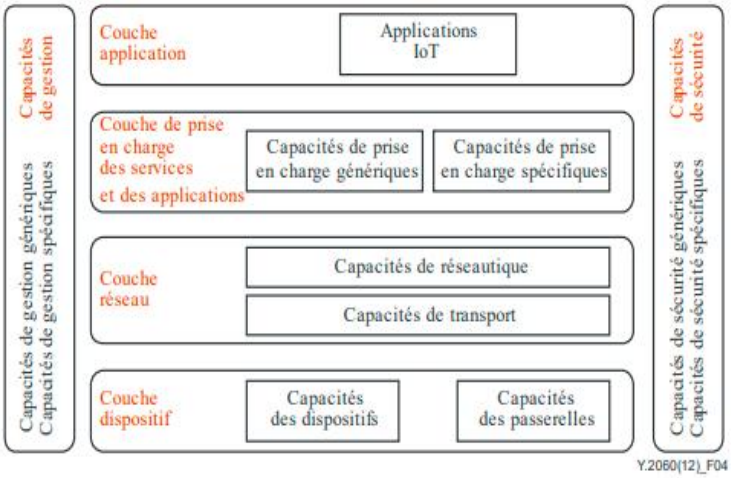


Figure 10: Architecture de référence de l'UIT-T [24]

III.2.1. La Couche de détection

Elle est la couche la plus importante de ce modèle et représente les objets physiques finaux de l'architecture possédant la capacité à communiquer, à collecter et à envoyer les données à leurs destinations. La collecte d'informations se fait à l'aide de différents appareils tels que la carte à puce, l'étiquette RFID, les réseaux de lecteurs et de capteurs, etc. Elle a une fonction de détection complète à travers le système RFID pour obtenir des informations sur les objets à tout moment et n'importe où. Elle veille aussi à la consommation de l'énergie en communiquant uniquement en cas de besoin. Elle est connectée à la couche supérieure (couche réseau) via des canaux de communications utilisant Wi-Fi, Zigbee, 3G/4G, les réseaux LTE, les réseaux Ethernet ou les lignes d'abonné numérique (DSL) [25]. La quantité de données générées par l'IoT provient de cette couche.

III.2.2. Couche réseau

Cette couche sert de relais entre les objets connectés et les deux couches (service et application). Elle a comme rôle de veiller à la communication des objets dans le réseau en assurant le contrôle d'accès et le contrôle des ressources de transport, la gestion de la mobilité

ou l'authentification et l'autorisation. Elle a aussi des capacités de transport destinées essentiellement à assurer la connectivité nécessaire pour le transport des informations propres à chaque service ou application IoT ainsi que pour le transport des informations de contrôle et de gestion relatives à l'IoT. À cette couche, les passerelles et les serveurs de pointe peuvent effectuer des analyses locales et transférer les données vers la couche services via diverses technologies, telles que RFID, 3G, 4G, GSM, UMTS, Wifi, LTE, ZigBee, Lora, Sigfox, etc.

III.2.3. Couche service

C'est la couche en charge des services et des applications a deux tâches :

- Les capacités de support générales en exécutant les tâches de traitement et de stockage des données pouvant être utilisées par les applications IoT.
- Les capacités de support spécifiées pouvant exécuter les besoins de service de divers objectifs spéciaux.

On retrouve dans cette couche des middlewares, de cloud computing, des plateformes M2M et aussi des technologies de fog computing qui sont capables de gérer des données générées dans la couche réseau.

III.2.4. Couche application

Elle est la couche supérieure de l'architecture. Elle est constituée des applications IoT, utilisateurs finaux des données acheminées par les objets. Ces applications doivent être créées de manière à répondre aux besoins de nombreux marchés dans différents domaines tels que la surveillance sanitaire, la construction intelligente, l'agriculture intelligente, le transport intelligent, les maisons intelligentes, la surveillance routière, l'industrie intelligente, etc.

III.3. Domaines d'applications

Le potentiel des objets connectés est énorme. A peine quelques années, ils ont envahi notre quotidien sans même que nous y prêtions attention. L'IoT touche maintenant presque tous les domaines (voir figure 11). Ces résultats fructueux ont abouti à l'élargissement du domaine et aussi une assurance pour le futur. Il est mis à profit à travers différents domaines d'utilisation. Voici quelques-uns :

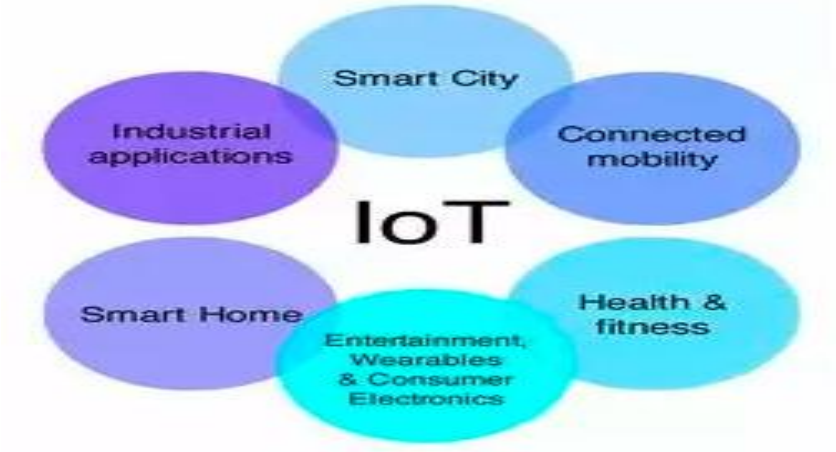


Figure 11: Domaine d'application de l'IoT [26]

III.3.1. Domaine de la santé

Les compteurs d'énergie, les moniteurs connectés, le système télémédecine, les machines à rayons X, tous ces dispositifs disponibles aujourd'hui dans la santé sont possibles grâce à l'internet des objets.

Les objets connectés offrent plusieurs services au quotidien dans ce domaine comme la surveillance au sein des établissements médicaux, la maintenance, les opérations chirurgicales, les services de géolocalisation et le contrôle à distance. En effet, les systèmes automatisés, en collectant et en analysant de manières active et passive les données nécessaires des patients, peuvent effectuer une importante partie du travail des médecins (diagnostics, tests, prescriptions, modifications de comportement) [26]. Pour cela, la base de données collectée est riche en renseignements permettant aux médecins d'avoir une vue globale sur l'état de chaque patient et d'être alertés en cas d'urgence ou de besoin.

Le e-sante est utilisé quasiment dans tous les hôpitaux mondiaux et sa normalisation dans cette sphère va permettre de créer de nouvelles applications de fonctionnement qui augmentent l'efficacité des personnels de santé, mais aussi la collaboration entre soignants ainsi que la liaison avec les patients.

III.3.2. L'agriculture

Les systèmes conçus dans l'agriculture intelligent basé sur l'IoT sont déployés pour surveiller le champ 24h sur 24 à l'aides des capteurs tel que la température, la lumière, l'humidité, l'état du sol, l'automatisation du système d'irrigation et les informations qu'ils recueillent, sont accessibles sur n'importe quel endroit. L'application la plus répandue dans l'agriculture

intelligente basée sur IoT est l'agriculture de précision permettant d'utiliser la quantité optimale d'eau, réduire les dépenses, des pesticides dont la culture a besoin et produire de meilleure récolte et plus saines.

III.3.3. Le smart city

Une ville est dite intelligente lorsque la ville développe les technologies de l'information et de la communication (TIC) pour améliorer la qualité des services urbains. Par conséquent, la ville intelligente devrait répondre aux enjeux comme la capacité à gérer des infrastructures connectés (foyers domotiques, éclairage citadine intelligent) adaptables, durables et plus efficaces en autonomisant pour faciliter la vie des citoyens et tout en respectant l'environnement.

III.3.4. L'industrie connecté

IIoT désigne l'application de la technologie IOT dans le domaine de l'industrie en intégrant des capteurs et des actionneurs dans des machines et des logiciels connectés les uns aux autres via le réseau. Et cela apporte à l'industrie une accélération des processus, l'auto-optimisation des équipements et des installations industrielles, l'augmentation de la productivité dans de nombreux secteurs tels que le transport, le ferroviaire, le pétrole, le carburant, l'aviation etc.

Parfois appelé aussi industrie 4.0, il utilise depuis peu une communication M2M (Machine à Machine), et fait appel à des technologies Cloud, le machine Learning ou l'analogie en temps réel pour générer de nouveaux bénéfices et de nouvelles innovations.

Conclusion

Ce chapitre a permis de poser les bases théoriques nécessaires pour comprendre les technologies fondamentales qui sous-tendent l'Internet des objets (IoT). Nous avons exploré en détail les réseaux de capteurs sans fil (RCSF), en examinant leur architecture, les différents types de modèles de communication ainsi que leurs applications dans divers domaines, tels que l'agriculture, la santé, ou encore la sécurité militaire. De plus, l'étude des technologies RFID et de leur architecture a mis en lumière l'importance de ces dispositifs dans l'identification et le suivi des objets.

L'architecture de l'IoT a également été abordée, notamment les différentes couches qui composent un système IoT (couche de détection, réseau, service, et application), ainsi que les technologies de communication qui permettent l'interconnexion entre ces différentes couches. Nous avons ainsi pu constater que l'IoT englobe une variété de technologies, chacune ayant

ses propres spécificités, mais partageant toutes le même objectif : offrir une connectivité intelligente à des objets de plus en plus nombreux.

L’étude de l’IoT nous a permis d’avoir une vue d'ensemble des technologies qui le composent, tout en soulignant les défis techniques liés à la gestion de ces réseaux hétérogènes. Ces fondations technologiques sont essentielles pour aborder les questions de sécurité dans les chapitres suivants, où nous examinerons comment protéger ces systèmes contre les menaces croissantes auxquelles ils sont confrontés.

Chapitre 2 : Etat de l’art sur la sécurité dans l’IOT

L'internet des objets est l'une des technologies les plus prometteuses qui vise à améliorer la qualité de vie de l'être humain. Il joue un rôle important dans plusieurs domaines dont certains énumérés dans le chapitre précédent. Toutefois, cette technologie IoT a également introduit de nouveaux types de risque pour la sécurité des organisations. De ce fait, la sécurité est très souvent abordée et discutée dans cette sphère et il est essentiel d'établir des mécanismes de sécurité car l'émergence de l'internet des objets a également modifié radicalement le paysage des cybermenaces. Le déploiement à grande échelle de ces dispositifs intrinsèquement vulnérables crée un nombre exponentiel de vecteurs d'attaques, qui, à leur tour, introduisent un ordre exponentiellement plus élevé de risque pour la sécurité. Ainsi le changement de paradigme provoqué par l'internet des objets semble avoir créé la situation parfaite pour la sécurité.

Pour bien comprendre les défis et les solutions de sécurité apportés dans l'IoT, une étude profonde est abordée dans ce chapitre en se basant sur des cas concrets d'applications avec une architecture bien précise.

Comme l'IoT présente une large gamme d'objets connectés et une variéte de domaines d'application, pour mieux adresser la question de séciruté, nous avons choisi la domotique plus communément appelée maison connectée. C'est l'une des domaines les plus risqués pour l'IoT pour protéger les données privées, qui sont en l'occurrence les données des utilisateurs et des propriétaires d'objets connectés.

I. Défis en matière de sécurité

La sécurité représente un élément essentiel pour un système domotique. Il est impératif d'identifier les menaces et attaques pouvant cibler les maisons connectées. On peut citer les menaces contre les périphériques et les menaces contre la communication. Pour limiter notre étude, nous nous concentrons sur la communication des objets connectés et le transfert des données jusqu'à leur stockage dans le point central. Il est difficile d'accéder physiquement aux objets connectés de la maison. En revanche, accéder à ces objets à distance au travers des communications depuis l'extérieur est tout à fait possible. En étant connectés à l'internet, ces réseaux domestiques peuvent être reliés à l'extérieur et être en contact avec les objets connectés. Ces genres de réseaux suscitent beaucoup d'intérêt car le flux et la quantité de données qui y circulent sont très importants d'où l'importance d'établir des solutions de sécurité pour la protection des données et de la vie privée des utilisateurs. Si un pirate arrive à accéder à un réseau domestique, il pourra [27] :

- Avoir des informations sur la vie privée d'un utilisateur ;
- Tirer des profits en faisant des modifications sur la configuration du réseau. Il peut par exemple désactiver les alarmes, ouvrir les fenêtres ou la porte puis accéder physiquement à la maison pour la cambrioler ;
- Causer des dégâts tels que l'augmentation de la facture de consommation en allumant et réglant aux valeurs maximales le chauffage, les lumières ou la climatisation. Il peut créer des dégâts plus considérables si le réseau est relié à des objets comme les robinets, elle peut désactiver, par exemple, les actionneurs et les détecteurs de fuites d'eau ou de gaz puis ouvre les robinets.

À cause de la multitude des menaces et des vulnérabilités qui s'en suivent et qui pèsent sur la sécurité de la vie privée des utilisateurs, de nombreuses attaques sur ces systèmes sont possibles. Dans ce contexte, diverses études ont souligné les besoins de sécurité de ces domaines en décrivant diverses attaques.

- **Min-in-the-Middle (MiTM) :** l'attaque de l'homme du milieu est une attaque d'usurpation d'identité avancée dans laquelle une entité malveillante s'interpose dans l'échange entre deux ou plusieurs objets IoT. Il emprunte l'identité des deux objets pour intercepter les données sensibles et ensuite les rejouer. Par cette démarche l'attaquant peut récupérer des données sensibles qui peuvent entraîner plusieurs conséquences. Pour notre architecture, cette attaque peut violer la vie privée des utilisateurs en s'octroyant des données sensibles comme le quotidien de l'utilisateur.

- **Déni de service :** Le DoS est l'une des attaques réseau les plus fréquentes et peut avoir un impact significatif sur les systèmes IoT. Les attaques DoS peuvent rapidement épuiser les ressources, provoquant l'indisponibilité des systèmes IoT et de graves conséquences, en compromettant les liens de communication et en inondant les réseaux IoT de données massives. Pour ce faire, il peut envoyer continuellement un nombre important de messages à l'objet jusqu'à ce que les ressources de l'objet s'épuisent (débordement de la mémoire, drainage de la batterie, etc.). De plus, ces attaques ont un impact beaucoup plus important lorsqu'elles sont effectuées de manière distribuée (DDos).

- **Attaques par phishing** : Les attaques de phishing sont souvent utilisées pour cibler les utilisateurs d'appareils IoT. Les attaquants peuvent envoyer des e-mails ou des

messages de Phishing pour inciter les utilisateurs à révéler leurs identifiants de connexion ou à télécharger du code malveillant.

- **L'attaque de gouffre** (Sinkhole) est une attaque qui a pour but d'amener les nœuds voisins à partager des informations à un nœud corrompu avec des informations trompeuses sur le chemin de routage. Ce qui a pour conséquence que les flux de données provenant d'un nœud particulier sont déviés vers le nœud compromis. Le trafic est alors réduit au silence pendant que le système est trompé, croyant que les données ont été reçues d'un nœud non compromis. De plus, cette attaque entraîne une consommation d'énergie plus élevée qui peut provoquer un déni de service (DOS).

- **L'accès aux étiquettes RFID** est parfois possible en raison de l'absence d'authentification appropriée. En effet, il existe un grand nombre de systèmes RFID qui peuvent être accessibles par une personne sans autorisation. Au-delà de simplement lire les données, l'attaquant pourrait aussi possiblement les modifier ou les supprimer. Par exemple le non accès au domicile de l'utilisateur sécurisé par une carte RFID.

- **Accès non autorisé au service** : un mécanisme d'authentification faible peut actionner les attaquants grâce aux services IoT proposés en particulier ceux exposés à l'internet, d'établir un contrôle à distance non autorisé en violant l'intégrité, la confidentialité, l'authentification des données. On peut citer comme attaque, l'attaque de contournement ou l'attaque de dégradation de service

Il est très important de prendre en compte les exigences de sécurité et de protection de la vie privée des personnes depuis la conception des systèmes jusqu'au déploiement pour contrecarrer ces attaques. Il est aussi important de sensibiliser les utilisateurs sur les enjeux de sécurité et de protection de la vie privée dans leur utilisation de l'IoT. Pour un système sécurisé, il existe des protocoles de communication qui implémentent des mécanismes de sécurité permettant de se protéger contre ces attaques.

II. La sécurité d'un système de surveillance domestique

Les réseaux de surveillance domestique utilisent des capteurs de mouvement, des caméras de surveillance et d'autres dispositifs connectés pour surveiller les maisons et les entreprises contre les intrusions et les incidents indésirables ou bien un réseau qui permet l'accès à un monde ambiant, où les objets domestiques collaborent entre eux pour améliorer l'habitat humain. Ce concept permet de gérer tous les équipements (tâches ménagères, verrouillage des portes, des fenêtres, vidéosurveillance, etc.). Les maisons connectées deviennent de plus en plus populaires avec l'introduction d'un grand nombre d'applications de l'Internet des objets (IoT) et leurs appareils intelligents respectifs, dont beaucoup sont connectés directement à Internet pour rendre possible la gestion à distance. La connectivité à Internet et la popularité croissante de ces systèmes, présente de sérieux défis en termes de sécurité et de gestion efficace de la maison connectée (intelligente).

II.1. L'architecture proposée pour une maison intelligente

Pour mieux limiter notre domaine d'étude, nous prenons en exemple une architecture domotique de la figure 12.

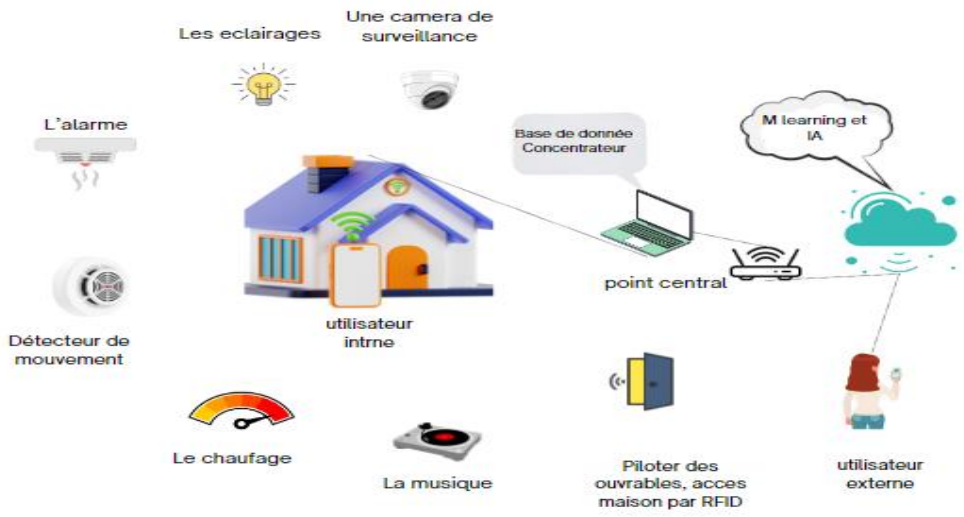


Figure 12 : Architecture Domotique

Elle offre un service de domotique composé de :

- D'équipements pour commander à distance via une application d'un smartphone (piloter les ouvrages, commander les éclairages de la maison, réguler le chauffage, diffuser la musique, etc.) ;

- D’une carte RFID pour l’accès à la maison ;
- D'un système de surveillance composé d’un détecteur de mouvement, d’un centre d’alarme (unité de gestion des informations transmises par le détecteur),
- D’une commande domotique qui permet en cas de déclenchement de l’alarme : l’allumage des lumières et éventuellement la mise en marche de l’enregistrement des vidéos, une ou plusieurs caméras, un écran distant (ordinateur, téléphone).
- D’un point central qui permettra de stocker les données collectées localement mais aussi qui agit en tant que passerelle entre les objets connectés et le point d’accès vers le monde extérieur (Internet) pour accéder au service Cloud et l’accès à distance des applications liées aux objets connectés. Son rôle est de s'occuper de la gestion des transmissions des données provenant des objets connectés, mais également des données provenant de l'extérieur de la maison connectées.

Pour le fonctionnement de cette architecture, le point central (ordinateur classique) est constitué d’une base de données permettant de stocker les données localement et un concentrateur (HUB), un Home Assistant libre pour la solution d’automatisation domotique des objets connectées comme décrit dans [28] et une gestion centrale de sécurité. Le hub a le rôle d’intercepter tout le trafic provenant des objets connectés et ceux aussi destinés aux objets connectés comme des mises à jour.

A titre illustratif, nous prenons le scénario où le détecteur de mouvement repère une anomalie relative à une intrusion par exemple qu’il transmet au centre d’alarme via une technologie de communication. Après la réception du message, il déclenche la commande domotique qui prend en charge la mise en route de l’alarme, l’envoi de messages aux numéros préprogrammés (mobiles) et le lancement des enregistrements des caméras et allumages des lampes. Elles filment des images qui sont instantanément disponibles à l’interne du réseau ou bien à distance grâce au flux vidéo envoyé au point central.

II.2. Les services de sécurité pour notre architecture

Les services de sécurité que nous considérons pertinents et nécessaires pour assurer la protection du réseau domestique sont : l'authentification des utilisateurs, le contrôle d’accès à la maison, l'intégrité, la confidentialité des données et la disponibilité du service à tout moment.

II.2.1 La Confidentialité

Le service de confidentialité protège les flux de données contre la divulgation et l'analyse du trafic par des entités non autorisées. Le chiffrement des données est le mécanisme de sécurité le plus efficace pour garantir ce service de sécurité. Il est utilisé dans ce cas pour protéger les données (des flux de vidéo) en transit et au repos dans le point central et aussi les mots de passe des utilisateurs. Avant d'être transmises sur le réseau, les données sensibles collectées par les appareils IoT doivent être chiffrées, empêchant les attaquants d'intercepter et d'analyser de trafic (connaître les habitudes des résidents). Ce service est très important pour la sécurité de la vie privée des utilisateurs.

II.2.2. L'authentification

L'établissement de l'identité de l'utilisateur d'un service IoT se repose sur l'utilisation d'un identifiant attribué de manière individuelle à un utilisateur. Dans ce cas l'utilisateur doit s'authentifier pour l'accès à la maison par une carte RFID qui est associé à un numéro d'identification unique et ce service est offert par le point central qui compare l'identifiant de la carte et ceux enregistré dans la base de données, l'accès aux autres services par le smartphone requiert de l'authentification et de l'autorisation et aussi les objets doivent s'identifier avant d'envoyer des messages. Par exemple, quand le détecteur envoie un message d'alerte lorsqu'il y'a présence d'intrus dans la maison, ou bien quand il faut déclencher l'alarme. Étant donné que les objets d'un réseau domestique communiquent, ils doivent s'authentifier mutuellement. Un objet ne doit accepter ni agir sur les messages provenant d'un autre objet ou envoyer des messages à un autre objet que s'il est sûr que l'objet fait partie du réseau et qu'il s'agit du bon objet. Un objet ne doit accepter que les messages provenant d'objets en qui il a entièrement confiance.

II.2.3. L'intégrité

L'intégrité des données et l'intégrité des objets sont les deux principales composantes de l'intégrité, qui est un service de sécurité indispensable dans ce contexte. L'intégrité des données garantit que les données échangées dans la maison connectée n'aient pas été détruites ou modifiées lors de leur acheminement d'une manière non autorisée. Cela est nécessaire pour fournir un service fiable en s'assurant que les commandes reçues par les objets et les informations collectées sont légitimes.

Le deuxième type d'intégrité est lié aux objets, ils peuvent être physiquement attaqués si l'attaquant a un accès à la maison, par exemple en modifiant le code d'exécution en cours de ces objets, ce qui nécessite l'intégrité des objets. Ce deuxième service d'intégrité, qui est à

garantir, permet la détection et l'empêchement des modifications apportées à la configuration des objets. L'intégrité des objets permet également l'élimination et le verrouillage des périphériques non conformes.

II.2.4. La disponibilité

La disponibilité est la capacité d'une entité autorisée, sur demande, d'accéder à des ressources et de les utiliser après un contrôle d'accès. Par conséquent, la disponibilité est un service de sécurité pour empêcher une attaque de type Dos et aussi permettre à l'utilisateur d'accéder au service de la gestion de la maison sans interruption.

Pour assurer tous ces services, il va falloir choisir le bon protocole assurant une communication de bout en bout en même temps la sécurité.

III. Les protocoles de communication de la couche application

Dans l'Internet des objets (IoT), des milliards de dispositifs hétérogènes doivent interagir entre eux. L'interopérabilité et la sécurité sont deux des principaux défis à relever pour assurer un fonctionnement harmonieux de l'IoT. La communication dans l'IoT s'effectue à l'aide de différents protocoles, et le type de protocole IoT à employer dépend de la couche et de l'architecture du système sur lequel les données doivent circuler.

Les protocoles de communication au niveau de la couche application font partie des éléments moteurs de l'écosystème IoT car ils sont à la base de toute communication entre les objets IoT et aussi le cloud. On peut les classer en deux catégories par rapport à leur fonctionnement :

- Les protocoles de messageries (MQTT, CoAP, AMQP, DDS, XMPP)
- Les protocoles de découverte de service (mDNS, SSDP)

III.1. Les protocoles de messageries

III.1.1. MQTT

MQTT (Message Queuing Telemetry Transport) est un protocole idéal pour la communication IoT et M2M (machine to machine) dans laquelle la priorité est d'économiser la puissance de calcul et la bande passante du réseau. Il dispose d'un mécanisme de qualité de service (QoS) qui garantit la livraison des données. MQTT s'appuie sur le protocole TCP/IP comme base de communication. Il a été publié en 2010 et est devenu une norme en 2014 selon OASIS (Organization for the Advancement of Structured Information Standards).

a. Architecture Du Protocole MQTT

Un protocole de messagerie push avec un modèle éditeur/abonné (pub-sub) répondant aux exigence IoT tels qu'un en-tête léger, une messagerie bidirectionnelle, une livraison fiable des messages, un temps de réponse rapide, la prise en charge de connexions d'appareils illimitées, une facilité d'installation et un besoin de ressources système minimale. Le modèle de messagerie MQTT se compose de deux composants principaux du réseau (voir figure 13) : un courtier (broker) de messages et des clients (Publisher, Subscriber), dans lesquels tous les messages publiés sont collectés et transmis aux clients abonnés respectifs par le courtier. Pour ce faire, les clients se connectent à un serveur central(broker) qui fait office de proxy. Afin de filtrer les messages envoyés à chaque client, les messages sont classés en rubriques organisées hiérarchiquement. Les clients peuvent publier des messages dans un sujet. D'autres clients peuvent s'abonner au sujet et le courtier leur enverra des messages abonnés.

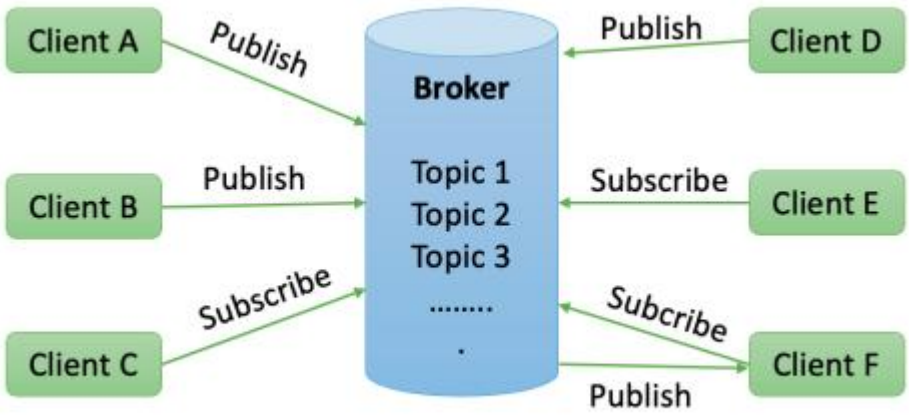


Figure 13: Architecture du protocole MQTT [29]

Pour se connecter au serveur, le client envoie un message CONNECT contenant les informations nécessaires (nom d'utilisateur, mot de passe, identifiant client, etc.). Le courtier répond par un message CONNACK contenant le résultat de la connexion (acceptée, rejetée, etc.) [30].

Pour envoyer des messages, les clients utilisent des messages PUBLISH, qui contiennent une rubrique et une charge utile. Pour s'abonner ou se désabonner, ils utilisent les messages SUBSCRIBE et UNSUSCRIBE et le serveur répond avec SUBACK et UNSUBACK. En revanche, pour garantir que la connexion est active, le client envoie périodiquement des

messages PINGREQ et le serveur répond avec PINGRESP. Enfin, le client se déconnecte en envoyant un message DISCONNECT [31].

b. La sécurité avec MQTT

À l'origine, le protocole MQTT (Message Queuing Telemetry Transport) avait pour but de permettre le transfert de données de manière efficace sur des liaisons de communication coûteuses et peu fiables. Par conséquent, la sécurité n'était pas une priorité absolue dans le développement et la mise en œuvre de MQTT [32].

En effet, il existe des mesures de sécurité pour MQTT, mais elles entraînent une surcharge en termes de transfert de données et d'empreinte mémoire. Autrement dit, la sécurité a un impact sur l'efficacité du protocole. MQTT prend en charge divers mécanismes d'authentification (OAuth, Customer Managed Certificates, etc) ainsi que le chiffrement basé sur TLS. Transport Layer Security (TLS) est une suite de protocoles Secure Socket Layer (SSL) et la version actuelle est TLS 1.3 [33]. Il permet une communication sécurisée de bout en bout entre deux machines en utilisant la cryptographie pour assurer l'authenticité des données et des points d'extrémité, ainsi que la confidentialité des données. Lors de l'établissement d'une connexion, le protocole de poignée de main permet aux deux parties de se mettre d'accord sur la manière dont elles s'identifient l'une de l'autre [21]. Par exemple l'authentification du serveur et du client avec un certificat dans la couche de transport. Pour s'authentifier auprès du courtier, un identifiant unique est fourni à chaque client. Cet identifiant unique est utilisé pour se connecter au courtier dans MQTT CONNECT.

L'authentification en fournissant un nom d'utilisateur et un mot de passe au niveau de la couche d'application est possible aussi avec MQTT mais ses données sont diffusées en texte clair.

L'autorisation est également un moyen de sécuriser la couche d'application, dans laquelle les droits d'accès sont attribués à une ressource particulière. Dans MQTT, la liste de contrôle d'accès (ACL)est utilisée pour l'autorisation et déployée du côté du courtier. La liste de contrôle d'accès permet d'attribuer les droits d'accès et des activités autorisées à une tâche particulière. Tous les noms d'utilisateurs et les mots de passe appartenant à un utilisateur et ses rubriques publiés ou abonnées sont enregistrés dans ACL dans MQTT.

III.1.2. CoAP

CoAP (Constrained Application Protocol) est un protocole de transfert web client/serveur, destiné à fonctionner sur des appareils contraints dont l'énergie, la puissance de traitement, l'espace de stockage et les capacités de transmission sont limitées. Pour cela, CoAP devient un protocole web léger supportant les communications de type M2M (Machine à Machine) entre capteurs connectés dans l'IoT. Le protocole CoAP a été conçu sur la base de REST (Representational state transfer) et opère sur le protocole de transport UDP avec prise en charge des interactions monodiffusion et multidiffusion et qui fournit un service de transport assez simple et dont l'adoption pour les réseaux de capteurs est bien approuvée. Pour la sécurité, CoAP utilise le schéma d'URI « coaps » pour une ressource sécurisée grâce au protocole DTLS sur un port UDP donné [34].

a. Architecture Du Protocole CoAP

Comme CoAP se focalise sur le modèle REST (voir figure 14), le serveur CoAP (un capteur) rend les ressources accessibles aux clients à travers des URI (Uniform Resource Identifier) identifiant les ressources, et des méthodes bien déterminées que les clients spécifient dans leurs requêtes. Pour envoyer une donnée, un client envoie une requête CoAP qui contient le type de message (CON ou NON), l'identifiant du message (ID) et une méthode (GET, POST, PUT ou DELETE). Le serveur répond aussi à la requête du client par un message contenant, le type de message (ACK), le même ID du client et un code réponse correspondant [19].

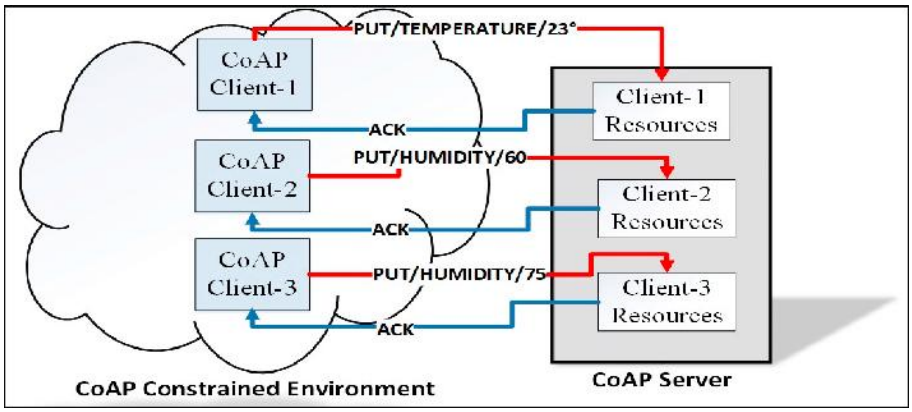


Figure 14 : Architecture du protocole CoAP[25]

Les transactions CoAP utilisent quatre types de messages différents pour combler la non fiabilité du protocole UDP [36] :

- Confirmable (CON) ;
- Non Confirmable (NON) ;

- Acknowledgment (ACK) ;
- Reset (RST)

Un message fiable s'obtient à l'aide du message confirmable (CON). Lorsqu'aucun paquet CON n'est reçu, il suscite exactement un message de retour de type Accusé de réception (ACK) ou de type Réinitialisation (RST). Les Non Confirmable (NON) sont des messages qui ne nécessitent pas d'accusé de réception de la part du serveur. Un message d'accusé de réception (ACK) confirme qu'un message confirmable spécifique (identifié par son ID de transaction) est arrivé. Un message de réinitialisation (RST) indique qu'un message confirmable spécifique a été reçu, mais qu'il manque un certain contexte pour le traiter correctement.

b. La sécurité avec CoAP

Comme MQTT, CoAPP n'a pas de mécanisme de sécurité directement intégré dans la propriété mais il est conçu pour fonctionner avec des mécanismes comme DTLS et OSCORE qui fournissent la sécurité nécessaire pour les applications IoT.

Le schéma d'URI « coaps » est utilisé pour une ressource sécurisée grâce au protocole DTLS sur un port UDP donné [37]. DTLS (Datagram Transport Layer Security), basé sur de nouvelles versions TLS optimisées et adaptées à l'IoT a été conçu pour sécuriser de bout en bout la communication entre deux équipements [36]. Il reprend les fonctionnalités de ce dernier mais utilisera la couche Transport fournie par UDP contrairement à TLS qui utilise TCP.

DTLS est un protocole composé de deux couches, une couche supérieure comprenant quatre protocoles (NoSec, PreSharedKey[RFC4279], RawPublicKey[RFC7250], certificat X.509[RFC5280]) qui va s'occuper de l'authentification des hôtes, du signalement des erreurs et du chiffrage des données; et une couche inférieure « Record Protocol » qui fournit un chiffrement par clé symétrique sécurisé pour assurer la confidentialité et/ou l'intégrité du message.

IPSec [38], protocole de sécurité de la couche 3, peut protéger les applications des couches d'application et de transport. Par exemple il peut être employé pour une sécurisation des flux CoAP dans des environnements contraints et plus particulièrement Encapsulating Security Payload Protocol [RFC2406] (IPSec-ESP) lorsque CoAP est utilisé sans DTLS. L'IPSec peut offrir divers services de sécurité tels que la confidentialité limitée des flux de trafic, le

mécanisme anti-rejeu, le contrôle d'accès, la confidentialité, la protection des données, l'intégrité sans connexion et l'authentification de l'origine des données.

III.1.3. XMPP

Le protocole XMPP (Extensible Messaging and Presence Protocol) est un protocole de communication ouvert basé sur XML (Extensible Markup Language). Il est utilisé principalement dans l'IoT pour sa messagerie instantanée, sa fonctionnalité de présence, de gestion des états, et sa communication en temps réel. XMPP met en œuvre une interaction client-serveur et une interaction pubish/subscriber, s'exécutant sur le protocole TCP. Il fonctionne sur une variété de plateformes basées sur Internet de manière décentralisée. Les clients XMPP se connectent à un serveur XMPP via une connexion TCP/IP.

a. Architecture du protocole XMPP

L'architecture du protocole XMPP (voir figure 15) est conçue pour offrir des communications en temps réel et des services de présence dans un réseau décentralisé. Les clients XMPP sont des applications ou dispositifs qui initient des connexions aux serveurs XMPP pour envoyer et recevoir des messages à l'aide des trophées XML [35]. Ils communiquent avec les serveurs via une connexion TCP/IP sécurisée. Les serveurs XMPP jouent un rôle central dans la gestion des connexions des clients, l'acheminement des messages, et la fourniture de services de présence (en ligne, hors ligne, occupé, etc.) et d'authentification. Les serveurs peuvent également interagir entre eux pour créer un réseau distribué. Une évolutivité élevée dans XMPP est fournie par architecture décentralisée. Il existe de nombreuses extensions pour le protocole XMPP, cela lui permet de travailler sur l'environnement sans infrastructure.

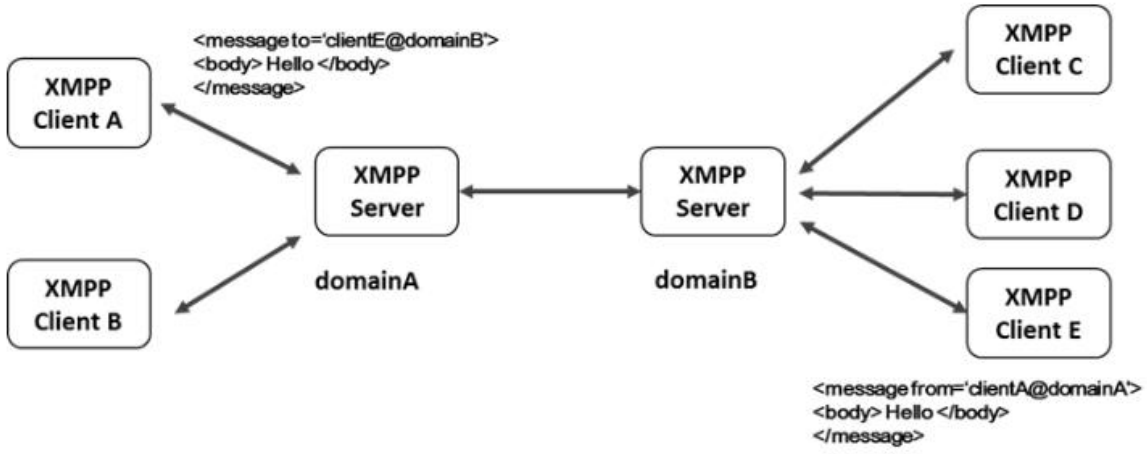


Figure 15 : Architecture du protocole XMPP [39]

b. La sécurité de XMPP

La connexion TCP/IP du protocole XMPP est généralement sécurisée avec TLS (Transport Layer Security) pour garantir la confidentialité et l'intégrité des données. XMPP implémente aussi SASL (Simple Authentication and Security Layer), qui garantit la validation du serveur, Cela peut inclure des noms d'utilisateur et des mots de passe, ou d'autres méthodes d'authentification plus sécurisées.

SASL (Simple Authentication and Security Layer) est un protocole conçu comme un cadre modulaire qui permet d'intégrer différentes méthodes d'authentification. Cela permet aux protocoles de communication de prendre en charge une variété de mécanismes d'authentification sans être limités à un seul. La couche d'authentification et de sécurité simple (SASL) [RFC4422] est une méthode permettant d'ajouter un support d'authentification aux protocoles basés sur la connexion. Pour utiliser cette spécification, un protocole inclut une commande permettant d'identifier et d'authentifier un utilisateur auprès d'un serveur et de négocier éventuellement une couche de sécurité pour les interactions ultérieures du protocole [40].

Apres l'établissement de la session TLS, le client XMPP se connecte à un serveur XMPP, en négocient un mécanisme SASL approprié à utiliser pour l'authentification. Le serveur XMPP annonce les mécanismes SASL qu'il supporte (comme le PLAIN, DIGEST-MD5, SCRAM-SHA1/ SCRAM-SHA-256, OAUTHBEARER) [40] et le client choisit parmi ces mécanismes et envoie la réponse appropriée au serveur. Ensuite le client et le serveur échangent les informations nécessaires pour compléter l'authentification selon le mécanisme choisi. A la fin, Le serveur informe le client de la réussite ou de l'échec de l'authentification. En cas de succès, la connexion est établie et sécurisée.

III.1.4. DDS

DDS (Data Distribution Service) est un protocole basé sur une communication entre pairs (des systèmes légers de gestion des données) utilisant les données au lieu des messages. Il offre une interopérabilité centrée sur les données en temps réel en utilisant le modèle d'interaction « publication/abonnement ». Par défaut, DDS est basé sur UDP, mais il est également possible de l'utiliser avec TCP. If offre un niveau de QoS important, tel que la fiabilité, la durabilité, la vivacité et bien d'autre idéal pour l'internet des objets [21]. L'une de ses principales caractéristiques est son évolutivité, qui découle de sa prise en charge de la découverte dynamique [41].

En matière de sécurité, le protocole DDS propose une variété étendue de mécanismes. Tout comme d'autres protocoles de messagerie, DDS prend en charge les protocoles TLS et DTLS. De plus, pour assurer la confidentialité, l'intégrité et l'authenticité des échanges, la dernière spécification de sécurité DDS de l'OMG définit une architecture basée sur un ensemble de modules intégrés. Par exemple, les plugins [42] fournissent des mécanismes d'authentification et d'autorisation pour les DataWriters et les DataReaders, ce qui permet d'éviter les publications et les abonnements non autorisés [31].

a. Architecture du protocole DDS

Le Domain Participant est l'élément principal de DDS qui représente une entité dans un domaine de communication. Il agit comme un conteneur pour les éditeurs (Publisher) et les abonnés (Subscriber)et gère leurs interactions (voir figure 16). Le Publisher est responsable de la gestion de la diffusion des données. Il peut contenir plusieurs DataWriters pour publier des instances de données sur un Topic spécifique. Le Subscriber lui, il est responsable de la réception des données contenant aussi plusieurs DataReaders pour recevoir des instances de données sur un Topic spécifique.

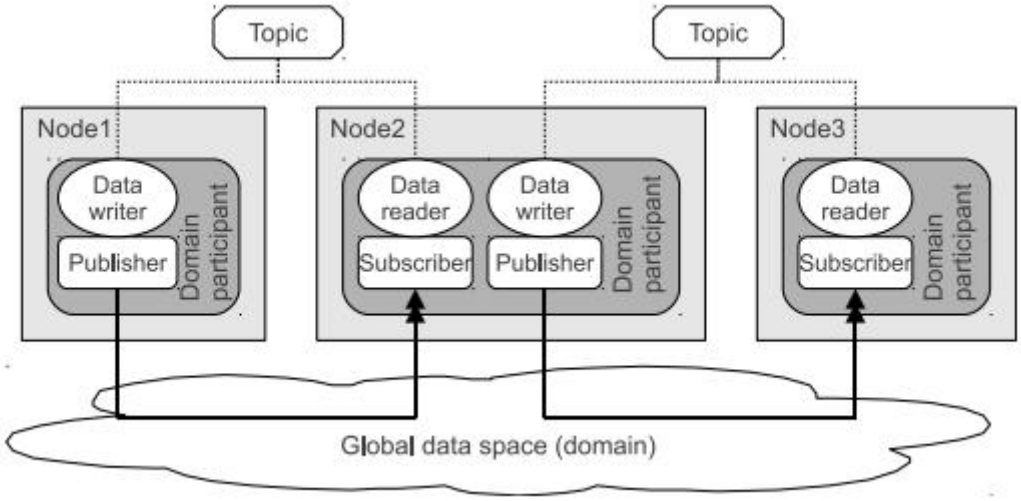


Figure 16: Architecture du protocole DDS [43]

III.2. Comparaison des protocoles de la couche application de l'IoT

Après avoir examiné divers protocoles de communication mentionnés précédemment, nous avons élaboré le tableau 1 qui présente leur modèle de communication, la prise en charge de la qualité de service, les services de sécurité qu'ils offrent et les mécanismes de sécurité qu'ils utilisent, ce qui permet de voir le plus adapté à notre architecture de réseau domotique IoT.

Protocole. Application	**MQTT**	**CoAP**	**AMQP**	**XMPP**	**DDS**
Modéle. Communication	Éditeur/abonné	Réq/réponse	Éditeur/abonné	Éditeur/abonné Réq/réponse	Éditeur/abonné
QoS	Oui	Oui	Oui	Non	Oui
Pro. Transport	TCP/IP	UDP	TCP/IP	TCP/IP	TCP/IP
Authentification	Certificats/ nom d'utilisateur et mots de passe	DTLS/IPsec	SASL	SASL	TLS/DTLS
Chiffrement	TLS	DTLS	TLS	TLS	TLS/DTLS
Consommation	Le plus bas	Moyen	Haut	Haut	Haut
Fiabilité	Haut	Modéré	Haut	Haut	Haut
Evolutivité	Haut	Modéré	Haut	Faible	Haut
Modéle. Architecture	Centraliser	Centraliser	Centraliser	Centraliser	Décentraliser

Tableau 1: Comparaison Des Protocoles de la couche application de L'IoT [31], [37], [44]

Tous ces protocoles sont utilisables dans l'écosystème IoT mais certains se distinguent par leur légèreté et leur facilité à intégrer les systèmes IoT. Dans la littérature, on trouve plusieurs comparaisons mettant en évidence les limites de certains protocoles par rapport à d'autres. Corak et al. [35] examinent les métriques de performances (temps de création des paquets, vitesse de livraison des paquets) des trois protocoles (MQTT, CoAP, et XMPP) et déduits que MQTT offre de meilleurs résultats que les autres car il délivre ses paquets 2 fois plus rapide que le CoAP et aussi un meilleur temps de création des paquets. Concernant XMPP, il présente une latence par rapport aux autres protocoles. Thangavel et al. [45] comparent les protocoles MQTT et CoAP en termes de performance ; les résultats montrent des similitudes entre les deux. Cependant, les chercheurs recommandent d'utiliser MQTT pour les applications liées à la surveillance et au contrôle des environnements. Dans ce contexte, les clients ne sont pas tenus d'être constamment connectés au broker, et la transmission des messages se fait généralement de manière périodique. Saadaoui et al. [41] montrent que le protocole DDS est plus adapté aux applications robotiques.

En nous appuyant sur ces résultats de recherche, nous choisissons d'utiliser le protocole MQTT comme protocole de communication pour notre architecture domotique et nous passons en revue sa sécurité dans la section suivante.

II.2.1. Les Vulnérabilités du protocoles MQTT

Les mécanismes de sécurité comme TLS sont par défaut désactivés pour sécuriser la communication du protocoles MQTT. De ce fait, certains développeurs négligent ces services dans la mise en œuvre et la configuration de leur application ou trop coûteux pour s'adapter aux capacités limitées (par exemple, la bande passante, la puissance de calcul) de nombreux appareils IoT. Cela permet aux attaquants d'exploiter ces failles et de perturber potentiellement les communications de ce protocole ou les fonctionnalités des appareils IoT. Même si ces mécanismes de sécurité sont déployés, il ne garantit pas totalement la sécurité de la communication du protocole MQTT car le problème peut survenir à une mauvaise pratique de mise en œuvre ou l'utilisation de chiffrement faible. Des options spécifiques sont proposées pour chaque protocole et des spécialisations requises pour une bonne sécurité mais les bibliothèques logicielles qui appliquent ces protocoles ne suivent pas réellement la spécialisation, ce qui permet aux attaquant de lancer diverses attaques, telles que le déni de service, contre les services fournis par le protocole de la couche application [44]. Le protocole MQTT, largement utilisé dans l'écosystème IoT présente des vulnérabilités et des risques de sécurité en particulier le courtier. Le tableau 3 met en évidence quelques vulnérabilités décrit dans la littérature.

Nebbione et Calzarossa [31]	• Le service d'authentification et l'autorisation : le courtier MQTT ne vérifie pas correctement l'identité de l'éditeur/abonné, ne bloque pas les accès répétés et aussi ne définit pas correctement les permissions d'éditer/publier ; • Validation des messages : L'envoi de messages (via un éditeur) contenant des caractères non autorisés, peuvent ne pas être correctement décodés par le courtier et l'abonné, présente une vulnérabilité exploitable pour réaliser diverses attaques malveillantes ; • Chiffrement des messages : Le client et le serveur échangent des messages non cryptés, ce qui permet à un pirate d'intercepter et d'usurper les messages en transit par une attaque de

	l'homme du milieu.
RaiKar et S M [46]	● Abonnement à tous les sujets : La confidentialité est violée par l'attaquant lorsqu'il s'abonne en utilisant le sujet. L'attaquant peut voir tous les sujets et les données échangées entre le courtier et l'appareil final IoT. ● Si le courtier n'effectue pas d'authentification, l'attaquant peut publier des données, ce qui entraîne une attaque par déni de service (DoS). ● Confidentialité : La règle de confidentialité est violée si les données ne sont pas cryptées pendant la communication entre les appareils utilisant le protocole MQTT. L'attaquant peut renifler les paquets échangés entre les nœuds pendant le transit.
Sahmi Imane[29]	● Vulnérabilités potentielles telles que les ports qui sont ouverts et peuvent être utilisés pour lancer des attaques de déni de service (DoS) aussi les attaques 'Buffer Overflow' qui peuvent se lancer à travers les réseaux et les appareils ● Lors de l'authentification, le nom d'utilisateur et le mot de passe seraient envoyés en texte clair

Tableau 2 : Les Vulnérabilités du protocole MQTT

Par conséquent, comme illustré dans ce tableau 2, ces vulnérabilités peuvent avoir un impact sérieux sur les environnements MQTT, compromettant à la fois leur disponibilité et la sécurité des données sensibles échangées et stockées. Donc il est crucial de protéger ces données en identifiant correctement les risques potentiels en matière de sécurité et en élaborant des mesures d'atténuation adéquates. Il existe aussi d'autres solutions de sécurité qui peuvent atténuer les risques de sécurité comme l'IA (Intelligence Artificielle).

L’IoT évolue de jour en jour et certaines structures reconnues dans l'écosystème IoT ont validé l’approche de ces méthodes d’IA essentielles au développement futur des systèmes intelligents. Ainsi, grâce à l’intelligence artificielle, notamment l’apprentissage automatique et l’apprentissage profond, nous sommes en mesure de détecter les comportements inattendus ou malveillants des objets connectés (IoT) et de proposer une solution de sécurité dynamique et adaptative.

Conclusion

Ce second chapitre a permis d'établir un état de l'art sur les enjeux de la sécurité dans l'Internet des objets (IoT). Nous avons mis en évidence les défis majeurs auxquels l'IoT est confronté en matière de sécurité, notamment en raison de la diversité des dispositifs connectés, de leur faible capacité de traitement et de stockage, ainsi que de leur interconnexion permanente avec Internet, les rendant vulnérables à diverses attaques.

L'analyse des protocoles de communication au niveau de la couche application, tels que MQTT et CoAP, a révélé des vulnérabilités spécifiques qui rendent ces protocoles particulièrement exposés aux cybermenaces. Même s’ils existent des mécanismes proposés pour assurer la sécurité de ces protocoles, ces mécanismes sont coûteux et ne conviennent pas trop aux dispositifs contraints.

En somme, ce chapitre a mis en lumière l'importance de développer des solutions de sécurité robustes et adaptées aux spécificités des réseaux IoT. Les protocoles utilisés à la couche application, bien qu’efficaces en termes de communication, nécessitent des mécanismes de sécurisation supplémentaires pour faire face aux menaces croissantes. Cette réflexion nous conduit naturellement à explorer dans les chapitres suivants les apports des méthodes d'intelligence artificielle pour renforcer la sécurité des systèmes IoT, en particulier dans le cadre des protocoles de communication comme MQTT.

Chapitre 3 : L'Intelligence Artificielle pour la sécurité dans l'IoT

L'idée de machines capables de penser remonte aux années 40, avec des figures clés comme Alan Turing et son "test de Turing" en 1950 et les travaux de McCulloch et Pitts sur les neurones artificiels en 1943. Le terme "intelligence artificielle" est apparu en 1956 à la conférence de Dartmouth [47]. Depuis, l'IA a évolué, notamment avec l'essor de l'apprentissage automatique et de l'apprentissage profond dans les années 1990-2000, avec des algorithmes comme SVM et les forêts aléatoires. L'IA permet aujourd'hui de traiter de grandes quantités de données, transformant des secteurs tels que la santé, les transports, et la cybersécurité. L'apprentissage profond, basé sur des réseaux neuronaux inspirés du cerveau humain, joue un rôle clé. Avec l'essor de l'IoT, l'IA est devenue essentielle pour la sécurité, en particulier pour détecter les anomalies dans les réseaux IoT et prévenir les cyberattaques. Ce chapitre explore l'application du machine learning à la sécurité de l'IoT.

I. L'Intelligence Artificielle

L'intelligence artificielle (IA) est un domaine de l'informatique qui vise à créer des systèmes capables d'accomplir des tâches qui nécessitent habituellement l'intelligence humaine. Ces tâches peuvent inclure la reconnaissance d'images, la compréhension du langage naturel, la prise de décisions, ou encore l'interaction avec l'environnement. L'IA repose sur la création d'algorithmes capables de traiter de grandes quantités de données pour tirer des conclusions, s'adapter et apprendre de manière autonome. Il peut apporter beaucoup de solutions à la sécurité de l'IoT notamment.

I.1. Surveillance et détection des anomalies

En raison des préoccupations croissantes concernant la sécurité et la sûreté, ainsi que de l'énorme quantité de données de surveillance collectées par des capteurs, bien au-delà des capacités humaines pour les analyser, l'IA devient essentielle. Ces données, qui reflètent le comportement des cibles de surveillance (suspectées de présenter des comportements anormaux), peuvent être exploitées par des algorithmes d'intelligence artificielle pour détecter si ces comportements sont normaux ou anormaux [48].

Par exemple, les modèles de ML (Machine learning) peuvent être déployés sur des flux de données en temps réel pour surveiller les dispositifs IoT de manière continue. Des techniques comme les forêts d'isolement (Isolation Forests) ou le SVM (Machine à vecteurs de support) peuvent être utilisées pour analyser en continu les séquences de données et identifier les comportements inhabituels qui pourraient indiquer une cyberattaque, un dysfonctionnement du système [49] ou bien pour prévoir des comportements futurs basés sur des tendances

historiques. Cette capacité prédictive permet d'anticiper les anomalies avant qu'elles ne se produisent, en fournissant des alertes précoces aux administrateurs système.

I.2. Classification des menaces

Dans le contexte de la sécurité de l'IoT, les menaces peuvent être de plusieurs types : attaques par déni de service distribué (DDoS), attaques par déni de service (DoS), attaques par force brute, attaques par injection de code, écoute clandestine, etc. et aussi varier selon différents critères tels que la nature des attaques (physiques, réseau, logiciel), les cibles (capteurs, passerelles, applications), ou les impacts (perte de données, déni de service, prise de contrôle). L'Intelligence artificielle peut être utilisée pour classer ces menaces en fonction de leurs caractéristiques. Ils peuvent surveiller les systèmes en continu et identifier des comportements anormaux ou suspects en temps réel.

Par exemple, un modèle d'IA peut détecter des pics de trafic inhabituels, des tentatives de connexion répétées, ou des accès à des ressources sensibles qui ne correspondent pas au profil d'utilisation habituel. Une fois un incident détecté, les modèles de machine learning peuvent classifier automatiquement le type de menace, comme une tentative d'exfiltration de données, un ransomware, ou une attaque par déni de service (DDoS) ; Et cela nécessite un ensemble de données étiquetées contenant des exemples de menaces et leurs catégories correspondantes pour prédire la catégorie d'une nouvelle menace. Par exemple les arbres de Décision, les forêts aléatoires (Random Forest) ou le Machine à vecteurs de Support (SVM) sont formés sur des jeux de données annotés et peuvent être utilisés pour classer les menaces en fonction des critères de décision [26].

I.3. Prévention des attaques

Les dispositifs IoT collectent souvent des données sensibles, telles que des informations personnelles, des habitudes de vie ou des données professionnelles. Attendre qu'une attaque se produise avant d'agir peut entraîner des dommages importants et coûteux. Adopter des stratégies proactives de prévention et de sécurité contribue à assurer la résilience des dispositifs IoT et à protéger les utilisateurs et les organisations contre les impacts négatifs des cyberattaques.

La prévention des attaques par le ML consiste à identifier des comportements ou des modèles inhabituels dans les données ou le trafic réseau qui pourraient indiquer une attaque. Les modèles de machine learning apprennent le comportement "normal" du système et détectent les écarts par rapport à ce comportement ou bien des structures inhérentes aux données sans

étiquettes. Comme les IDS (Intrusion Detection Système) [50] ou le Clustering (K-means, DBSCAN) qui peut contrôler le trafic du réseau en temps réel pour détecter les comportements malveillants. Cette approche est de plus en plus adoptée pour renforcer la sécurité des systèmes, y compris dans le contexte de l'Internet des objets (IoT).

I.4. Réponse aux incidents

La réponse aux incidents de sécurité en utilisant des méthodes de machine learning (ML) représente une avancée importante dans la gestion proactive et réactive des menaces. Les techniques de machine learning permettent non seulement de détecter et de prévenir les attaques, mais aussi de réagir efficacement lorsqu'un incident se produit [51]. Lorsqu'un incident est détecté, des systèmes de ML peuvent déclencher des actions automatiques pour contenir la menace, comme isoler les dispositifs compromis, bloquer des adresses IP malveillantes [52], ou révoquer les accès compromis. Ils peuvent recommander ou appliquer des mesures de correction, comme le déploiement de correctifs de sécurité, la restauration des systèmes à partir de sauvegardes, ou la réinitialisation des mots de passe et aussi apprendre des incidents précédents pour améliorer la détection future. Par exemple, un modèle pourrait identifier des signatures de nouvelles menaces basées sur des incidents similaires traités par le passé.

I.5. Renforcement de la Cryptographie

L'Internet des objets (IoT) repose sur des milliers de dispositifs interconnectés, souvent vulnérables à des attaques. La cryptographie est un outil clé pour assurer la sécurité de ces dispositifs en garantissant la confidentialité, l'intégrité et l'authenticité des communications. Cependant, les algorithmes cryptographiques traditionnels peuvent ne pas être suffisants pour répondre aux défis spécifiques de l'IoT, notamment en raison de la faible puissance de calcul des dispositifs et de la nature dynamique de l'environnement [53].

Le machine learning peut être utilisé pour générer ou sélectionner des clés cryptographiques optimisées en fonction des spécificités de l'environnement IoT, ou pour alléger les opérations cryptographiques, ce qui est crucial pour les dispositifs IoT à ressources limitées. Par exemple, ces modèles peuvent apprendre à compresser les clés ou ajuster les processus cryptographiques afin de réduire la consommation d'énergie, tout en maintenant un niveau de sécurité élevé. En surveillant les communications chiffrées, les systèmes basés sur le machine learning sont capables d'identifier des schémas d'attaques spécifiques, comme les tentatives de cryptanalyse, et d'émettre des alertes en cas de menace. Ils peuvent aussi améliorer les

protocoles de négociation de clés en détectant les tentatives d'interception ou automatiser la gestion du cycle de vie des clés cryptographiques dans les environnements IoT, en assurant leur génération, distribution, renouvellement, et révocation, tout en garantissant l'utilisation de clés valides et sécurisées.

II. L'apprentissage automatique (Machine Learning)

L'apprentissage automatique, également appelé en anglais machine learning, est une méthode qui permet de transformer automatiquement des données brutes en connaissances exploitables dans le but d'optimiser la prise de décision. Il repose sur des techniques telles que les statistiques, la classification et le partitionnement des données pour découvrir des modèles et faire des prédictions à partir des informations disponibles. En exploitant de grandes quantités de données, qu'elles soient structurées ou catégorisées, l'apprentissage automatique aide à comprendre ces informations et à traiter de nouvelles données similaires pour orienter les décisions de manière efficace.

Face aux défis complexes et en perpétuelle évolution des systèmes de sécurité modernes, le machine Learning peut être utilisé pour renforcer la sécurité des dispositifs IoT.

II.1. Algorithmes de Machine Learning Courants

Pour mieux comprendre le rôle du machine Learning, nous l'étudions à travers un certain nombre d'algorithmes ou modèles les plus couramment utilisés.

a. Arbres de décision

Un arbre de décision est un algorithme d'apprentissage supervisé utilisé pour la classification et la régression. Il possède une structure hiérarchique et arborescente, qui se compose d'un nœud racine, de branches, de nœuds internes et de nœuds feuille. Chaque nœud de l'arbre représente une condition à évaluer, et les branches indiquent les résultats possibles, menant à des feuilles qui représentent les décisions ou les classes [54].

- **Nœud racine** : C'est le point de départ de l'arbre. Il contient le premier critère de décision qui divise les données en groupes.
- **Nœud internes** : Chaque nœud interne pose un nouveau critère basé sur une caractéristique des données. Ces nœuds sont connectés par des branches qui représentent les réponses possibles.
- **Branches** : Elles relient les nœuds entre eux et indiquent les résultats des tests ou des décisions aux différents niveaux.

- **Feuilles** : Ce sont les nœuds finaux qui donnent la prédiction ou le résultat final. Une feuille indique la classe ou l'action à entreprendre.

Les arbres de décision peuvent être utilisé dans l'IoT pour la détection d'intrusions, les arbres de décision permettent de classifier les activités réseau en comportements normaux ou suspects, en se basant sur des règles simples.

b. Régression Linéaire

La régression linéaire est une méthode statistique et d'apprentissage automatique qui sert à modéliser la relation entre une valeur que l'on souhaite prédire (appelée variable cible) et une ou plusieurs valeurs qui influencent cette prédiction (appelées variables explicatives). Elle est surtout utilisée pour prédire des résultats et analyser des données qui suivent une tendance linéaire, c'est-à-dire lorsque les changements dans les variables explicatives entraînent des changements proportionnels dans la variable cible.

Le modèle de régression linéaire cherche à ajuster une ligne droite (ou un plan dans le cas de plusieurs variables) qui minimise la différence entre les valeurs réelles et les valeurs prédites par le modèle. On peut citer comme type de régression : la régression simple, multiple ou logistique dont certains sont plus adaptés que d'autres par rapport à la complexité du jeu de données.

Les appareils IoT génèrent continuellement des données, et la régression linéaire peut être utilisée pour modéliser le trafic réseau normal, prédire les usages (ressources système comme bande passante, CPU, mémoire) attendus sous différentes conditions, prédire les comportements utilisateurs en fonction des historiques d'accès ou d'utilisation d'appareils IoT etc.

c. Réseaux de neurones artificiels (ANN)

Les réseaux de neurones artificiels sont des modèles inspirés du fonctionnement du cerveau humain. Ils sont constitués de couches de neurones artificiels, organisés en couches d'entrée (les données brutes), cachées (traitent les données grâce à leur connexion), et de sortie (résultat final). Chaque neurone reçoit des informations, les traite et transmet un signal à d'autres neurones, permettant au réseau d'apprendre des relations complexes entre les données [54], [55].

Ces algorithmes apprennent à partir de données d'intrusions passées pour reconnaître des motifs similaires dans les nouveaux flux de données IoT, renforçant ainsi la détection des menaces.

Il existe de nombreux type de réseaux de neurones artificiels tels que les réseaux de neurones récurrents, les auto-encodeurs, les réseaux Transformers ou encore les réseaux antagonistes génératifs (generative adversarial networks).

d. K-Nearest Neighbors (K-NN)

Le K-Nearest Neighbors est utilisé principalement pour la classification, mais aussi pour la régression. Il est basé sur le principe de la proximité entre les données, où la classe d'un point est déterminée par les classes de ses k voisins les plus proches. C'est-à-dire, Il stocke les exemples d'entraînement (les données et leurs classes associées). Lorsque le modèle doit classer un nouveau point, il calcule la distance (exemple distance euclidienne) entre ce point et tous les autres points du jeu de données d'entraînement. Il sélectionne ses k_voisins pour ensuite le classer selon la majorité des classes de ses voisins [26].

KNN peut être utilisé dans la sécurité de l'IoT pour classer des activités réseau en fonction des comportements similaires observés dans le passé, il est efficace dans la classification des attaques et des intrusions [56].

e. K-means

Le K-Means est utilisé pour le clustering (groupement) de données. Il regroupe les données en un nombre prédéfini de clusters (k), en fonction de leur similarité. Initialement l'algorithme Choisit k centres de clusters, appelés centroids, de manière aléatoire dans l'espace des données. Chaque point de données est assigné au centroid le plus proche en utilisant une mesure de distance (par exemple la distance euclidienne). Une fois que tous les points sont assignés à un cluster, le centroid de chaque cluster est recalculé comme étant la moyenne de toutes les données du cluster. Les étapes d'assignation des points et de mise à jour des centroids sont répétées jusqu'à convergence. C'est-à-dire que les points de données restent dans les mêmes clusters [47].

K-Means est souvent utilisé pour identifier des anomalies dans les réseaux IoT. Les points qui ne correspondent à aucun cluster ou sont loin des centroids peuvent être considérés comme des comportements anormaux ou suspects. En analysant le trafic réseau IoT, K-Means peut

détecter des schémas inhabituels qui pourraient indiquer des attaques ou des activités malveillantes [56].

f. Isolement foret

La méthode des "forêts d'isolement" (ou Isolation Forest) est une technique de machine learning utilisée principalement pour la détection d'anomalies. L'algorithme des forêts d'isolement crée plusieurs arbres de décision (des sous-ensembles d'arbres appelés "arbres d'isolement") pour essayer d'isoler chaque point de données. C'est-à-dire, chaque arbre est construit en sélectionnant un attribut et un seuil de manière aléatoire. Les données sont divisées en fonction de ce seuil [56]. Ce processus est répété jusqu'à ce que chaque point soit isolé dans un nœud feuille ou qu'une profondeur maximale soit atteinte. Les points qui sont des anomalies sont isolés plus rapidement car ils sont éloignés des autres points.

II.2. L'apprentissage profond (Deep Learning)

L'apprentissage profond (deep learning) est une sous-catégorie de l'apprentissage automatique qui repose sur des réseaux de neurones artificiels à plusieurs couches, appelés réseaux de neurones profonds (DNN). L'apprentissage profond se distingue par sa capacité à apprendre des représentations complexes des données à partir d'énormes volumes d'informations, en particulier pour des tâches où les relations entre les variables sont complexes ou non linéaires. Il fait appel à la fois aux connaissances en neurosciences, aux mathématiques et aux progrès technologiques, et est aujourd'hui désigné comme une véritable révolution dans le domaine de l'intelligence artificielle. Les architectures des modèles profondes sont relativement récentes où de nombreuses étapes de traitements non linéaires de l'information sont exploitées, dans lesquelles les informations sont traitées en couches hiérarchiques, chacune recevant et interprétant les informations de la couche précédente pour l'apprentissage des représentations de données. Il a déjà permis d'immenses progrès et de multiples applications dans les domaines de la reconnaissance faciale et vocale, de l'étiquetage d'images, du traitement automatique du langage ou encore de la vision par ordinateur [57].

Nous présenterons un bref aperçu des structures que l'on retrouve dans des réseaux profonds.

a. RNN

Un RNN (Recurrent Neural Network) est un type de réseau de neurones utilisé en apprentissage profond, particulièrement adapté pour traiter des données séquentielles ou des données de série temporelles afin de créer un modèle de machine learning capable de tirer des prédictions séquentielles ou tirer des conclusions sur la base d'entrées séquentielles. Il est

souvent utilisé dans des domaines tels que la reconnaissance vocale, la traduction automatique et le traitement du langage naturel, où la relation entre les éléments d'une séquence est importante.

Ils ont pour avantage :

- Les RNN disposent d'une mémoire à court terme (état mémoire) qui leur permet de prendre en compte les informations récentes lors de la prédiction des séquences futures.
- Ils peuvent gérer des séquences de longueurs variables, ce qui les rend adaptés à des tâches où la longueur de la séquence varie, telle que la traduction automatique.
- Comme les RNN sont largement utilisés, il existe de nombreux modèles pré-entraînés disponibles, qui peuvent être utilisés comme point de départ pour des tâches spécifiques (voir figure 17).

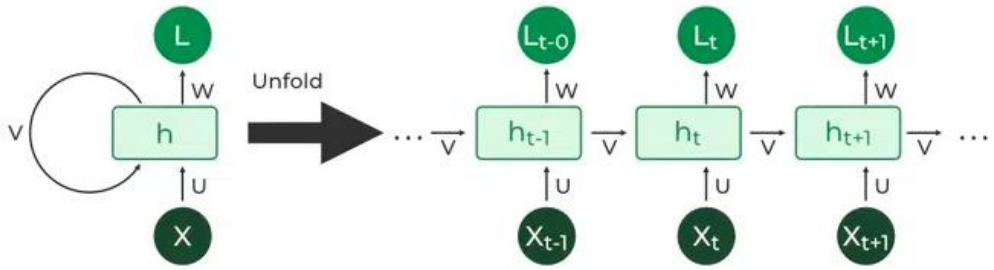


Figure 17 : Architecture RNN [58]

b. CNN-LSTM (Convolutional Neural Networks-Long Short-Term Memory)

CNN-LSTM est un modèle hybride combinant deux architectures profondes (voir figure 19) : les réseaux de neurones convolutifs et les réseaux de neurones à mémoire à long court terme (LSTM). Il est souvent utilisé pour traiter des données ayant à la fois des structures spatiales et temporelles, comme les videos, les séries temporelles ou les signaux séquentiels.

- Réseaux LSTM (Long Short-Term Memory)

Les LSTM sont un type de réseau de neurones récurrents (RNN) particulièrement adaptés au traitement des séquences, car ils sont capables de maintenir une mémoire des états passés sur de longues périodes. Ils sont donc utilisés pour modéliser les dépendances temporelles dans

les séries de données ou les séquences. La cellule mémoire d'un LSTM est composée de plusieurs portes : une porte d'entrée, une porte de sortie et une porte d'oubli**.** Les LSTM introduisent un second état, *c_t* (état actuel) en plus du h_t de l'état caché du RNN. Dans l'architecture LSTM, *h_t* représente la mémoire à court terme du neurone, tandis que c_t correspond à la mémoire à long terme (voir figure 18).

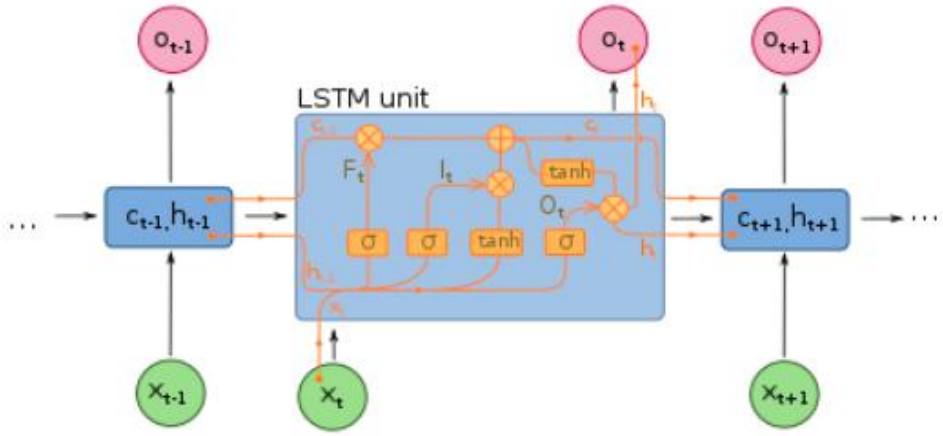


Figure 18: Architecture LSTM [59]

La porte d'entrée *l_t* détermine quelles informations doivent être intégrées dans la cellule mémoire. Elle prend en compte l'entrée actuelle *x_t* ainsi que l'état précédent de la cellule mémoire *h_(t-1)* et génère un vecteur d'activation qui indique les informations à ajouter à la cellule *c_(t-1).* Cet ajout se réalise par une opération mathématique entre le vecteur d'activation et l'état précédent *c_(t-1).*

La porte d'oubli *F_t* permet au LSTM de supprimer les informations obsolètes de la cellule mémoire.Il utilise à la fois l'entrée actuelle et l'état précédent pour créer un vecteur d'activation qui définit quelles informations doivent être oubliées. Cela se traduit également par une opération mathématique appliquée entre ce vecteur d'activation et l'état précédent *c_(t-1).* À partir de ces deux opérations, on obtient l'état actuel *c_t.*

Enfin, la porte de sortie *O_t* détermine la sortie du LSTM à un instant donné. Elle utilise l'entrée actuelle *x_t* et l'état actuel de la cellule mémoire *c_t* pour produire un vecteur d'activation représentant la sortie du LSTM, d'où résulte *h_*t.

La synergie de ces trois portes permet au réseau LSTM de gérer efficacement les dépendances à long terme. Lors de la rétropropagation du gradient, les LSTM parviennent à maintenir un

flux d'informations stable au fil du temps, ce qui conduit à un apprentissage plus stable et précis [60].

- Combinaison des deux architecture CNN-LSTM

Le CNN gère l'extraction de motifs à courts termes dans les flux des données, c'est-à-dire l'extraction des caractéristiques à partir des vecteurs, de caractérisés pour chaque séquence ensuite passez la sortie du CNN dans un ou plusieurs LSTM pour capturer les dépendances temporelles pour enfin classer les séquences.

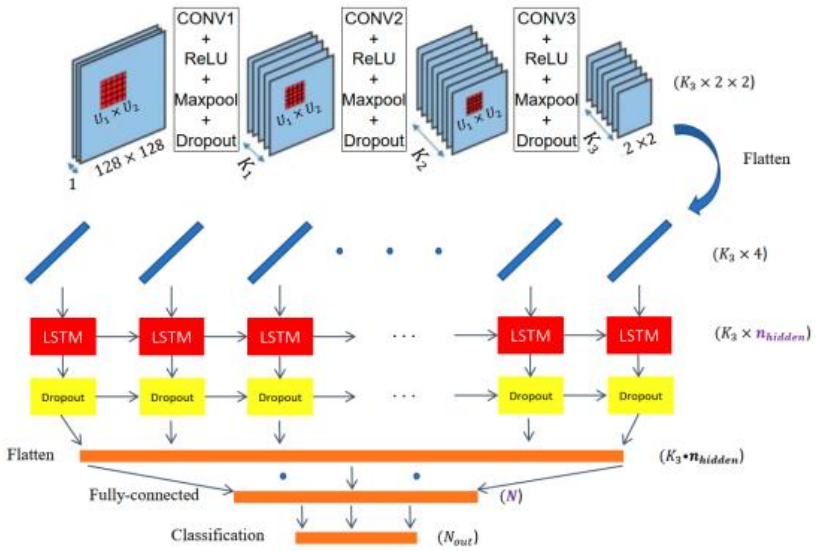


Figure 19 : Architecture CNN-LSTM [50]

III. Technique d'apprentissage

Différentes techniques d'apprentissages sont utilisées, chacune étant adaptée à des tâches spécifiques, comme l'apprentissage supervisé, l'apprentissage non supervisé, et l'apprentissage par renforcement.

- Apprentissage Supervisé :

L'apprentissage supervisé (voir figure 20) consiste à former un modèle à partir d'un ensemble de données constituées de paires (x, y), où x représente les caractéristiques d'un exemple, et y est l'étiquette associée ou le résultat attendu. Le but est de construire une fonction f capable de prédire y à partir de nouvelles observations (x). Pour ce faire, le modèle ajuste ses paramètres internes lors de l'entraînement, en apprenant les relations entre les caractéristiques (x) et les

étiquettes (*y*). Une fois formé, le modèle est utilisé pour prédire des étiquettes pour des données nouvelles, non observées auparavant [62].

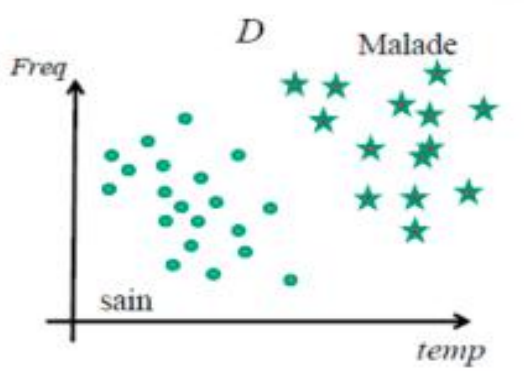


Figure 20 : Apprentissage supervisé

- Apprentissage non supervisé :

Avec l'apprentissage non supervisé (voir figure 21), les algorithmes apprennent à partir de données non étiquetées. Contrairement à l'apprentissage supervisé, il n'y a pas de labels ou de résultats prédéfinis pour guider le modèle. L'objectif principal de l'apprentissage non supervisé est de découvrir des structures cachées, des relations ou des motifs dans les données [63]. C'est-à-dire, les algorithmes analysent les données d'entrée pour identifier des similarités, des groupes (clusters), ou des régularités sans intervention humaine. Il s'agit principalement de comprendre la distribution des données ou de détecter des motifs sous-jacents.

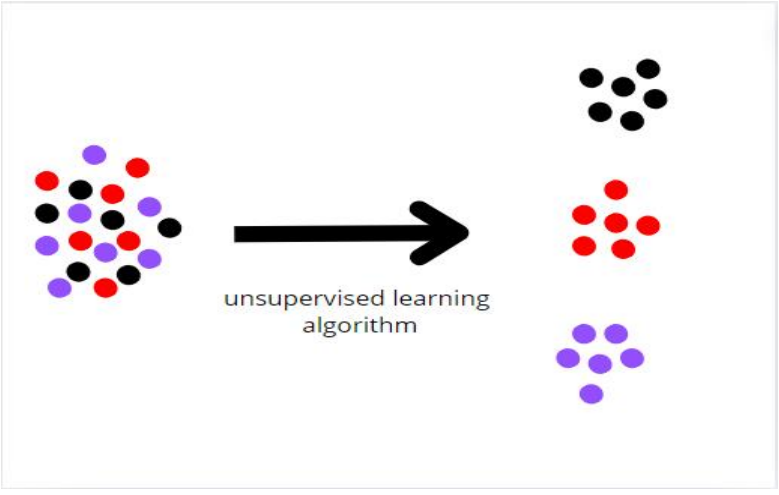


Figure 21 : Apprentissage non supervisé

Dans le contexte de la sécurité de l'Internet des objets (IoT), l'apprentissage non supervisé est particulièrement utile pour détecter des comportements anormaux ou inconnus sans avoir besoin d'un ensemble de données étiquetées [64].

- Apprentissage par renforcement :

L'apprentissage par renforcement (ou Reinforcement Learning, RL) est une branche du machine learning dans laquelle un agent apprend à prendre des décisions optimales en interagissant avec un environnement. Contrairement à l'apprentissage supervisé ou non supervisé, l'apprentissage par renforcement (voir figure 22) se base sur la notion de récompense pour évaluer les actions entreprises par l'agent. Au travers de son expérience, l'agent cherche à trouver la stratégie décisionnelle optimale qui puisse lui permettre de maximiser les récompenses accumulées au cours du temps.

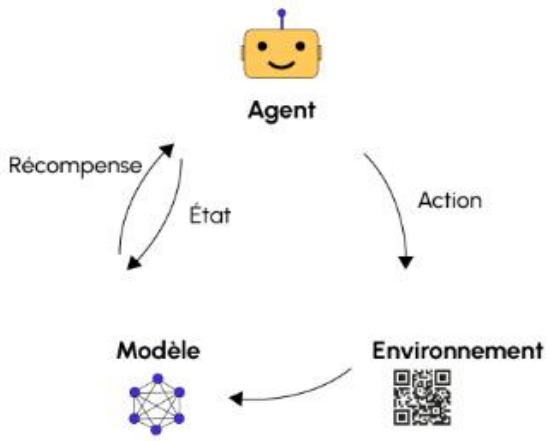


Figure 22 : Apprentissage par renforcement [65]

L'apprentissage par renforcement peut jouer un rôle croissant dans la sécurité des systèmes IoT en raison de sa capacité à s'adapter et à prendre des décisions autonomes dans des environnements dynamiques et complexes [66]. Les dispositifs IoT étant souvent déployés dans des environnements hétérogènes et vulnérables aux cyberattaques. RL peut offrir des solutions pour renforcer leur sécurité comme par exemple :

- Les agents de RL (pour les IDS) peuvent surveiller les flux de données des appareils IoT en temps réel et apprendre à identifier des menaces telles que les attaques DDoS, les intrusions réseau ou les violations de protocoles de sécurité.
- Il permet aussi de développer des stratégies de contrôle d'accès dynamique, qui ajustent les permissions et les niveaux d'accès des utilisateurs ou des appareils en fonction des conditions en temps réel [67], [68].
- Agents de RL pour la réponse aux incidents, peuvent être formés pour prendre des décisions en cas d'incidents de sécurité, telles que la fermeture des connexions, le confinement d'appareils compromis, ou l'ajustement des paramètres de sécurité sans intervention humaine.
- Les systèmes IoT utilisent des protocoles de communication qui doivent être à la fois efficaces et sécurisés. L'apprentissage par renforcement peut aider à améliorer ces protocoles en optimisant des paramètres tels que le chiffrement, la gestion des clés ou la configuration des pares-feux [53].

III.1. Technique de prédiction en apprentissage supervisé

III.1.1. La Classification

La classification est une tâche de l'apprentissage supervisé dont l'objectif est de prédire une étiquette ou une catégorie à partir des données d'entrée. Plus précisément, le modèle doit attribuer une observation x à l'une des classes prédéfinies y. Formellement, cela implique que la fonction $f(x)$, que le modèle apprend, doit projeter les caractéristiques d'entrée x sur une classe y appartenant à un ensemble fini de catégories $\{C1, C2, ..., Ck\}$. Des algorithmes tels que les forêts aléatoires, les machines à vecteurs de support (SVM) et les réseaux de neurones sont souvent utilisés pour résoudre des problèmes de classification.

La classification peut être utilisée dans un réseau IoT pour détecter les intrusions, identifier les types d'attaques, repérer les dispositifs compromis et renforcer l'authentification. Elle permet aussi de filtrer les spams et malwares en classant les comportements ou les communications comme légitimes ou malveillants.

III.1.2. La Régression

La régression, en revanche, vise à prédire une valeur numérique continue plutôt qu'une catégorie discrète. Dans ce cas, le modèle apprend à établir une relation entre les caractéristiques d'entrée x et une sortie y, où y est un nombre réel. La fonction $f(x)$ cherche

donc à estimer ou approximativement la valeur de *y*. Les algorithmes couramment utilisés pour les problèmes de régression incluant la régression linéaire, les réseaux de neurones (pour les régressions non linéaires) et les forêts aléatoires.

La régression peut être utile dans l'IoT en permettant la modélisation des comportements des dispositifs et de détecter des anomalies. En prédisant les valeurs attendues pour diverses variables, elle aide à identifier les comportements suspects, à prévoir les risques potentiels, et à optimiser les mécanismes de sécurité en ajustant les paramètres en fonction des données historiques.

III.2. Les algorithmes d'apprentissage supervisé les plus couramment utilisés

En plus des k-NN, des arbres de décision, des réseaux de neurones artificiels, on peut également citer d'autres algorithmes supervisés tels que :

a. Random foret (forets Aléatoires)

Le Random Forest (voir figure 23) est un algorithme qui combine plusieurs arbres de décision pour améliorer la précision et réduire le surapprentissage. Il fonctionne en créant plusieurs arbres à partir de sous-échantillons de données, puis en prenant la décision finale par vote ou moyenne des prédictions [56], [69]. Random Forests est utile pour détecter les anomalies ou classer des menaces dans des environnements IoT complexes. Il nécessite trois hyper paramètre définis avant l'entrainement :

- le nombre de nœuds maximal (taille des arbres) ;
- le nombre d'arbres à utiliser ;
- le nombre de caractéristiques échantillonnés.

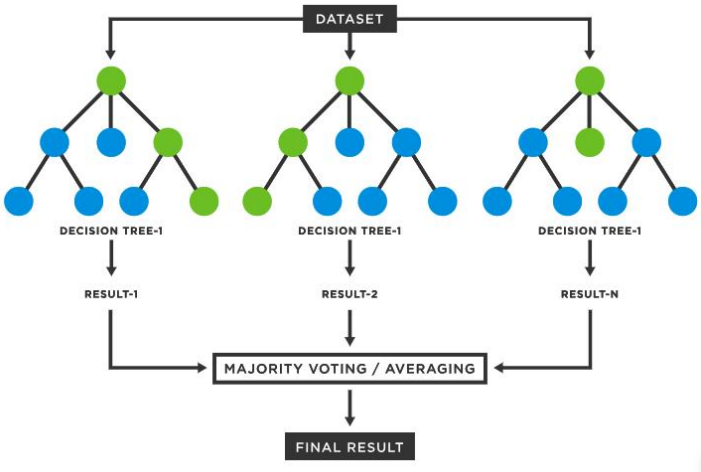


Figure 23: Illustration du modèle Random Forest [70]

Random Forest combine le concept de bagging (pour bootstrap aggregating) avec des arbres de décision. Cela lui permet de générer un ensemble de modèles prédictifs diversifié et solide. Ce concept est une technique de rééchantillonnage avec remplacement. Pour entraîner chaque arbre décisionnel dans la forêt, un échantillon aléatoire avec remplacement est sélectionné à partir du jeu de données d'origine. Cette méthode permet de générer plusieurs ensembles d'entraînement distincts, de sorte que chaque arbre est entraîné sur un ensemble de données légèrement différent.

Chaque arbre décisionnel est construit en itérant de manière récursive à travers l'ensemble du jeu de données d'entraînement. À chaque nœud de l'arbre, on choisit une séparation optimale en se basant sur un critère. Une fois que tous les arbres ont été construits, l'étape suivante consiste à combiner les prédictions de chaque arbre afin d'obtenir une prédiction globale.

b. SVM (Machine à vecteur de support)

Utilisées pour la classification binaire et multiclass, leur fonctionnement (voir figure 24) repose sur la recherche d'un hyperplan optimal qui sépare les données de différentes classes avec la plus grande marge possible, c'est-à-dire la distance entre l'hyperplan et les points les plus proches de chaque classe, appelés vecteurs de support.

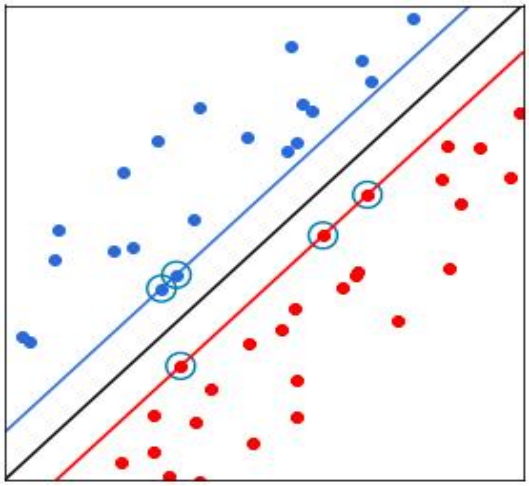


Figure 24: Illustration de l'algorithme du Machine a vecteur de support [71]

Dans cette image (un espace à deux dimensions), la frontière est la droite noire, les vecteurs de support sont les points entourés (les plus proches de la frontière) et la marge est la distance entre la frontière et les droites bleue et rouge. Il existe des jeux d'hyper paramètres par défaut pour l'algorithme SVM pour la classification, la régression ou la détection d'anomalie. On peut citer le one-class qui est utilisé principalement pour la classification.

Les SVM sont efficaces pour détecter des attaques dans les systèmes IoT en séparant les comportements malveillants des comportements normaux [26], [54].

c. XGBoost

XGBoost (Extreme Gradient Boosting), est un algorithme d'apprentissage automatique très populaire utilisé pour la régression et la classification. Il fait partie des méthodes d'ensemble (voir figure 25), qui consistent à combiner les prédictions de plusieurs modèles simples (arbres de décision) pour obtenir des prédictions plus précises et robustes [72].

- **Ensemble de modèles faibles** : XGBoost utilise des arbres de décision peu profonds (souvent 1 ou 2 niveaux) comme modèles de base, appelés "modèles faibles" car ils ont une faible capacité à faire des prédictions précises.
- **Construction séquentielle des arbres** : Les arbres sont construits séquentiellement, l'un après l'autre. À chaque itération, XGBoost tente de corriger les erreurs des arbres précédents. Chaque nouvel arbre est entraîné sur les erreurs commises par le modèle précédent pour améliorer les prédictions. Les arbres suivants se concentrent principalement sur les exemples mal classés par les arbres précédents.
- **Ces caractéristiques** : XGBoost minimise une fonction de coût qui mesure l'erreur entre les prédictions du modèle et les véritables étiquettes (valeurs cibles). Cette fonction de coût comprend : Le terme de perte (l'erreur de prédiction) et le terme de régularisation (limitant la profondeur des arbres pour éviter le sur apprentissage (overfitting). Cela aide à améliorer la généralisation du modèle sur de nouvelles données.
- **Performance du modèle** : XGBoost offre des fonctionnalités avancées pour améliorer les performances du modèle. Il sélectionne automatiquement les caractéristiques importantes, réduisant ainsi la dimensionnalité et accélérant l'apprentissage. Pour prévenir le surapprentissage, XGBoost utilise des techniques de régularisation comme la limitation de la profondeur des arbres, la réduction du taux d'apprentissage, et la diminution du nombre d'arbres.
- **Vitesse d'apprentissage** : Le taux d'apprentissage contrôle la vitesse à laquelle les poids des arbres sont ajustés, et un taux plus bas améliore souvent la généralisation mais nécessite plus d'itérations.

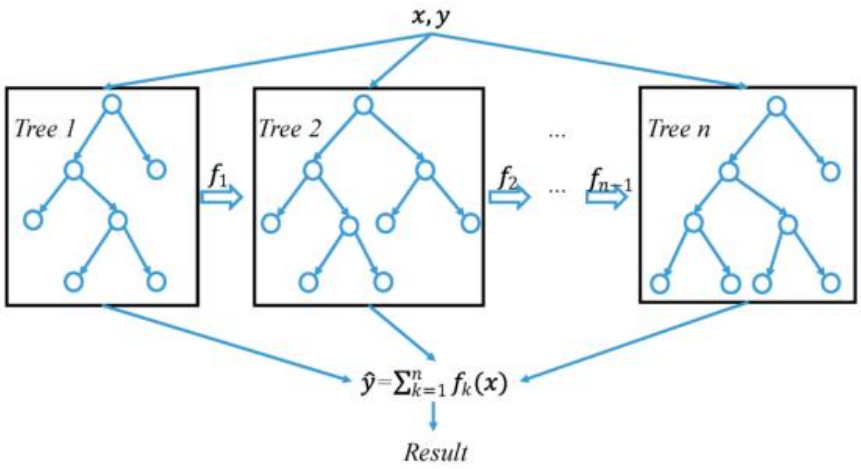


Figure 25: Algorithme XGBoost [73]

Une fois tous les arbres construits, leurs prédictions sont combinées pour produire la prédiction finale. XGBoost est très apprécié pour ses performances élevées, sa flexibilité, sa capacité à gérer des jeux de données volumineux, et ses fonctionnalités avancées comme la classification multi classe, la régression logistique, et la détection d'anomalies [74].

III.3. Les algorithmes d'optimisation

Les algorithmes d'optimisation sont essentiels pour ajuster les paramètres internes des modèles d'apprentissage automatique afin de minimiser l'erreur entre les prédictions du modèle et les données réelles.

III.3.1. Descente de gradient

La descente de gradient est une méthode d'optimisation classique largement utilisée pour minimiser une fonction de coût dans les modèles d'apprentissage automatique. Son objectif est de trouver les valeurs des paramètres qui minimisent l'erreur en ajustant les poids du modèle dans la direction opposée au gradient de la fonction de coût. Cela se fait en itérant progressivement, étape par étape, jusqu'à atteindre un minimum local ou global. La taille des étapes est contrôlée par le taux d'apprentissage [75].

III.3.2. Descente de gradient stochastique

La descente de gradient stochastique est une variante de la descente de gradient qui met à jour les paramètres du modèle à chaque itération en utilisant une seule observation aléatoire ou un petit sous-ensemble des données, plutôt que l'ensemble complet des données. Cette approche rend l'optimisation plus rapide, en particulier pour les grands ensembles de données, mais

introduit plus de bruit dans les mises à jour, ce qui peut aider à sortir des minima locaux mais nécessite plus d'itérations pour guider le modèle à des paramètres plus pertinents.

III.3.3. Adam (ADAptive du Moment)

Adam est un algorithme d'optimisation qui combine les avantages de la descente de gradient stochastique avec l'adaptativité des méthodes basées sur le moment. Il ajuste les taux d'apprentissage pour chaque paramètre individuellement, en fonction des premières et secondes moyennes des gradients (moyenne et variance), ce qui le rend particulièrement efficace pour traiter des problèmes complexes et des données bruyantes. Adam est largement utilisé dans le domaine de l'apprentissage profond en raison de sa capacité à converger rapidement et de son efficacité sur des modèles à grande échelle.

IV. Solutions d'intelligence artificielle pour la sécurité de l'IoT

Il existe de nombreuses solutions faisant appel à l'intelligence artificielle, notamment l'apprentissage automatique, pour renforcer la sécurité de l'Internet des Objets, et cela s'étend à divers autres secteurs. Les dispositifs IoT peuvent exploiter l'intelligence artificielle, notamment l'apprentissage automatique, pour prendre des décisions intelligentes après avoir analysé les données collectées. Grâce à des modèles d'apprentissage, il devient possible d'identifier des tendances significatives dans les événements de sécurité, souvent à travers des règles, des algorithmes ou des fonctions de transfert. Ces modèles de sécurité basés sur les données peuvent être conçus à l'aide de techniques de classification, de régression, de clustering ou encore d'approches basées sur des règles, favorisant l'optimisation et la prévention des risques.

Dans [76], l'auteur présente un système intelligent de détection des DDoS et de localisation physique des appareils dans les SE (environnement intelligent), en appliquant des techniques d'apprentissage automatique (ML) et de trilatération. Les expériences réalisées, en utilisant le trafic réseau réel et la simulation, montre que le système proposé est capable de détecter les attaques et de trouver des appareils malveillants.

D'autres chercheurs ont proposé et intégré la similarité de gradient avec le cryptage homomorphe pour concevoir un cadre d'apprentissage de régression logistique multipartite dans lequel aucune information privée n'est divulguée pendant la formation du modèle tout en filtrant les données de mauvaise qualité des contributeurs de données IoT. Ils ont démontré l'efficacité du cadre proposé en expérimentant à l'aide d'ensembles de données du monde réel [77].

Dans [78] les auteurs proposent une approche de sécurité centrée sur l'utilisateur basée sur un arbre de décision appelée DecisionTSec qui fournit un canal sécurisé pour la communication dans les réseaux IoT, en combinant les centres de données périphériques dans les périphéries du réseau. Cette solution adopte un mécanisme d'arbre de décision et ensuite l'intégre au système de cryptographie pour placer la barre plus haute pour les attaquants.

Les auteurs de [79] ont utilisé deux classificateurs d'apprentissage automatique dans leur travail : KNN et DT (Decision Tree) pour protéger les appareils IoT aux pirates. Ils ont calculé le taux d'erreur, l'exactitude, la précision, le rappel et le F1-score pour chaque méthode. Ils ont combiné les deux classificateurs et obtenu des résultats exceptionnels (100 %).

Au final, nous présentons dans le tableau 10 une synthèse des travaux de recherche sur la sécurisation des systèmes IoT par l'intelligence artificielle. Quatre grandes catégories de travaux cherchant à améliorer la sécurité ont été identifiés. Parmi ces catégories, deux s'adressent en priorité aux questions de confidentialité et d'intégrité des données, tandis qu'un autre se focalise sur la disponibilité du système. La troisième catégorie comporte des travaux cherchant à détecter les anomalies dans les systèmes et peut donc à la fois mettre en lumière les attaques. Ces travaux relatifs à la détection d'intrus et d'anomalies nous intéressent particulièrement dans la suite de ce mémoire. Les deux premières et la dernière catégorie (tableau 3) comportent des travaux concernant le partage et la sécurité des données, le contrôle d'accès, l'authentification, la sécurité réseau ainsi que les modèles et méthodes de sécurité.

Catégorie	**Sous- Catégorie**	**Références**
Contrôle d'accès		[2], [80], [81], [82]
Authentification		[24], [82]
Surveillance de sécurité		[3], [76]
Partage sécurisé des données	Contrôle de flux des données	[77]
	Transport des données	[83]

	Confidentialité des données	[78], [84], [85]
Détection d'intrusion	IDS basé sur l'anomalie	[86], [87], [88], [89]
	IDS basé sur le routage	[90], [91]

Tableau 3: Tableau récapitulatif de différentes approches de sécurisation des systèmes IoT par l'IA

La détection d'anomalies ou d'intrusion est une approche qui intéresse les chercheurs et les professionnels de la sécurité, notamment dans le domaine de l'IoT. Et les travaux dans ce sens s'orientent dans les différents domaines et applications de l'IoT.

IV.1. L'agriculture intelligente

L'agriculture intelligente repose sur des dispositifs IoT pour surveiller les conditions environnementales, les niveaux d'irrigation et la santé des cultures. L'apprentissage automatique renforce la sécurité de ces dispositifs en détectant les anomalies dans les données des capteurs, en identifiant des comportements inhabituels ou des pannes pouvant indiquer une cyberattaque. Par exemple, les algorithmes de détection des anomalies peuvent signaler toute tentative de manipulation des capteurs, affectant ainsi la gestion de l'irrigation ou la santé des cultures, et garantissant la fiabilité des systèmes agricoles [92], [93].

IV.2. La santé

Dans le secteur de la santé, les dispositifs IoT (comme les pacemakers connectés ou les moniteurs de santé) sont très sensibles aux attaques, car ils gèrent des données critiques. L'apprentissage automatique est utilisé pour détecter les anomalies dans le comportement des dispositifs et les communications réseau. En apprenant à partir de données passées, les modèles de ML peuvent repérer des tentatives d'attaques, comme des accès non autorisés ou des manipulations des dispositifs médicaux, assurant ainsi la sécurité des patients et de leurs données [84].

IV.3. La domotique

Dans les maisons intelligentes, les dispositifs IoT, tels que les caméras de sécurité, les thermostats et les serrures connectées, sont particulièrement exposés aux vulnérabilités. L'apprentissage automatique améliore la sécurité des réseaux domestiques en surveillant les comportements normaux des appareils pour détecter des accès inhabituels ou des tentatives

d’intrusion. Par exemple, les systèmes d’apprentissage supervisé peuvent identifier des anomalies dans les flux vidéo des caméras de surveillance, garantissant ainsi que seules les personnes autorisées peuvent accéder à la maison [94].

IV.4. L’industrie

Les systèmes IoT industriels (Industrie 4.0) connectent des machines pour améliorer la production, mais aussi les rendre vulnérables aux cyberattaques. Les algorithmes de machine learning analysent les flux de données des équipements industriels pour identifier les anomalies, prédire les défaillances et renforcer la sécurité en temps réel. Ces systèmes permettent aussi de détecter les tentatives d’accès non autorisé aux infrastructures critiques et de prévenir les interruptions de production [3], [83].

Conclusion

Dans ce chapitre, nous avons examiné en profondeur l'apport des méthodes d'intelligence artificielle, et plus particulièrement du machine learning, pour renforcer la sécurité des systèmes IoT. Nous avons mis en évidence comment ces techniques permettent non seulement de surveiller et de détecter les anomalies, mais aussi de classifier les menaces, de prévenir les attaques et de renforcer les mécanismes cryptographiques. Nous avons répertorié et passé en revue plusieurs algorithmes qui contribuent à renforcer la sécurité des réseaux IoT, ainsi que quelques solutions de sécurité proposées pour mettre en œuvre différents services de sécurité dans des applications variées de l’IoT. Enfin nous avons porté une attention particulière sur la détection d’intrus et d’anomalies sur laquelle nous souhaitons tester l’efficacité des algorithmes d’intelligence artificielle dans le chapitre suivant. C’est ainsi que nous allons l’examiner dans le contexte du protocole MQTT sur une application domotique.

Chapitre 4 : Contribution à la performance de la détection d'intrus dans MQTT

Le protocole MQTT (Message Queuing Telemetry Transport) est devenu l'un des protocoles de communication les plus utilisés dans l'Internet des objets (IoT), en raison de sa légèreté et de son efficacité dans des environnements à faible bande passante. Cependant, comme détaillé dans le Chapitre 2, bien que MQTT soit adapté aux contraintes des systèmes IoT, il présente des vulnérabilités importantes en matière de sécurité, notamment une absence d'authentification et de chiffrement intégrés. Ces faiblesses exposent les systèmes utilisant MQTT à diverses menaces, telles que les attaques "Man-in-the-Middle", la falsification de données ou encore l'accès non autorisé aux informations sensibles.

Pour répondre à ces défis, plusieurs solutions de sécurité ont été proposées au fil des années, telles que l'intégration de protocoles de chiffrement comme TLS ou l'authentification via certificats. Toutefois, ces approches, bien que efficaces, ne sont pas toujours adaptées aux environnements IoT fortement contraints en termes de ressources, où la puissance de calcul, la mémoire et la bande passante sont souvent limitées. Dans ce chapitre, nous allons examiner les solutions actuelles pour sécuriser les communications MQTT et analyser leurs limites, en particulier dans le contexte des dispositifs IoT à ressources limitées. Ensuite, nous présenterons une approche innovante basée sur l'intelligence artificielle (IA) pour renforcer la sécurité de ce protocole. En effet, l'IA offre de nouvelles perspectives en matière de détection d'anomalies et d'optimisation des mécanismes de sécurité, permettant ainsi une approche plus dynamique et adaptée aux contraintes des systèmes IoT. Ainsi nous avons accès notre contribution sur la performance des algorithmes d'I.A face à la détection d'intrus dans le protocole MQTT utilisé dans le contexte d'une application domotique pour surveiller les comportements normaux des appareils tels que les caméras de sécurité, les thermostats et les serrures connectées pour détecter des accès inhabituels ou des tentatives d'intrusion ou encore identifier des anomalies dans les flux vidéo des caméras de surveillance, garantissant ainsi que seules les personnes autorisées peuvent accéder à la maison [79].

I. Solutions Cryptographiques légères pour le protocole MQTT

MQTT présente bien des vulnérabilités concernant l'authentification lors de la connexion ou la confidentialité des données envoyées au niveau du courtier. Même si le protocole MQTT dispose bien detoutes les caractéristiques de sécurité mentionnées à la section III.1.1.b.

Dans [95], les auteurs présentent différentes options (algorithmes) de sécurité cryptographiques (AES-CCM, AES, AES-CBC, AES-OCB,) qui ont été développées et évaluées sur un banc d'essai réel pour les nœuds MQTT, comprenant des capteurs sans fil

fonctionnant avec le système d'exploitation Contiki. Les options évaluées comprenaient trois implémentations de la couche application, fournissant une sécurité de bout en bout, et un mécanisme de la couche liaison, fournissant une protection saut par saut.

Dans l'article [96], les chercheurs utilisent les techniques et les algorithmes standard de la cryptographie notamment la signature numérique, les fonctions de hachage et l'algorithme RSA, afin de sécuriser la communication et l'échange des données dans un réseau de capteurs.

Une solution de schéma de signature numérique post-quantique CRYSTALS-Dilithium est utilisée dans [97] pour fournir l'authentification pour MQTT et déterminer l'utilisation du processeur, de la mémoire et du disque lors de cette opération. Ils étudient en outre une autre possibilité de fournir une authentification lors de l'utilisation de MQTT, à savoir une astuce de mécanisme d'encapsulation de clé (KEM) proposée en 2020 pour la sécurité au niveau du transport (TLS).

Des travaux visant à obtenir à la fois l'autorisation et la confidentialité basés sur des variantes ABE ou CP-ABE adaptées aux dispositifs contraints sont présentées dans ces articles : [98], [99]. Une solution appelée MQTT-Auth est basée sur l'algorithme de sécurité AugPAKE pour garantir la confidentialité, et sur deux jetons qui permettent d'authentifier l'utilisation d'un sujet et de garantir l'autorisation d'accès à un sujet respectivement [85].

MQTTSec est proposé dans [101] pour une amélioration du protocole MQTT traditionnel. Il permet aux appareils communicants de sélectionner la technique cryptographique appropriée en fonction des ressources disponibles.

II. Solutions d'IA pour la sécurité du protocole MQTT

Il existe bien dans la littérature des solutions de l'IA apportant une contribution dans la sécurité de l'internet des objets. La majorité propose des solutions de détection d'intrusion ou pour contrecarrer des attaques comme le man-in-the -middle ou l'attaque de DOS. Dans [102], les auteurs proposent un modèle non-supervisé appelé GAN-AE pour la détection d'intrusion. Les performances du modèle proposé sont comparées à celles d'autres modèles non supervisés populaires, à savoir autoencoder, Classe unique SVM(OCSVM) et Isolation Forest (IF). Le modèle GAN-AE a obtenu des résultats supérieurs aux autres modèles en termes de précision et de F1-Score de 0,97 sur leur ensemble de données MQTT personnalisé et public. Un réseau neuronal profond (DNN) pour la détection d'intrusions dans le protocole MQTT est proposé dans [103] dont les performances sont comparées à d'autres algorithmes d'apprentissage

automatique (ML) traditionnels, tels que Naive Bayes (NB), Random Forest (RF), k-Nearest Neighbour (kNN), Decision Tree (DT), Long Short-Term Memory (LSTM) et Gated Recurrent Units (GRU). Les performances sont prouvées à l'aide de deux ensembles de données différents accessibles au public. L'article [104] compare plusieurs algorithmes, notamment KNN (k-plus proches voisins), LDA (analyse discriminante linéaire), CNN (réseau neuronal convolutif), et CNN-LSTM (réseau neuronal convolutif avec mémoire à court terme) pour détecter les intrusions dans le protocole MQTT de l'IoT. L'ensemble de données, extrait de Kaggle, a été testé contre cinq types d'attaques : force brute, inondation, paquets malformés, SlowITe et des paquets normaux. Les résultats montrent que les algorithmes d'apprentissage profond, en particulier CNN et CNN-LSTM, surpassent les modèles d'apprentissage automatique traditionnels en termes de précision pour l'identification des intrusions. Deux travaux [105], [106] se focalisent spécialement sur les deux attaques les plus répandues dans la couche application qui sont l'attaque Man-in-the-middle et le DoS en proposant des solutions pour les contrecarrer. Ils mettent en œuvre différents modèles d'apprentissage automatique sur un ensemble de données open source et évaluent différents paramètres.

Dans le tableau 4, nous présentons quelques solutions de détection d'intrusion pour le protocole MQTT.

Les attaques	Solutions proposées
DoS, Bruteforce, SlowITe, Flooting (inundation), données malformées	[107], Une solution appelé Edge IDS basé sur une architecture d'apprentissage contradictoire utilisant le réseau contradictoire génératif (GAN)
Bruteforce, SlowITe, DoS, les paquets normaux, les paquets mal formées	[104], CNN-LSTM est utiliser pour identifier les intrusions IoT du protocole MQTT
Man in the Middle	[106], Compare les modèles ANN, NB, XGB, DT, et KNN et à la fin XGBoost est choisie comme modèle obtenant les meilleurs résultats
MitM, Dos, intrusion dans le réseau	[108], Compare les modèles DNN, KNN, NB, LSTM, GRU, et au final le DNN est obtenue la meilleur résultat

Tableau 4: Des Solutions de Détection d'intrusions pour le protocole MQTT

III. Limites des solutions actuelles

MQTT a été initialement développé sans fonctionnalités de sécurité, et les solutions de sécurité actuelles, telles que TLS, l'authentification par certificat X.509 et OAuth 2.0, peuvent entraîner des coûts de communication qui ne sont pas toujours adaptés aux appareils à ressources limitées. Ce défi pourrait être amplifié par l'utilisation d'algorithmes traditionnels, qui exigent souvent une puissance de traitement plus élevée.

Pour apporter une solution efficace à la sécurité du protocole MQTT, il faut prendre en compte certains paramètres comme la capacité de calcul des appareils IoT et il existe bien des solutions dans la littérature pour sécuriser le protocole MQTT :

- Bien que certaines études aient examiné des algorithmes cryptographiques plus légers dans le cadre de MQTT [96], [97], ces recherches sont restées relativement limitées.
- Des solutions combinant la cryptographie pour l'authentification et des méthodes d'IA sont proposées pour une sécurisation robuste du protocoles MQTT [73], mais ces solutions peuvent exiger des calculs intensifs si certaines approches adaptées aux limites des appareils connectés ne sont pas prises en compte.
- Des mécanismes d'IA ont été proposés pour la détection d'intrusions dans l'IoT, notamment dans le protocole MQTT. Il s'avère que ces derniers sont plus efficaces par rapport au solution cryptographique pour la détection d'intrusions car la cryptographie se concentre sur la protection des données et l'authentification mais ne peut pas détecter de manière active les comportements anormaux [109].
- Les performances des méthodes d'IA dépendent des modèles qui sont utilisés

Notre objectif principal est de mesurer la performance des modèles d'IA face à la détection d'intrusions dans le protocole MQTT.

IV. Contribution : Accroître la performance de la détection d'intrus dans MQTT

Notre idée consiste à analyser les performances des approches existantes afin d'en proposer une combinaison d'approches basée sur l'Intelligence Artificielle (IA) pour accroître la détection d'anomalies dans la communication MQTT, afin de prévenir les attaques potentielles. Contrairement aux solutions traditionnelles basées sur des règles statiques, les algorithmes d'IA offrent la capacité d'apprendre à partir des données et de s'adapter aux

menaces émergentes, les rendant ainsi plus efficaces pour identifier des comportements anormaux et prédire des attaques dans des environnements IoT évolutifs. Pour ce faire nous examinons d'abord les performances de quelques techniques d'IA existantes en vue d'apprécier leur degré de détection pour ensuite faire des ajustements pour obtenir de meilleures performances.

IV.1. Choix de la méthode de l'intelligence Artificielle pour renforcer la sécurité de MQTT

Avant de choisir une méthode, le tableau 5 nous présente un survol des différentes techniques d'IA appliquées à la sécurité des systèmes réseau afin d'apprécier leur degré de performance en matière de détection.

Article	**Technique de sécurité**	**Technique d'apprentissage**	**Performance**
[78]	Contrôle d'accès	Arbre de décision	Taux d'erreur moyen
[79]	Détection d'intrusion	KNN et DT (Decision Tree)	Combinant les deux Résultat 100%
[4]	Détection d'intrusion	Un modèle d'apprentissage profond hybride, incorporant des réseaux neuronaux convolutifs (CNN), des mémoires à long terme (LSTM) et des unités récurrentes à portes (GRU)	Le modèle atteint une précision de 96,68 %
[110]	Authentification	KNN, naive bayésien, Random Forest, Decision Tree	Decision Tree a obtenu la meilleur résultat

[111]	Détection et la classification des attaques	KNN, une analyse discriminante linéaire (LDA), CNN et CNN-LSTM	La précision de la méthode KNN était de 80,82 %, tandis que la précision de l'algorithme LDA était de 76,60 %. Le modèle CNN-LSTM a atteint un niveau de précision élevé (98,94 %) et est donc très efficace pour détecter les intrusions dans les paramètres IoT.
[76]	Détection intrusion	KNN, naive bayésien, Random Forest, Decision Tree, logistic regression, SVM	Decision Tree étant plus adapté pour la détection d'intrusion
[5]	Détection intrusion	Forêt aléatoire, GLM, GBM, XGBoost, réseau neuronal profond	XGBoost donne le meilleur résultat 99,74% de précision
[6]	Détection intrusion	Arbre de décision (DT)	Le résultat obtenu est de 96%
[7]	Détection d'attaques de botnet (BASHLITE et Mirai)	CNN-LSTM	Les résultats expérimentaux ont montré la supériorité du modèle CNN-LSTM avec des précisions importantes dans la détection des attaques de botnet provenant de sonnettes, avec des taux de précision de 90,88% et 88,61%. De même, l'algorithme proposé atteint un taux de précision acceptable de 88,53% pour identifier les attaques de botnet à partir de dispositifs de thermostat.

Tableau 5 : Solutions de sécurité IoT par l'apprentissage automatique

De nombreux algorithmes de machine learning sont proposés pour renforcer la sécurité des réseaux, et en analysant le tableau 5, nous constatons que certains, tels que les arbres de décision, XGBoost et les architectures neuronales CNN-LSTM [7], [5], [111], [4], se révèlent plus performants que d'autres, notamment pour des tâches comme la détection d'intrusions. Par exemple, l'arbre de décision est souvent efficace pour les décisions basées sur des règles simples et bien définies. Cependant, XGBoost, qui repose sur un ensemble d'arbres de décision amélioré par des techniques de boosting, montre des performances encore plus élevées grâce à sa capacité à corriger les erreurs des prédictions précédentes, offrant ainsi une grande précision pour la détection des anomalies et des intrusions complexes [7], [106]. Concernant le modèle CNN-LSTM, sa performance repose sur sa capacité à capturer à la fois les caractéristiques spatiales (via les couches CNN) et les dépendances temporelles (via les couches LSTM) des données réseau.

Ces performances pourraient être exploitées pour sécuriser la communication au sein du protocole MQTT. Le MQTT étant largement utilisé dans les environnements IoT, où la sécurité est un enjeu crucial, l'utilisation de modèles comme XGBoost ou le CNN-LSTM pourrait permettre de détecter les anomalies dans les flux de messages, les tentatives d'intrusions ou les comportements malveillants.

En somme, l'utilisation d'algorithmes avancés comme XGBoost ou CNN-LSTM pourrait renforcer de manière significative la sécurité des communications MQTT, en détectant plus efficacement les anomalies et les attaques dans ces réseaux souvent sensibles.

Étant donné que ces deux modèles sont efficaces pour la détection d'intrusions dans l'IoT, il est préférable de les comparer en utilisant les mêmes données afin de déterminer lequel des deux est le mieux adapté pour assurer la sécurité du protocole MQTT en matière de détection d'intrusions.

IV.2. Evaluation des performances des méthodes choisies

Comme le montre le Tableau 5, XGBoost et CNN-LSTM se révèlent prometteurs de par leur performance et leur degré de précision atteignant respectivement une précision de 99,74% et 98,94% dans la détection d'intrus. Notre premier plan est de tester leurs performances pour la détection d'anomalies dans la communication MQTT, afin de prévenir les attaques potentielles.

Pour ce faire, nous avons entrainé ces modèles sur des données obtenues à l'aide d'un ensemble de données standard extrait du référentiel Kaggle [5]. Pour entraîner les modèles XGBoost et CNN-LSTM, nous avons besoin d'un ensemble de données qui contient à la fois le comportement réseau des appareils IoT et des scénarios d'attaques. Les données sont présentées sous forme d'un tableau de 123117 lignes et 85 colonnes y compris des fonctionnalités qui décrivent le comportement du trafic réseau normal et contradictoire. Le Tableau 6 est un extrait du tableau de données et comporte les colonnes clés avec les cinq premières lignes relatives aux différentes attaques liées aux intrusions à détecter avec une certaine précision.

- Fonctionnalités liées au trafic : proto, service, flow_duration, fwd_pkts_tot, bwd_pkts_tot, fwd_data_pkts_tot, bwd_data_pkts_tot, etc.
- Statistiques de paquets et de flux : fwd_pkts_per_sec, bwd_pkts_per_sec, flow_pkts_per_sec et bien d'autres détaillant les en-têtes de paquets, la charge utile et les flux.
- Attributs temporels : active.min, active.max, ralenti.min, ralenti.max, etc.
- Attack_type : il s'agit de la colonne cible, contenant des étiquettes telles que MQTT_Publish, DOS_SYN_Hping, etc.

	no	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot	bwd_pkts_tot	fwd_data_pkts_tot	bwd_data_pkts_tot	...	active.std	idle.min	i
0	0	38667	1883	tcp	mqtt	32.011598	9	5	3	3	...	0.0	2.972918e+07	2.972
1	1	51143	1883	tcp	mqtt	31.883584	9	5	3	3	...	0.0	2.985528e+07	2.985
2	2	44761	1883	tcp	mqtt	32.124053	9	5	3	3	...	0.0	2.984215e+07	2.984
3	3	60893	1883	tcp	mqtt	31.961063	9	5	3	3	...	0.0	2.991377e+07	2.991
4	4	51087	1883	tcp	mqtt	31.902362	9	5	3	3	...	0.0	2.981470e+07	2.981

5 rows × 85 columns

Tableau 6 : Données de traitement

Dans la suite, nous donnons une description du processus d'acquisition et de mise à disposition des données.

- Le RT-IoT2022, un ensemble de données englobant les comportements normaux et adverses du réseau, fournissant une représentation générale des scénarios du monde réel intégrant des données provenant d'appareils IoT tels que ThingSpeak-LED, Wipro-Bulb et MQTT-Temp, ainsi que des scénarios d'attaque simulés impliquant des

attaques SSH par force brute, des attaques DDoS utilisant Hping et Slowloris et des modèles Nmap.

- RT-IoT2022 offre une perspective détaillée sur la nature complexe du trafic réseau. Les attributs bidirectionnels du trafic réseau sont méticuleusement capturés à l'aide de l'outil de surveillance réseau Zeek et du plug-in Flowmeter.
- L'infrastructure se compose de deux parties, à savoir les appareils victimes IoT et les appareils attaquants IoT, tous deux connectés via un routeur. La collection du trafic réseau est obtenu via un routeur à l'aide de Wireshark, un outil de surveillance open source du trafic réseau qui permet d'extraire les traces et de les convertir en fichier PCAP.
- L'infrastructure attaquée comprend 50 machines, et l'organisation victime compte 5 départements et comprend 420 machines et 30 serveurs. L'ensemble de données comprend les captures du trafic réseau et les journaux système de chaque machine, ainsi que 80 fonctionnalités extraites du trafic capturé.
- Il comprend 9 scénarios d'attaque différents : DOS_SYN_Hping, ARP_poisioning, NMAP_UDP_SCAN, NMAP_XMAS_TREE_SCAN, NMAP_OS_DETECTION, NMAP_TCP_scan, DDOS_Slowloris, Metasploit_Brute_Force_SSH, NMAP_FIN_SCAN et 3 modèles normaux MQTT, Thing_speak et Wipro_bulb_Dataset.

IV.3. L'environnement de développement

Dans cette section, nous présentons de façon succincte l'environnement de développement que nous avons mis en place pour simuler les modèles XGBoost et CNN-LSTM.

IV.3.1. Langage python

Python est un langage de programmation de haut niveau, polyvalent et largement utilisé. Il est open source, gratuit, et fonctionne sur plusieurs plateformes. Il prend en charge différents types de données, notamment les nombres, les chaînes de caractères, les listes, les tuples, et les dictionnaires. Contrairement aux langages compilés comme C ou C++, Python est un langage interprété, ce qui signifie que son code est exécuté ligne par ligne, ce qui peut parfois entraîner une exécution plus lente. Python supporte également plusieurs bases de données, comme MySQL et MSSQL, et est utilisé pour créer une variété d'applications, allant du

développement web à l'intelligence artificielle où y trouve des implémentations de modèle tels que XGBoost et CNN-LSTM [112].

IV.3.2. Anaconda

Anaconda est une plateforme logicielle gratuite et multiplateforme compatible avec Windows, Linux et macOS. Elle inclut des distributions de Python et R, et s'accompagne de Conda, un gestionnaire de paquets et d'environnements virtuels. Anaconda propose une vaste collection de bibliothèques et de packages déjà préinstallés, facilitant ainsi le développement en science des données et en apprentissage automatique. Parmi ces packages, on retrouve notamment NumPy, SciPy, Pandas, Scikit-learn, NLTK et Jupyter, offrant une solution clé en main pour les chercheurs et développeurs travaillant sur des projets data-driven[114].

- NumPy : NumPy est une bibliothèque Python utilisée pour le calcul numérique. Elle permet de manipuler des tableaux multidimensionnels (ou matrices) et d'exécuter des opérations mathématiques rapides sur ces données. NumPy est essentiel en apprentissage automatique car il offre des outils optimisés pour le traitement de grandes quantités de données, rendant les algorithmes plus efficaces en termes de vitesse et d'utilisation de la mémoire [118].
- Pandas : Pandas, une bibliothèque Python utilisée pour la manipulation et l'analyse de données. Elle fournit des structures de données puissantes comme les DataFrames qui permettent de gérer facilement des ensembles de données structurés. En apprentissage automatique, Pandas est crucial pour la préparation des données (nettoyage, transformation, gestion des données manquantes), facilitant ainsi l'analyse exploratoire avant l'entraînement des modèles [119].
- Seaborn : Seaborn est une bibliothèque de visualisation de données en Python, construite sur Matplotlib, qui facilite la création de graphiques statistiques attractifs et informatifs. En apprentissage automatique, Seaborn est essentiel pour l'exploration et la visualisation des données, permettant de comprendre les relations entre les variables, détecter des anomalies ou des tendances, et ainsi améliorer la préparation des données et l'interprétation des résultats des modèles [120].
- Sklearn : Scikit-learn (sklearn) est une bibliothèque Python largement utilisée pour l'apprentissage automatique. Elle fournit des outils simples et efficaces pour l'entraînement de modèles de classification, régression, clustering, et réduction de dimension. En apprentissage automatique, Sklearn est important car il propose des algorithmes de machine learning prêts à l'emploi, ainsi que des outils pour la

préparation des données, l'évaluation des modèles, et le tuning des hyperparamètres, facilitant ainsi le développement complet de pipelines de machine learning[121].

- TensorFlow : TensorFlow est une bibliothèque open-source de machine learning (apprentissage automatique) et de deep learning (apprentissage profond) développée par Google. Elle est largement utilisée pour construire, entraîner et déployer des modèles d'apprentissage automatique, notamment des réseaux de neurones profonds comme les CNN, LSTM, et bien d'autres. TensorFlow est principalement utilisé avec Python, bien qu'il existe également des interfaces pour d'autres langages comme C++, JavaScript, et Java. Elle fonctionne en construisant un graphe de calcul, où chaque nœud représente une opération mathématique (comme une multiplication ou une addition), et les bords représentent les données qui circulent entre ces opérations. TensorFlow intègre **Keras**, une API de haut niveau pour construire et entraîner des modèles. Cela simplifie grandement le développement des modèles complexes comme les réseaux de neurones profonds [122].

IV.3.3. Jupyter

Jupyter est une application web conçue pour faciliter la programmation dans plusieurs langages, incluant Python, Julia, Ruby, R, Scala etc. Issu du projet IPython, Jupyter vise à promouvoir le développement de logiciels libres et de formats ouverts pour l'informatique interactive. Il permet de créer des notebooks, qui combinent du code exécutable et du texte au format Markdown. Ces notebooks sont particulièrement utiles en science des données, car ils permettent d'explorer, de visualiser et d'analyser des données de manière interactive et reproductible, tout en intégrant texte, graphiques et code dans un seul document [116].

IV.4. Importation de l'ensemble des données

Les données sont réparties en 80% de données d'entraînement et de 20% de données de test. Les ensembles de données sont fournis au format CSV, un format qui stocke les données tabulaires sous forme de texte brut. Chaque ligne d'un fichier CSV représente un enregistrement individuel de données.

Pour charger un fichier CSV local en tant que DataFrame, nous avons utilisé la méthode read_csv de la bibliothèque Pandas (voir figure 26).

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from xgboost import XGBClassifier
from sklearn.metrics import classification_report, accuracy_score

# chargeons le fichier CSV
df = pd.read_csv('RT_IOT2022.csv')
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, LabelEncoder
from sklearn.utils.class_weight import compute_class_weight
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, LSTM, Dense, Dropout, Flatten
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.optimizers import Adam
from tensorflow.keras import Input

# 1. Chargons et Prétraiter les Données

df = pd.read_csv('RT_IOT2022.csv')
```

Figure 26: Importation des données

IV.5. Prétraitement des données

Nous allons maintenant préparer les données pour entraîner le modèle XGBoost et le CNN-LSTM en le prétraitant i.e en supprimant les données nulles et en codant certaines colonnes au format numérique (voir figure 27).

La méthode d'encodage des étiquettes fit_transform est utilisée ici pour convertir les valeurs catégorielles de la cible y (qui caractérise les attaques) en valeurs numériques, que l'algorithme de machine learning XGBoost peut comprendre et traiter. Par exemple :

- **le = LabelEncoder()** : On crée un objet LabelEncoder de la bibliothèque sklearn. Il est utilisé pour transformer des labels (ou cibles) catégoriels en valeurs numériques. Cela signifie que, si la colonne y contient des classes comme MQTT_Publish, DOS_SYN_Hping, Slowloris, etc., elles seront converties en des entiers comme 0, 1, 2, etc.
- **y_encoded = le.fit_transform(y)** : Cette commande fait deux choses :Ajuste **(fit)** l'encodeur à la série de labels (y) pour savoir quelles sont les différentes classes présentes dans les données et transforme **(transform)** ces classes catégorielles en entiers, et renvoie une version encodée de y. Par exemple, si y contient les classes

MQTT_Publish, ***DOS_SYN_Hping***, et Slowloris, elles seront transformées en 0, 1, et 2.

Ces valeurs sont des chaînes de caractères (valeurs catégorielles). Cependant, XGBoost et CNN-LSTM, ne peut pas traiter directement les valeurs textuelles. Ils nécessitent que la cible (les classes à prédire) soit représentée par des valeurs numériques. ***y_encoded*** contient les labels convertis en valeurs numériques.

train_test_split(X, y_encoded, test_size=0.2, random_state=42) :

- **X** : Les caractéristiques du trafic réseau capturées.
- **y_encoded** : La variable cible (le comportement ou l'attaque à prédire), après encodage par LabelEncoder.
- **test_size=0.2** : Indique que 20 % des données seront réservées pour le **test**, et 80 % seront utilisées pour l'entraînement du modèle. Cette division permet d'évaluer la performance du modèle sur un ensemble de données distinct de celles sur lesquelles il a été entraîné.
- **random_state=42** : Un paramètre utilisé pour garantir que la séparation des données soit reproductible. Chaque fois que le code sera exécuté avec cette valeur, les ensembles d'entraînement et de test seront divisés de la même manière.

X_train et **y_train** contiennent respectivement les caractéristiques et les labels cibles pour l'entraînement du modèle (80 % des données).
X_test et **y_test** contiennent les caractéristiques et les labels cibles pour tester le modèle (20 % des données).

```
# Entraîner le modèle XGBoost sur l'ensemble de données
from sklearn.utils.class_weight import compute_class_weight
# Séparons les entités et la cible des données échantillonnées
X = df.drop('Attack_type', axis=1)
y = df['Attack_type']

# Initialisons l'encodeur d'étiquette
le_proto = LabelEncoder()
le_service = LabelEncoder()

# Appliquons le codage des étiquettes aux colonnes « proto » et « service »
# Appliquons l'encodage sur les colonnes 'proto' et 'service'
# tranformations  des classes catégorielles en entiers
X['proto'] = le_proto.fit_transform(X['proto'])
X['service'] = le_service.fit_transform(X['service'])

# Encodage des étiquettes cibles
# tranformations  des classes catégorielles en entiers
le = LabelEncoder()
y_encoded = le.fit_transform(y)

# Divisons les données échantillonnées en ensembles de formation et de test (80 % d'entraînement, 20 % de test)
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)

# Calculer les poids des classes basés sur y_train pour equilibrer les classes
class_weights = compute_class_weight(class_weight='balanced', classes=np.unique(y_train), y=y_train)
class_weights_dict = dict(enumerate(class_weights))
```

Figure 27: Prétraitement des données

IV.6. Performance de XGBoost

Après la phase de prétraitement, une fois que le modèle est entraîné, nous l'avons évalué en utilisant les données de test et nous avons obtenu une précision de 99,88% qui est légèrement meilleur que la précision de [5] qui est de 99,74, et à laquelle nous confrontons nos résultats. Le F1-score (moyenne macro) est de 95% et le F1-score (moyenne pondérée) est de 99,88%

Afin de rendre le modèle XGBoost plus performant, nous avons procédé à un ajustement des hyperparamètres. Cela se fait en utilisant une technique comme la recherche par grille (GridSearchCV) pour ajuster ces hyperparamètres que sont.

- **learning_rate**: taux d'apprentissage (plus bas = convergence plus lente mais plus stable).C'est à dire la **taille des pas** effectués par l'algorithme lorsqu'il met à jour les poids pour minimiser l'erreur
- **max_depth**: profondeur maximale des arbres.
- **n_estimators**: nombre d'arbres dans le modèle.

Ainsi on obtient un meilleur résultat 99,9% (voir figure 28) lorsque le modèle est entraîné avec un taux d'apprentissage de 0.2, une profondeur maximale de 3 des arbres et un nombre d'arbres de 200. On constate que plus la profondeur des arbres augmente, plus on obtient une précision qui devient maximale lorsqu'on atteint une profondeur de 3 et un temps d'entraînement. Le tableau 7 modélise les performances du modèle.

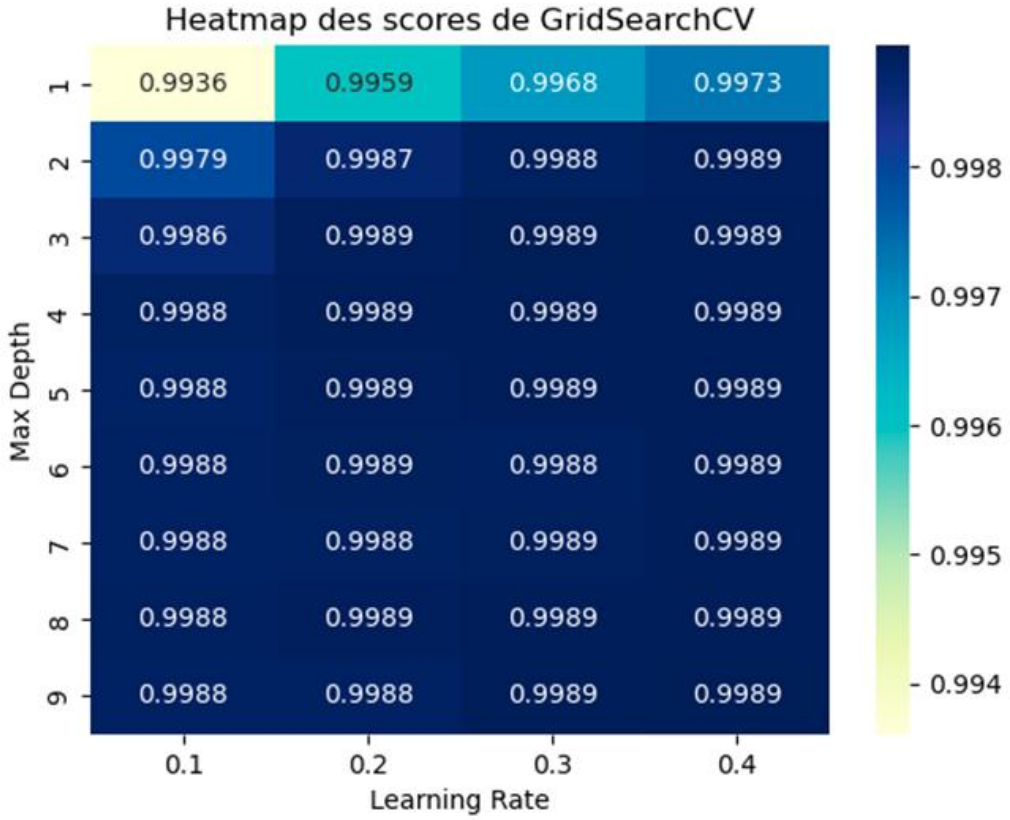


Figure 28 :

Accuracy	Précision	Recall	F1-Score	Temps d'Entraînement	Temps de reponse	Mémoire
99,89%	92.20%	96.83%	93.99%	38.17s	0.34s	442.12 MB

Visualisation des résultats sous forme de matrice colorée

Tableau 7 : Performance du modèle XGBoost

IV.7. Performance du CNN-LSTM

Ce modèle, conçu pour des tâches de classification sur des données séquentielles, commence par une couche de convolution (Convid2) pour capturer les caractéristiques spatiales, puis utilise des couches LSTM (Lstm_4 et Lstm_5) pour capturer les dépendances temporelles. Ensuite, des couches fully-connected (Dense_4 et Dense_5) sont utilisées pour la sortie finale. Le Tableau 8 présente le sommaire des différentes couches.

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 8, 64)	16,192
max_pooling1d_2 (MaxPooling1D)	(None, 4, 64)	0
dropout_6 (Dropout)	(None, 4, 64)	0
lstm_4 (LSTM)	(None, 4, 100)	66,000
dropout_7 (Dropout)	(None, 4, 100)	0
lstm_5 (LSTM)	(None, 50)	30,200
dropout_8 (Dropout)	(None, 50)	0
dense_4 (Dense)	(None, 50)	2,550
dense_5 (Dense)	(None, 12)	612

Total params: 115,554 (451.38 KB)
Trainable params: 115,554 (451.38 KB)
Non-trainable params: 0 (0.00 B)

Tableau 8 : Résumé du Modèle CNN-LSTM pour 2 couches de LSTM

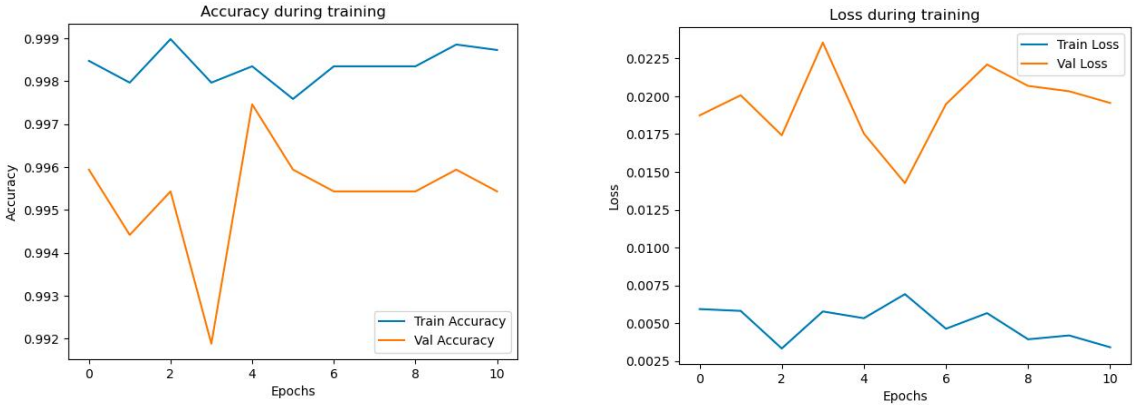
Après la définition des couches, nous procédons à l'ajustement des paramètres du modèle **CNN-LSTM** pour améliorer ses performances. A ce propos, plusieurs points peuvent être modifiés ou optimisés, en fonction des objectifs du modèle.

- Ajouter des couches LSTM empilées (avec return_sequences=True pour toutes sauf la dernière) peut permettre au modèle de mieux capturer les relations temporelles complexes.
- Augmenter ou diminuer le nombre d'unités dans la couche LSTM (par exemple, de 100 à 200) peut affecter la capacité du modèle à apprendre des séquences plus complexes.
- Manipuler le Dropout qui est à 0.2 avec des valeurs plus élevées ou plus faibles (0.3, 0.5, etc.) ou ajouter une régularisation pour les couches denses afin de réduire le surapprentissage.

- Ajoutez des couches de Batch Normalization après les couches convolutionnelles ou LSTM pour stabiliser et accélérer l'entraînement.

Nous avons entraîné plusieurs modèles avec différentes configurations de couches LSTM, puis nous les avons évalués en utilisant les données de validation. Les résultats obtenus sont consignés sur les figures 29, 30, 31, suivantes avec les paramètres de prédiction tels que Train Accuracy, Val Accuracy, Train Lost et Val Lost.

- **Train Accuracy:** c'est l'efficacité ou la précision du modèle à prédire correctement pour chaque Epochs (passage complet sur l'ensemble des données d'entraînement par le modèle) sur les données de test
- **Val Accuracy**: c'est l'efficacité du modèle à prédire sur les données de validation
- **Train Lost**: c'est le manque de précision du modèle sur les données de test, il doit être complémentaire à **Train Accuracy.**
- **Val Lost:** c'est le manque de précision du modèle sur les données de validation, il doit être complémentaire à **Val Accuracy.**



(a) La précision du modèle pour chaque passage des données d'entrainement et de validation

(b) L'erreur du modèle pour chaque passage des données d'entrainement et de validation

Figure 29 : Performance du modèle avec une couche de LSTM pour chaque epochs

On peut constater qu'avec une seule couche LSTM, le **Train Accuracy est** supérieur à 99,75% de précision sur les données de tests, alors que le **Val Accuracy** est inférieur 99,6 avec deux pics qui atteignent respectivement 99,2% et 99.7%. Ce qui montre que le modèle a encore à apprendre et peut être amélioré pour atteindre la même précision que sur les données de test. Les valeurs moyennes sont consignées dans le Tableau 9.

Précision	Rappel	F1-Score	Accuracy	Temps d'entraînement	Temps de réponse	Mémoire
95,98%	98,41%	96,82%	99,63%	26,61s	1,16 s	1102MB

Tableau 9: Performance du modèle avec une couche de LSTM

Globalement, on peut dire qu'avec une seule couche, le modèle apprend très vite et que le temps d'entraînement n'influence pas trop la précision qui resté quasiment constante. Il faut noter que la meilleure précision reste 99,9%.

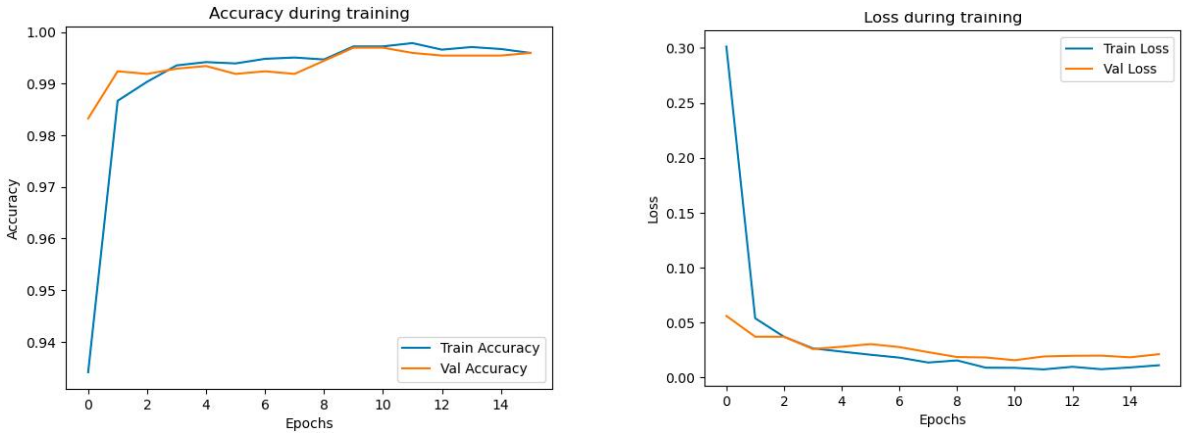


Figure 30: Performance du modèle avec deux couches de LSTM pour chaque epochs

On peut constater qu'avec deux couches LSTM, le **Train Accuracy est** supérieur à 99,90% et un temps de **Val Accuracy** élevé de 99,80%. On a une précision plus élevée sur les données de test et sur les données de validation. Ce qui

(a) La précision du modèle pour chaque passage des données d'entrainement et de validation

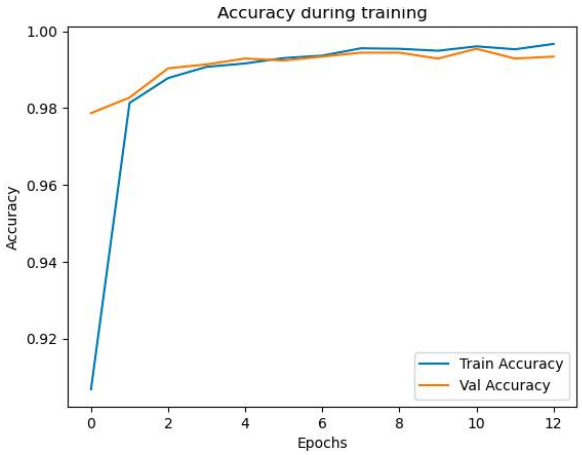
(b) L'erreur du modèle pour chaque passage des données d'entrainement et de validation

montre que le modèle est bien ajusté et fait de bonnes prédictions sur les données de test et de validation. Les valeurs moyennes sont consignées dans le Tableau 10.

Précision	Rappel	F1-Score	Accuracy	Temps d'entraînement	Temps de réponse	Mémoire
94,37%	98,34%	96,29%	99,59%	66,08s	1,94 s	1059MB

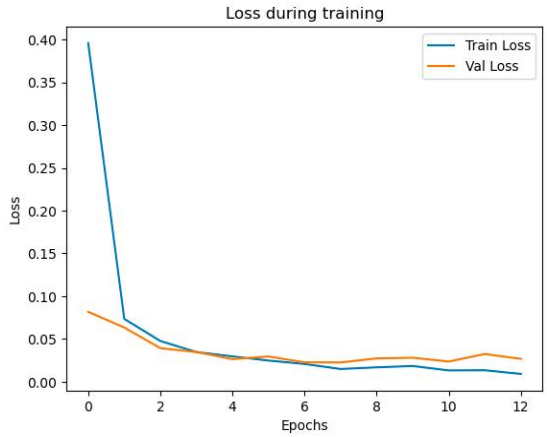
Tableau 10: Performance du modèle avec deux couches de LSTM

Globalement, on peut dire qu'avec deux couches, le modèle apprend lentement au voisinage d'une epoche, et qu'à partir de la 2ème Epochs on a plus de précision qu'avec une seule couche et que toutes les précisions dépassent 99,9%.



(a) La précision du modèle pour chaque passage des données d'entrainement et de validation

Figure 31 :



(b) L'erreur du modèle pour chaque passage des données d'entrainement et de validation

Performance du modèle avec trois couches de LSTM pour chaque epochs

On peut constater qu'avec trois couches LSTM, le **Train Accuracy est** au supérieur à 99,90% et un temps de **Val Accuracy** élevé de 99,80%. On a approximativement les mêmes précisions qu'avec deux couches. Ce qui montre que le meilleur ajustement du modèle est de deux couches LSTM. Les valeurs moyennes sont consignées dans le Tableau 11.

Précision	Rappel	F1-Score	Accuracy	Temps d'entraînement	Temps de réponse	Mémoire
95,85%	98,28%	96,70%	99,55%	60,06s	2,40 s	1162MB

Tableau 11: Performance du modèle avec trois couches de LSTM

Globalement avec 3 couches, il n'y a pas eu d'évolution sur les résultats par rapport à deux couches.

En conclusion, on peut noter qu'au niveau de la Figure 30-a la performance du modèle avec une couche montre que la précision de l'entraînement est très élevée, restant presque constante autour de 99,9%, tandis que la précision de validation est légèrement plus basse, avec quelques oscillations mais sans baisse notable. La Figure 30-b montre une perte d'entraînement très basse et stable, tandis que la perte de validation est un peu plus élevée et varie davantage. Ces précisions sont globalement bonnes. Cependant, cet écart entre les performances d'entraînement et de validation pourrait être un signe de légère sur-apprentissage. Il faut noter qu'avec une seule couche la précision est meilleur avec un temps d'apprentissage faible tandis qu'avec plus de couches on atteint de meilleures précisions mais avec un temps d'apprentissage relativement plus élevé. Nous privilégions donc le modèle avec deux couches, qui présente une meilleure précision aussi bien sur les données de test que de validation avec moins de ressources mémoire qu'un modèle à 3 couches. Dans la suite, nous comparons XGBoost avec le meilleur modèle de CNN-LSTM.

IV.8. Comparaison des deux modèles

Le Tableau 11 présente les résultats des deux modèles, nous pouvons sans doute remarquer

Modèle	Accuracy	Précision	Recall	F1-Score	Temps d'Entraînement	Temps de reponse	Mémoire
XGBoost	99,89%	92.20%	96.83%	93.99%	38.17s	0.34s	442.12 MB
CNN-LSTM	99,59%	94..37%	99.08%	96.29%	66.08s	1.94s	1059 MB

que, XGBoost présente de meilleures précisions.

Tableau 12: Résultats des deux modèles

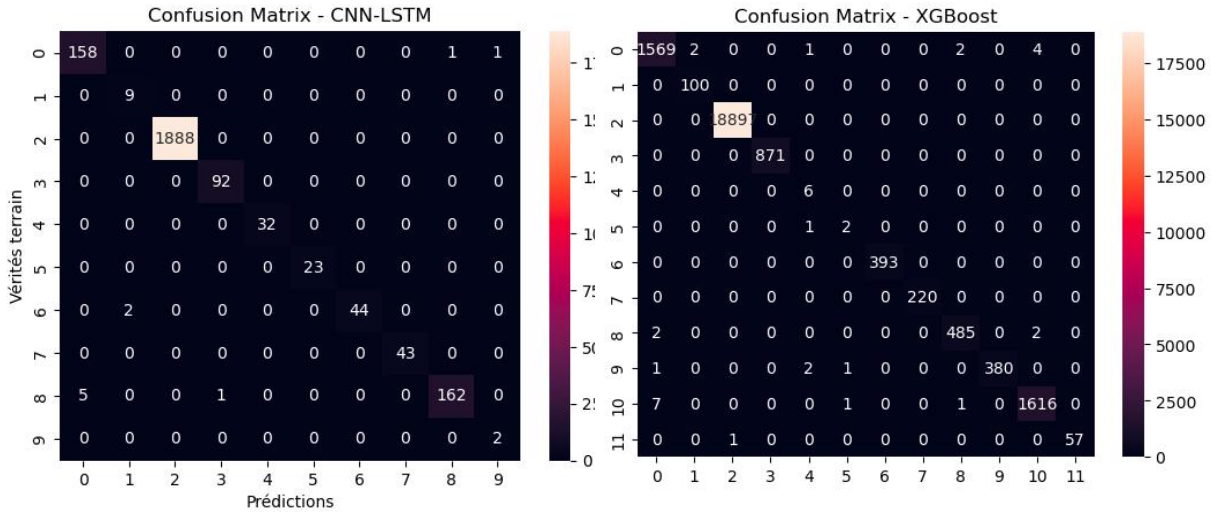


Tableau 13 : Matrice de confusion

Ces deux matrices de confusion pour les deux modèles montrent deux choses :

- Diagonale : Les valeurs sur la diagonale (de haut à gauche à bas à droite) représentent les cas où le modèle a fait une prédiction correcte, c'est-à-dire où la classe prédite correspond à la classe réelle.
- Cellules hors diagonale : Elles montrent les erreurs du modèle, où il a prédit une mauvaise classe.

On peut constater que globalement XGboost présente significativement plus de Vrai Positifs et légèrement plus de Faux positifs.

Avantages et inconvénients de XGBoost

- **Avantages** :
    - **Faible consommation en mémoire** : Par rapport au CNN-LSTM, il utilise beaucoup moins de mémoire (404.53 MB contre 1059 MB).
    - **Temps d'inférence rapide** : Avec un temps de réponse rapide de 0.33s, il est mieux adapté pour des décisions en temps réel dans des environnements à ressources limitées comme l'IoT.

- **Efficace pour des systèmes limités** : XGBoost est un modèle qui peut être optimisé pour fonctionner efficacement sur des dispositifs avec peu de ressources comme dans l'IoT.

- **Inconvénients** :
  - **Moins flexible que CNN-LSTM** : Pour des attaques complexes ou des séquences temporelles plus longues, XGBoost pourrait ne pas capturer autant d'informations que le CNN-LSTM.

Avantage et inconvénients de CNN-LSTM

- **Avantages** :
  - **Détection de séquences temporelles complexes** : L'avantage clé du CNN-LSTM est qu'il est capable de capturer les relations temporelles dans les données.
  - **Précision élevée sur le rappel et F1-Score** : Avec un rappel de 99.08% et un F1-Score de 96.29%, le CNN-LSTM peut exceller dans la détection des attaques sophistiquées et récurrentes.
- **Inconvénients** :
  - **Consommation en ressources** : Il nécessite plus de mémoire (1059 MB) et a un temps d'inférence plus long (1.94s), ce qui peut poser problème pour les dispositifs IoT à ressources limitées.
  - **Temps d'entraînement plus long** : Avec un temps d'entraînement plus élevé (73.08s), il pourrait ne pas être aussi efficace pour des systèmes qui nécessitent des mises à jour fréquentes ou des réentraînements.

XGBoost semble être la meilleure option pour notre solution de sécurité pour le protocole MQTT. Il est léger, rapide et tout aussi performant dans les tâches de détection d'anomalies. Il peut être optimisé pour détecter efficacement des comportements malveillants sans surcharger les appareils IoT. CNN-LSTM, bien que plus précis pour la détection d'attaques complexes, peut s'avérer trop gourmand en ressources pour les environnements IoT. Il serait adapté pour des infrastructures plus robustes où les ressources sont moins limitées (par exemple, sur des passerelles IoT ou des serveurs centraux qui collectent les données).

Conclusion

Dans ce chapitre, nous avons exploré deux modèles de machine learning, XGBoost et CNN-LSTM, afin de choisir celui qui convient le mieux pour renforcer la sécurité du protocole MQTT dans les environnements IoT.

Après une comparaison détaillée basée sur plusieurs critères tels que la précision, le rappel, le F1-Score, la consommation de mémoire, le temps d'inférence et les contraintes des dispositifs IoT, il apparaît que le modèle XGBoost est le plus adapté pour cette application.

Bien que le modèle CNN-LSTM offre de meilleures performances en termes de détection de séquences temporelles complexes et un rappel supérieur, sa consommation élevée de ressources et son temps d'inférence plus long le rendent moins viable pour des dispositifs IoT à faible capacité. En revanche, XGBoost offre une précision quasi identique, tout en étant rapide, léger et facile à déployer sur des appareils IoT, ce qui est crucial dans un environnement où les ressources sont limitées.

Ainsi, le modèle XGBoost a été retenu pour la sécurisation du protocole MQTT. Ce modèle, grâce à son efficacité en temps réel et sa faible consommation de ressources, constitue une solution optimale pour détecter des anomalies et prévenir les attaques dans les communications via MQTT, tout en garantissant la fiabilité et la viabilité du système IoT.

Conclusion Générale

Au cours de ce travail, nous avons exploré les technologies et défis qui façonnent la sécurité de l'Internet des Objets (IoT). Nous avons examiné les technologies de base de l'IoT, telles que les réseaux de capteurs sans fil et les technologies RFID, ainsi que les défis techniques liés à leur gestion. Nous avons également analysé les vulnérabilités des protocoles de communication IoT, comme MQTT et CoAP, en soulignant la nécessité de développer des mécanismes de sécurité adaptés aux contraintes de ces dispositifs, afin de protéger les systèmes IoT contre les cybermenaces.

Nous avons ensuite exploré l'impact de l'intelligence artificielle, en particulier du machine learning, pour renforcer la sécurité de l'IoT, en mettant l'accent sur la détection d'anomalies et d'intrusions. Enfin, une comparaison entre les modèles XGBoost et CNN-LSTM a révélé que XGBoost, plus léger et rapide, est mieux adapté aux environnements IoT avec des ressources limitées, offrant ainsi une solution efficace pour sécuriser les communications MQTT.

En conclusion, ce travail a permis de jeter les bases d'une approche intégrant intelligence artificielle et optimisation des ressources pour renforcer la sécurité des systèmes IoT. Les contributions proposées, notamment l'utilisation d'algorithmes de machine learning pour la détection d'anomalies dans le protocole MQTT, ouvrent des perspectives prometteuses pour l'amélioration continue de la sécurité dans ce domaine. Toutefois, la sécurité de l'IoT reste un défi évolutif, exigeant des recherches constantes pour faire face aux nouvelles menaces et répondre aux besoins croissants en matière de connectivité intelligente.

Ce travail ouvre plusieurs perspectives de recherche, étant donné que notre proposition est théorique, nous souhaitons la tester en pratique dans une architecture domotique réelle, afin d'évaluer leur performance en environnement IoT, en tenant compte des contraintes de ressources et de rapidité, et de confronter les résultats de la simulation à la réalité. Une analyse comparative entre XGBoost et d'autres modèles neuronaux avancés sera également réalisée pour identifier des solutions plus performantes. Par ailleurs, des approches émergentes telles que l'apprentissage par renforcement et les modèles fédérés seront explorées pour améliorer la sécurité des systèmes IoT tout en préservant la confidentialité des données sensibles.

BIBLIOGRAPHIE

[1] S. SAHRAOUI, « Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things) », doctoral, Université de Batna 2, 2016. Consulté le: 19 juillet 2024. [En ligne]. Disponible sur: http://eprints.univ-batna2.dz/308/

[2] K. Al-hammuri, F. Gebali, et A. Kanan, « ZTCloudGuard: Zero Trust Context-Aware Access Management Framework to Avoid Medical Errors in the Era of Generative AI and Cloud-Based Health Information Ecosystems », *AI*, vol. 5, n° 3, Art. n° 3, sept. 2024, doi: 10.3390/ai5030055.

[3] R. Soleymanzadeh, M. Aljasim, M. W. Qadeer, et R. Kashef, « Cyberattack and Fraud Detection Using Ensemble Stacking », *AI*, vol. 3, n° 1, Art. n° 1, mars 2022, doi: 10.3390/ai3010002.

[4] I. Dh. Belmiloud, « Intrusion Detection System based on Deep learning and Complex event processing for IoT environments », Thesis, 2024. Consulté le: 8 novembre 2024. [En ligne]. Disponible sur: https://repository.esi-sba.dz/jspui/handle/123456789/761

[5] N. Alotaibi, H. Ahmed, S. Kamel, et G. ElKabbany, « Secure Enhancement for MQTT Protocol Using Distributed Machine Learning Framework », *Sensors*, vol. 24, p. 1638, mars 2024, doi: 10.3390/s24051638.

[6] H. Mezili, « Vers une amélioration de la détection d'intrusion par les méthodes de sélection des fonctionnalités à l'aide des arbres de décision », Thesis, Université Ibn Khaldoun -Tiaret-, 2021. Consulté le: 29 septembre 2024. [En ligne]. Disponible sur: http://dspace.univ-tiaret.dz:80/handle/123456789/5509

[7] H. Alkahtani et T. H. H. Aldhyani, « Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications », *Security and Communication Networks*, vol. 2021, p. 1-23, sept. 2021, doi: 10.1155/2021/3806459.

[8] « RT-IoT2022(Real Time Internet Of Things) ». Consulté le: 3 octobre 2024. [En ligne]. Disponible sur: https://www.kaggle.com/datasets/supplejade/rt-iot2022real-time-internet-of-things

[9] P. E. Ceruzzi, « Les origines sociales et culturelles d'Internet », in *Les ingénieurs des Télécommunications dans la France contemporaine*, Institut de la gestion publique et du développement économique;, 2013. doi: 10.4000/books.igpde.3188.

[10] « Qu'est-ce que l'IoT ? », Oracle France. Consulté le: 30 janvier 2023. [En ligne]. Disponible sur: https://www.oracle.com/fr/internet-of-things/definition-internet-of-things-iot/

[11] C.-T. Kone, « Conception de l'architecture d'un réseau de capteurs sans fil de grande dimension », phdthesis, Université Henri Poincaré - Nancy I, 2011. Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://theses.hal.science/tel-00650839

[12] M. Bouallegue, « Protocoles de communication et optimisation de l'énergie dans les réseaux de capteurs sans fil », PhD Thesis, Université du Maine, 2016. Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://theses.hal.science/tel-01400679/

[13] A. Chafik, « Architecture de réseau de capteurs pour la surveillance de grands systèmes physiques à mobilité cyclique », PhD Thesis, Université de Lorraine, 2014. Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://hal.science/tel-01081865/

[14] « Les réseaux de capteurs : concepts de base - Recherche Google ». Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://www.google.com/search?q=Les+r%C3%A9seaux+de+capteurs+%3A+concepts+de+base&oq=Les+r%C3%A9seaux+de+capteurs+%3A+concepts+de+base&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQRRg80gEJMjIzMWowajE1qAIIsAIB&sourceid=chrome&ie=UTF-8

[15] R. Weinstein, « RFID: a technical overview and its application to the enterprise », *IT Prof.*, vol. 7, nº 3, p. 27-33, mai 2005, doi: 10.1109/MITP.2005.69.

[16] S. F. Wamba, « LES IMPACTS DE LA TECHNOLOGIE RFID ET DU RÉSEAU EPC SUR LA GESTION DE LA CHAÎNE D'APPROVISIONNEMENT: LE CAS DE L'INDUSTRIE DU COMMERCE DE DÉTAIL ».

[17] Dipole, « Lecteur RFID Portable Technology Solutions 2128 | Dipole », DipoleRFID. Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://www.dipolerfid.fr/produit/lecteur-rfid-portable-technology-solutions-2128

[18] A. Ghiotto, « Conception d'antennes de tags RFID UHF, application a la réalisation par jet de matière. », PhD Thesis, Institut National Polytechnique de Grenoble-INPG, 2008. Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://theses.hal.science/tel-00389807/

[19] A. Douidi et O. Chikh, « Conception et simulation de tags RFID. », PhD Thesis, 2018. Consulté le: 8 août 2024. [En ligne]. Disponible sur: http://dspace1.univ-tlemcen.dz/handle/112/13302

[20] « Fonctionnement d'un système RFID ». Consulté le: 6 décembre 2023. [En ligne]. Disponible sur: https://www.connectwave.fr/techno-appli-iot/rfid/fonctionnement-dun-systeme-rfid/

[21] « Sécurisation des services de maintenance intelligents : sécurité matérielle et TLS pour MQTT | Publication de la conférence IEEE | IEEE Xplore ». Consulté le: 12 juillet 2024. [En ligne]. Disponible sur: https://ieeexplore.ieee.org/abstract/document/7281913

[22] Y. Abbassi et H. Benlahmer, « Un aperçu sur la sécurité de l'internet des objets (IOT) », in *Colloque sur les Objets et systèmes Connectés-COC'2021*, 2021. Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://hal.science/hal-03593723/

[23] D. Aksu et M. A. Aydin, « A Survey of IoT Architectural Reference Models », in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, Istanbul, Turkey: IEEE, mars 2019, p. 413-417. doi: 10.1109/SSD.2019.8893170.

[24] T. Sylla, « Sécurité et vie privée centrées sur l'utilisateur dans l'IoT », phdthesis, Université de Bordeaux ; Université des sciences, des techniques et des technologies de Bamako (Mali), 2021. Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://theses.hal.science/tel-03529415

[25] « Infrastructure mondiale de l'information, protocole Internet, réseaux de prochaine génération, Internet des objets et villes intelligentes ». Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://www.itu.int/rec/T-REC-Y/fr

[26] « yasmine labio Mécanisme de Sécurité pour L'internet des Objets - Recherche Google ». Consulté le: 8 août 2024. [En ligne]. Disponible sur: https://www.google.com/search?q=yasmine

[27] A. Larab, P. Martineau, et P. Gaucher, « Un module de sécurité pour sécuriser les communications domestiques », in *Third International Conference: Sciences of Electronic, Technologies of Information and Telecommunications SETIT*, 2005. Consulté le: 8 août 2024. [En ligne]. Disponible sur: http://www.setit.rnu.tn/last_edition/setit2005/reseau/265.pdf

[28] A. Martin, « IOTFLA : une architecture de domotique sécurisée respectueuse de la vie privée ». Consulté le: 22 novembre 2024. [En ligne]. Disponible sur: https://archipel.uqam.ca/13808/

[29] « Amélioration de la sécurité des Objets Connectés... - Google Scholar ». Consulté le: 12 juillet 2024. [En ligne]. Disponible sur: https://scholar.google.com/scholar?hl=fr&as_sdt=0%2C5&q=Am%C3%A9lioration+de+la+s%C3%A9curit%C3%A9+des+Objets+Connect%C3%A9s+%28IoT%29+utilisant+le+protocole+MQTT+dans+le+domaine+de+la+E-sant%C3%A9&btnG=

[30] « Comprendre le protocole MQTT ». Consulté le: 12 juillet 2024. [En ligne]. Disponible sur: https://www.ip-system

es.com/details-comprendre+le+protocole+mqtt-795.html

[31] G. Nebbione et M. C. Calzarossa, « Security of IoT Application Layer Protocols: Challenges and Findings », *Future Internet*, vol. 12, n° 3, p. 55, mars 2020, doi: 10.3390/fi12030055.

[32] R. Chakravarthy, « An experimental study of iot-based topologies on MQTT protocol for agriculture intrusion detection », *Measurement: Sensors*, vol. 24, p. 100470, 2022.

[33] « Security and Performance Analysis of MQTT Protocol with TLS in IoT Networks | IEEE Conference Publication | IEEE Xplore ». Consulté le: 12 juillet 2024. [En ligne]. Disponible sur: https://ieeexplore.ieee.org/document/9717495

[34] Z. Shelby, K. Hartke, et C. Bormann, « The Constrained Application Protocol (CoAP) », Art. n° rfc7252, juin 2014, doi: 10.17487/RFC7252.

[35] B. H. Çorak, F. Y. Okay, M. Güzel, Ş. Murt, et S. Ozdemir, « Comparative Analysis of IoT Communication Protocols », in *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, juin 2018, p. 1-6. doi: 10.1109/ISNCC.2018.8530963.

[36] « Protocole d'application contraint (CoAP) ». Consulté le: 17 juillet 2024. [En ligne]. Disponible sur: https://www.ietf.org/archive/id/draft-ietf-core-coap-02.html#appendix_codes

[37] H. B. Rebah, « Gateway IoT de pilotage et de surveillance des capteurs domestiques via le protocole MQTT », présenté à colloque international sur les objets et systèmes connectés 2022, mai 2022. Consulté le: 19 juillet 2024. [En ligne]. Disponible sur: https://hal.science/hal-04258670

[38] A. Bhattacharjya, X. Zhong, J. Wang, et X. Li, « CoAP—Application Layer Connection-Less Lightweight Protocol for the Internet of Things (IoT) and CoAP-IPSEC Security with DTLS Supporting CoAP », in *Digital Twin Technologies and Smart Cities*, M. Farsi, A. Daneshkhah, A. Hosseinian-Far, et

H. Jahankhani, Éd., Cham: Springer International Publishing, 2020, p. 151-175. doi: 10.1007/978-3-030-18732-3_9.

[39] « Quels protocoles applicatifs pour l'Internet des Objets ? - FRUGAL PROTOTYPE ». Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://www.frugalprototype.com/quels-protocoles-applicatifs-pour-linternet-des-objets/

[40] « Mécanismes de la couche d'authentification et de sécurité simple (SASL) ». Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://www.iana.org/assignments/sasl-mechanisms/sasl-mechanisms.xhtml

[41] F. Z. Saadaoui, A. Maizate, et M. Ouzzif, « État d'art sur les protocoles en temps réel pour l'internet des objets sous le réseau NDN », in *Colloque sur les Objets et systèmes Connectés*, CASABLANCA, Morocco: Ecole Supérieure de Technologie de Casablanca (Maroc), Institut Universitaire de Technologie d'Aix-Marseille (France), juin 2019. Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://hal.science/hal-02296848

[42] « À propos de la spécification de sécurité DDS version 1.1 ». Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://www.omg.org/spec/DDS-SECURITY/1.1/About-DDS-SECURITY/

[43] « Fig. 1. Essential DDS architecture », ResearchGate. Consulté le: 6 août 2024. [En ligne]. Disponible sur: https://www.researchgate.net/figure/Essential-DDS-architecture_fig1_261486849

[44] S. Lakshminarayana, A. Praseed, et P. S. Thilagam, « Securing the IoT Application Layer from an MQTT Protocol Perspective: Challenges and Research Prospects », *IEEE Commun. Surv. Tutorials*, p. 1-1, 2024, doi: 10.1109/COMST.2024.3372630.

[45] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, et C. Tan, « Performance evaluation of MQTT and CoAP via a common middleware », avr. 2014. doi: 10.1109/ISSNIP.2014.6827678.

[46] M. M. Raikar et M. S M, « Vulnerability assessment of MQTT protocol in Internet of Things (IoT) », in *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, mai 2021, p. 535-540. doi: 10.1109/ICSCCC51823.2021.9478156.

[47] P. Ka, « Proposition d'une architecture basée sur l'intelligence artificielle pour le contrôle de congestion dans l'internet des objets », 2022, Consulté le: 20 août 2024. [En ligne]. Disponible sur: http://rivieresdusud.uasz.sn/xmlui/handle/123456789/1484

[48] A. A. Sodemann, M. P. Ross, et B. J. Borghetti, « A Review of Anomaly Detection in Automated Surveillance », *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, n° 6, p. 1257-1272, nov. 2012, doi: 10.1109/TSMCC.2012.2215319.

[49] J. Bzai *et al.*, « Machine Learning-Enabled Internet of Things (IoT): Data, Applications, and Industry Perspective », *Electronics*, vol. 11, n° 17, Art. n° 17, janv. 2022, doi: 10.3390/electronics11172676.

[50] M. A. CHALOUF, « Intelligence artificielle pour la sécurité en e-santé », *Gestion de la sécurité en e-santé: Sécurité des communications, sécurité du traitement des données et respect de la vie privée des patients*, p. 213, 2024.

[51] T. P. CONIX Consultant Sénior, « Cybersurveillance : Risques et opportunités », Global Security Mag Online. Consulté le: 5 septembre 2024. [En ligne]. Disponible sur: https://www.globalsecuritymag.fr/Cybersurveillance-Risques-et,20161201,67355.html

[52] A. Gasmi, « Proposition d'une Architecture réseaux compus sécurise par un firewall opensource pfsense », Thesis, Université Echahid Chikh Larbi Tébessi -Tébessa, 2023. Consulté le: 5 septembre 2024. [En ligne]. Disponible sur: http://dspace.univ-tebessa.dz:8080/jspui/handle/123456789/http//localhost:8080/jspui/handle/123456789/10968

[53] G. Milan, L. Vassio, I. Drago, et M. Mellia, « RL-IoT: Reinforcement Learning to Interact with IoT Devices », in *2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS)*, Barcelona, Spain: IEEE, août 2021, p. 1-6. doi: 10.1109/COINS51742.2021.9524260.

[54] M.-A. Chalouf, *La gestion et le contrôle intelligents des performances et de la sécurité dans l'IoT*. ISTE Group, 2022.

[55] A. TALBA, « ALLOCATION DYNAMIQUE DES FREQUENCES POUR L'INTERNET DES OBJETS (IoT) », Consulté le: 9 septembre 2024. [En ligne]. Disponible sur: https://www.researchgate.net/profile/Talba-Adam/publication/363611102_Allocation_Dynamique_des_frequences_pour_l'Internet_des_Objets_IoT/links/63259643873eca0c009288c9/Allocation-Dynamique-des-frequences-pour-lInternet-des-Objets-IoT.pdf

[56] A. Hemmer, « Méthodes de détection pour la sécurité des systèmes IoT hétérogènes », PhD Thesis, Université de Lorraine, 2023. Consulté le: 9 septembre 2024. [En ligne]. Disponible sur: http://docnum.univ-lorraine.fr/public/DDOC_T_2023_0020_HEMMER.pdf

[57] Y. LeCun, « L'apprentissage profond, une révolution en intelligence artificielle », *La lettre du Collège de France*, n° 41, Art. n° 41, nov. 2016, doi: 10.4000/lettre-cdf.3227.

[58] « Introduction to Recurrent Neural Network », GeeksforGeeks. Consulté le: 15 octobre 2024. [En ligne]. Disponible sur: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/

[59] S. Serardi et M. E. houcine Benhamouda, « Vers un système de détection d'intrusion dans l'Internet des Objets », Thesis, Université Ibn Khaldoun, 2023. Consulté le: 29 septembre 2024. [En ligne]. Disponible sur: http://dspace.univ-tiaret.dz:80/handle/123456789/13453

[60] Daniel, « Long Short Term Memory (LSTM) : de quoi s'agit-il ? », Formation Data Science | DataScientest.com. Consulté le: 16 octobre 2024. [En ligne]. Disponible sur: https://datascientest.com/long-short-term-memory-tout-savoir

[61] A. Toumi, J.-C. Cexus, M. Abid, et A. Khenchaf, « Un réseau hybride CNN-LSTM pour la classification de navires à partir d'une base frugale des images SAR », présenté à GRETSI - Groupe de Recherche en Traitement du Signal et des Images, août 2023. Consulté le: 16 octobre 2024. [En ligne]. Disponible sur: https://ensta-bretagne.hal.science/hal-04320593

[62] « Application des algorithmes d'apprentissage automatique pour la détection de défauts de roulements sur les machines tournantes dans le cadre de l'Industrie 4.0 ». Consulté le: 9 septembre 2024. [En ligne]. Disponible sur: https://constellation.uqac.ca/id/eprint/5857/

[63] « Systéme de surveillance intelligent basé sur l'IoT pour la prédiction des maladies cardiaques ». Consulté le: 9 septembre 2024. [En ligne]. Disponible sur: http://dspace.univ-tebessa.dz:8080/xmlui/handle/123456789/10940

[64] O. Lourme et M. Hauspie, « Contribution à l'adoption des IDS dans l'IoT », 2021, Consulté le: 9 septembre 2024. [En ligne]. Disponible sur: https://hal.science/hal-03486267/

[65] J. Robert, « Reinforcement Learning : Définition et application », Formation Data Science | DataScientest.com. Consulté le: 9 septembre 2024. [En ligne]. Disponible sur: https://datascientest.com/reinforcement-learning

[66] D. Godfrey, B. Suh, B. H. Lim, K.-C. Lee, et K.-I. Kim, « An Energy-Efficient Routing Protocol with Reinforcement Learning in Software-Defined Wireless Sensor Networks », *Sensors*, vol. 23, n° 20, Art. n° 20, janv. 2023, doi: 10.3390/s23208435.

[67] A. Outchakoucht, H. Es-Samaali, A. Abou El Kalam, et S. Benhadou, « Apprentissage par Renforcement et Blockchain: Nouvelle approche pour sécuriser l'IoT », 2019, Consulté le: 9 septembre 2024. [En ligne]. Disponible sur: http://www.openscience.fr/IMG/pdf/iste_ido19v3n1_3.pdf

[68] Y. Hadjadj-Aoul et S. Aït-Chellouche, « Gestion des accès massifs des équipements dans les réseaux NB-IoT: une stratégie basée sur l'apprentissage par renforcement », *La gestion et le contrôle intelligents des performances et de la sécurité dans l'IoT*, p. 1-29, 2021.

[69] C. Maniraguha, « THESIS/THÈSE », Consulté le: 9 septembre 2024. [En ligne]. Disponible sur: https://core.ac.uk/download/pdf/326317993.pdf

[70] « Spotfire | Demystifying the Random Forest Algorithm for Accurate Predictions », Spotfire. Consulté le: 14 octobre 2024. [En ligne]. Disponible sur: http://www.spotfire.com/glossary/what-is-a-random-forest

[71] « SVM », Data Analytics Post. Consulté le: 14 octobre 2024. [En ligne]. Disponible sur: https://dataanalyticspost.com/Lexique/svm/

[72] L. Ponthier, « Application des approches de modelisation et de machine learning a l'individualisation des doses d'anti-infectieux en pediatrie », phdthesis, Université de Limoges, 2023. Consulté le: 30 septembre 2024. [En ligne]. Disponible sur: https://theses.hal.science/tel-04404574

[73] Y. Wang, Z. Pan, J. Zheng, L. Qian, et L. Mingtao, « A hybrid ensemble method for pulsar candidate classification », *Astrophysics and Space Science*, vol. 364, août 2019, doi: 10.1007/s10509-019-3602-4.

[74] T. Wang, P. Reiffsteck, C. Chevalier, C.-W. Chen, et F. Schmidt, « Maintenance Prédictive des ouvrages d'art avec des fondations en sites aquatiques », in *11èmes journées nationales de géotechnique et de géologie de l'ingénieur*, 2022. Consulté le: 30 septembre 2024. [En ligne]. Disponible sur: https://hal.science/hal-03719846/document

[75] « descente de gradient - Recherche Google ». Consulté le: 14 octobre 2024. [En ligne]. Disponible sur: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://exo7math.github.io/deepmath-exo7/descente/descente.pdf

[76] E. C. Júnior, W. L. Costa, A. L. C. Portela, L. S. Rocha, R. L. Gomes, et R. M. C. Andrade, « Detecting Attacks and Locating Malicious Devices Using Unmanned Air Vehicles and Machine Learning », *Journal of Internet Services and Applications*, vol. 13, nº 1, Art. nº 1, sept. 2022, doi: 10.5753/jisa.2022.2327.

[77] K. Edemacu et J. W. Kim, « Multi-Party Privacy-Preserving Logistic Regression with Poor Quality Data Filtering for IoT Contributors », *Electronics*, vol. 10, nº 17, Art. nº 17, janv. 2021, doi: 10.3390/electronics10172049.

[78] D. Puthal *et al.*, « Decision tree based user-centric security solution for critical IoT infrastructure », *Computers and Electrical Engineering*, vol. 99, p. 107754, avr. 2022, doi: 10.1016/j.compeleceng.2022.107754.

[79] Z. H. Abdaljabar, O. N. Ucan, et K. M. Ali Alheeti, « An Intrusion Detection System for IoT Using KNN and Decision-Tree Based Classification », in *2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI)*, déc. 2021, p. 1-5. doi: 10.1109/MTICTI53925.2021.9664772.

[80] Y. Hadjadj-Aoul et S. Ait Chellouche, « Access Control in NB-IoT Networks: A Deep Reinforcement Learning Strategy », *Information*, vol. 11, p. 541, nov. 2020, doi: 10.3390/info11110541.

[81] A. Sammoud, O. Hamdi, M.-A. Chalouf, et N. MONTAVONT, « Apports de la biométrie et de l'intelligence artificielle dans la sécurisation de l'IoT », 2022, p. 205-230. doi: 10.51926/ISTE.9053.ch8.

[82] T. Sylla, M. Chalouf, F. Krief, et K. Samaké, « SETUCOM: Secure and Trustworthy Context Management for Context-Aware Security and Privacy in the Internet of Things », *Security and Communication Networks*, vol. 2021, p. 1-24, avr. 2021, doi: 10.1155/2021/6632747.

[83] M. A. Abid *et al.*, « Evolution towards Smart and Software-Defined Internet of Things », *AI*, vol. 3, nº 1, Art. nº 1, mars 2022, doi: 10.3390/ai3010007.

[84] M. ABDELHEDI et O. HAMDI, « Sécurité du traitement des données médicales », 2024, p. 183-211. doi: 10.51926/ISTE.9179.ch8.

[85] R. Lazzarini, H. Tianfield, et V. Charissis, « Federated Learning for IoT Intrusion Detection », *AI*, vol. 4, nº 3, Art. nº 3, sept. 2023, doi: 10.3390/ai4030028.

[86] S. Deshmukh-Bhosale et S. S. Sonavane, « A real-time intrusion detection system for wormhole attack in the RPL based Internet of Things », *Procedia Manufacturing*, vol. 32, p. 840-847, 2019.

[87] J. Ren, Y. Zhang, K. Zhang, et X. Shen, « Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks », *IEEE Transactions on Wireless Communications*, vol. 15, nº 5, p. 3718-3731, 2016.

[88] B. Chen, D. W. Ho, G. Hu, et L. Yu, « Secure fusion estimation for bandwidth constrained cyber-physical systems under replay attacks », *IEEE transactions on cybernetics*, vol. 48, n° 6, p. 1862-1876, 2017.

[89] S. Smys, A. Basar, et H. Wang, « Hybrid intrusion detection system for internet of things (IoT) », *Journal of ISMAC*, vol. 2, n° 04, p. 190-199, 2020.

[90] S. Rathore et J. H. Park, « Semi-supervised learning based distributed attack detection framework for IoT », *Applied Soft Computing*, vol. 72, p. 79-89, 2018.

[91] A. K. Mishra, A. K. Tripathy, D. Puthal, et L. T. Yang, « Analytical model for sybil attack phases in internet of things », *IEEE Internet of Things Journal*, vol. 6, n° 1, p. 379-387, 2018.

[92] « Capteurs | Texte intégral gratuit | Enquête sur les menaces de sécurité dans l'IoT agricole et l'agriculture intelligente ». Consulté le: 14 septembre 2024. [En ligne]. Disponible sur: https://www.mdpi.com/1424-8220/20/22/6458

[93] R. NOUAR, « Une approche d'apprentissage automatique pour la détection des maladies dans les fermes intelligentes », Consulté le: 14 septembre 2024. [En ligne]. Disponible sur: http://archives.univ-biskra.dz/bitstream/123456789/15744/1/Rabie_NOUAR.pdf

[94] R. Khatoun, *Cybersécurité des maisons intelligentes: Architectures, solutions et technologies*. ISTE Group, 2024.

[95] S. Katsikeas *et al.*, *Lightweight & Secure Industrial IoT Communications via the MQ Telemetry Transport Protocol*. 2017. doi: 10.1109/ISCC.2017.8024687.

[96] A. Mektoubi, H. L. Hassani, A. Zakari, et O. Malassé, « NOUVELLE APPROCHE DE COMMUNICATION SÉCURISÉE DES OBJETS CONNECTÉS BASÉE SUR LE PROTOCOLE MQTT », *Revue Méditerranéenne des Télécommunications*, vol. 6, n° 2, Art. n° 2, juill. 2016, Consulté le: 15 septembre 2024. [En ligne]. Disponible sur: https://revues.imist.ma/index.php/RMT/article/view/5838

[97] J. Samandari et C. Gritti, « Post-Quantum Authentication in the MQTT Protocol », *Journal of Cybersecurity and Privacy*, vol. 3, p. 416-434, juill. 2023, doi: 10.3390/jcp3030021.

[98] V. Gupta, S. Khera, et N. Turk, « MQTT protocol employing IOT based home safety system with ABE encryption », *Multimedia Tools and Applications*, vol. 80, p. 1-19, janv. 2021, doi: 10.1007/s11042-020-09750-4.

[99] F. Mendoza-Cardenas, R. S. Leon-Aguilar, et J. L. Quiroz-Arroyo, « CP-ABE encryption over MQTT for an IoT system with Raspberry Pi », in *2022 56th Annual Conference on Information Sciences and Systems (CISS)*, mars 2022, p. 236-239. doi: 10.1109/CISS53076.2022.9751194.

[100] M. Calabretta, R. Pecori, M. Vecchio, et L. Veltri, « MQTT-Auth: a Token-based Solution to Endow MQTT with Authentication and Authorization Capabilities », *Journal of Communications Software and Systems*, vol. 14, n° 4, p. 320-331, oct. 2018, doi: 10.24138/jcomss.v14i4.604.

[101] M. Massad et B. Alsaify, *MQTTSec Based on Context-Aware Cryptographic Selection Algorithm (CASA) for Resource-Constrained IoT Devices*. 2020. doi: 10.1109/ICICS49469.2020.239541.

[102] T. K. Boppana et P. Bagade, « GAN-AE: An unsupervised intrusion detection system for MQTT networks », *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105805, mars 2023, doi: 10.1016/j.engappai.2022.105805.

[103] M. A. Khan *et al.*, « A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT », *Sensors*, vol. 21, oct. 2021, doi: 10.3390/s21217016.

[104] A. Alzahrani et T. Aldhyani, « Artificial Intelligence Algorithms for Detecting and Classifying MQTT Protocol Internet of Things Attacks », *Electronics*, vol. 11, nov. 2022, doi: 10.3390/electronics11223837.

[105] Z. Hayette, B. Mehdi, et R. Chikh, « Detection of DoS Attacks in MQTT Environment », 2023, p. 129-140. doi: 10.1007/978-3-031-46338-9_10.

[106] A. Sultan, S. Mehmood, et H. Zahid, *Man in the Middle Attack Detection for MQTT based IoT devices using different Machine Learning Algorithms*. 2022, p. 121. doi: 10.1109/ICAI55435.2022.9773590.

[107] I. Idrissi, M. Azizi, et O. Moussaoui, « An unsupervised generative adversarial network based-host intrusion detection system for IoT devices », *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, févr. 2022, doi: 10.11591/ijeecs.v25.i2.pp1140-1150.

[108] D. Zitouni et A. Barka, « Un système de détection d'intrusion basés sur l'apprentissage profond pour la cybersécurité », Thesis, Kasdi Merbah University OUARGLA ALGERIA, 2024. Consulté le: 22 novembre 2024. [En ligne]. Disponible sur: http://dspace.univ-ouargla.dz/jspui/handle/123456789/37016

[109] F. Mosaiyebzadeh, L. G. Araujo Rodriguez, D. Macedo Batista, et R. Hirata, « A Network Intrusion Detection System using Deep Learning against MQTT Attacks in IoT », in *2021 IEEE Latin-American Conference on Communications (LATINCOM)*, Santo Domingo, Dominican Republic: IEEE, nov. 2021, p. 1-6. doi: 10.1109/LATINCOM53176.2021.9647850.

[110] A. Z. Agghey, L. J. Mwinuka, S. M. Pandhare, M. A. Dida, et J. D. Ndibwile, « Detection of Username Enumeration Attack on SSH Protocol: Machine Learning Approach », *Symmetry*, vol. 13, n° 11, Art. n° 11, nov. 2021, doi: 10.3390/sym13112192.

[111] « Algorithmes d'intelligence artificielle pour la détection et la classification des attaques contre l'Internet des objets du protocole MQTT ». Consulté le: 11 novembre 2024. [En ligne]. Disponible sur: https://www.mdpi.com/2079-9292/11/22/3837

[112] « Qu'est-ce que Python ? – Le langage Python expliqué – AWS », Amazon Web Services, Inc. Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://aws.amazon.com/fr/what-is/python/

[113] « Welcome to Python.org », Python.org. Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://www.python.org/

[114] « Installer Anaconda pour Python : comment faire ? » Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://datascientest.com/installer-anaconda-tout-savoir

[115] « Download Anaconda Distribution », Anaconda. Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://www.anaconda.com/download

[116] R. Kassel, « Jupyter Notebook : Un outil indispensable en partage de code », Formation Data Science | DataScientest.com. Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://datascientest.com/jupyter-notebook-tout-savoir

[117] « Project Jupyter ». Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://jupyter.org

[118] « Qu'est-ce que NumPy ? — Manuel NumPy v2.2.dev0 ». Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://numpy.org/devdocs/user/whatisnumpy.html

[119] « pandas - Bibliothèque d'analyse de données Python ». Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://pandas.pydata.org/

[120] « Introduction à Seaborn — documentation de Seaborn 0.13.2 ». Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://seaborn.pydata.org/tutorial/introduction.html

[121] « Glossary of Common Terms and API Elements », scikit-learn. Consulté le: 2 octobre 2024. [En ligne]. Disponible sur: https://scikit-learn/stable/glossary.html

[122] « TensorFlow », TensorFlow. Consulté le: 17 octobre 2024. [En ligne]. Disponible sur: https://www.tensorflow.org/?hl=fr