الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
Université Akli Mohand Oulhadj - Bouira -
Tasdawit Akli Muḥend Ulḥağ - Tubirett -
Faculté des Sciences Appliquées

وزارة التعليم العالي والبحث العلمي
جامعة أكلي محند أولحاج
-البويرة-
كلية العلوم التطبيقية

Référence :……../**MM**/**2025** المرجع :……./م م/**2025**

Mémoire de Master(ESE)

Présenté au

Département : Génie Électrique
Domaine : Sciences et Technologies
Filière : Electronique
Spécialité : Electronique des systèmes embarqués

Réalisé par :

CHEKKOUR AZIZA

Et

ZADI FATIMA ZOUHRA

Thème

Détection d'Intrusions dans les Réseaux IOT

Soutenu le: **..26…/…06./2025**

Mr : …***Madjdoub smail***…… Prof. Univ. Bouira encadreur

Année Universitaire: 2024-2025

Remerciements

Ce projet de fin d'études marque l'aboutissement de mon parcours de Master en Électronique, spécialité Systèmes Embarqués, à université aklimhand el hadj et je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à sa réalisation.

Je remercie chaleureusement Monsieur madjdoubsmail, mon encadrant universitaire, pour sa disponibilité, ses conseils pertinents, et l'encadrement rigoureux qu'il m' apporté tout au long de ce travail. Ses orientations méthodologiques m'ont permis d'avancer avec clarté et structure.

Je remercie également toute l'équipe technique et les collaborateurs de département génie électrique , pour leur accueil, leur soutien et leur esprit de partage, qui ont fortement contribué à la qualité de mon expérience professionnelle.

Enfin, je tiens à exprimer toute ma reconnaissance à ma famille sourtout mon père et à mes amis, pour leur soutien moral constant et leur encouragement indéfectible tout au long de mes études.

Dédicace.

Je dédie ce travail de fin d'études à toutes les personnes qui m'ont soutenu(e) tout au long de ce parcours.

À mes parents, pour leur amour inconditionnel, leur patience, leurs encouragements constants et leur confiance inébranlable. Votre soutien a été ma plus grande force.

À ma famille, qui ont toujours été présents dans les moments de doute comme dans les moments de joie.

À mes amis proches, pour leur aide, leurs conseils, leurs encouragements, et pour avoir rendu ce parcours plus agréable.

Enfin, je dédie également ce travail à tous ceux qui, de près ou de loin, ont contribué à mon évolution personnelle et professionnelle.

Chekkouraziza.

Dédicace.

Je dédie ce modeste travail Au meilleur des pèreset à ma très chère maman qu'ils trouvent en moila source de leur fierté qui ne cessent de me

donner avec amour le nécessaire pour que jepuisse arriver à ce que je suis aujourd'hui.

Que dieu les protège et que la réussite soittoujours à ma portée pour que je puisse vous

combler de bonheur.

À mes sœurs Nawal et Meriem et mes frèresAbdlwahab Mouhamed et Bilal

A mes belles sœurs Nabila et Mounia et mes

beaux-frères Lyes et AymenA mes petits neveux et mes petites nièces.

À toute ma famille Et A toutes mes amiesA tous les gens qui me connaissent et que je

connais

À tous ceux qui me sont chers, aux personnes quim'ont aidé et encouragé de près ou de loin, quiétaient toujours à mes côtés et qui m'ont

accompagné durant mon chemin d'études.

ZADI FATIMA ZOHRA

Introduction générale

Introduction Générale :

Avec l'essor rapide des technologies de l'information et de la communication, l'Internet des objets (IoT - *Internet of Things*) a connu un développement considérable au cours de la dernière décennie. Il s'agit d'un paradigme technologique qui permet à des objets physiques d'interagir, de collecter et d'échanger des données à travers des réseaux, sans intervention humaine directe. Ces objets connectés sont aujourd'hui omniprésents, que ce soit dans les domaines de la santé, de la domotique, de l'agriculture intelligente, des villes intelligentes ou encore dans l'industrie 4.0.

Cependant, cette prolifération des dispositifs IoT s'accompagne de nombreux défis, notamment en matière de sécurité. En effet, la nature hétérogène et distribuée des réseaux IoT, combinée à leurs ressources limitées (puissance de calcul, mémoire, bande passante, etc.), les rend particulièrement vulnérables à diverses attaques. Parmi ces menaces, les intrusions constituent un danger majeur, pouvant compromettre la confidentialité, l'intégrité et la disponibilité des systèmes.

C'est dans ce contexte que la détection d'intrusions dans les réseaux IoT prend toute son importance. Elle vise à identifier de manière proactive toute activité suspecte ou malveillante afin de prévenir d'éventuels dommages. La mise en œuvre de mécanismes de détection efficaces et adaptés aux contraintes spécifiques des environnements IoT représente donc un enjeu critique pour garantir la sécurité de ces réseaux.

Ce travail s'inscrit dans cette problématique. Il a pour objectif d'analyser les approches existantes en matière de détection d'intrusions dans les réseaux IoT, d'en identifier les limites, et de proposer une solution adaptée au contexte des systèmes embarqués et des réseaux contraints.

Pour répondre à cette problématique, ce document est structuré comme suit :

Chapitre 1 : Réseaux IoT
Ce chapitre présente les concepts fondamentaux de l'Internet des objets, son architecture, ses composants, ses protocoles de communication ainsi que ses applications principales.

Chapitre 2 : Détection d'Intrusion (IDS) dans les Réseaux IoT

Il s'agit ici de passer en revue les différentes techniques de détection d'intrusions existantes, en mettant l'accent sur celles adaptées aux environnements IoT : systèmes à base de signatures, systèmes à base d'anomalies, et approches hybrides.

Chapitre 3 : Simulation et Évaluation
Ce chapitre présente soit une étude de cas pratique on utilise logiciel MATLAB, soit une proposition de solution originale pour améliorer la détection des intrusions dans un réseau IoT typique.

Conclusion générale :
Une synthèse des résultats obtenus, des limites rencontrées et des perspectives de recherche futures

Chapiter01 :

Réseaux IoT

1.1 Introduction :

L'Internet des Objets, plus connu sous le nom d'IoT (Internet of Things)**,** représente une avancée majeure dans le domaine des technologies de l'information et des communications. Il s'agit d'un réseau d'objets physiques interconnectés, capables de collecter, de transmettre et parfois de traiter des données via Internet ou d'autres infrastructures de communication. Ces objets peuvent être des capteurs, des actionneurs, des caméras, des dispositifs portables, des équipements industriels, etc.

Dans un réseau IoT, chaque objet est équipé d'un microcontrôleur, d'un module de communication (Wi-Fi, Bluetooth, ZigBee, LoRa, NB-IoT, etc.) et d'un ou plusieurs capteurs ou actionneurs. Cette architecture permet aux objets de communiquer entre eux et avec des plateformes cloud, sans intervention humaine directe.

L'IoT s'impose aujourd'hui comme un enjeu stratégique dans de nombreux secteurs **:** santé connectée, agriculture intelligente, domotique, transports intelligents, villes durables, industrie 4.0**,** etc. Ces systèmes connectés permettent une meilleure prise de décision, une automatisation des processus, une réduction des coûts opérationnels et une amélioration de la qualité de vie.

Cependant, le développement rapide de ces réseaux connectés soulève également de nouveaux défis, notamment en matière de sécurité, d'interopérabilité, de gestion de l'énergie et de traitement de données en temps réel**.** L'hétérogénéité des objets, les protocoles multiples et les contraintes de ressources rendent la conception des réseaux IoT complexe, mais aussi particulièrement stimulante pour les ingénieurs.

Dans ce contexte, il est essentiel de bien comprendre l'architecture des réseaux IoT, leurs composants, les technologies sous-jacentes, ainsi que les problématiques spécifiques liées à leur déploiement.[1][2]

1.2 Aperçu des Réseaux IoT :

1.2.1 Définition et importance de l'Internet des Objets (IoT) :

L'Internet des Objets, ou IoT (Internet of Things), désigne un réseau de dispositifs physiques connectés à Internet capables de collecter, d'échanger et de traiter des données sans intervention humaine directe. Ces objets peuvent être très variés : capteurs industriels, caméras de surveillance, montres

connectées, thermostats intelligents, dispositifs médicaux, ou encore véhicules autonomes. Leur caractéristique commune est d'être dotés d'une capacité de communication, généralement via des technologies sans fil comme le Wi-Fi, le Bluetooth, la 5G, ou encore des protocoles spécifiques bas débit comme LoRaWAN ou Zigbee.[1]

L'importance de l'IoT réside dans sa capacité à transformer notre manière d'interagir avec le monde physique. En connectant des milliards d'objets et en automatisant la collecte d'informations, l'IoT permet d'optimiser les processus, d'améliorer l'efficacité énergétique, de réduire les coûts opérationnels, et de créer de nouveaux services à forte valeur ajoutée. Par exemple, dans l'industrie (Industrie 4.0), l'IoT permet de surveiller en temps réel l'état des machines, d'anticiper les pannes et d'améliorer la production grâce à l'analyse prédictive. Dans le secteur de la santé, il facilite le suivi à distance des patients et permet une réponse médicale plus rapide et mieux ciblée.

Cependant, l'IoT apporte aussi son lot de défis, notamment en matière de sécurité. Chaque objet connecté représente une porte d'entrée potentielle pour les cyberattaques. La grande hétérogénéité des dispositifs, souvent peu sécurisés et difficiles à mettre à jour, rend la tâche encore plus complexe. C'est pourquoi il est crucial de développer des stratégies robustes pour protéger les réseaux IoT et assurer la confidentialité, l'intégrité et la disponibilité des données échangées.[3]

En résumé, l'Internet des Objets est une révolution technologique majeure qui touche tous les aspects de notre vie quotidienne et industrielle. Sa croissance rapide souligne l'importance d'une réflexion approfondie sur les enjeux techniques, économiques et sécuritaires qu'il soulève.

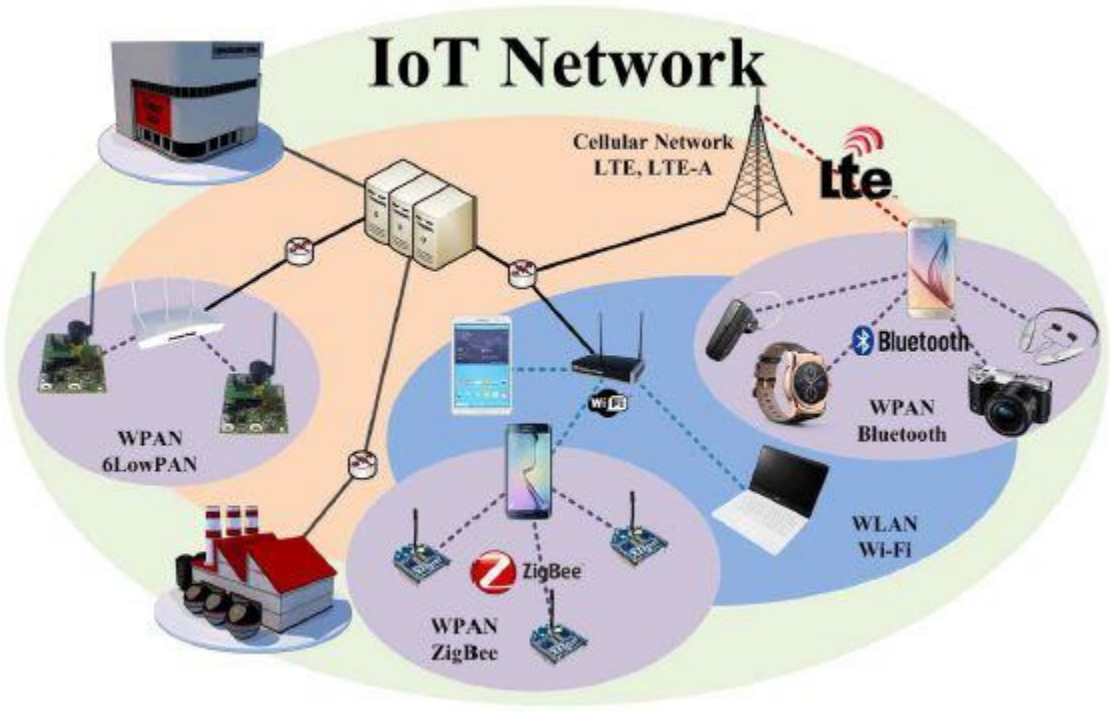


Figure.1.:Réseau d'IoT

1.2.2 Applications des réseaux IoT :

L'IoT est devenu un pilier essentiel de la transformation numérique dans de nombreux secteurs d'activité. Grâce à la capacité des objets connectés à capter, transmettre et analyser des données en temps réel, l'IoT ouvre la voie à une multitude d'applications innovantes. Voici quelques-uns des domaines majeurs où les réseaux IoT jouent un rôle fondamental :[3][4]

1.2.2.1 Villes intelligentes (Smart Cities)

Dans le contexte des villes intelligentes, les réseaux IoT sont déployés pour rendre les espaces urbains plus efficaces, durables et agréables à vivre.Quelques exemples d'applications :

- **Gestion intelligente de l'éclairage public** : Les lampadaires connectés adaptent leur intensité selon la luminosité ambiante ou la présence de piétons, ce qui permet de réduire la consommation énergétique.
- **Surveillance de la qualité de l'air** : Des capteurs déployés à travers la ville mesurent en temps réel les niveaux de pollution pour informer les citoyens et orienter les politiques publiques.
- **Gestion du stationnement** : Des capteurs au sol détectent les places de stationnement disponibles et partagent l'information via des applications mobiles pour réduire le temps de recherche de stationnement.

- **Collecte des déchets intelligente** : Les bennes à ordures connectées signalent leur niveau de remplissage, permettant une optimisation des tournées de ramassage.

L'IoT contribue ainsi à améliorer la qualité de vie, à réduire l'empreinte carbone des villes et à optimiser l'utilisation des ressources urbaines.

1.2.2.2 Santé (e-Santé ou Smart Health) :

Le secteur de la santé bénéficie grandement de l'IoT en améliorant la qualité des soins, en facilitant le suivi des patients et en réduisant les coûts. Quelques applications clés :

- **Télésurveillance médicale** : Des capteurs portables, comme les montres intelligentes ou les implants, mesurent des constantes vitales (fréquence cardiaque, tension artérielle, taux de glucose) et transmettent les données aux professionnels de santé en temps réel.
- **Assistance aux personnes âgées** : Les dispositifs IoT permettent de surveiller l'activité des personnes vulnérables, de détecter des chutes ou des anomalies comportementales, et d'alerter immédiatement les services d'urgence ou les familles.
- **Gestion des équipements hospitaliers** : Les hôpitaux utilisent l'IoT pour localiser les équipements médicaux, optimiser leur maintenance, et assurer leur disponibilité en cas de besoin.

En santé, l'IoT améliore non seulement la réactivité des soins mais contribue aussi à rendre la médecine plus personnalisée et préventive .[5]

1.2.2.3 Transport intelligent :

Les réseaux IoT transforment profondément les systèmes de transport en les rendant plus sûrs, fluides et respectueux de l'environnement. Exemples d'applications :

- **Véhicules connectés** : Les voitures modernes échangent des informations entre elles (V2V - Vehicle-to-Vehicle) ou avec l'infrastructure routière (V2I - Vehicle-to-Infrastructure) pour éviter les collisions, adapter la vitesse aux conditions de circulation et améliorer la sécurité.
- **Gestion du trafic en temps réel** : Les capteurs sur les routes analysent les flux de circulation et ajustent automatiquement les feux de signalisation pour limiter les embouteillages.
- **Suivi logistique** : Dans le transport de marchandises, des balises IoT suivent l'emplacement des colis, surveillent les conditions de transport (température, humidité) et préviennent tout incident.

- **Transports publics optimisés** : Les bus et trains intelligents fournissent des informations en temps réel sur leur position aux voyageurs, améliorant ainsi la planification des trajets.

Grâce à l'IoT, les transports deviennent plus intelligents, plus efficaces et mieux adaptés aux besoins des usagers.

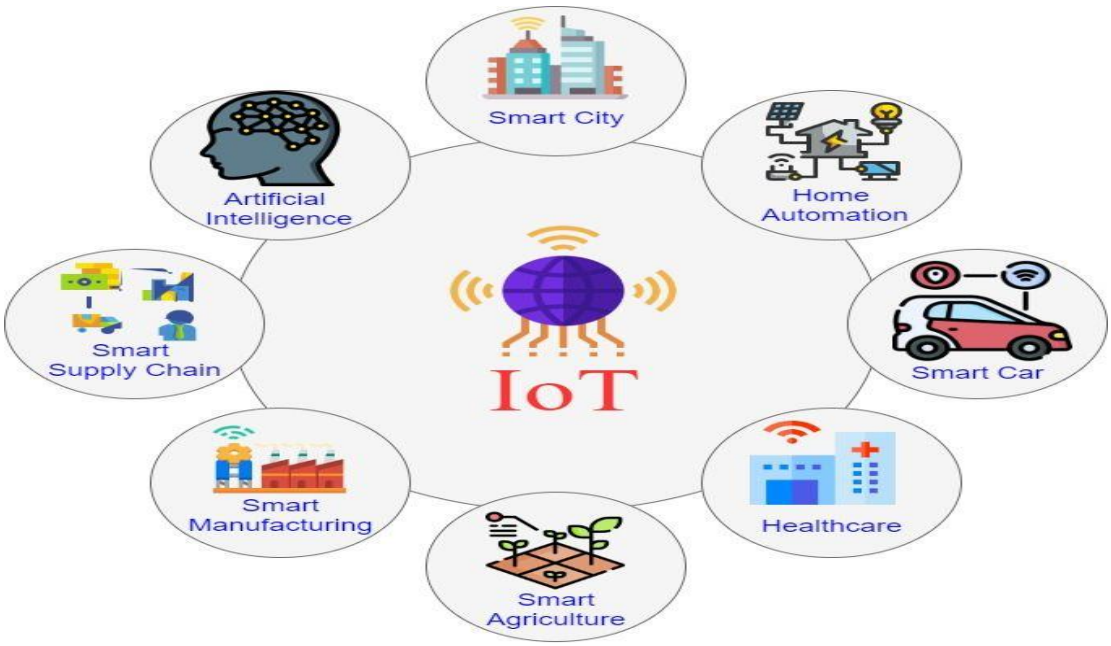


Figure.2 :Applications des réseaux IoT

.

1.3 Architecture des Réseaux IoT :

1.3.1 Les couches des réseaux IoT :

Pour comprendre l'architecture des réseaux IoT, il est essentiel d'analyser leur structure en couches, similaire à celle utilisée dans les réseaux traditionnels (comme le modèle OSI). Cette approche facilite la conception, l'optimisation et la sécurisation des systèmes. En IoT, on distingue principalement trois grandes couches fonctionnelles : la couche de perception, la couche réseau et la couche applicative. Chacune joue un rôle spécifique et crucial dans le cycle de vie de l'information [6][7].

1.3.1.1 Couche de perception (Perception Layer) :

La couche de perception est la première interface entre le monde physique et le monde numérique. Elle est chargée de détecter, capter, collecter et parfois prétraiter les données issues de l'environnement réel.

Fonctions principales : les fonctions principales de la couche perception sont les suivantes :

- Capturer des données physiques comme la température, l'humidité, la pression, la lumière, la vitesse ou la position.
- Identifier et authentifier les objets connectés grâce, par exemple, à des RFID, des codes QR, ou des technologies biométriques.

Dispositifs utilisés dans la couche perception :[7]

les dispositifs utilisés au niveau de la couche perception peuvent être résumé comme suit :

- **Capteurs** : température, mouvement, pression, gaz, …etc.
- **Actionneurs :**éléments mécaniques ou électriques déclenchant une actionphysique(moteur, relais, électrovanne, …etc.)
- **Caméras** : pour la capture d'images ou de vidéos, utilisées notamment dans les systèmes de surveillance ou de vision artificielle.
- **Lecteurs RFID** : pour l'identification automatique d'objets via des étiquettes RFID.
- **Modules GPS** : pour la localisation des objets ou des utilisateurs.
- **Modules NFC, LoRa ou ZigBee** : pour la communication à courte portée.
- **Microphones** : pour la détection de sons ou la commande vocale.
- **Capteurs biométriques** : empreintes digitales, reconnaissance faciale, capteurs de rythme cardiaque, etc.

Défis majeurs :La couche perception joue un rôle critique dans la chaîne d'acquisition de données, car elle constitue le point d'entrée de l'information dans un système intelligent. Les dispositifs à ce niveau sont confrontés à plusieurs contraintes et défis techniques et opérationnelles tel que :

- Minimiser la consommation d'énergie (capteurs souvent sur batterie).
- Assurer l'exactitude et la fiabilité des données collectées.
- Garantir la sécurité physique et logique des dispositifs de perception.
- Maintenir la connectivité dans des environnements hostiles ou à faible couverture réseau.

En résumé, la couche de perception permet aux objets IoT de "sentir" leur environnement.[7][8]

1.3.1.2 Couche réseau (Network Layer) :

La couche réseau assure la transmission des données collectées par la couche de perception vers les plateformes de traitement et d'analyse. Elle est responsable de la connectivité, du routage et parfois du stockage temporaire des données[7].Cette couche permet également d'accéder à la couche de perception via différentes normes et protocoles tels que IEEE 802.x,GPS et NFC[10]. Cependant, les mécanismes de sécurité les plus courants au sein de la couche réseau des architectures IoT incluent la technologie blockchain, les systèmes intelligents de détection d'intrusion et les systèmes de gestion de clés et de chiffrement

Fonctions principales :la couche réseau est une composante essentielle des systèmes IoT, permette l'établissement de communications et l'échange de données entre les éléments du réseau IoT. Elle assure :

- Le transport des données de manière fiable et sécurisée.
- L'Adressage des dispositifs IoT pour assurer leur communication (ex. IPv6 pour l'IoT).
- La Sélection optimale des chemins de transmission dans les réseaux souvent dynamiques et hétérogènes.

Technologies utilisées :Différentstechnologies de communication sont utilisés dans la couche réseaux pour assurer la communication entre tous les objets, qu'ils soient filaires ou sans fil :

- **Protocoles sans fil:** Wi-Fi, Bluetooth Low Energy (BLE), Zigbee, LoRaWAN, NB-IoT, 5G, etc.
- **Protocoles IP spécifiques** : 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks).
- **Réseaux cellulaires et satellites** : dans certains cas pour les objets distants.

Défis majeurs :La couche réseau de l'IoT fait face à de nombreux défis liés à la nature dynamique, distribuée et contrainte des objets connectés à savoir :

- Gérer un très grand nombre de dispositifs simultanément.
- Réduire la latence et la consommation énergétique pendant la transmission.
- Sécuriser les communications contre les interceptions et les attaques.

La couche réseau est donc le "système circulatoire" du réseau IoT, permettant le déplacement de l'information.[9][18]

1.3.1.3 Couche applicative (Application Layer) :

La couche applicative est celle qui interagit directement avec les utilisateurs ou les autres systèmes. Elle traite les données reçues, les analyse, et présente des résultats exploitables sous forme de services ou d'applications.

Fonctions principales :La couche applicative est l'interface entre l'utilisateur final et les fonctionnalités offertes par l'IoT.

- Fournir des services intelligents basés sur l'analyse des données (alertes, recommandations, automatisations).
- Permettre aux utilisateurs de contrôler et de superviser les objets IoT.
- Intégrer l'IoT avec d'autres systèmes d'information (ERP, cloud computing, IA, etc.).

Exemples d'applications :

- Gestion intelligente de l'énergie (Smart Grid).
- Applications de santé connectée (e-santé).
- Solutions de transport intelligent (gestion du trafic, véhicules autonomes).
- Automatisation industrielle (Industrie 4.0).

Défis majeurs : la couche applicative doit relever plusieurs défis techniques et organisationnels pour garantir des services efficaces, sûrs et compatibles avec les standards et réglementations en vigueur :

- Adapter les interfaces aux contraintes de l'IoT (simplicité, latence minimale).
- Garantir la confidentialité et la conformité légale (ex: RGPD pour les données personnelles).
- Assurer l'interopérabilité entre des systèmes souvent très hétérogènes.

La couche applicative donne donc une "intelligence d'usage" aux données brutes et transforme l'information en valeur ajoutée pour les utilisateurs.[11][12]

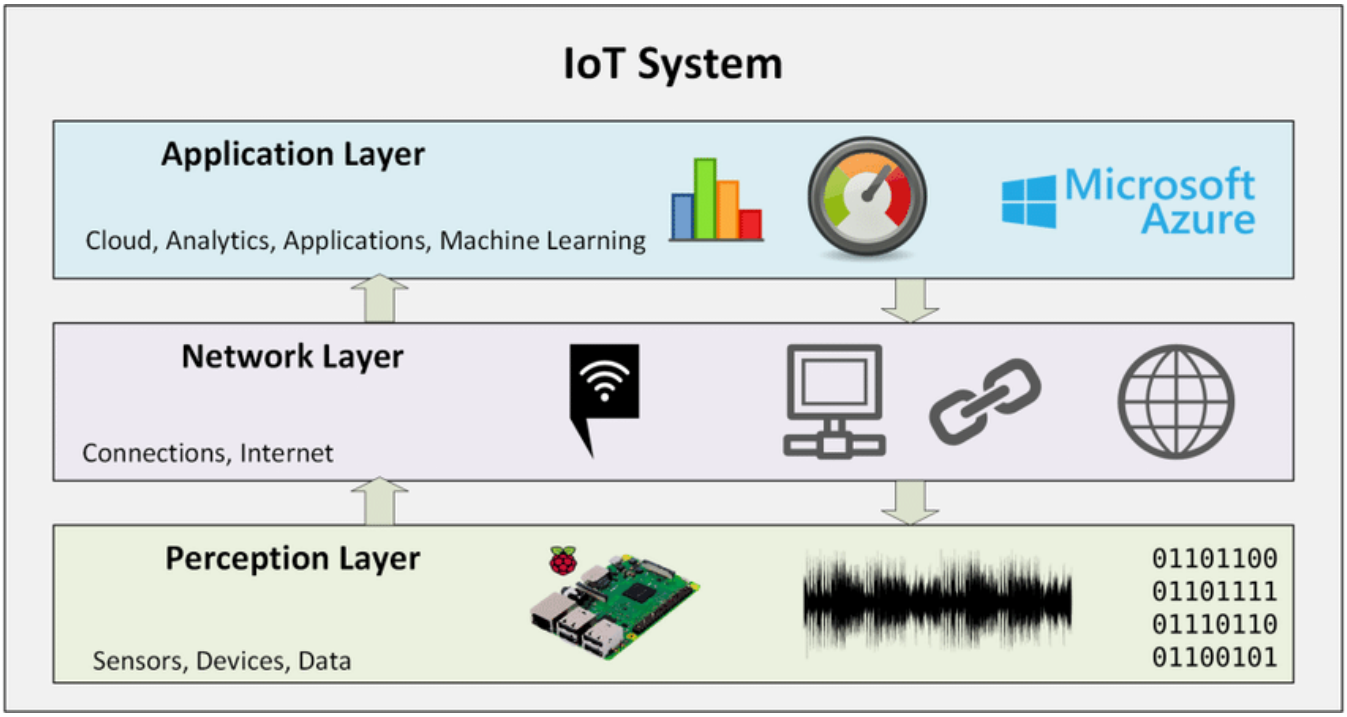


Figure3: les 03 couches des réseaux IOT

.

1.3.2 Topologies courantes des réseaux IoT :

Dans un réseau IoT, la **topologie** désigne la manière dont les dispositifs (capteurs, actionneurs, passerelles, serveurs) sont organisés et interconnectés pour échanger des données. Le choix de la topologie influence fortement la performance, la robustesse, la consommation d'énergie et la latence du réseau. Voici les principales topologies couramment utilisées dans les réseaux IoT :[12]

Topologie en étoile (Star Topology)

Définition : Dans cette configuration, tous les dispositifs IoT (nœuds) communiquent directement avec une unité centrale appelée **passerelle** ou **concentrateur** (gateway ou hub).

Caractéristiques :

- Communication directe entre le nœud et la passerelle.
- La passerelle collecte, traite, et transmet les données vers le serveur ou le cloud.

Avantage :

- Simplicité d'installation et de gestion.

- Faible temps de latence pour les communications.
- Facilité de maintenance.

Inconvénients :

- Dépendance forte à la passerelle : si elle tombe en panne, tout le réseau est paralysé.
- Portée limitée par la distance entre les nœuds et la passerelle.

Utilisation typique :

- Réseaux LoRaWAN.
- Applications de villes intelligentes où les capteurs sont dispersés sur de grandes zones mais doivent communiquer avec un point central.

Topologie en maillage (MeshTopology)

Définition : Dans un réseau maillé, chaque nœud peut communiquer non seulement avec la passerelle, mais aussi avec d'autres nœuds à proximité. Les données peuvent être transmises de nœud en nœud jusqu'à atteindre la destination.

Caractéristiques :

- Les nœuds peuvent agir comme des **relais** pour transmettre les données d'autres nœuds.

Avantages :

- Grande robustesse : si un nœud tombe, les données peuvent être redirigées par d'autres chemins.
- Extension facile de la couverture du réseau.
- Optimisation automatique des routes de communication.

Inconvénients :

- Complexité plus élevée dans la gestion du réseau (routage, synchronisation).
- Consommation d'énergie plus importante pour les nœuds relais.

Utilisation typique :

- Réseaux Zigbee.
- Applications industrielles (Industrie 4.0) et domotiques (bâtiments intelligents).

Topologie en arbre (Tree ou Cluster Topology) :

Définition : La topologie en arbre est une combinaison d'une structure hiérarchique avec des nœuds organisés en "branches" autour d'une racine (souvent la passerelle).

Caractéristiques :

- Les nœuds sont regroupés en **clusters** autour de chefs de groupe (**cluster heads**), qui communiquent avec la passerelle.
- Les chefs de groupe gèrent et agrègent les données des nœuds enfants avant de les transmettre.

Avantages :

- Réduction de la charge de communication sur le réseau global.
- Bonne adaptation aux réseaux à grande échelle.

Inconvénients :

- Si un cluster head tombe, toute la branche associée est affectée.
- Complexité de la gestion des clusters et de l'équilibrage de la charge.

Utilisation typique :

- Réseaux de capteurs sans fil (Wireless Sensor Networks - WSN) dans l'agriculture de précision ou l'environnement.

Topologie point à point (Point-to-Point Topology)

Définition : Dans cette topologie, chaque dispositif IoT communique uniquement avec un autre dispositif spécifique.

Caractéristiques :

- Liaison directe sans intermédiaire entre deux nœuds.

Avantages :

- Communication très fiable et sécurisée.
- Très faible latence.

Inconvénients :

- Non scalable pour un grand nombre de dispositifs.
- Impraticable pour des réseaux complexes.

Utilisation typique :

- Applications critiques nécessitant des communications rapides et sécurisées, comme certaines applications médicales ou industrielles.

Résumé:

Tableau 1.1 : Table récapitulatif des topologies réseau IoT

Topologie	Avantages	Inconvénients	Usage typique
Étoile	Simplicité, faible latence	Dépendance à la passerelle	LoRaWAN, villes intelligentes
Maillage	Robustesse, couverture étendue	Complexité, consommation accrue	Zigbee, Industrie 4.0
Arbre	Scalabilité, hiérarchie structurée	Vulnérabilité des chefs de groupe	Agriculture connectée
Point à Point	Fiabilité, faible latence	Non adapté aux grands réseaux	Santé, industriel critique

1.4 Défis de Sécurité dans les Réseaux IoT :

1.4.1 Vulnérabilités des réseaux IoT :

Avec la croissance rapide des objets connectés dans tous les secteurs — villes intelligentes, santé, industrie, agriculture — les réseaux IoT sont devenus des cibles privilégiées pour les cyberattaques. Leur architecture particulière, leur grande hétérogénéité et souvent leur faible niveau de sécurité intrinsèque les rendent particulièrement vulnérables. Comprendre ces vulnérabilités est essentiel pour concevoir des réseaux plus sûrs et plus résilients[13].

Faible sécurité embarquée sur les dispositifs :

De nombreux objets IoT sont conçus avec des ressources limitées (mémoire, puissance de calcul, batterie). Par conséquent, ils embarquent souvent des mécanismes de sécurité très basiques, voire inexistants.

Problèmes fréquents :

- Absence de chiffrement des données échangées.
- Utilisation de mots de passe par défaut faciles à deviner.
- Impossibilité ou difficulté de mise à jour logicielle (patching).

Exemple : Une caméra de surveillance connectée sans mise à jour de sécurité peut être facilement compromise pour intégrer un réseau de bots (botnet).

Vulnérabilités au niveau des communications :

La transmission des données sur des réseaux sans fil expose les communications à différents types d'attaques.[k][j]

Risques courants :

- **Interception de données** : écoute des transmissions (attaque par sniffing).
- **Attaque de type Man-in-the-Middle** (MITM) : un attaquant intercepte et modifie les communications entre deux objets.
- **Jamming** : brouillage des signaux radio pour perturber la communication.

Conséquence : Fuite d'informations sensibles, perte de contrôle sur les dispositifs.

Problèmes d'authentification et de gestion des accès

Dans de nombreux réseaux IoT, les mécanismes d'authentification et de contrôle des accès sont faibles ou mal implémentés.

Vulnérabilités :

- Accès non autorisé aux dispositifs.
- Détournement d'objets connectés pour exécuter des actions malveillantes.
- Escalade de privilèges dans le réseau.

Exemple : Un simple capteur de température compromis peut servir de porte d'entrée pour attaquer d'autres équipements critiques du réseau.

Vulnérabilités liées à l'architecture réseau

Certaines topologies, comme la topologie en étoile, concentrent toute l'intelligence et la communication via une passerelle centrale. Cela crée un **point de défaillance unique**.

Risques associés :

- Si la passerelle est compromise, tout le réseau est affecté.
- Difficulté de maintenir un service continu lors d'attaques de déni de service (DoS).

Risques liés aux mises à jour et au cycle de vie des objets :

Beaucoup d'objets IoT sont déployés pour de longues périodes (plusieurs années), mais sans mécanismes sûrs et automatiques pour être mis à jour.

Conséquences :

- Objets vulnérables longtemps exposés à des failles connues.
- Accumulation de failles sur des infrastructures critiques.

1.4.2 Principales menaces de sécurité dans les réseaux IoT :

Les réseaux IoT, par leur nature distribuée, leur usage massif de communications sans fil et leur diversité d'objets connectés, sont particulièrement exposés à de nombreuses menaces de sécurité. Parmi les plus préoccupantes, on retrouve l'écoute clandestine, les attaques par rejeu, ainsi que les attaques par DDoS et ransomwares. Comprendre ces menaces est une étape clé pour pouvoir mieux protéger les infrastructures IoT[14]..

1.4.2.1 Écoute clandestine (Eavesdropping) :

Définition : L'écoute clandestine consiste à intercepter discrètement des communications échangées entre les dispositifs IoT sans autorisation.

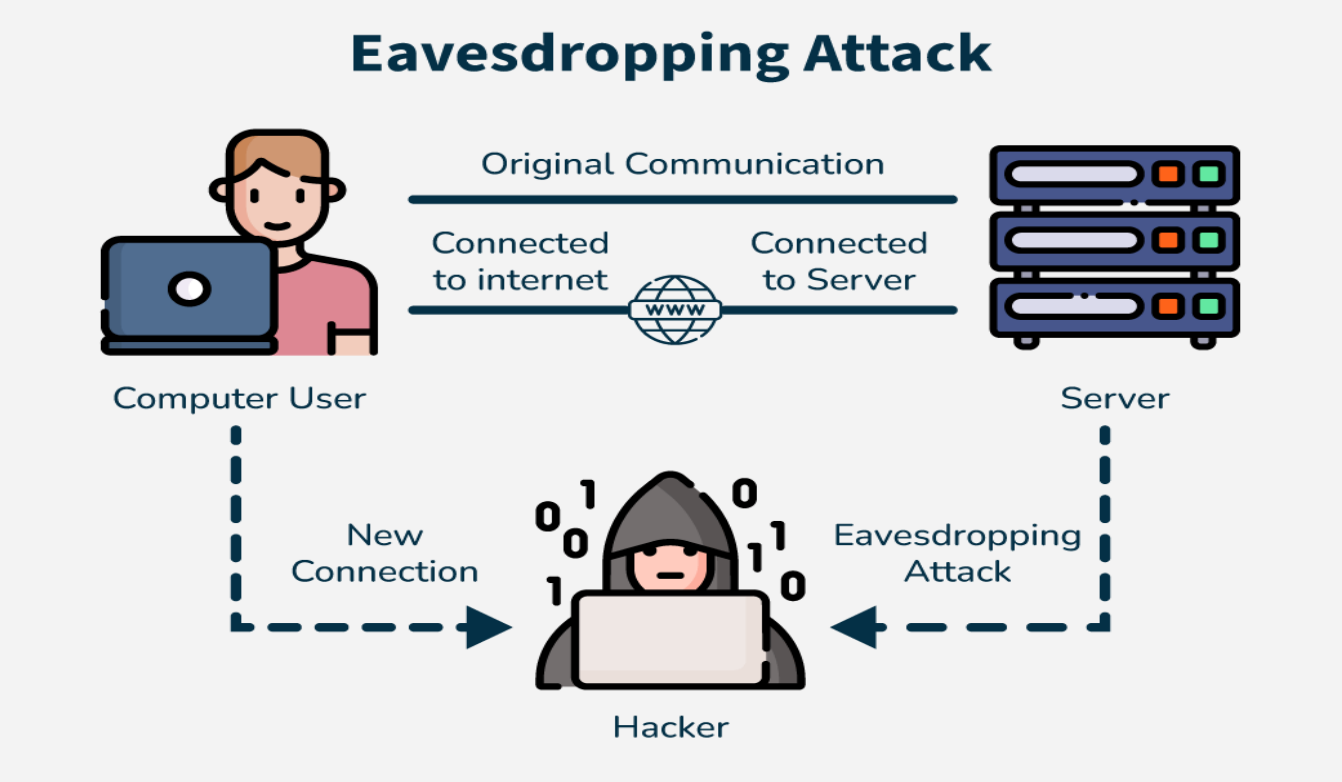


Figure 4:illustration d'une attaque par Écoute clandestine

Fonctionnement :

- Un attaquant se positionne à proximité d'un réseau sans fil (Wi-Fi, Zigbee, LoRaWAN, etc.).
- Il capte les paquets de données en transit pour les analyser ou extraire des informations sensibles (mots de passe, identifiants, données personnelles, etc.).

Conséquences :

- Vol de données confidentielles.
- Compromission de la vie privée des utilisateurs.
- Facilitation d'autres attaques plus ciblées.

Pourquoi c'est un problème en IoT :

- De nombreux objets connectés n'intègrent pas de chiffrement robuste.
- La diversité des protocoles multiplie les vulnérabilités.

1.4.2.2 Attaques par rejeu (Replay Attacks)

Définition :Dans une attaque par rejeu, l'attaquant enregistre des communications légitimes entre deux dispositifs pour ensuite les rejouer à un moment ultérieur afin de tromper le système[15]..

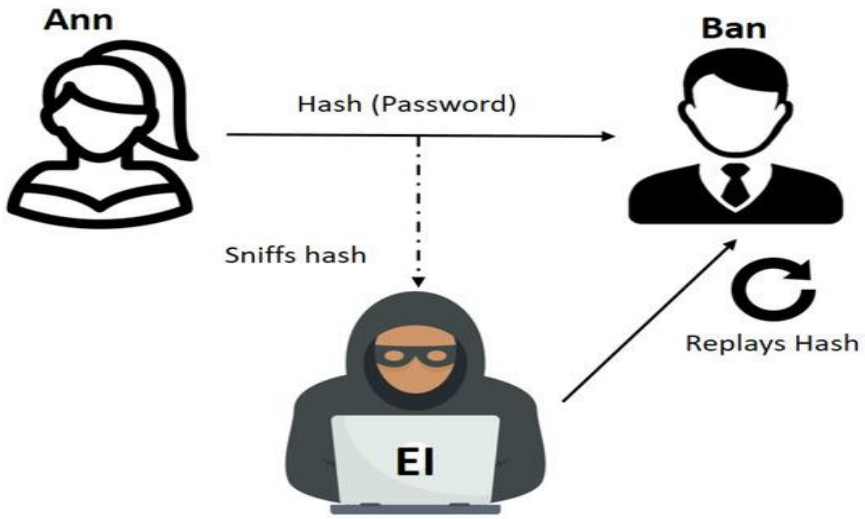


Figure .5:illustration d'une attaquepar rejeu

Fonctionnement :

- L'attaquant capture un message authentique (par exemple, un ordre d'ouverture de porte envoyé à une serrure connectée).
- Il retransmet ce message pour forcer l'action sans autorisation réelle.

Conséquences :

- Prise de contrôle non autorisée sur les dispositifs.
- Usurpation d'identité.
- Perturbation des opérations normales du réseau.

Pourquoi c'est critique en IoT :

- Beaucoup de dispositifs utilisent des communications simples sans mécanisme d'authentification forte ni d'horodatage sécurisé.
- Les systèmes de contrôle d'accès (domotique, alarmes, badges connectés) sont particulièrement vulnérables.

. **Attaques par Déni de Service Distribué (DDoS) et ransomwares :**

1.4.2.3 DDoS (Distributed Denial of Service):[16]. .[17]

Définition : Une attaque DDoS vise à rendre un service ou un réseau indisponible en le submergeant de trafic malveillant provenant de multiples sources.

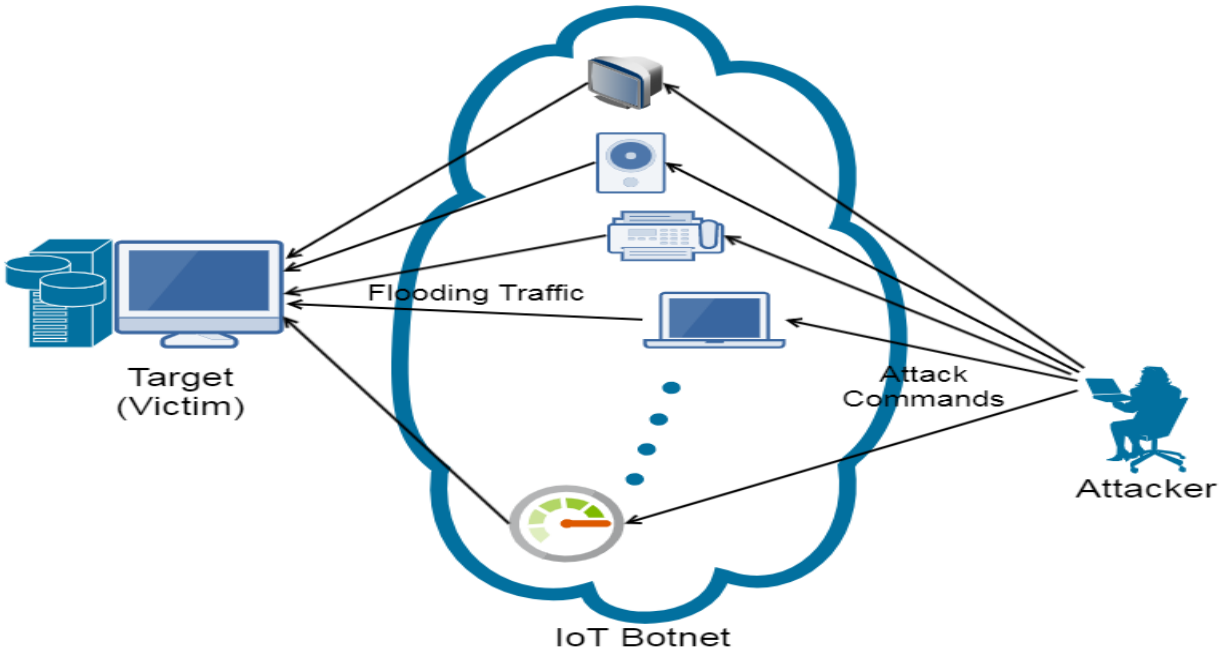


Figure 6:Schéma d'une attaque DDoS [20].

Fonctionnement :

- Des milliers, voire des millions de dispositifs compromis (par exemple, via un botnet IoT) bombardent un serveur ou une passerelle de requêtes simultanées.
- La surcharge entraîne l'interruption ou la dégradation du service.

Conséquences :

- Indisponibilité du service IoT (alarmes qui ne fonctionnent plus, perte de contrôle des équipements industriels, etc.).
- Perte financière pour les opérateurs et perte de confiance des utilisateurs.

Cas célèbre :

- Le botnet Mirai (2016) a utilisé des millions d'objets IoT infectés pour mener une des plus grandes attaques DDoS jamais vues, paralysant des services majeurs comme Twitter, Netflix et Reddit.

Ransomwares :

Définition :Un ransomware est un logiciel malveillant qui chiffre les données d'un dispositif ou bloque son fonctionnement, exigeant une rançon pour restaurer l'accès.

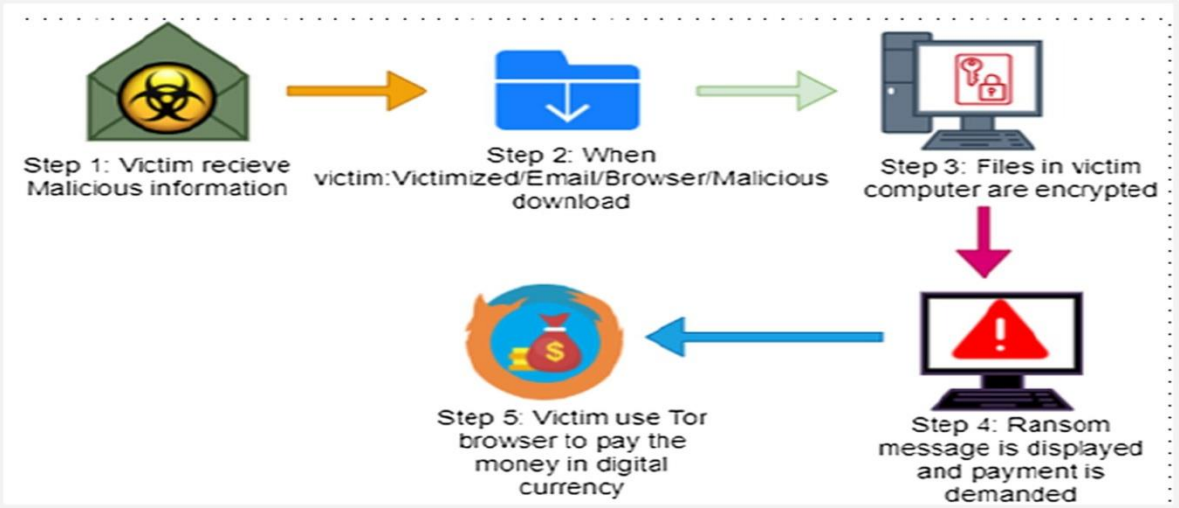


Figure 7 :étapes d'une attaque Ransomwares[21]

Fonctionnement :

- Infection du dispositif IoT via une faille de sécurité ou un accès non protégé.
- Blocage des équipements critiques (caméras de surveillance, capteurs industriels, équipements médicaux).
- Demande de rançon (souvent en cryptomonnaie) pour récupérer l'accès.

Conséquences :

- Interruption d'activités essentielles.
- Perte de données stratégiques ou sensibles.
- Coûts élevés liés au paiement des rançons et à la remise en service.

Pourquoi c'est préoccupant en IoT :

- Beaucoup d'objets connectés n'ont pas de capacités avancées de détection des malwares.

- L'impact est souvent plus grave que dans l'informatique classique, car il touche des systèmes physiques (usines, hôpitaux, villes).

1.5 Conclusion :

À travers cette exploration de l'univers des réseaux IoT, nous avons mis en lumière l'énorme potentiel qu'offrent ces technologies pour transformer nos villes, nos industries, nos transports et nos systèmes de santé. Aujourd'hui, l'Internet des objets permet de connecter le monde physique au monde numérique de manière innovante, ce qui améliore notre qualité de vie, optimise la gestion des ressources et ouvre la porte à de nouveaux services intelligents. Cependant, cette révolution technologique ne se fera pas sans difficultés. L'architecture même des réseaux IoT — leur diversité, leur ouverture, leur dépendance aux communications sans fil — les expose à de nombreuses vulnérabilités. Du manque de sécurité embarquée à l'écoute clandestine, en passant par les attaques par rejeu, les DDoS massifs ou encore les ransomwares, les menaces sont multiples et évolutives.Il est donc essentiel de mettre en place des mécanismes de protection adaptés aux spécificités des environnements IoT.Dans le chapitre suivant, nous aborderons la problématique de la Détection d'Intrusion (IDS) dans les Réseaux IoT.

Chapiter02 :

Détection Intrusion (IDS) dans les Réseaux IoT.

1.6 Introduction :

Avec l'essor rapide de l'Internet des Objets (IoT), des milliards d'appareils intelligents sont désormais interconnectés, allant des capteurs environnementaux aux dispositifs médicaux, en passant par les systèmes domotiques et industriels. Cette prolifération a amélioré l'efficacité des systèmes automatisés, mais elle a aussi introduit de nouvelles vulnérabilités. En effet, la nature hétérogène et souvent limitée en ressources des dispositifs IoT rend leur sécurisation complexe.

Parmi les mécanismes de défense adoptés, les systèmes de détection d'intrusion (IDS) jouent un rôle clé. Dans le contexte de l'IoT, un IDS efficace doit fonctionner dans des environnements contraints tout en conservant un haut niveau de précision.

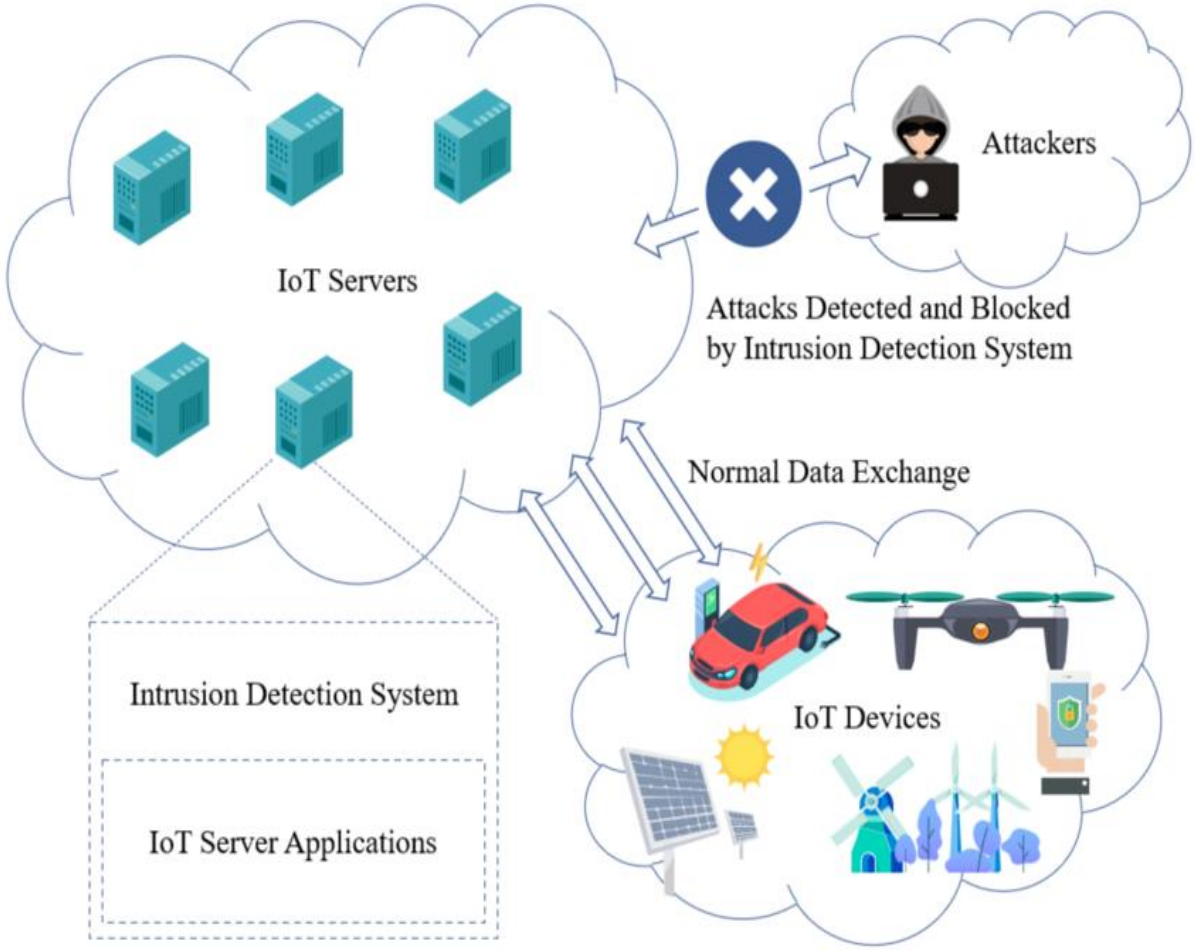
Ce chapitre s'inscrit donc dans le cadre de la mise en œuvre et de l'évaluation de méthodes de détection d'intrusions adaptées aux architectures IoT, avec un accent particulier sur les approches intelligentes comme l'apprentissage automatique, qui permettent d'anticiper les attaques complexes de manière autonome et évolutive.

1.7 Aperçu sur l' IDS :

1.7.1 Définition et rôle des IDS dans la sécurité réseau

Les Systèmes de Détection d'Intrusion (IDS – *Intrusion DetectionSystems*) sont des dispositifs ou logiciels conçus pour surveiller en temps réel l'activité d'un réseau ou d'un système, dans le but de détecter toute tentative d'accès non autorisé, de modification de données, ou de comportement anormal pouvant indiquer une attaque. Contrairement aux pare-feux, qui agissent de manière préventive en filtrant les flux entrants et sortants selon des règles prédéfinies, les IDS ont une fonction de détection et d'analyse. Ils n'interviennent généralement pas pour bloquer une attaque, mais ils permettent d'alerter l'administrateur ou d'enregistrer l'événement pour une réaction appropriée.

Dans les architectures réseau modernes – et plus particulièrement dans les systèmes embarqués connectés à l'IoT – la complexité et la diversité des équipements rendent difficile la surveillance continue par des méthodes classiques. Les IDS jouent donc un rôle central en tant que première ligne de défense secondaire, capable d'identifier des attaques qui échappent aux protections traditionnelles. Ils peuvent détecter divers types de menaces, telles que les attaques par déni de service (DoS), les intrusions internes, les communications anormales entre dispositifs, ou encore les tentatives de falsification de données.[22][23]



.

Figure 8:l'IDS appliqué dans le réseau IoT.

D'un point de vue ingénierie télécom, l'implémentation d'un IDS dans un réseau repose sur des considérations techniques précises : compatibilité avec les protocoles de communication, impact sur la latence, consommation énergétique minimale, et capacité à traiter des volumes importants de données en temps réel. Plusieurs approches existent, notamment les IDS basés sur les signatures (qui comparent le trafic à une base d'attaques connues) et ceux basés sur le comportement (qui identifient les écarts par rapport à un fonctionnement normal). De plus en plus, des techniques hybrides ou basées sur l'intelligence artificielle sont adoptées pour améliorer la détection d'intrusions dans des environnements IoT complexes.[23]

1.7.2 Types de Systèmes de Détection d'Intrusion (IDS) :

Dans le domaine de la cybersécurité des réseaux, notamment dans les environnements contraints comme ceux de l'Internet des Objets (IoT), il existe plusieurs types d'IDS, chacun ayant des caractéristiques spécifiques en matière de fonctionnement, d'efficacité et de complexité de déploiement. Voici les trois grandes catégories d'IDS généralement utilisées :

1.7.2.1 IDS basés sur les signatures (Signature-Based IDS) :

Ce type d'IDS fonctionne à partir d'une base de données contenant les empreintes ou signatures connues d'attaques précédentes. Le système analyse le trafic réseau ou les journaux d'activité, et compare chaque événement observé aux signatures enregistrées. Dès qu'une correspondance est détectée, une alerte est générée.[24]

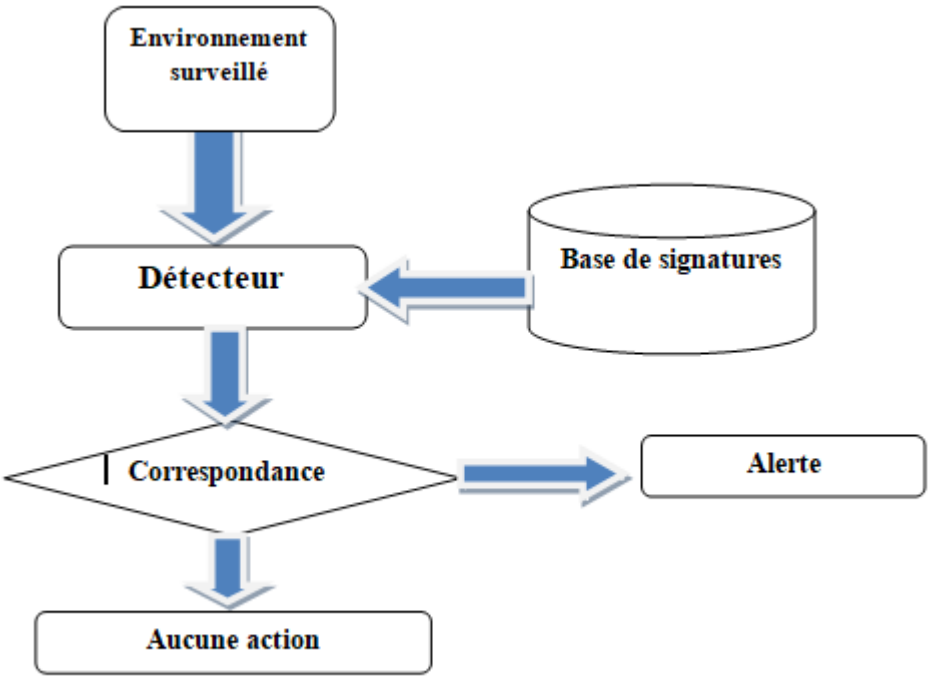


Figure 9:Architecture IDS basée sur la signature

Avantages : Très efficace pour détecter les attaques connues ; faible taux de fausses alertes.

Limites : Incapable de détecter des attaques nouvelles ou inconnues (zero-day) ; nécessite une mise à jour régulière de la base de signatures.

1.7.2.2 . IDS basés sur les anomalies (Anomaly-Based IDS) :[24]

Ce type d'IDS établit un modèle de comportement « normal » du système ou du réseau. Toute activité déviant significativement de ce modèle est considérée comme suspecte. Ces systèmes utilisent souvent des techniques statistiques ou d'apprentissage automatique pour créer et ajuster leur modèle de référence.

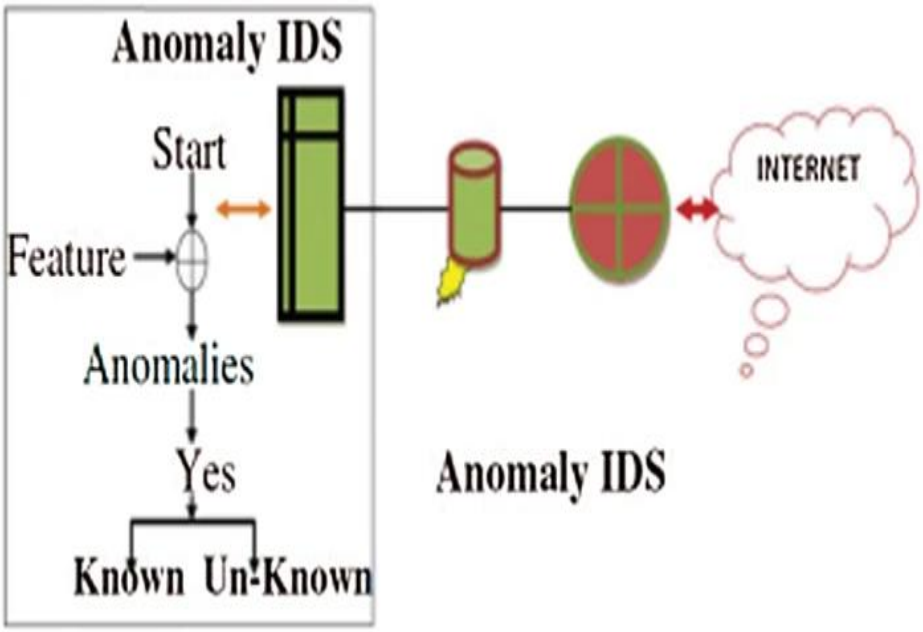


Figure 10:Architecture IDS basée sur l'anomalie

- **Avantages** : Capables de détecter des menaces nouvelles ou inconnues ; adaptabilité à des environnements dynamiques.
- **Limites** : Taux de fausses alertes potentiellement élevé, surtout si le modèle est mal entraîné ; complexité de mise en œuvre.

1.7.2.3 . IDS hybrides :

Les IDS hybrides combinent les approches par signature et par anomalie, dans le but de profiter des avantages des deux méthodes. Ils assurent une couverture plus complète en détectant à la fois les attaques connues et les comportements suspects inconnus.[24][25]

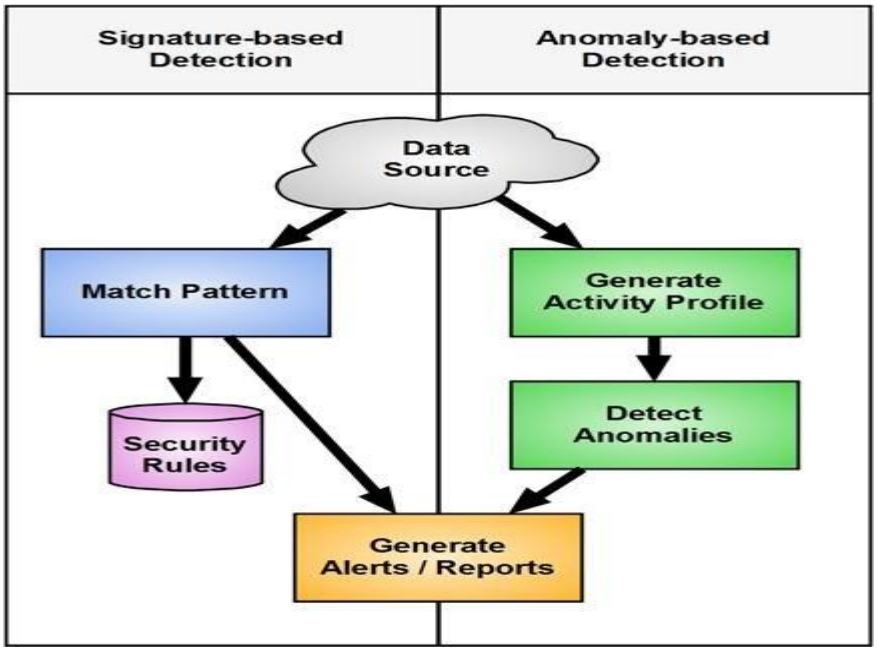


Figure 11:Architecture IDS hybride

- **Avantages** : Meilleure précision globale ; réduction du taux de fausses alertes ; capacité à détecter des attaques variées.
- **Limites** : Plus gourmands en ressources ; complexité accrue en matière de conception et de maintenance.

1.8 Les défis des IDS dans l'IoT :

L'intégration de systèmes de détection d'intrusion (IDS) dans les environnements IoT soulève plusieurs défis techniques majeurs. En tant qu'ingénieur en électronique et télécommunications, il est essentiel de prendre en compte les spécificités de ces réseaux, où les nœuds sont souvent limités en puissance de calcul, en mémoire et en autonomie énergétique. Trois enjeux clés se dégagent dans le déploiement efficace des IDS au sein de ces architectures distribuées :

1.8.1 Scalabilité (Évolutivité) :

Les réseaux IoT peuvent inclure des centaines, voire des milliers de dispositifs interconnectés, répartis sur de vastes zones géographiques. Concevoir un IDS capable de s'adapter à cette échelle constitue un véritable défi. Un système non scalable devient rapidement inefficace à mesure que le nombre de nœuds augmente. Il faut donc concevoir des architectures distribuées et modulaires, capables de traiter des volumes croissants de données tout en conservant des performances stables. Cela implique également

une gestion intelligente des mises à jour, de la synchronisation des données et de la communication entre IDS locaux ou hiérarchisés.

1.8.2 . Détection en temps réel :

La capacité à détecter une intrusion en temps réel est cruciale pour limiter l'impact d'une attaque. Dans le contexte de l'IoT, cette exigence se heurte aux limitations des canaux de communication (débits faibles, latence élevée) et à la variabilité des flux de données. Les algorithmes de détection doivent être suffisamment rapides et optimisés pour fonctionner en continu, sans engendrer de surcharge réseau ni compromettre la réactivité du système. L'équilibre entre rapidité de détection et précision du diagnostic est ici primordial.

1.8.3 Contraintes de ressources des dispositifs IoT. :

Les dispositifs IoT, notamment les capteurs et actionneurs embarqués, sont conçus pour être compacts, économes en énergie et peu coûteux. Ils disposent donc de capacités très limitées en termes de traitement, de mémoire et d'autonomie énergétique. Intégrer des IDS dans de tels environnements nécessite des algorithmes légers, souvent basés sur des modèles simplifiés ou décentralisés, capables de fonctionner localement sans dépendre excessivement du cloud ou d'une unité centrale. De plus, le choix des protocoles de communication sécurisés doit être compatible avec ces contraintes.

1.9 Introduction au Machine Learning (ML) et au Deep Learning (DL) :

Dans le monde numérique actuel, marqué par l'explosion des données et la puissance croissante des systèmes de calcul, les technologies comme le Machine Learning (ML) et le Deep Learning (DL) occupent une place centrale dans de nombreuses applications, allant de la reconnaissance vocale aux systèmes de conduite autonome.Le Machine Learning, ou apprentissage automatique, est une branche de l'intelligence artificielle qui permet à une machine d'apprendre à partir de données, sans être explicitement programmée pour chaque tâche. Le principe de base est simple : on fournit à un algorithme un grand nombre d'exemples (des données d'entrée avec leurs résultats attendus), et il apprend à établir des modèles ou des relations qu'il pourra ensuite généraliser à de nouveaux cas.[26]

Le Deep Learning, ou apprentissage profond, est une sous-catégorie du ML qui repose sur des réseaux de neurones artificiels composés de plusieurs couches (d'où le terme "profond"). Ces architectures complexes sont particulièrement efficaces pour traiter de grandes quantités de données non structurées comme des images, des vidéos, du son ou du texte. C'est grâce au DL, par exemple, que les

assistants vocaux peuvent comprendre nos requêtes ou que les systèmes de vision artificielle reconnaissent des objets dans une image.

1.9.1 Machine learning :

1.9.1.1 Définition :

La Machine Learning, également connue sous le terme d'apprentissage automatique, est un domaine d'intelligence artificielle qui consiste à permettre à une machine d'apprendre d'exemples sans y être programmée explicitement au préalable. Il se base sur des algorithmes qui peuvent analyser des données, en extraire des modèles, puis faire des prévisions ou prendre des décisions sur la base de données nouvelles. Il s'applique avec succès dans tous types d'applications telles que la reconnaissance vocale, la détection de fraude ou la suggestion d'informations.

1.9.1.2 Type de Machine learning :

1.9.1.2.1 . Apprentissage supervisé (Supervised Learning)

C’est la forme la plus courante de machine learning. Dans ce type d’apprentissage, l’algorithme est entraîné à partir d’un jeu de données **étiquetées**, c’est-à-dire que chaque exemple d’entrée est associé à une sortie correcte (une classe ou une valeur à prédire). L’objectif est que le modèle apprenne à généraliser les relations entre les données d’entrée et les sorties correspondantes pour pouvoir faire des prédictions sur de nouvelles données non vues.

Exemples d’application :

- Classification d’e-mails en spam ou non spam
- Reconnaissance d’objets dans des images
- Prédiction de la consommation d’énergie ou du prix d’un bien immobilier

1.9.1.2.2 . Apprentissage non supervisé (Unsupervised Learning)

Contrairement à l’apprentissage supervisé, ici les données d’entraînement **ne possèdent pas d’étiquettes**. L’algorithme explore les données pour y détecter des structures cachées ou des relations naturelles. Le but est généralement de **regrouper**, **réduire la dimensionnalité** ou **détecter des anomalies** dans les données.

Exemples d’application :

- Regroupement de clients par comportement d'achat (clustering)
- Compression d'images
- Détection de comportements anormaux dans un réseau

1.9.1.2.3 Apprentissage par renforcement (Reinforcement Learning) :

Dans ce type d'apprentissage, l'algorithme (appelé agent) apprend en interagissant avec un environnement**.** Il prend des décisions, reçoit des récompenses ou punitions selon la qualité de ses actions, et ajuste sa stratégie (sa politique) pour maximiser la récompense cumulée au fil du temps. C'est un apprentissage basé sur l'essai-erreur, souvent utilisé dans des contextes dynamiques ou séquentiels.

Exemples d'application :

- Jeux vidéo ou échecs (AlphaGo de Google)
- Robots mobiles qui apprennent à se déplacer
- Optimisation du trafic réseau ou des systèmes industriels autonomes.

Le tableau2.1 donne une comparaisondes trois principaux types d'apprentissage en Machine Learning

Tableau 2.1:Comparaison entre 03 types de la ML

Type d'apprentissage	Données d'entrée	But principal	Exemples d'applications
Supervisé	Données étiquetées (entrée + sortie connue)	Apprendre une fonction de prédiction	Classification d'e-mails, prédiction de prix, OCR
Non supervisé	Données non étiquetées	Trouver des structures ou des regroupements	Segmentation client, détection d'anomalies, clustering
Par renforcement	Environnement interactif + récompenses	Apprendre une stratégie d'action optimale	Jeux, robots autonomes, gestion de ressources réseau

1.9.1.3 Algorithmes pour classification des intrusions:

1.9.1.3.1 1. SVM (Support Vector Machine)

Le SVM est un algorithme supervisé utilisé pour la **classification binaire**, très populaire dans la détection d'intrusions. Il cherche à séparer les données en classes (ex. : trafic normal vs intrusion) en traçant une **hyperplane optimal** dans un espace multidimensionnel.

- **Avantages :** Très efficace pour les jeux de données avec peu de bruit et lorsque les classes sont bien séparables.
- **Inconvénients :** Moins performant avec des données très volumineuses ou non linéaires, sauf en utilisant des noyaux (*kernel trick*).

Application IDS : Détection d'anomalies dans le trafic réseau, classification des types d'attaques (DoS, probe, etc.).

2. Random Forest :

C'est un algorithme d'apprentissage supervisé basé sur un ensemble d'arbres de décision. Chaque arbre prend une décision, et la forêt "vote" pour donner une classification finale.

- **Avantages :** Robuste, gère bien les données bruitées et les grands jeux de données. Très bon pour l'interprétation.
- **Inconvénients :** Peut devenir complexe et coûteux en ressources si la forêt est trop grande.

Application IDS : Utilisé pour distinguer les attaques du trafic normal avec une grande précision, en analysant plusieurs variables à la fois.

3. K-Means

K-Means est un algorithme **non supervisé** de **clustering** qui regroupe les données en *k* groupes selon leur similarité. Il n'a pas besoin d'étiquettes dans les données.

- **Avantages :** Simple, rapide, efficace sur des données bien séparées.
- **Inconvénients :** Doit connaître à l'avance le nombre de clusters, sensible aux valeurs aberrantes.

Application IDS : Utilisé pour découvrir automatiquement des comportements anormaux dans le réseau sans avoir besoin de données étiquetées.

4. K-Nearest Neighbors (KNN)

KNN est un algorithme supervisé qui classe une donnée en fonction des **k points les plus proches** dans l'espace de caractéristiques.

- **Avantages :** Simple à implémenter, bon pour les petites bases.
- **Inconvénients :** Lourd en calcul pour de grandes bases de données.

Application IDS : Identifier les comportements anormaux en comparant avec les données historiques connues.

5. Réseaux de Neurones Artificiels (ANN)

Les ANN sont des algorithmes inspirés du cerveau humain. Ils sont capables de détecter des relations complexes dans les données.

- **Avantages :** Très bon pour les problèmes non linéaires complexes.
- **Inconvénients :** Nécessite beaucoup de données et de puissance de calcul.

Application IDS : Détection d'intrusions multi-classes, détection de nouveaux types d'attaques.

6. Deep Learning (ex. : LSTM, CNN)

Les réseaux de neurones profonds, comme les **LSTM** (Long Short-Term Memory) ou **CNN** (Convolutional Neural Networks), sont très puissants pour traiter des séquences temporelles (trafic réseau dans le temps) ou des flux complexes.

- **Avantages :** Capacité d'apprentissage très élevée, détection automatique des caractéristiques.
- **Inconvénients :** Très gourmands en ressources et nécessitent beaucoup de données.

Applications IDS : détection de comportements malveillants dans des flux des données continus

Un tableau comparatif (tableau 2.2) synthétique des algorithmes courants utilisés dans les IDS, selon leurs principales caractéristiques et leur usage en détection d'intrusion :

Tableau2.2 :Comparaison entre les algorithmes utilisés dans l'IDS.

Algorithme	Type d'apprentissage	Avantages clés	Inconvénients	Utilisation en IDS
SVM (Support Vector Machine)	Supervisé	Haute précision, efficace pour séparation nette des classes	Moins adapté aux grands jeux de données non linéaires	Classification des attaques (ex. DoS, probe)
Random Forest	Supervisé	Robuste, gère bruit et variables multiples	Complexité élevée avec grand nombre d'arbres	Détection multi-classes, filtrage intelligent
K-Means	Non supervisé	Rapide, simple, efficace pour clustering	Sensible aux anomalies, nécessite k défini	Clustering de comportements réseau
KNN (K-Nearest Neighbors)	Supervisé	Facile à implémenter, pas besoin d'entraînement complexe	Coûteux en calcul, sensible au choix de k	Détection par similarité avec comportements connus
Réseaux de Neurones (ANN)	Supervisé	Capte les relations complexes, bonne généralisation	Besoin de beaucoup de données, ajustement délicat	Détection d'intrusions complexes
Deep Learning (LSTM, CNN)	Supervisé	Excellente capacité d'apprentissage, détecte patterns fins	Très gourmand en ressources, besoin massif de données	Analyse de trafic temps réel, détection avancée

1.9.2 Deep Learning :

Définition : Le Deep Learning, ou apprentissage profond, est un sous-domaine avancé du Machine Learning fondé sur des réseaux de neurones artificiels multi-couches. Ces réseaux possèdent la capacité à manipuler d'immenses quantités de données non structurées telles que des images, du son ou du texte, et à extraire automatiquement des renseignements sans intervention humaine à leur niveau donné. Ainsi, grâce à cette aptitude, le Deep Learning est désormais à l'origine de nombreuses avancées dans des domaines comme la reconnaissance vocale, la vision par ordinateur, ou encore les véhicules autonomes.[27]

1.9.2.1 Avantages du DL pour les IDS dans l'IoT.

Le Deep Learning (DL) a beaucoup à offrir aux systèmes de détection d'intrusion (IDS) dans les environnements IoT (Internet des objets). Grâce à son aptitude à analyser des masses gigantesques de

données non structurées et à apprendre automatiquement des propriétés subtiles, le DL est capable d'identifier des activités anormales ou malveillantes, même si elles ne correspondent pas à des modèles d'attaques connus.

Cela est particulièrement utile dans les réseaux IoT, où les appareils sont nombreux, hétérogènes et souvent vulnérables. En intégrant le Deep Learning dans les IDS pour l'IoT, on peut :

- Améliorer la détection en temps réel grâce à l'analyse continue des flux de données,
- Réduire les faux positifs, car le modèle apprend à mieux distinguer entre activité normale et réelle menace,
- Adapter dynamiquement la sécurité face aux menaces nouvelles ou en évolution,
- Et garantir une meilleure scalabilité du système de sécurité, capable de s'étendre à des milliers de nœuds IoT.

1.9.2.2 Les Architectures courantes :

1.9.2.2.1 1. RNN (Recurrent Neural Network) – Pour l'analyse de séries temporelles

Les RNN sont des réseaux de neurones spécialement conçus pour traiter des données séquentielles, comme des séries temporelles. Contrairement aux réseaux classiques, les RNN intègrent une mémoire interne qui leur permet de prendre en compte les dépendances temporelles entre les données. Cela les rend particulièrement utiles dans les IDS pour analyser des flux de données réseau continus, comme les logs de connexion, les paquets réseau ou le comportement d'un appareil IoT sur une période donnée.

Exemple typique : Détecter une attaque lente et progressive (comme un scan réseau ou une attaque de type slowloris), en observant une séquence d'événements au lieu d'un événement isolé.

Cependant, les RNN classiques ont des limitations lorsqu'il s'agit de mémoriser des relations à long terme. C'est pourquoi on utilise souvent des variantes comme les LSTM (Long Short-Term Memory) ou les GRU (GatedRecurrent Unit) dans ce type de tâches.

1.9.2.2.2 2. CNN (Convolutional Neural Network) – Pour l'analyse de données spatiales

Bien connus pour leur rôle en vision par ordinateur, les CNN sont également très performants dans les IDS, surtout lorsqu'on transforme les données réseau ou les logs système en représentations matricielles (comme des images ou des "heatmaps" d'activité).

Les CNN sont capables de capturer des motifs locaux (patterns) à travers des opérations de convolution, ce qui permet de repérer des comportements suspects, même dans des structures de données complexes ou multidimensionnelles. On les utilise souvent pour :

- Analyser la distribution spatiale des paquets réseau,
- Détecter des patterns fixes associés à des intrusions.

Exemple typique : Identifier une signature d'attaque à partir d'un profil de trafic visualisé sous forme de matrice.

Les CNN sont aussi efficaces sur des systèmes embarqués, car ils peuvent être optimisés en termes de calcul et de mémoire.

1.9.2.2.3 3. Autoencodeurs – Pour la détection d'anomalies

Les autoencodeurs sont des réseaux neuronaux non supervisés conçus pour apprendre une représentation compacte (encodée) des données. Ils fonctionnent en deux phases :

- Une phase d'encodage qui réduit la dimensionnalité des données,
- Une phase de décodage qui tente de reconstruire les données d'origine à partir de cette version compressée.

Dans un IDS, un autoencodeur peut être entraîné uniquement sur des données normales. Ensuite, lorsqu'une anomalie (comme un comportement malveillant) est détectée, le modèle n'arrive pas à bien la reconstruire, ce qui génère une erreur de reconstruction élevée. Cette erreur peut alors être utilisée comme signal d'alerte.

Exemple typique : Détection d'un comportement réseau anormal sur un capteur IoT qui commence à communiquer de manière inhabituelle avec un serveur inconnu.

Les autoencodeurs sont puissants pour la détection de menaces nouvelles ou non étiquetées, ce qui est un grand avantage dans des environnements IoT souvent dynamiques et imprévisibles.

1.10 Les IDS dans l'IoT avec le ML et le DL :

Avec la croissance massive des dispositifs connectés dans l'Internet des Objets (IoT), les enjeux de sécurité sont devenus critiques. Les IDS (Intrusion DetectionSystems) jouent un rôle central dans la

surveillance de ces réseaux, mais leur conception évolue. Aujourd'hui, on distingue principalement trois grandes approches : les IDS traditionnels, les IDS intégrant le ML/DL pour l'IoT, et les IDS génériques basés sur ML/DL. La figure 2.4 montre le principe d'un système IDS basé sur ML/DL.[28]

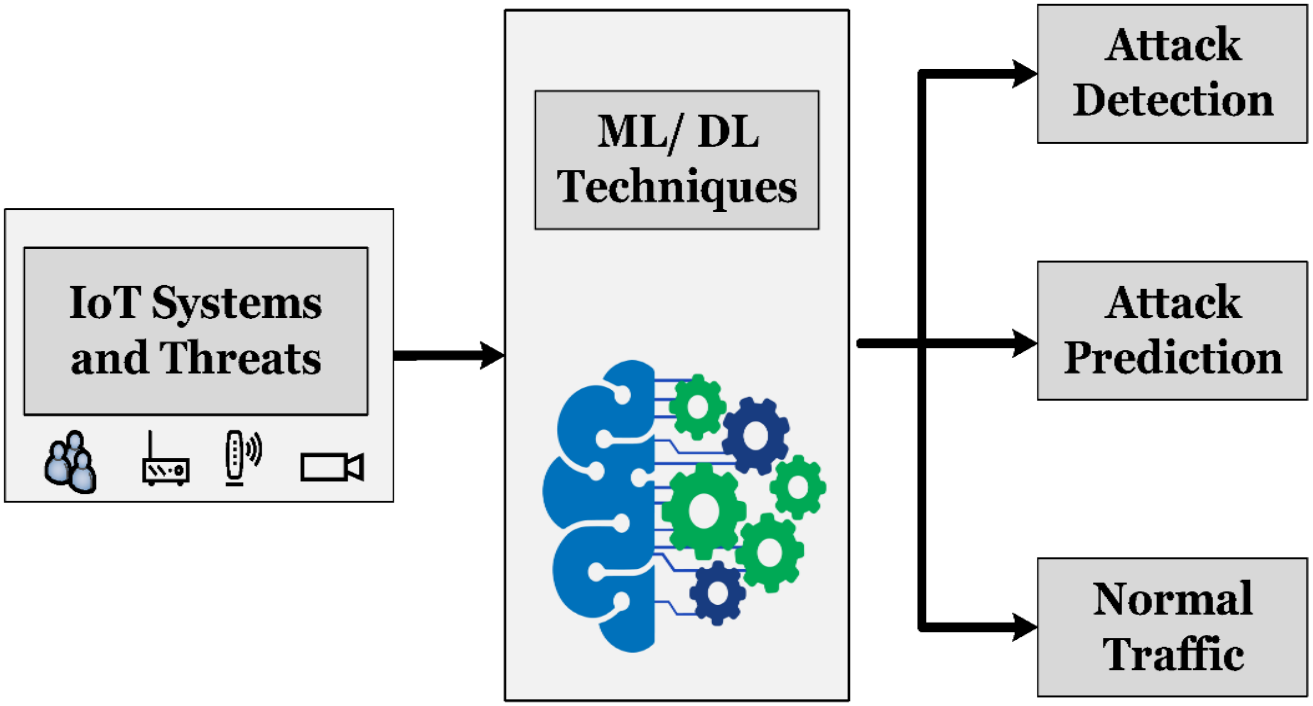


Figure 12:IDS basés sur ML/DL

1. **IDS traditionnels**

Les IDS classiques s'appuient généralement sur des méthodes statiques telles que :

- La détection par signature, qui compare le trafic à une base d'attaques connues.
- La détection par règles, qui repose sur des seuils ou comportements définis manuellement.

Limites : Ces systèmes sont efficaces pour les attaques connues, mais montrent vite leurs limites face aux attaques nouvelles ou polymorphes, surtout dans les réseaux IoT, où la diversité des appareils rend difficile l'application d'un modèle unique.

1.10.1 2. IDS basés sur le ML/DL pour l'IoT

Dans les environnements IoT, les IDS modernes intègrent le Machine Learning et le Deep Learning pour s'adapter automatiquement à la nature dynamique du trafic. Ces systèmes peuvent :

- Apprendre des comportements normaux d'un objet connecté,
- Détecter des écarts subtils, souvent invisibles pour les méthodes classiques,
- S'adapter aux évolutions du réseau sans reconfiguration manuelle.

Exemples :

- Utilisation de Random Forest ou SVM pour classifier des paquets comme normaux ou malveillants.
- Déploiement de RNN/LSTM pour analyser le trafic temporel d'un capteur industriel.
- Autoencodeurs pour détecter automatiquement les anomalies dans les données issues de capteurs.

Avantages : Précision accrue, détection de menaces zero-day, adaptation automatique, réduction des faux positifs.

1.10.2 3.IDS basés sur ML/DL (générique)

Cette catégorie regroupe les systèmes IDS intelligents mais qui ne sont pas conçus spécifiquement pour l'IoT. Ils s'appliquent souvent à des infrastructures IT classiques (serveurs, PC, réseaux d'entreprise) avec :

- Des algorithmes ML/DL puissants,
- Une capacité de traitement élevée,
- Une forte dépendance à des jeux de données volumineux et étiquetés.

Limite : Dans l'IoT, où les ressources (CPU, mémoire, batterie) sont limitées, ces systèmes peuvent être inadaptés sans optimisation spécifique (réduction de modèle, calcul en périphérie, etc.)[29]

1.11 Proposition d'un Cadre d'IDS pour l'IoT :

L'objectif d'un cadre IDS dans le contexte IoT est de garantir une surveillance intelligente et efficace du trafic, en tenant compte des contraintes spécifiques des objets connectés : ressources limitées, hétérogénéité des protocoles, et nature dynamique des flux. Voici une approche structurée et détaillée :

1.11.1 Caractéristiques à surveiller :

Pour qu'un IDS soit pertinent et performant dans un environnement IoT, il doit être capable d'analyser plusieurs caractéristiques réseau en temps réel. Les principales à prendre en compte sont :

1.11.2 Taille des paquets (Packet Size) :

La taille des paquets est un indicateur fondamental dans la détection d'anomalies. En IoT, les dispositifs envoient souvent des paquets courts et réguliers. Une variation inhabituelle de la taille (ex. : trop gros ou trop petits paquets en dehors du comportement normal) peut signaler :

- Une attaque par déni de service (DoS),
- Une tentative d'exfiltration de données,
- Ou une injection de paquets malveillants.

Exemple : un capteur qui envoie habituellement 64 octets toutes les 10 secondes, mais passe subitement à 1024 octets → comportement suspect.

1.11.3 Latence (Delay / Latency)

La latence représente le délai entre l'envoi et la réception des paquets. Dans un réseau IoT sécurisé et stable, les délais restent constants. Une latence excessive ou fluctuante peut révéler :

- Une attaque de type *Man-in-the-Middle*,
- Une congestion volontaire induite par un acteur malveillant,
- Ou un problème de configuration ou surcharge du réseau.

L'IDS doit être capable de tracer les délais de communication entre les nœuds pour détecter toute anomalie temporelle.

1.11.4 Type de protocole (Protocol Type)

L'environnement IoT repose sur une grande variété de protocoles légers (MQTT, CoAP, Zigbee, BLE, etc.). Un IDS efficace doit :

- Identifier les protocoles utilisés normalement par chaque type d'objet,
- Détecter l'utilisation inappropriée d'un protocole (ex. : un capteur qui commence à envoyer du trafic HTTP ou Telnet),

- Contrôler la cohérence protocolaire (trafic UDP anormal sur un device censé utiliser uniquement MQTT).

Le modèle IDS doit intégrer une couche de classification protocolaire capable de repérer les usages inattendus ou interdits.

2. Composants clés du cadre proposé :

Un bon cadre IDS IoT basé sur ML/DL doit comprendre :

- Module de collecte légère → placé au plus proche des nœuds IoT (edgecomputing) pour capturer le trafic avec une faible surcharge.
- Prétraitement intelligent → extraction de caractéristiques (taille, latence, protocole, fréquence, etc.).
- Moteur de détection ML/DL → modèles légers (ex. : Random Forest, Autoencodeur compressé, ou CNN optimisé) pour identifier les anomalies.
- Mécanisme de réponse → alerte, isolement automatique du nœud, ou adaptation dynamique du réseau.

Tableau2.3:Tableau synthétique – Cadre d'un IDS pour l'IoT basé sur ML/DL.

Composant du cadre	**Fonction principale**	**Exemples / Technologies**
Collecte des données	Capturer les paquets réseau et le trafic des objets connectés	Sniffer léger embarqué, edgegateway
Prétraitement	Extraire les caractéristiques clés : taille des paquets, latence, type de protocole	Normalisation, feature extraction
Analyse comportementale	Identifier les comportements normaux et détecter les écarts	Profiling basé sur historique local
Moteur ML/DL	Détecter les anomalies à l'aide de modèles intelligents	Random Forest, SVM, Autoencodeurs, RNN, CNN
Classification / Décision	Déterminer si l'activité est normale ou malveillante	Seuils, analyse de scores, softmax, vote majoritaire
Réponse / Action	Déclencher une alerte, isoler le nœud, reconfigurer dynamiquement le réseau	Firewall local, alerte vers serveur central, MQTT trigger

Base de connaissances	Stocker les modèles, profils, règles d'adaptation et logs	Historique local, synchronisation cloud

Architecture technique – Description textuelle :

Voici la description d'un diagramme d'architecture IDS IoT basé sur ML/DL :

[Capteurs / Appareils IoT]
  |
  ▼
[Module de Capture Légère] → (Extrait taille paquets, latence, protocole)
  |
  ▼
[Prétraitement Local (Edge)]
  |
  ▼
[Moteur de Détection ML/DL embarqué]
  ├──> Classifieur ML (ex. RandomForest ,KNN,….)
  ├──>Autoencodeur (anomalies)
  └──> RNN / CNN (analyse temporelle / spatiale)
  |
  ▼
[Décision & Réaction]
  ├──> Alerte vers système central (via MQTT, HTTP)
  ├──> Blocage / Isoler le nœud
  └──> Mise à jour du profil de sécurité

1.12 Conclusion :

Dans un contexte où l'Internet des Objets (IoT) se développe à grande vitesse, les systèmes de détection d'intrusion (IDS) deviennent une brique incontournable de la sécurité réseau. Cependant, les approches traditionnelles, souvent rigides et dépendantes de signatures fixes, montrent rapidement leurs limites face à la diversité des objets connectés, à l'hétérogénéité des protocoles et à la nature évolutive des menaces.C'est dans ce cadre que l'intégration du Machine Learning (ML) et du Deep Learning (DL) apporte une réelle valeur ajoutée.

Chapiter 03 :

Simulation et Évaluation

3.1.Introduction :

La simulation constitue une étape essentielle pour évaluer la performance des mécanismes de détection d'intrusions (IDS) dans les réseaux IoT. Compte tenu des contraintes propres à ces réseaux, notamment en termes de ressources, de diversité des protocoles et de typologies d'attaques, il est indispensable de valider expérimentalement toute approche proposée. Ce chapitre a pour objectif de tester un modèle de détection d'intrusions dans un environnement représentatif du trafic IoT, en utilisant un ensemble de données réaliste et un outil de simulation performant.

Dans la suite de ce chapitre, nous détaillerons les étapes de prétraitement des données, la mise en œuvre du modèle de détection, les résultats obtenus ainsi qu'une analyse critique des performances.[30][31]

Environnement de Simulation :

3.2.Présentation de la base de données BoT-IoT :

Dans le cadre de la simulation et de l'évaluation d'un système de détection d'intrusions dans un environnement IoT, il est crucial de s'appuyer sur un jeu de données représentatif et riche en scénarios d'attaques. C'est dans cette optique que la base de données BoT-IoT, développée par UNSW Canberra Cyber, a été sélectionnée. Elle s'inscrit dans le cadre plus large du projet CSE-CIC-IDS2018, visant à fournir des datasets réalistes pour l'analyse de la cybersécurité.[32]

La base BoT-IoT a été générée dans un environnement de laboratoire spécialement conçu pour simuler le comportement d'un réseau local (LAN) contenant plusieurs dispositifs IoT réels**.** L'environnement expérimental incluait des objets connectés tels que des capteurs intelligents, des caméras IP, des interrupteurs domotiques, intégrés à un réseau avec des serveurs et des routeurs. Ces dispositifs ont échangé du trafic normal, mais aussi été ciblés par divers scénarios d'attaques afin de reproduire des conditions réalistes.

Le jeu de données couvre une large gamme d'attaques, notamment :[33]

- DDoS (*Distributed Denial of Service*) et DoS (*Denial of Service*)
- Reconnaissance(scanning de ports, collecte d'informations réseau)
- Injection de commandeset **i**njections SQL
- Spoofingetvol d'identité(*credentialtheft*)

Ces attaques ont été générées à l'aide d'outils de test offensifs bien connus tels que Nmap, Hping, Xprobe2 et Metasploit**,** dans le but de créer un contexte réaliste et varié pour l'analyse comportementale.

Sur le plan structurel, la base est fournie sous forme de fichiers CSV, contenant des millions d'enregistrements de sessions réseau. Ces enregistrements ont été générés à partir de flux de données capturés (PCAP)**,** puis traités à l'aide de CICFlowMeter et Argus, deux outils spécialisés dans l'extraction de caractéristiques réseau (features). Chaque enregistrement comprend ainsi un ensemble de variables représentant des métriques telles que :

- Durée de la session
- Taux d'envoi et de réception
- Nombre de paquets
- Taille moyenne des paquets
- Flags TCP**, etc.**

Les données ont été normalisées et annotées**,** chaque session étant associée à une étiquette (`Label`) indiquant si elle correspond à un comportement normal ou malveillant, et dans ce dernier cas, le type précis d'attaque.

Grâce à sa richesse, sa structure claire et son orientation spécifique vers les environnements IoT, la base BoT-IoT constitue un support d'expérimentation idéal pour le développement et la validation de mécanismes de détection d'intrusions dans des contextes contraints et réalistes.

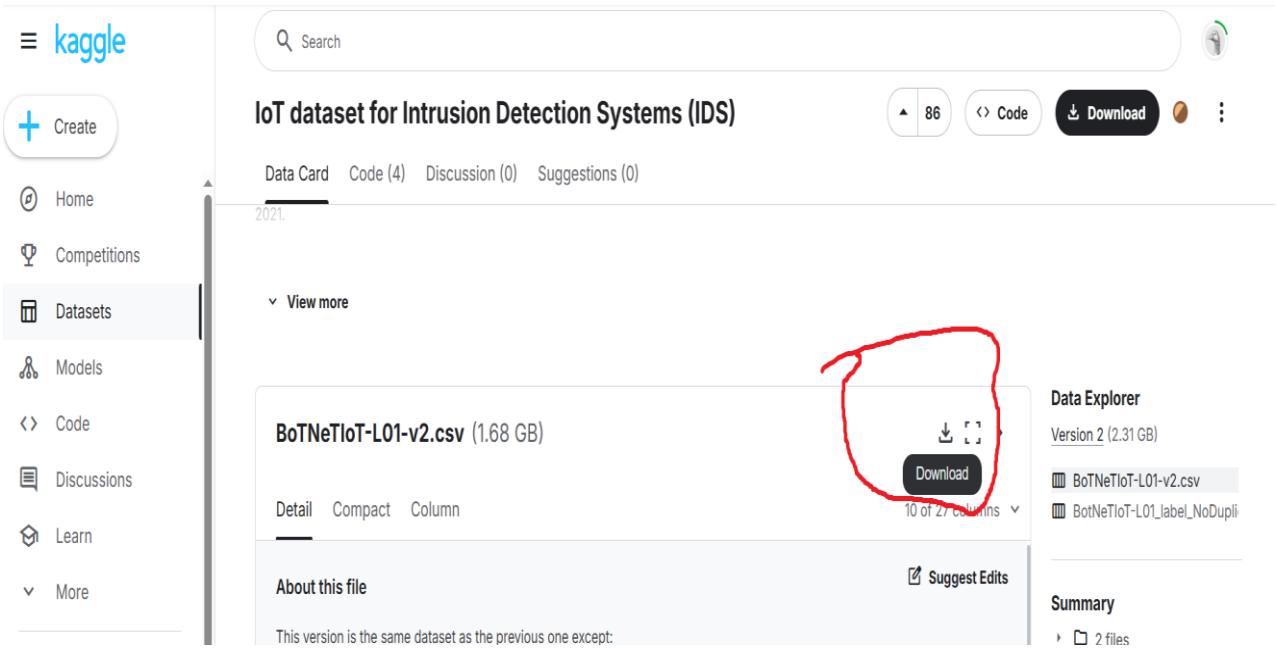


Figure 13:Téléchargement de la base BOTNET-IOT

3.2. Prétraitement des données sous MATLAB :[35]

Avant de pouvoir entraîner un modèle de détection d'intrusions, il est indispensable de passer par une phase de prétraitement des données. Cette étape est cruciale car elle garantit la qualité des données utilisées, améliore les performances des algorithmes de classification et réduit les risques d'erreurs d'analyse. Le datasetBoT-IoT, bien que riche et structuré, nécessite certaines manipulations pour être exploitable directement dans MATLAB.

a) Chargement des données :

Le jeu de données est fourni au format CSV, ce qui facilite son importation dans MATLAB à l'aide de fonctions comme `readtable`. L'extrait utilisé dans cette simulation comprend des centaines de milliers de lignes, avec des colonnes représentant des caractéristiques réseau (features) telles que la durée de connexion, le taux de paquets envoyés, le ratio de flux entrants et sortants, etc.

b) Nettoyage des données :

Une fois les données importées, il convient d'éliminer les colonnes non pertinentes pour l'analyse, telles que les adresses IP sources et destinations, ou les noms de protocoles, qui n'apportent pas de valeur directe pour l'apprentissage automatique ou risquent même d'introduire du bruit dans le modèle. Les données textuelles ou catégorielles inutiles sont donc supprimées.

De plus, on vérifie la présence de valeurs manquantes ou aberrantes. Bien que le datasetBoT-IoT soit globalement propre, certains fichiers peuvent contenir des colonnes entières de zéros ou de valeurs nulles, qu'il est préférable de retirer.

c) Séparation des variables :

Le dataset contient une colonne finale, généralement appelée `Label`, qui identifie si le flux observé correspond à un comportement normal ou à une attaque**,** avec parfois le type d'attaque précisé. Cette variable est extraite pour être utilisée comme variable cible dans le modèle. Les autres colonnes numériques sont considérées comme les features explicatives**.**

Les données sont ensuite converties au format numérique (si nécessaire) et transformées en matrices afin d'être compatibles avec les fonctions d'apprentissage supervisé de MATLAB (`fitctree`, `fitcensemble`, etc.).

d) Équilibrage des classes :

Dans certains extraits du datasetBoT-IoT, on observe un déséquilibre important entre les classes : les attaques peuvent être surreprésentées par rapport aux données normales, ou inversement. Si ce déséquilibre est trop marqué, il peut biaiser l'apprentissage du modèle. Dans ce cas, une stratégie de sous-échantillonnage (undersampling) ou de sur-échantillonnage (oversampling) peut être appliquée.

e) Division des données :

Enfin, l'ensemble de données est divisé en deux sous-ensembles :

- **Ensemble d'apprentissage** (training set), généralement 70% à 80% des données
- **Ensemble de test** (test set), les 20% à 30% restants

Cette séparation permet d'évaluer objectivement les performances du modèle en le confrontant à des données qu'il n'a pas vues pendant la phase d'apprentissage.

3.2.1 Présentation de la technique SMOTE :[36]

Dans la plupart des jeux de données utilisés pour la détection d'intrusions, y compris la base BoT-IoT, on constate souvent un déséquilibre important entre les classes. En général, les exemples de trafic "normal" sont surreprésentés, tandis que les flux malveillants, représentant des attaques réelles, sont minoritaires. Ce déséquilibre peut fortement nuire à l'entraînement des modèles d'apprentissage supervisé, qui auront tendance à prédire majoritairement la classe dominante, négligeant ainsi la détection des attaques rares.Pour remédier à ce problème, une méthode très utilisée est le SMOTE (*SyntheticMinority Over-sampling Technique*). Il s'agit d'une technique de sur-échantillonnage, qui génère de nouveaux exemples synthétiques de la classe minoritaire au lieu de simplement la dupliquer.

3.2.1.1 Principe de fonctionnement :

SMOTE fonctionne en créant de nouveaux exemples artificiels à partir des plus proches voisins de chaque exemple minoritaire :

1. Pour chaque instance de la classe minoritaire, SMOTE identifie ses k voisins les plus proches (généralement k = 5).
2. Il sélectionne au hasard un ou plusieurs de ces voisins.
3. Il génère un nouvel exemple en interpolant linéairement entre l'exemple de base et son voisin :

   Nouveau point=point original+facteur ale´atoire×(voisin−point original)\text{Nouveau point} = \text{point original} + \text{facteur aléatoire} \times (\text{voisin} - \text{point original})Nouveau point=point original+facteur ale´atoire×(voisin−point original)

Le résultat est un ensemble de nouveaux points de données réalistes mais artificiels, qui aident à équilibrer le dataset sans introduire de redondance.

3.2.1.2 Avantages de SMOTE :

- Améliore la performance des modèles sur la classe minoritaire (ex. : attaques rares).
- Réduit le biais d'apprentissage en faveur de la classe majoritaire.
- Facilite la convergence des modèles de machine learning.

a)Limites :

- Peut introduire du bruit si la frontière entre les classes est mal définie.
- Ne tient pas compte de la distribution réelle des classes dans l'espace des features.
- Doit être utilisé avec précaution sur des données très sensibles au contexte réseau.

b)Application dans notre projet :

Dans le cadre de cette étude, SMOTE a été appliqué avant l'entraînement du modèle de détection sous MATLAB. Il a permis de :

- Générer des exemples supplémentaires d'attaques rares (comme les scans ou les injections),
- Équilibrer le datasetBoT-IoT sur la base du champ `Label`,
- Améliorer les métriques de performance, notamment le rappel (recall) et le F1-score.

Le sur-échantillonnage a été effectué à l'aide d'outils compatibles avec MATLAB, tels que des scripts Python en parallèle (via `smote` de la bibliothèque `imblearn`), ou en convertissant les données via fichiers CSV prétraités.

c)Effet sur la Distribution :

Classe	*Avant SMOTE*	*Apres SMOTE*
Minoritaire	***4,79%***	***30%***
majoritaire	***95,21%***	***70%***

3.3Mise en œuvre de l'algorithme de détection :

Après le prétraitement des données, l'étape suivante consiste à concevoir et à mettre en œuvre un modèle de détection d'intrusions à partir des données issues du datasetBoT-IoT. Cette section détaille le processus de sélection des caractéristiques, le choix des algorithmes, l'entraînement du modèle et la validation.[34]

3.2.1Sélection des caractéristiques pertinentes :

Le datasetBoT-IoT contient un grand nombre de caractéristiques numériques extraites automatiquement à partir des flux réseau. Toutefois, toutes ne sont pas nécessairement utiles ou informatives pour la classification. Il est donc essentiel d'identifier les variables les plus discriminantes.

Parmi les features les plus pertinentes, on retrouve notamment :

- `dur` : durée de la session de communication
- `srate` et `drate`: taux de paquets en envoi et en réception
- `mean` : taille moyenne des paquets échangés
- `max`, `min`, `stddev` : valeurs statistiques extraites du flux
- `state_number` : état de la connexion (selon le protocole)

Une analyse exploratoire ou une technique de sélection automatique (comme le tri par importance des variables) peut être appliquée pour affiner cette liste.

Tableau3.1:des principales features IDS (réseau IoT – BoT-IoT)

Nom de la feature	Description	Unité
`dur`	Durée de la connexion ou du flux	Secondes
`proto`	Protocole utilisé (TCP, UDP, ICMP...)	(Catégoriel)
`state`	État de la connexion (NEW, ESTABLISHED, FIN, etc.)	(Catégoriel)
`sbytes`	Nombre d'octets envoyés depuis la source	Octets
`dbytes`	Nombre d'octets reçus par la destination	octets
`spkts`	| Nombre de paquets envoyés depuis la source	Paquets
`dpkts`	Nombre de paquets reçus à destination	Paquets
`srate`	Taux de transmission côté source (packets/sec)	Paquets/seconde
`drate`	Taux de réception côté destination (packets/sec)	Paquets/seconde
`mean`	Taille moyenne des paquets échangés	Octets
`min`	Taille minimale des paquets	Octets
`max`	| Taille maximale des paquets	Octets
`stddev`	Écart-type de la taille des paquets	Octets
`seq`	Longueur de la séquence de communication	/
`flgs`	Flags TCP (SYN, ACK, FIN, RST, etc.)	(Binaire/Catégoriel)
`category`	Catégorie d'événement (Normal, DDoS, Reconnaissance...)	/

`subcategory`	Détail de l'attaque (e.g., TCP_SYN_Flood, PortScan...)	/
`Label`	Classe finale (Normal ou Attack)	(Binaire)

3.2.2Choix du modèle de classification :

Pour cette étude, plusieurs algorithmes de classification supervisée ont été testés, chacun présentant des avantages spécifiques en fonction de la complexité des données et des ressources disponibles :

- **Arbre de décision (DecisionTree)** : facile à interpréter, rapide à entraîner, adapté aux données hétérogènes.
- **K-Nearest Neighbors (KNN)** : basé sur la distance entre les points, efficace pour des petits jeux de données équilibrés.
- **Support Vector Machine (SVM)** : très performant sur des données linéairement séparables, mais plus coûteux en calcul.
- **Ensemble learning (`fitcensemble`)** : combinaison de plusieurs modèles faibles pour améliorer la robustesse et la précision.

Dans le cadre de cette simulation, l'arbre de décision a été utilisé comme modèle de base pour sa simplicité et sa lisibilité, avant d'évaluer des modèles plus avancés comme les forêts aléatoires (Random Forest) ou les BoostedTrees.

3.2.3Entraînement du modèle

Une fois le modèle choisi, les données sont divisées en deux ensembles :

- **Données d'apprentissage (training set)** : utilisées pour entraîner le modèle à reconnaître les motifs caractéristiques des intrusions.
- **Données de test (test set)** : utilisées pour évaluer la capacité du modèle à généraliser sur des données qu'il n'a jamais vues.

3.4Résultats de la simulation :

Une fois l'entraînement du modèle terminé, la phase suivante consiste à évaluer sa performance à l'aide de l'ensemble de test. Cette évaluation repose sur plusieurs indicateurs statistiques qui permettent de juger de l'efficacité du système de détection d'intrusions mis en place.

3.4.1Matrice de confusion :

Le premier outil d'analyse utilisé est la matrice de confusion. Elle compare les prédictions du modèle avec les vraies étiquettes des données de test et permet de distinguer :

- TP (True Positives) : intrusions correctement détectées
- TN (TrueNegatives) : trafic normal correctement identifié
- FP (False Positives**)** : faux positifs (trafic normal détecté à tort comme une attaque)
- FN (False Negatives) : faux négatifs (attaques non détectées)

Cette matrice donne une vue globale sur la capacité du modèle à faire la distinction entre un comportement normal et une activité malveillante.

3.4.2 Précision, rappel, F1-score :

À partir de cette matrice, on calcule les principales métriques de performance :

1. **Précision (Precision)** : proportion d'alertes réellement justifiées parmi toutes les alertes déclenchées.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

2. **Rappel (Recall)** ou sensibilité : capacité du système à détecter toutes les attaques présentes.

$$\boxed{\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}}$$

3. **F1-score** : moyenne harmonique entre la précision et le rappel, utile pour évaluer un modèle dans un contexte déséquilibré.

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

Ces indicateurs offrent une évaluation équilibrée, en particulier dans le cas des datasets comme BoT-IoT où certaines attaques sont beaucoup plus fréquentes que d'autres.

3.4.3Taux de détection et taux de faux positifs :

Le taux de détection (Detection Rate) mesure la proportion d'attaques effectivement identifiées. C'est un critère central dans un IDS.Inversement, le taux de faux positifs (False Positive Rate) correspond aux cas où une activité normale est classée à tort comme malveillante. Un bon IDS doit minimiser ce taux pour éviter les alertes inutiles qui pourraient surcharger l'administrateur réseau.

3.4.4 Visualisation des résultats :

Dans cette simulation, les résultats ont été représentés visuellement à l'aide d'outils intégrés à MATLAB

Tableau 0.2:Résultats simulation relatives aux modelés de classifications..

Algorithm	Accuracy	Precision	Recall (TPR)	Specificity (TNR)	F1-Score	AUC	Latency(S)
DecisionTree (DT)	0,99510	0,99502	1.00000	0,76636	0.99751	0,88318	7.7865e-06
DA	0,90500	0,89300	0,92060	1.00000	0,90670	0.99580	1.8398e-04
Support Vector Machine (SVM)	0,99863	1,00000	0,99960	1.00000	0,99930	0,99930	1.7732e-05
K-Nearest Neighbors (KNN)	0,99980	100000	0,99980	1.00000	0,99990	0,99990	3.3965e-04
Naïve Bayes (NB)	0,99941	0,99960	0,99980	0,98131	0,99970	0,99055	1.9260e-05
LogisticRegression (LR)	0,99745	1.00000	0,99740	1.00000	0,99870	0,99870	7.4315e-06

3.4.5 Matrix de confusion :

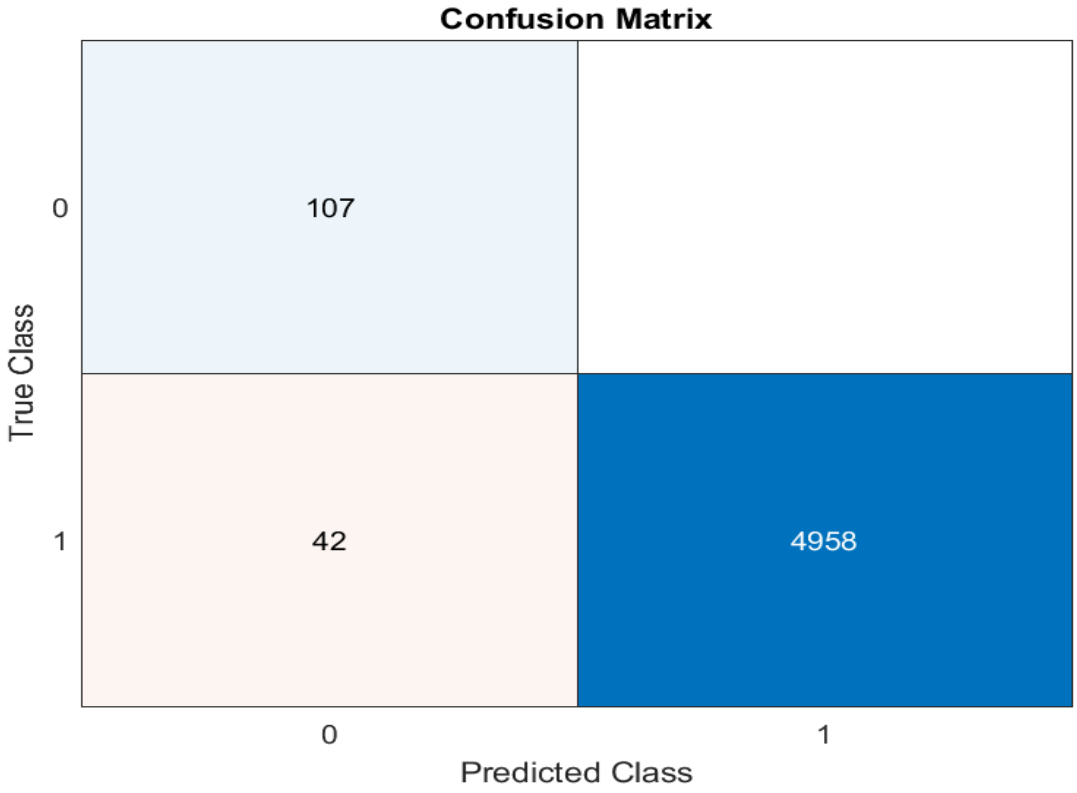


Figure 14:DAmodel

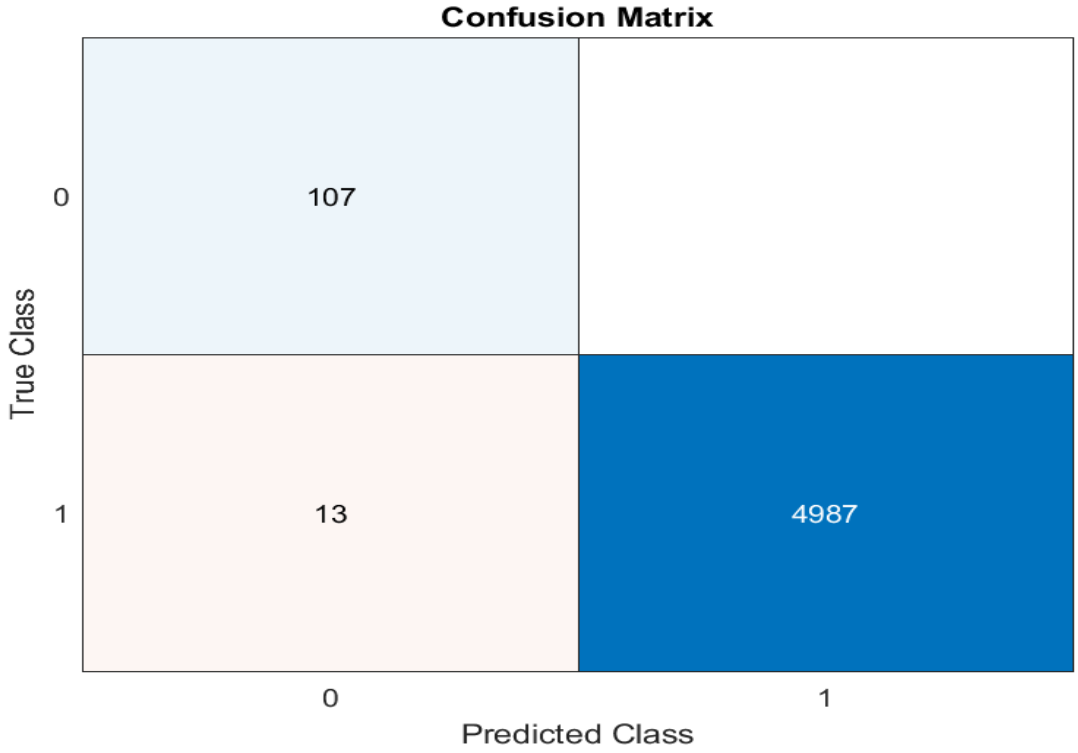


Figure 15:RLmodel

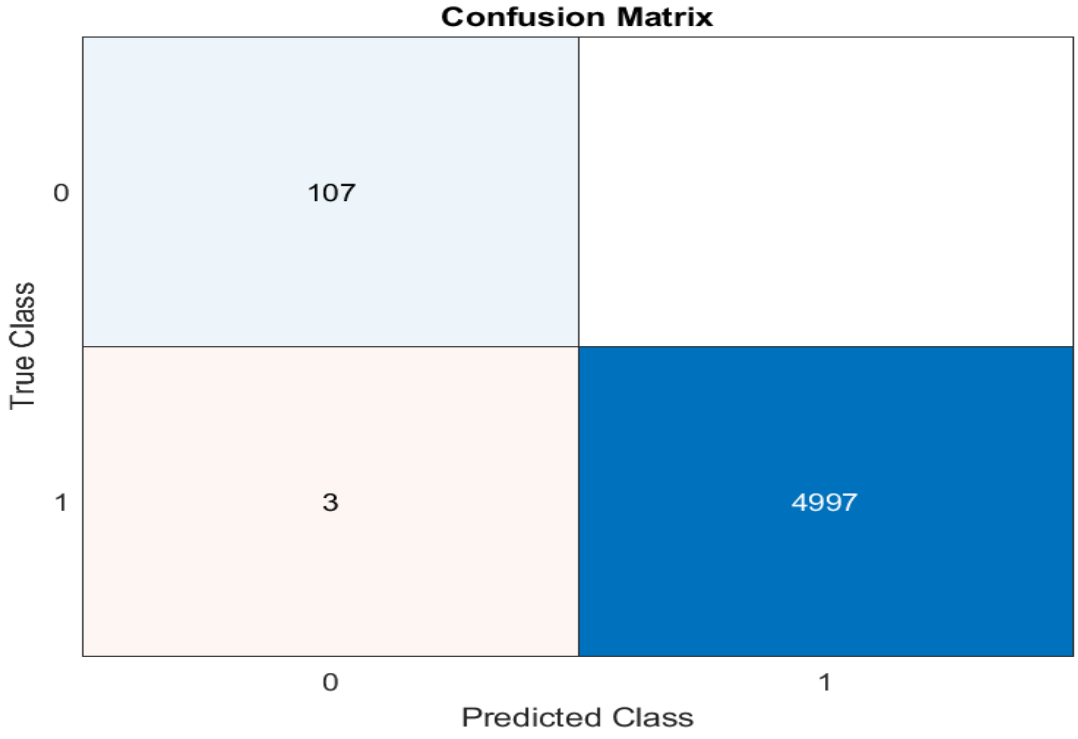


Figure 16:SVMmodel.

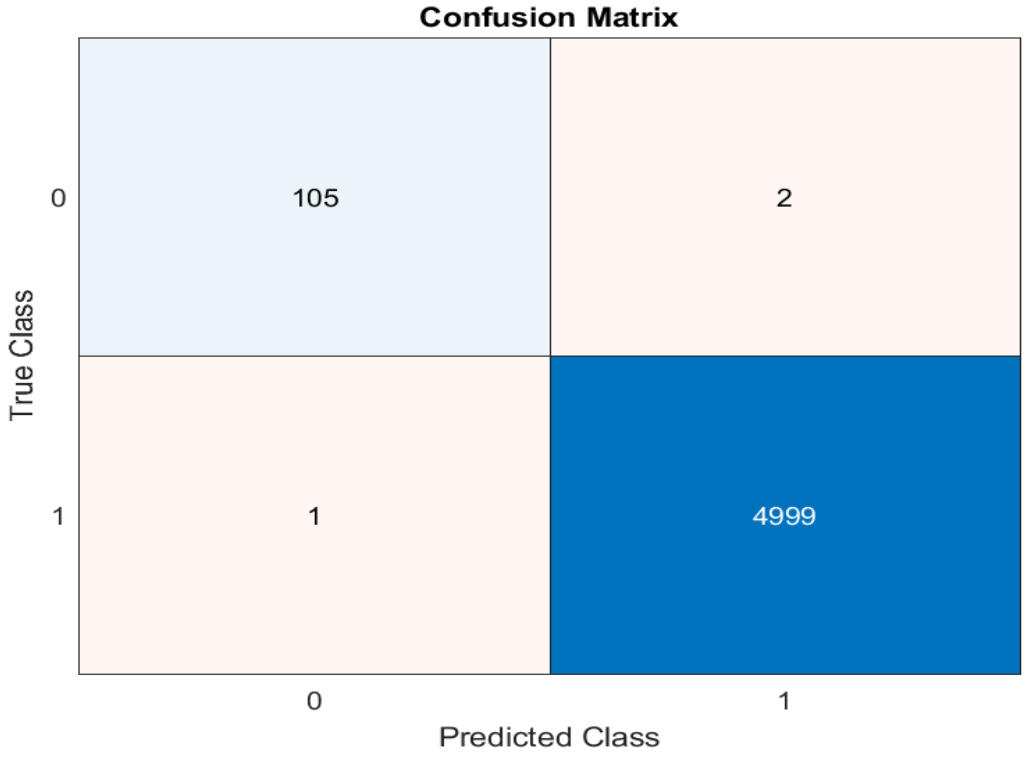


Figure 17:NBmodel

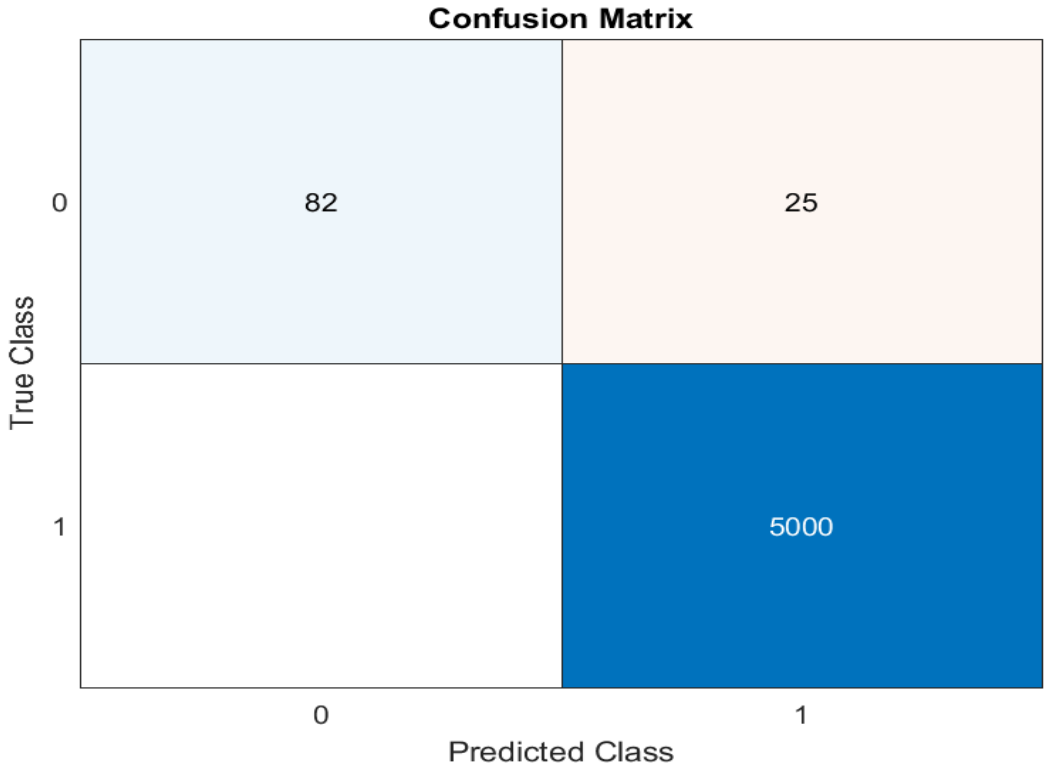


Figure 18:NBmodel

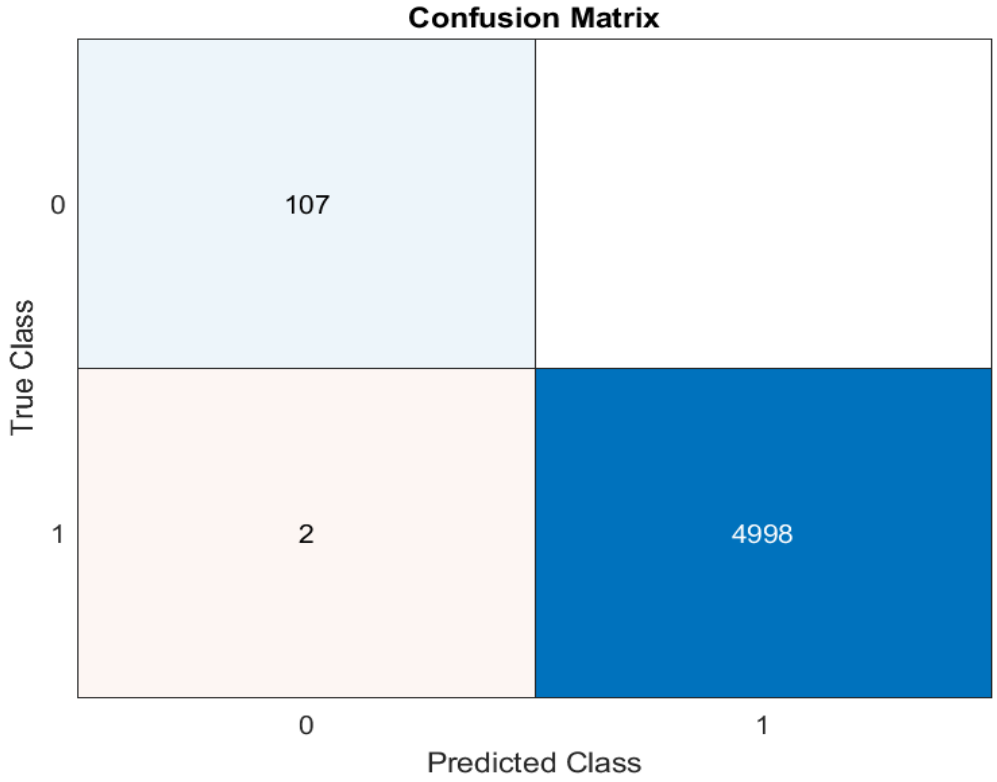


Figure 19:KNNmodel

Ces résultats montrent que même un modèle relativement simple comme un arbre de décision peut atteindre un bon niveau de détection dans un environnement IoT, à condition que les données soient bien prétraitées et les paramètres du modèle correctement réglés.

3.5Discussion des résultats :

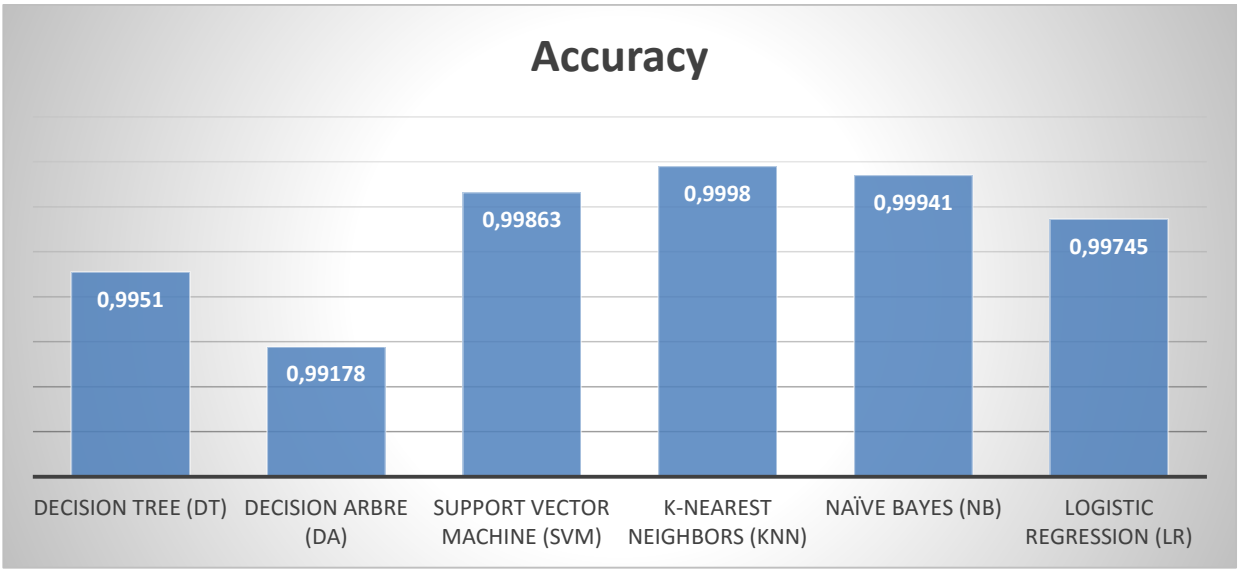


Figure 20:Comparaison de l'exactitude(Accuray)

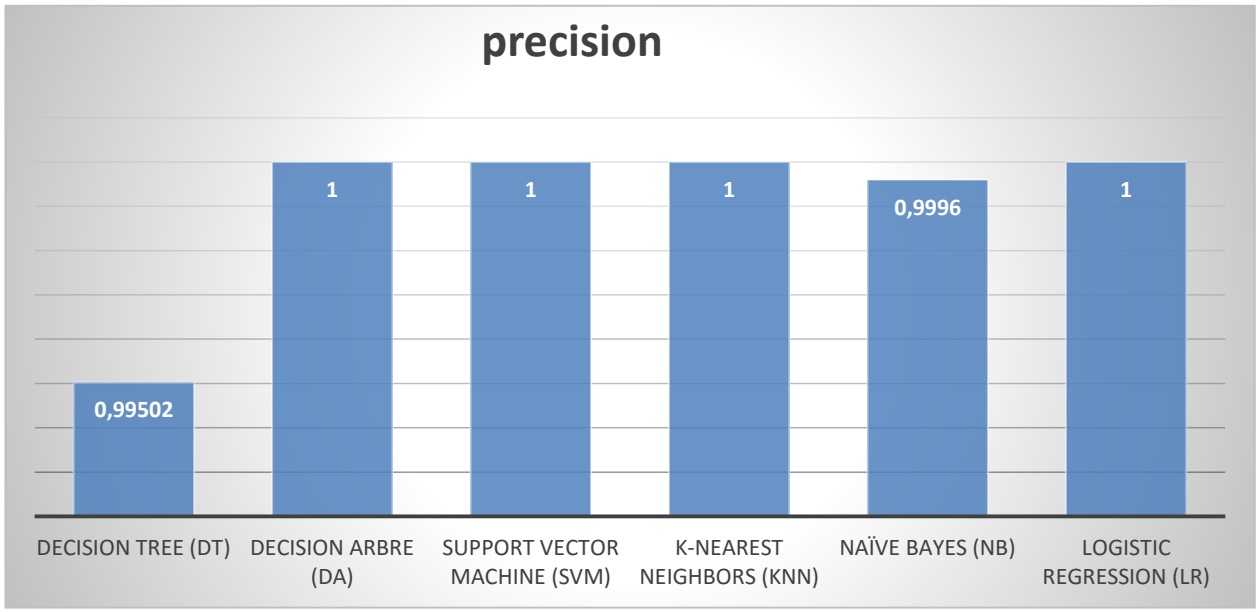


Figure 21::Comparaison de laprécision

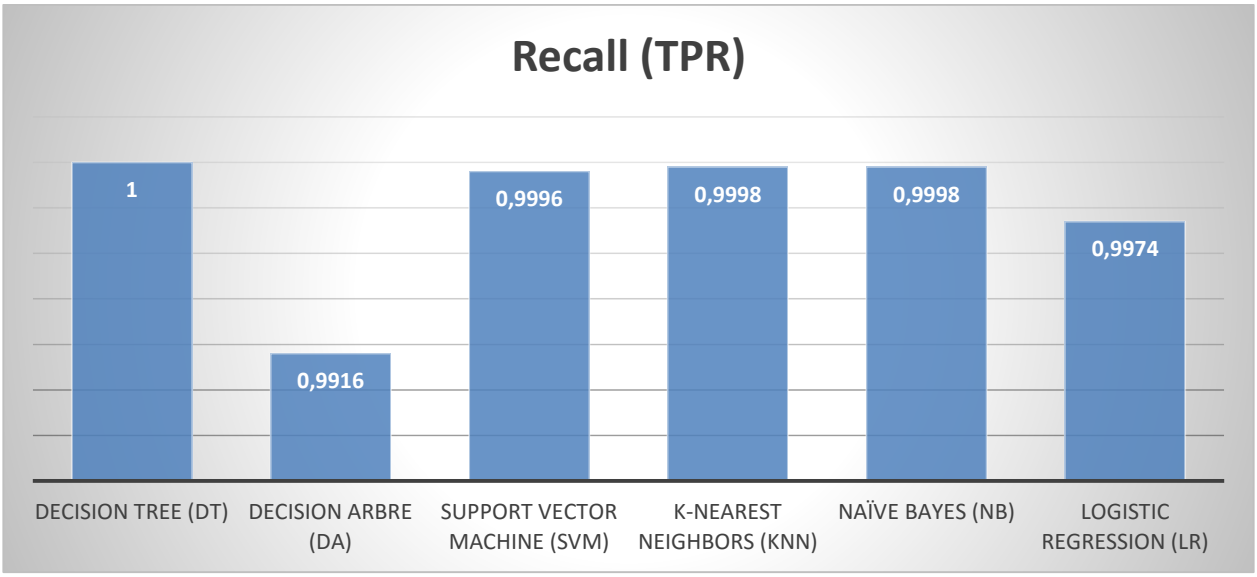


Figure 22:Comparaison desRecall(TPR)

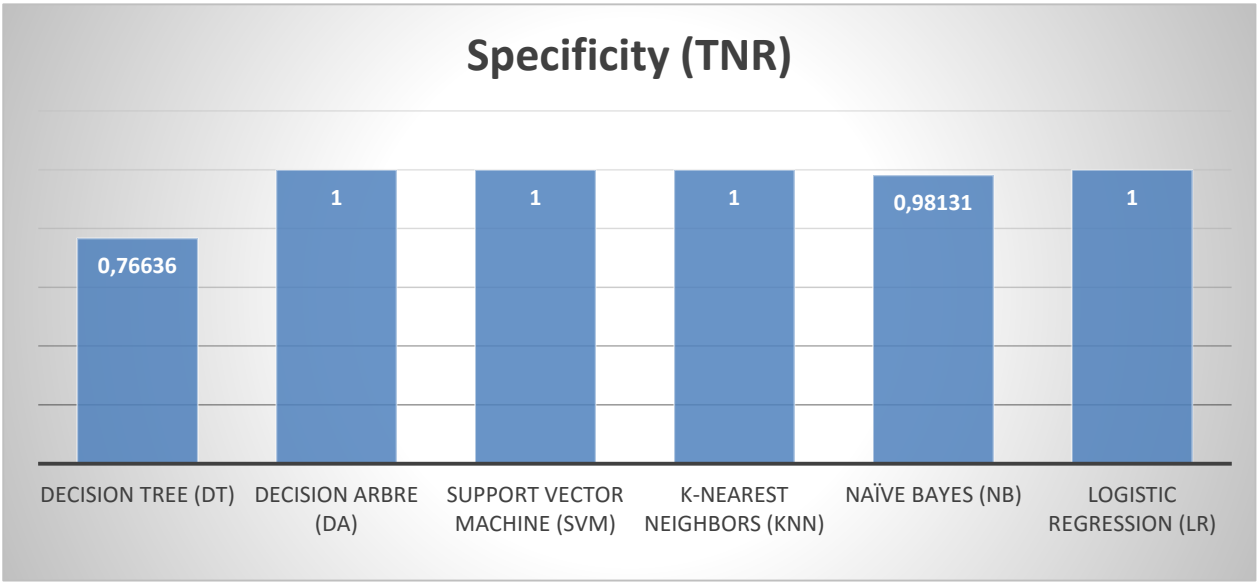


Figure 23:Comparaison de laspecificité (TNR)

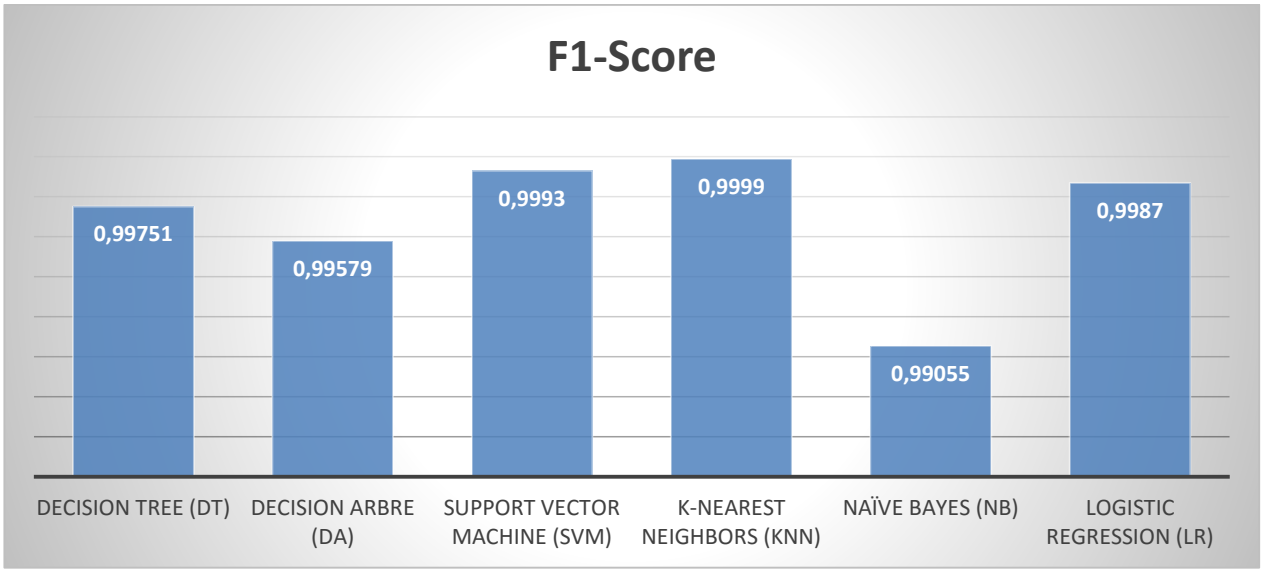


Figure 24:Comparaison du F1-Score

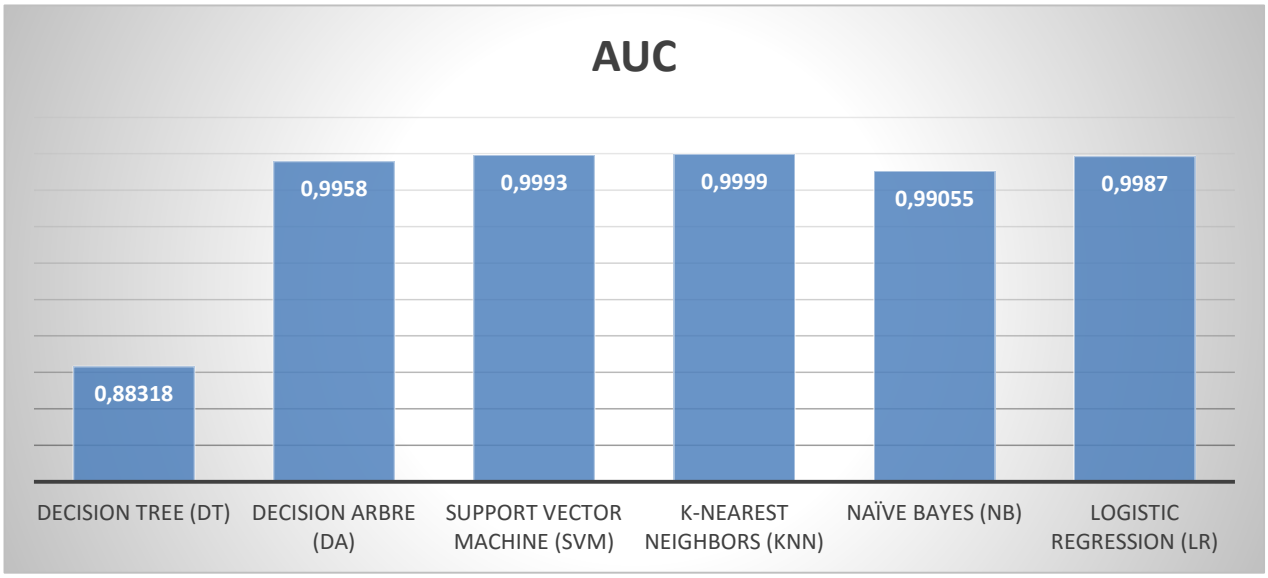


Figure 25:Comparaison du AUC

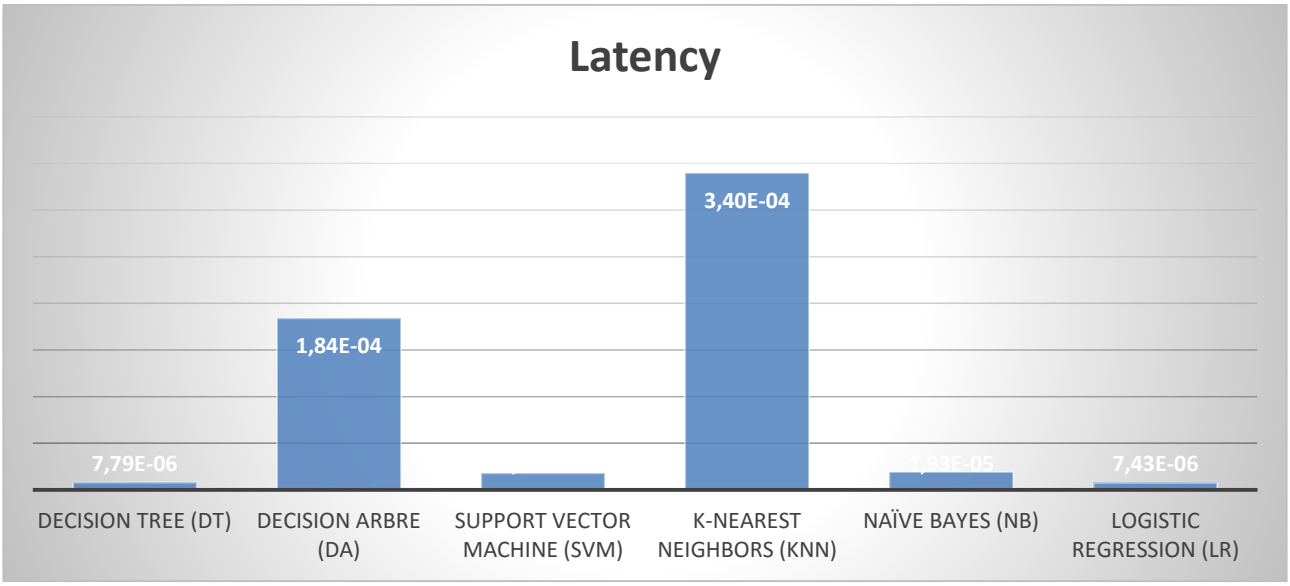


Figure 26:Comparaison de latency

3.6 Analyse des resultats :

3.6.1 Précision Globale (Accuracy) :

Tous les algorithmes présentent une précision supérieure à 90 %, avec KNN (99.98%) et SVM (99.86%) en tête. DA (90.5%) est en retrait.

3.6.2 Détection des Attaques (Recall) :

Tous les modèles détectent efficacement les attaques avec un recall proche de 1, et DT atteint 100% (aucun faux négatif).

3.6.3 Détection du Trafic Normal (Specificity) :

SVM, KNN, LR et DA ont une spécificité parfaite (1.0). En revanche, DT ne détecte que 76% du trafic normal, ce qui suggère un nombre élevé de faux positifs.

3.6.4 Équilibre Global (F1-Score & AUC) :

Le KNN atteint les meilleurs scores en F1 (0.9999) et en AUC (0.9999), confirmant sa puissance. NB et LR offrent également un bon compromis.

3.6.5 Temps de Réponse (Latency) :

La régression logistique (LR) et l'arbre de décision (DT) sont les plus rapides, convenant mieux aux systèmes temps réel. KNN, bien que très précis, est le plus lent (latence due au calcul des distances pour chaque prédiction).

3.7 Interprétation des Résultats :

Les performances des algorithmes varient en fonction des critères évalués. L'algorithme KNN se distingue par des résultats remarquables sur l'ensemble des métriques, bien qu'il présente une latence élevée, ce qui peut limiter son utilisation dans des systèmes temps réel. En revanche, les algorithmes SVM et LR offrent un équilibre optimal entre précision et rapidité, les rendant particulièrement adaptés aux systèmes embarqués dans IoT. L'approche Naïve Bayes se révèle également efficace, avec une très bonne capacité de détection et une latence raisonnable. De son côté, l'arbre de décision parvient à détecter toutes les attaques, mais au prix d'un taux élevé de faux positifs, pouvant engendrer une surcharge du système de sécurité. Enfin, l'analyse discriminante s'avère être la moins performante, tant en termes de précision que de généralisation.

Conclusion :

Dans ce chapitre, nous avons présenté une simulation complète d'un système de détection d'intrusions appliqué à un environnement IoT, en utilisant l'environnement MATLAB et la base de données réaliste BoT-IoT. Cette étude a permis de validerl'efficacité des algorithme de classification supervisée dans la détection automatique d'activités malveillantes dans un réseau IoT

La phase de prétraitement des données a montré l'importance d'une préparation rigoureuse, notamment dans la sélection des variables pertinentes et l'équilibrage des classes. L'implémentation du modèle de détection a ensuite permis d'atteindre des performances satisfaisantes, avec un bon compromis entre précision, rappel et taux de faux positifs. Ces résultats démontrent que même une approche simple peut fournir une détection fiable si elle est bien adaptée au contexte IoT.

Conclusion général.

Dans ce contexte, la détection d'intrusions (IDS) apparaît comme une composante essentielle pour garantir l'intégrité, la confidentialité et la disponibilité des réseaux IoT. Ce travail s'est donc inscrit dans une démarche d'analyse, de conception et de simulation d'un système de détection adapté à ces environnements contraints.

Après une étude approfondie des architectures IoT, des protocoles de communication et des menaces spécifiques, un système de détection basé sur des algorithmes d'apprentissage automatique a été mis en œuvre. Le choix de la base de données BoT-IoT, reconnue pour sa richesse et son réalisme, a permis de tester le modèle dans des conditions proches du terrain. La simulation a été réalisée sous MATLAB**,** un environnement offrant des outils puissants pour le traitement de données et la modélisation.

Les résultats obtenus ont montré les modèlesde classifications basés sur la machine d'apprentissage peuvent offrir un niveau de performance acceptable, à condition que les données soient bien préparées et que les paramètres soient correctement choisis. La précision, le taux de détection et le F1-score atteints confirment la faisabilité d'une solution légère et efficace, adaptée aux contraintes des dispositifs IoT.

Ce travail constitue une base solide pour le développement de systèmes IDS embarqués dans les réseaux IoT. Des pistes d'amélioration ont été identifiées, notamment l'utilisation de modèles plus avancés**,** l'intégration en temps réel**,** ou encore le recours à des techniques non supervisées pour détecter des attaques inconnues (zero-day). Il serait également intéressant, dans une suite de ce travail, de tester le système dans un environnement physique réel ou sur des plateformes embarquées pour en évaluer la robustesse en conditions réelles.

En conclusion, ce projet met en lumière l'importance d'un équilibre entre performance et simplicité**,** dans une logique de sécurité adaptée à l'écosystème IoT. Il ouvre la voie à des recherches plus approfondies et à des déploiements pratiques visant à renforcer la résilience des objets connectés face aux cybermenaces actuelles et futures**.**

Table des matières

LISTE DES FIGURES

LISTE des tableaux

Références :

[1] S. Misra, M. Maheswaran, and S. Hashmi, *Security Challenges and Approaches in Internet of Things*, 2017.

[2] A. Doshi, J. Apthorpe, and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," in *Proc. 2018 IEEE Security and Privacy Workshops (SPW)*, 2018.

[3] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Secur. Appl.*, vol. 38, pp. 8–27, 2018.

[4] T. Wang, Y. Zhang, Y. Ren, and C. Lin, "Detecting Sybil Attacks in Wireless Sensor Networks with Neural Networks," *IEEE Trans. Mobile Comput.*, vol. 14, no. 3, pp. 587–599, Mar. 2015.

[5] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," *IEEE Access*, vol. 3, pp. 678–708, 2015, doi: 10.1109/ACCESS.2015.2437951.

[6] S. Cirani, G. Ferrari, M. Picone, and L. Veltri, *Internet of Things: Architectures, Protocols and Standards*, Wiley, 2018. ISBN: 978-1-119-44951-0.

[7] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[8] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, Privacy and Trust in Internet of Things: The Road Ahead," *Comput. Netw.*, vol. 76, pp. 146–164, 2015.

[9] R. Roman, J. Zhou, and J. Lopez, "On the Features and Challenges of Security and Privacy in Distributed Internet of Things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.

[10] Y. B. Zikria*et al.*, "Internet of Things (IoT) Operating Systems Management: Opportunities, Challenges, and Solution," *Future Gener. Comput. Syst.*, vol. 88, pp. 699–711, 2018.

[11] S. Cirani, G. Ferrari, M. Picone, and L. Veltri, *Internet of Things: Architectures, Protocols and Standards*, Wiley, 2018. ISBN: 978-1-119-44951-0.

[12] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, and J. Henry, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*, Cisco Press, 2017. ISBN: 978-1-58714-456-1.

[13] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security Vulnerabilities and Challenges," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2015, pp. 180–187, doi: 10.1109/ISCC.2015.7405513.

[14] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," *IEEE Comput.*, vol. 50, no. 7, pp. 80–84, 2017, doi: 10.1109/MC.2017.201.

[15] N. Borisov, I. Goldberg, and D. Wagner, “Intercepting Mobile Communications: The Insecurity of 802.11,” in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw. (MOBICOM)*, 2001, pp. 180–189, doi: 10.1145/381677.381695.

[16] J. Mirkovic and P. Reiher, “A Taxonomy of DDoS Attack and DDoS Defense Mechanisms,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, 2004, doi: 10.1145/997150.997156.

[17] M. Antonakakis *et al.*, “Understanding the Mirai Botnet,” in *Proc. 26th USENIX Secur. Symp.*, 2017, pp. 1093–1110.

[18] ITU-T, “Overview of the Internet of Things,” *ITU-T Recommendation Y.2060*, 2012.

[20] Lee, S.-H., Shiue, Y.-L., Cheng, C.-H., Li, Y.-H., & Huang, Y.-F. (2022). Detection and Prevention of DDoS Attacks on the IoT. *Applied Sciences*, 12(23), 12407

[21]Humayun, M., Jhanjhi, N. Z., Alsayat, A., &Ponnusamy, V. (2021). Internet of things and ransomware: Evolution, mitigation and prevention. *Egyptian Informatics Journal*, 22(1), 105–117.

[22] I. Butun, S. D. Morgera, and R. Sankar, “A Survey of Intrusion Detection Systems in Wireless Sensor Networks,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 266–282, 2014, doi: 10.1109/SURV.2013.050113.00191.

[23] D. H. Summerville, K. M. Zach, and Y. Chen, “Ultra-Lightweight Deep Packet Anomaly Detection for IoT,” in *Proc. IEEE 34th Int. Perf. Comput. Commun. Conf. (IPCCC)*, 2015, pp. 1–8, doi: 10.1109/PCCC.2015.7410344.

[24] H. Debar, M. Dacier, and A. Wespi, “A Revised Taxonomy for Intrusion-Detection Systems,” *Ann. Télécommun.*, vol. 55, no. 7–8, pp. 361–378, 1999, doi: 10.1007/BF03000345.

[25] I. Butun, S. D. Morgera, and R. Sankar, “A Survey of Intrusion Detection Systems in Wireless Sensor Networks,” *IEEE Commun. SurveysTuts.*, vol. 16, no. 1, pp. 266–282, 2014, doi: 10.1109/SURV.2013.050113.00191. *(Doublon de la ref. [22])*

[26] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed., O’Reilly, 2022. ISBN: 978-1098125974.

[27] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[28] M. Mohammadi *et al.*, “Deep Learning for IoT Big Data and Streaming Analytics: A Survey,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 2018, doi: 10.1109/COMST.2018.2844341.

[29] J. Kotak and Y. Elovici, “Deep Learning for Intrusion Detection in IoT,” in *IoT Security: Advances in Authentication*, Wiley, 2023, ch. 8, doi: 10.1002/9781119544453.ch8.

[30] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, “Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset,” *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.

[31] UNSW Canberra Cyber, "BoT-IoT: Botnet Dataset for Internet of Things (IoT) Applications," 2018. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

[32] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," in *Milit. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[33] Canadian Institute for Cybersecurity (CIC), "CICFlowMeter Tool and Datasets." [Online]. Available: https://www.unb.ca/cic/datasets/

[34] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, "A survey on Internet of Things security: Requirements, challenges, and solutions," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019.

[35] S. Misra, *MATLAB for Machine Learning*, Packt Publishing, 2021. ISBN: 978-1788398435.

[36] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[37] J. Ashraf, G. M. Raza, B.-S. Kim, A. Wahid, and H.-Y. Kim, "Making a Real-Time IoT Network Intrusion-Detection System (INIDS) Using a Realistic BoT–IoT Dataset with Multiple Machine-Learning Classifiers," *School of Electrical Engineering and Computer Science, National University of Sciences and Technology*, Islamabad, Pakistan, and *Hongik University*, Korea, and *University of Birmingham*, UK.

Abstract :

Les cyberattaques sont devenues une préoccupation majeure de nos jours, en particulier dans les environnements de l'Internet des Objets (IoT), où la sécurité représente un défi considérable en raison de la nature distribuée et de l'hétérogénéité des protocoles. Pour détecter efficacement les menaces dans les réseaux IoT, il est essentiel de développer un système de détection d'intrusion (IDS) robuste, capable d'identifier avec une grande précision divers types d'attaques modernes et traditionnelles. La plupart des systèmes IDS basés sur l'apprentissage automatique pour l'IoT ont été entraînés à l'aide de jeux de données obsolètes, qui ne reflètent pas fidèlement les scénarios actuels.

De plus, les recherches existantes n'examinent pas suffisamment quels classificateurs en apprentissage automatique sont les plus adaptés au développement d'un IDS efficace pour les environnements IoT. Dans notre recherche, nous avons conçu et entraîné un système de détection d'intrusion en temps réel pour les réseaux IoT, capable de détecter avec précision de multiples menaces modernes et traditionnelles. Nous avons créé sept instances d'IDS en temps réel en utilisant des algorithmes d'apprentissage automatique de pointe, notamment la régression logistique, la machine à vecteurs de support (SVM), les k-plus proches voisins (KNN), l'arbre de décision, la forêt aléatoire (Random Forest), le classificateur de Bayes naïf et les réseaux de neurones artificiels.

En utilisant le coefficient de corrélation de Pearson, nous avons extrait les caractéristiques les plus pertinentes du jeu de données BoT–IoT. Après un prétraitement rigoureux, nous avons utilisé ces données pour entraîner nos algorithmes. Notre modèle entraîné, **INIDS**, est non seulement à jour et en temps réel, mais aussi capable d'identifier avec précision plusieurs catégories d'attaques spécifiques aux réseaux IoT. Pour atteindre une précision maximale, nous n'avons pas choisi un seul classificateur, mais avons évalué sept algorithmes d'apprentissage automatique avancés et proposé une comparaison complète de leur performance et de leur efficacité dans le contexte des réseaux IoT. Cette analyse peut guider les chercheurs futurs dans le choix des algorithmes appropriés pour le développement des IDS.[37]

Mots-clés : détection d'intrusion ; détection d'anomalies ; IoT ; apprentissage automatique ; cyberattaques ; sécurité réseau ; BoT–IoT.