UNIVERSITE GASTON BERGER DE SAINT LOUIS

Ecole **D**octorale des **S**ciences et des **T**echnologies

UFR de **S**ciences **A**ppliquées et de **T**echnologie



THÈSE DE DOCTORAT UNIQUE

Mention : Informatique

Spécialité : Cybersécurité et Intelligence Artificielle

SÉCURITE INTELLIGENTE POUR LES RÉSEAUX IoT : MÉTHODES D'APPRENTISSAGE AUTOMATIQUE ET PROFOND

Présentée et soutenue publiquement le 3 août 2024 par

Abdoulaye DIALLO

Directeur de Thèse : **Professeur Chérif DIALLO**

Composition du Jury

M. Moussa	LO	Professeur Titulaire, Université Gaston Berger, Sénégal	Président
M. Riadh	DHAOU	Professeur Titulaire, Institut National Polytechnique, Toulouse, France	Rapporteur
M. Karim	KONATE	Professeur Titulaire, Université Cheikh Anta Diop, Dakar, Sénégal	Rapporteur
M. Cheikh	SARR	Professeur Titulaire, Université Iba Der Thiam, Thiès, Sénégal	Rapporteur
M. Maissa	MBAYE	Maître de Conférence (CAMES), Université Gaston Berger, Saint-Louis, Sénégal	Examinateur
M. Eugène C.	EZIN	Professeur Titulaire, Université d'Abomey-Calavi, Cotonou, Bénin	Examinateur
M. Chérif	DIALLO	Professeur Titulaire, Université Gaston Berger, Sénégal	Directeur

Thèse Numéro EDST018-2024

Préparée au sein du **L**aboratoire d'**A**lgèbre, **C**ryptographie, **C**odes et **A**pplications (LACCA)

Remerciements

Louange à ALLAH, le maître des mondes. Que la paix et la bénédiction soient sur son noble prophète Mouhamed et sa famille.

Avant tout, je tiens à adresser mes remerciements les plus sincères à mon directeur de thèse, le Professeur Chérif DIALLO, ainsi qu'au Professeur Eugène C. EZIN co-directeur du projet IoT-SDC, à travers lequel cette thèse a été financée. Je garde en mémoire nos différentes réunions, nos différents échanges. Ce furent des moments riches pendant lesquels j'ai beaucoup appris. Votre rigueur scientifique, votre disponibilité infaillible et vos précieux conseils ont été essentiels à la réalisation de ce travail. Votre soutien constant et vos orientations judicieuses ont grandement facilité ma progression et ont enrichi ma recherche. Encore une fois, merci chers Professeurs, je vous resterai éternellement reconnaissant.

Je souhaite également exprimer ma reconnaissance aux membres de mon jury pour l'intérêt marqué qu'ils ont porté à mon travail. Je remercie le Professeur Moussa LO de m'avoir fait l'honneur de présider mon jury de thèse. Je remercie les Professeurs Riadh DHAOU, Karim KONATE et Cheikh SARR qui ont accepté d'être les rapporteurs de cette thèse. Merci pour votre lecture attentive, vos remarques pertinentes et vos recommandations, qui contribuent largement à la qualité de cette thèse. J'adresse mes remerciements au Professeur Maissa MBAYE qui a bien voulu examiner cette thèse. C'est un immense honneur pour moi de bénéficier de votre expertise et de vos suggestions éclairées.

Mes remerciements vont également au DSTN à travers son coordonnateur, le Professeur Gaoussou Camara, pour le soutien financier sans lequel cette thèse n'aurait pu voir le jour. Je tiens à exprimer ma profonde reconnaissance au CEA-MITIC pour le cadre idéal mis à notre disposition, qui a largement contribué au succès de cette recherche.

Je remercie toute l'équipe pédagogique de l'UFR SAT d'avoir contribué de façon très significative à ma formation. Mention spéciale à la section informatique dans laquelle j'ai trouvé ma voie et ma passion.

Je souhaite également témoigner ma gratitude infinie à mes très chers parents pour leur soutien indéfectible, leurs conseils avisés et leurs prières constantes, ainsi qu'à mon adorable épouse pour son appui inconditionnel. Mes frères et sœurs méritent également toute ma reconnaissance pour leur présence réconfortante.

Je n'oublie pas de remercier chaleureusement mes collègues du CEA-MITIC et ceux du DSTN pour les moments d'échange et de partage, avec une mention spéciale pour mon binôme Lionel, dont la collaboration a été particulièrement fructueuse.

Enfin, je tiens à exprimer ma gratitude envers tous mes amis, dont certains ont été des compagnons de route inestimables durant mon cursus scolaire et universitaire, ainsi qu'à mes voisins du G3D, dont la compagnie a été source de nombreuses joies. À toute personne qui, de près ou de loin, a contribué à la réalisation de cette thèse, je vous dis un grand merci.

Résumé

L'internet des objets (IoT) est une technologie émergente qui connecte des milliards de dispositifs dans divers domaines, allant de la santé aux villes intelligentes, en passant par l'industrie et l'agriculture. Cependant, cette interconnectivité croissante présente des défis majeurs en matière de sécurité. Les dispositifs IoT sont souvent vulnérables à une variété d'attaques, notamment les attaques par déni de service (DoS) [1], les intrusions et les manipulations à distance, ce qui peut entraîner des conséquences graves pour la confidentialité, l'intégrité et la disponibilité des données.

Pour tirer pleinement profit de cette technologie dont les domaines d'application sont souvent très sensibles, la mise en place de mécanismes de sécurité adaptés constitue une nécessité. Les avancées récentes dans le domaine de l'intelligence artificielle ont permis d'apporter des solutions à de nombreux problèmes. Les méthodes d'apprentissage automatique et profond ne cessent de repousser les limites de l'IA. Leurs applications dans le domaine de la cyber-sécurité ont suscité beaucoup d'intérêt et d'espoir.

Ainsi, cette thèse propose une solution basée sur l'apprentissage automatique et l'apprentissage profond pour améliorer la sécurité des réseaux IoT. Elle propose d'abord une classification du trafic IoT, comparant l'efficacité de divers modèles, dont les forêts aléatoires, XGBoost et les réseaux de neurones à mémoire à court-long terme (LSTM), pour différencier le trafic normal des attaques. Ensuite, elle utilise des autoencodeurs pour détecter des anomalies, démontrant leur capacité à identifier des comportements anormaux. Le modèle distingue le trafic normal du trafic malveillant avec une grande précision. Enfin, un Framework intelligent de détection d'intrusions à trois niveaux est développé. Ce dernier intégre un premier auto-encodeur au niveau 1 pour la détection d'anomalie, un deuxième auto-encodeur au niveau 2 pour classer l'anomalie détectée en anomalie connue ou anomalie nouvelle. Au niveau 3 est implémenté un perceptron multi-couches pour identifier les attaques détectées. Les performances de ces modèles sont évaluées sur des jeux de données expérimentaux. Les résultats obtenus démontrent une précision de 99%, 99,60% et presque 100% respectivement au niveau 1, niveau 2 et niveau 3. Les résultats de cette thèse confirment la robustesse des techniques d'apprentissage automatique et profond dans l'amélioration de la sécurité des réseaux IoT, avec des implications pratiques pour le déploiement de ces technologies dans des environnements réels.

Mots clés : IoT, Cyber-attaques, Apprentissage automatique, Apprentissage profond, Détection d'intrusions, Classification du trafic, Détection d'anomalies

1. Denial of service

Abstract

The internet of things (IoT) is an emerging technology that connects billions of devices across various fields, ranging from healthcare to smart cities, as well as industry and agriculture. However, this growing interconnectivity presents significant security challenges. IoT devices are often vulnerable to a variety of attacks, including denial of service (DoS) attacks, intrusions, and remote manipulations, which can lead to serious consequences for data confidentiality, integrity, and availability.

Implementing appropriate security mechanisms is essential to fully benefit from this technology, which often involves very sensitive application areas. Recent advances in artificial intelligence have provided solutions to many problems. Machine learning and deep learning methods continue to push the boundaries of AI. Their applications in the field of cybersecurity have generated much interest and hope.

Thus, this thesis proposes a solution based on machine learning (ML) and deep learning (DL) to enhance IoT network security. It first proposes an IoT traffic classification, comparing the effectiveness of various models, including random forests, XGBoost, and long short-term memory (LSTM) neural networks, to differentiate normal traffic from attacks. Then, it uses autoencoders to detect anomalies, demonstrating their ability to identify abnormal behaviours. The model distinguishes normal traffic from malicious traffic with high accuracy. Finally, an intelligent three-level intrusion detection framework is developed. This framework integrates a first autoencoder at level 1 for anomaly detection, a second autoencoder at level 2 to classify the detected anomaly as a known or new anomaly, and a multilayer perceptron at level 3 to identify the detected attacks. The performance of these models is evaluated on experimental datasets. The results obtained demonstrate an accuracy of 99%, 99.60%, and almost 100% at levels 1, 2, and 3 respectively. The results of this thesis confirm the robustness of ML and DL techniques in improving IoT network security, with practical implications for deploying these technologies in real-world environments.

Keywords : IoT, Cyber-attacks, Machine learning, Deep learning, Intrusion detection, Traffic classification, Anomaly detection

Table des matières

Remerciements i

Résumé iii

Abstract v

Table des matières ix

Liste des figures xiv

Liste des tableaux xv

Liste des Acronymes xvii

Introduction Générale 1

1 Généralités sur l'IoT 9

1.1 Introduction . . . 10
1.2 Les domaines d'applications IoT . . . 11
1.3 Architecture des systèmes IoT . . . 13
1.3.1 Architecture à trois couches . . . 13
1.3.2 Architecture à cinq couches . . . 14
1.3.3 Architecture à sept Couches . . . 15
1.4 Technologies et protocoles de communication IoT . . . 17
1.4.1 Réseaux à courte portée . . . 17
1.4.2 Réseaux à longue portée (LPWAN) . . . 19
1.4.3 Technologies cellulaires . . . 20
1.4.4 Protocoles applicatifs pour pour l'IoT . . . 23
1.5 Les vulnérabilités de sécurité dans l'IoT . . . 26
1.6 Les attaques courantes contre l'IoT . . . 29
1.7 Conclusion . . . 32

2 IA et Cybersécurité 35

2.1 Introduction . . . 37
2.2 Historique et évolution de l'IA . . . 38
2.3 Apprentissage automatique . . . 39
2.3.1 Apprentissage supervisé . . . 39
2.3.2 Apprentissage non supervisé . . . 41

2.3.3 Apprentissage par renforcement . . . 43
2.3.4 Exemple d'algorithmes d'apprentissage automatique . . . 44
2.4 Apprentissage profond . . . 46
2.4.1 Architecture des réseaux de neurones profonds . . . 47
2.4.2 Algorithmes et techniques avancées . . . 52
2.5 Les principes fondamentaux de la sécurité informatique . . . 57
2.5.1 Définition . . . 57
2.5.2 Principes de base de la sécurité informatique . . . 57
2.5.3 Défis de la sécurité dans le domaine de l'IoT . . . 58
2.6 Applications de l'IA en cybersécurité . . . 59
2.6.1 Détection d'anomalies . . . 59
2.6.2 Détection d'intrusions et classification d'attaques . . . 63
2.6.3 Identification des dispositifs IoT . . . 67
2.6.4 Discussion . . . 69
2.7 Conclusion . . . 70

3 Classification du trafic réseau IoT à l'aide d'apprentissage automatique 71
3.1 Introduction . . . 73
3.2 Brève présentation des modèles implémentés . . . 74
3.2.1 Arbre de décision . . . 75
3.2.2 Forêt aléatoire . . . 75
3.2.3 AdaBoost . . . 76
3.2.4 XGBoost . . . 77
3.2.5 Perceptron multicouche . . . 77
3.2.6 LSTM . . . 78
3.3 Le jeu de données expérimental . . . 79
3.3.1 Description du jeu de données . . . 79
3.3.2 Prétraitement des données . . . 81
3.4 Classification binaire . . . 83
3.4.1 Définition et métriques . . . 83
3.4.2 Résultats du modèle d'arbre de de décision binaire . . . 85
3.4.3 Résultats du modèle de forêt aléatoire binaire . . . 86
3.4.4 Résultats du modèle d'AdaBoost binaire . . . 87
3.4.5 Résultats du modèle de XGBoost binaire . . . 88
3.4.6 Résultats du modèle MLP binaire . . . 89
3.4.7 Résultats du modèle LSTM binaire . . . 90
3.5 Classification multi-classe . . . 93
3.5.1 Définition et métriques . . . 93

3.5.2 Résultats du modèle d'arbre de décision multi-classe 95
3.5.3 Résultats du modèle de forêt aléatoire multi-classe 96
3.5.4 Résultats du modèle d'AdaBoost multi-classe 97
3.5.5 Résultats du modèle de XGBoost multi-classe 98
3.5.6 Résultats du modèle MLP multi-classe 99
3.5.7 Résultats du modèle LSTM multi-classe 100
3.6 Comparaison des techniques de classification binaire et multi-classe 103
3.7 Conclusion 109

4 Détection d'anomalie au sein d'un réseau IoT à l'aide des auto-encodeurs 111
4.1 Introduction 112
4.2 Le jeu de données expérimental 113
4.3 Auto-encodeurs et détection d'anomalie 115
4.4 Implémentation d'un auto-encodeur 119
4.4.1 Auto-encodeur 1 121
4.4.2 Auto-encodeur 2 123
4.4.3 Auto-encodeur 3 125
4.5 Conclusion 128

5 Un Framework intelligent pour la sécurité des réseaux IoT : un système de détection d'intrusions à trois niveaux basé sur l'apprentissage profond 131
5.1 Introduction 132
5.2 Travaux connexes 133
5.3 Méthodologie 134
5.3.1 Jeu de données expérimental 134
5.3.2 Framework proposé 135
5.3.3 Les modèles implémentés 137
5.4 Résultats 141
5.4.1 Résultats du niveau 1 141
5.4.2 Résultats du niveau 2 144
5.4.3 Résultats du niveau 3 147
5.5 Conclusion 148

Conclusion Générale et Perspectives 151

Liste des Publications 153

Bibliographie 172

Table des figures

1.1 Applications IoT. Naxoo 2022 . . . 11
1.2 Architecture IoT à 3 couches. Marian-Vladut 2021 . . . 14
1.3 Architecture IoT à 5 couches. GeeksforGeeks 2022 . . . 15
1.4 Architecture IoT à 7 couches : modèle de référence du Forum Mondial de l'IoT. Bauer et al. 2013 . . . 16
1.5 Architecture réseau de LoraWan. Akpakwu et al. 2018 . . . 20
1.6 Architecture réseau Sigfox. Akpakwu et al. 2018 . . . 21
1.7 Evolution de la connectivité mobile. Digital Matter 2022 . . . 21
1.8 Architecture réseau de LTE-M. Borkar 2020 . . . 22
1.9 Architecture réseau de NB-IoT. Iqbal, Abdullah et Shabnam 2020 . . . 23
1.10 Nombre annuel d'attaques de logiciels malveillants liés à l'Internet des objets (IoT) dans le monde de 2018 à 2022 (Lemeš 2023) . . . 30

2.1 Exemple de déroulement d'un algorithme d'apprentissage supervisé. Le modèle est entraîné pour la reconnaissance des bananes et des pommes . . . 40
2.2 Exemple de déroulement d'un algorithme d'apprentissage non supervisé. Le modèle est entraîné sur des données non étiquetées. Il apprend lui même à trouver des similarité pour former des groupes. Dans cet exemple le modèle s'est basé sur la couleur pour créer les groupes. Ainsi, on a le groupe des "vert" et celui des "jaune". . . . 42
2.3 Le principe de l'apprentissage par renforcement . . . 44
2.4 Exemple d'un réseau de neurones avec trois couches cachées. . . . 48
2.5 Un neurone simple avec les différentes opérations . . . 48
2.6 Les fonctions d'activation les plus utilisées . . . 50

3.1 Exemples de classification . . . 73
3.2 Banc d'essai de Botnet IoT dataset. Source image, Koroniotis et al. 2018 . . . 80
3.3 Taxonomie des attaques dans Botnet IoT dataset . . . 81
3.4 Performances du modèle d'arbre de décision binaire avec : A) le rapport de classification et B) la matrice de confusion . . . 86
3.5 Performances du modèle de forêt aléatoire binaire avec : A) le rapport de classification et B) la matrice de confusion . . . 87
3.6 Performances du modèle AdaBoost binaire avec : A) le rapport de classification et B) la matrice de confusion . . . 88
3.7 Performances du modèle XGBoost binaire avec : A) le rapport de classification et B) la matrice de confusion . . . 89

3.8 Résultats du modèle mlp binaire avec : A) le rapport de classification et B) la matrice de confusion 90
3.9 Performances du modèle lstm binaire avec : A) le rapport de classification et B) la matrice de confusion 91
3.10 Analyse comparative des performances du modèles binaires en terme de f1-score et de durée d'entrainement et de test 92
3.11 Performances du modèle d'arbre de décision multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b) 96
3.12 Performances du modèle de forêt aléatoire multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b) 97
3.13 Performances du modèle d'AdaBoost multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b) 98
3.14 Performances du modèle de XGBoost multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b) 99
3.15 Performances du modèle MLP multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b) 100
3.16 Performances du modèle LSTM multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b) 101
3.17 Performances comparatives des six modèles multi-classes avec la métrique f1-score pour chacune des 5 classes 101
3.18 Comparaison des temps d'entrainement et de test des six modèles multi-classes 102
3.19 Performances des modèles binaires 103
3.20 Performances des modèles multi-classes 104
3.21 Format de la matrice des classificateurs multi-classes avec n classes 104
3.22 Format de la matrice des classificateurs binaires 104
3.23 Comparaison des matrices de confusion des modèles arbres de décision binaire et multi-classe aprés transformation. Dans la version binaire, l'arbre de décision présente un nombre de TP élevé et un nombre très faible de FP et FN. La version multi-classe montre une performance similaire avec des chiffres proches, indiquant une bonne généralisation aux classes multiples. 106
3.24 Comparaison des matrices de confusion des modèles de forêts aléatoires binaire et multi-classe aprés transformation. Le modèle binaire présente une performance satisfaisante avec un nombre élevé de TP et un faible nombre de FP et FN. La version multi-classe démontre également de bons résultats, bien que certains indicateurs montrent une légère dégradation par rapport au modèle binaire. 106

3.25 Comparaison des matrices de confusion des modèles Adaboost binaire et multi-classe aprés transformation. Dans le modèle binaire, AdaBoost a montré une précision élevée avec un nombre limité de FP et FN. La version multi-classe a des résultats contrastés, avec une diminution de la précision due à une augmentation du nombre de FN et FP. 107
3.26 Comparaison des matrices de confusion des modèles XGBoost binaire et multi-classe aprés transformation. Les performances du modèle binaire sont bonnes, avec un TP élevé et un nombre très faible de FP et FN. La version multi-classe conserve une performance élevée, montrant une bonne capacité à gérer plusieurs classes. . 107
3.27 Comparaison des matrices de confusion des modèles MLP binaire et multi-classe aprés transformation. Le modèle binaire MLP affiche une performance solide, avec un TP élevé et un faible nombre de FP et FN. Dans la version multi-classe, bien que la précision reste élevée, cependant les FP sont légèrement plus élevés. 108
3.28 Comparaison des matrices de confusion des modèles LSTM binaire et multi-classe aprés transformation. Le modèle binaire LSTM présente des résultats remarquables, avec un TP élevé et un nombre très faible de FP et FN. La version multi-classe conserve une performance élevée, indiquant une bonne capacité à traiter plusieurs classes tout en maintenant une précision élevée. 108

4.1 Le banc d'essai du jeu de données MQTT-IoT-IDS2020 (Hindy et al. 2020) . 114
4.2 Taxonomie des attaques dans MQTT-IoT-IDS2020 115
4.3 Exemple d'auto-encodeur avec ces trois parties mise en évidence 116
4.4 Exemple 1 d'une entrée bénigne avec sa reconstruction 116
4.5 Exemple 2 d'une entrée bénigne avec sa reconstruction 116
4.6 Exemple 1 d'une entrée malicieuse avec sa reconstruction 117
4.7 Exemple 2 d'une entrée malicieuse x avec sa reconstruction x' 117
4.8 Les performances de l'$auto-encodeur\ 1 avec les différents seuils$ 122
4.9 Variation des performances de l'$auto-encodeur\ 1$ en fonction du seuil . . 123
4.10 Les performances de l'$auto-encodeur\ 2$ avec les différents seuils 124
4.11 Variation des performances de l'$auto-encodeur\ 2$ en fonction du seuil . . 125
4.12 Les performances de l'auto-encodeur 3 avec les différents seuils 126
4.13 Variation des performances de l'$auto-encodeur\ 3$ en fonction du seuil . . 127

5.1 Le banc d'essai du jeu de données MQTTSet 135
5.2 Description du Framework proposé 138
5.3 Architecture de l'auto-encodeur implémenté comme MODEL 1 et MODEL 2. Source image, Nicholaus et al. 2021 139
5.4 Exemple d'architecture d'un MLP avec Trois couches cachées 140

5.5 MODEL 1 : Exemple d'entrée normale 141
5.6 MODEL 1 : Exemple d'entrée malicieuse 141
5.7 MODEL 1 : Taux de détection du trafic normal et du trafic malicieux, en fonction du seuil 142
5.8 MODEL 1 : Taux de détection globale du type de trafic (Accuracy), en fonction du seuil 143
5.9 MODEL 2 : Exemple d'attaque connue 144
5.10 MODEL 2 : Exemple d'attaque nouvelle 145
5.11 MODEL 2 : Taux de détection des attaque connues et des attaques nouvelles, en fonction du seuil 145
5.12 MODEL 2 : Taux de détection globale des deux types d'attaques (Accuracy), en fonction du seuil 146
5.13 MODEL 3 : Matrice de Confusion 147
5.14 MODEL 3 : Rapport de classification 147

Liste des tableaux

1.1 Tableau comparatif des technologies de communication IoT à courte portée 19
1.2 Protocoles applicatifs pour l'IoT : Comparaison détaillée (Saritha et Sarasvathi 2017) . . . 25
1.3 Liste des vulnérabilités et leurs descriptions selon OWASP IoT Top 10 2018 27
1.4 Les vulnérabilités de sécurité dans l'IoT . . . 29
1.5 Typologie des attaques l'IoT . . . 32
1.6 Classification des attaques en fonction des catégories . . . 32

4.1 Paramètres de la grille de recherche du nombre de neurones pour les différentes couches cachées symétriques . . . 120
4.2 Paramètres de la grille de recherche de la meilleure combinaison de fonction perte et d'optimiseur . . . 120

5.1 Proportions des types de données dans MQTTSet . . . 135
5.2 Variation de la précision globale (Accuracy) en fonction du seuil . . . 142
5.3 Performances de $MODEL\ 2$ en fonction du seuil . . . 145

Liste des Acronymes

2G	Deuxième génération de technologies de réseaux mobiles
3G	Troisième génération de technologies de réseaux mobiles
4G	Quatrième génération de technologies de réseaux mobiles
5G	Cinquième génération de technologies de réseaux mobiles
6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks
AES	Advanced Encryption Standard
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
AUC	Area Under Curve
AWS	Amazon Web Services
BoT-IoT	A dataset for IoT Botnet attacks
CASAGRAS	Coordination and Support Action (CSA) for Global RFID-related Activities and Standard
CICIDS	Canadian Institute for Cybersecurity Intrusion Detection System dataset
CNN	Convolutional Neural Network
CoAP	Constrained Application Protocol
DDS	Data Distribution Service
DDoS	Distributed Denial of Service
DL	Deep Learning
DNN	Deep Neural Network
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
GAN	Generative Adversarial Network
GSM	Global System for Mobile Communications
GRU	Gated Recurrent Unit
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IA	Intelligence Artificielle
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IPS	Intrusion Prevention Systems
ISO/IEC	International Organization for Standardization / International Electrotechnical Commiss
JIS	Japanese Industrial Standards
LPWAN	Low Power Wide Area Network
LoRaWAN	Long Range Wide Area Network

LTE	Long-Term Evolution
LTE-M	Long Term Evolution for Machines
LSTM	Long Short-Term Memory
M2M	Machine-to-Machine
MIMO	Multiple Input Multiple Output
MITM	Man-In-The-Middle
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrowband Internet of Things
NFC	Near Field Communication
NIDS	Network Intrusion Detection Systems
NSL-KDD	A dataset used for evaluating intrusion detection systems
QoS	Quality of Service
R2L	Remote to Local
RBM	Restricted Boltzmann Machine
REST	Representational State Transfer
RESTful	Representational State Transfer
RFID	Radio-Frequency Identification
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
RSA	Rivest–Shamir–Adleman encryption algorithm
SAP	Systems, Applications, and Products in Data
SIGFOX	Service de réseau LPWAN propriétaire pour la communication M2M et IoT
SMQTT	Secure MQTT
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS/SSL	Transport Layer Security / Secure Sockets Layer
UDP	User Datagram Protocol
U2R	User to Root
Wi-Fi	Wireless Fidelity
WiFi HaLow	IEEE 802.11ah standard for low power, long-range WiFi suitable for IoT
WLAN	Wireless Local Area Network
XMPP	Extensible Messaging and Presence Protocol
XSS	Cross-Site Scripting

Introduction Générale

Contexte et justification

L'Internet des Objets (IoT) représente une avancée technologique majeure qui connecte le monde physique et numérique de manière sans précédent. Les dispositifs IoT collectent, échangent et analysent des données pour automatiser et améliorer les processus dans divers secteurs, notamment la santé, l'industrie, l'agriculture, les villes intelligentes, et bien d'autres. Cette intégration de la technologie IoT permet non seulement d'optimiser l'efficacité opérationnelle mais également d'introduire de nouvelles méthodes de gestion et de prise de décision basées sur des données en temps réel (Khanna et Kaur 2020 ; Ding et al. 2020). Les dispositifs IoT utilisent une variété de capteurs et de technologies de communication pour interagir avec leur environnement. Par exemple, des capteurs de température, de pression, d'humidité, et des caméras peuvent être utilisés pour surveiller les conditions environnementales et collecter des données. Ces données sont ensuite transmises via des réseaux de communication, tels que Wi-Fi, Bluetooth, Zigbee, ou des réseaux cellulaires, à des serveurs où elles sont analysées pour fournir des informations utiles et des alertes en temps réel. Le potentiel de l'IoT réside dans sa capacité à transformer des données brutes en informations exploitables. Les applications de l'IoT sont vastes et incluent la gestion de la chaîne d'approvisionnement, la surveillance de la santé à distance, l'automatisation industrielle, la gestion de l'énergie, et les villes intelligentes. Par exemple, dans le domaine de la santé, les dispositifs IoT peuvent surveiller les signes vitaux des patients et alerter les médecins en cas d'anomalies. Dans l'industrie, les capteurs IoT peuvent surveiller les équipements pour détecter les pannes avant qu'elles ne se produisent, réduisant ainsi les temps d'arrêt et les coûts de maintenance. Les applications de l'IoT dans le domaine de la santé ont permis des avancées significatives dans la surveillance des patients et la gestion des maladies chroniques. Les dispositifs IoT, tels que les moniteurs de fréquence cardiaque, les glucomètres connectés et les capteurs de pression artérielle, permettent une surveillance continue des patients à domicile. Ces dispositifs envoient des données en temps réel aux professionnels de la santé, qui peuvent intervenir rapidement en cas d'anomalies. Par exemple, les patients atteints de maladies chroniques comme le diabète ou l'hypertension peuvent bénéficier d'une surveillance constante, ce qui réduit les risques de complications graves et améliore la qualité de vie.

Un autre exemple est l'utilisation de capteurs portables pour suivre les signes vitaux des

patients après une intervention chirurgicale. Ces dispositifs peuvent détecter des signes précoces d'infection ou d'autres complications post-opératoires, permettant une intervention rapide. En outre, les hôpitaux intelligents utilisent des dispositifs IoT pour suivre l'emplacement et l'état des équipements médicaux, améliorer la gestion des stocks et optimiser les opérations quotidiennes. L'IoT transforme également le secteur industriel en permettant une surveillance et une gestion plus efficaces des processus de production. Les capteurs IoT peuvent surveiller les machines et les équipements pour détecter des signes de défaillance imminente, permettant ainsi une maintenance prédictive. Cette approche réduit les temps d'arrêt imprévus et prolonge la durée de vie des équipements. Dans les usines intelligentes, les dispositifs IoT sont utilisés pour surveiller les conditions environnementales, telles que la température et l'humidité, qui peuvent affecter la qualité des produits. Les données collectées sont analysées en temps réel pour optimiser les conditions de production et garantir la qualité des produits finis. De plus, les systèmes de gestion de la chaîne d'approvisionnement basés sur l'IoT permettent une visibilité complète sur les stocks et les expéditions, améliorant ainsi l'efficacité et réduisant les coûts.

L'agriculture de précision est un autre domaine où l'IoT a un impact significatif. Les capteurs IoT sont utilisés pour surveiller les conditions du sol, la température, l'humidité et d'autres facteurs environnementaux qui influencent la croissance des cultures. Les agriculteurs peuvent utiliser ces données pour prendre des décisions éclairées sur l'irrigation, la fertilisation et la protection des cultures, ce qui améliore les rendements et réduit les coûts. Par exemple, des capteurs de sol peuvent mesurer l'humidité à différentes profondeurs et envoyer ces données à une plateforme de gestion agricole. Les agriculteurs peuvent ensuite ajuster leurs systèmes d'irrigation en fonction des besoins spécifiques de chaque champ, évitant ainsi le gaspillage d'eau et optimisant l'utilisation des ressources.

Les villes intelligentes utilisent l'IoT pour améliorer la qualité de vie des citoyens et optimiser les services municipaux. Des capteurs intelligents sont installés dans les infrastructures urbaines pour surveiller la qualité de l'air, la circulation, les niveaux de bruit, et la gestion des déchets. Ces données sont analysées pour prendre des décisions informées sur l'amélioration des services publics et la gestion des ressources. Par exemple, les systèmes de gestion du trafic basés sur l'IoT peuvent ajuster les feux de signalisation en temps réel pour fluidifier la circulation et réduire les embouteillages. Les capteurs de stationnement intelligents peuvent indiquer aux conducteurs les emplacements de stationnement disponibles, réduisant ainsi le temps passé à chercher une place et diminuant les émissions de gaz à effet de serre.

Cependant, cette interconnectivité accrue présente également des défis importants, en particulier en matière de sécurité. Les dispositifs IoT, souvent déployés dans des environ-

nements ouverts et variés, sont vulnérables à une multitude de menaces. Les attaquants peuvent exploiter ces vulnérabilités pour mener des attaques variées, allant du vol de données personnelles à la perturbation des services critiques. Ces problèmes sont aggravés par l'absence de normes de sécurité unifiées et la diversité des technologies et des protocoles utilisés. En outre, la confidentialité des données est une préoccupation majeure dans l'IoT. Les dispositifs IoT collectent souvent des données sensibles, telles que des informations médicales ou des données de localisation, qui doivent être protégées contre les accès non autorisés. Les pratiques de sécurité inadéquates, telles que l'utilisation de mots de passe par défaut ou de configurations de sécurité faibles, peuvent exposer ces données à des risques importants.

Problématique

L'Internet des Objets (IoT) introduit une complexité sans précédent dans le domaine de la sécurité informatique. Les vulnérabilités présentes dans les dispositifs et les réseaux IoT peuvent être exploitées pour diverses attaques, telles que les attaques par déni de service (DoS), les intrusions, et les manipulations à distance. Les conséquences de ces attaques peuvent être graves, affectant non seulement la confidentialité et l'intégrité des données mais aussi la disponibilité des services. Par exemple, une attaque sur des dispositifs médicaux connectés peut mettre en danger la vie des patients, tandis qu'une attaque sur des systèmes industriels peut entraîner des pertes financières significatives.

En premier lieu, les attaques par déni de service (DoS) visent à rendre un service ou un réseau indisponible en submergeant la cible de trafic superflu. Les attaques par déni de service distribué (DDoS) sont une forme particulièrement dangereuse de DoS, où un grand nombre de dispositifs compromis (bots) participent à l'attaque, rendant la défense beaucoup plus difficile. Le botnet Mirai, qui a émergé en 2016, est un exemple notable de la façon dont les dispositifs IoT peuvent être compromis et utilisés pour lancer des attaques massives. Mirai infecte les dispositifs IoT en utilisant des mots de passe par défaut ou faibles, puis les utilise pour générer un trafic réseau énorme destiné à submerger les cibles et les rendre inaccessibles.Les attaques Mirai ont causé des perturbations majeures, affectant des services en ligne tels que Twitter, Netflix, et Reddit. Ces attaques démontrent la vulnérabilité des dispositifs IoT aux attaques DDoS et l'importance de renforcer la sécurité pour prévenir de telles menaces. Les dispositifs IoT sont souvent connectés en permanence et peuvent être exploités à tout moment, ce qui en fait des cibles privilégiées pour les attaquants.

Par ailleurs, les intrusions dans les systèmes IoT peuvent permettre aux attaquants d'ac-

céder à des données sensibles et de compromettre l'intégrité et la confidentialité des informations. Les dispositifs IoT collectent souvent des données personnelles, telles que des informations médicales, des enregistrements audio et vidéo, et des données de localisation. Une fois compromis, ces dispositifs peuvent être utilisés pour espionner les utilisateurs, voler des informations sensibles, ou même manipuler des dispositifs pour causer des dommages physiques. Par exemple, une intrusion dans un système de sécurité domestique IoT pourrait permettre à un attaquant de désactiver les alarmes de sécurité et de surveiller les activités de la maison à distance. De même, les dispositifs médicaux connectés, tels que les pompes à insuline et les stimulateurs cardiaques, peuvent être manipulés pour administrer des doses incorrectes ou provoquer des défaillances critiques, mettant ainsi en danger la vie des patients.

En outre, les dispositifs IoT peuvent être manipulés à distance pour prendre le contrôle des systèmes et des infrastructures critiques. Les attaquants peuvent exploiter des vulnérabilités pour envoyer des commandes malveillantes aux dispositifs IoT, provoquant des dysfonctionnements ou des interruptions de service. Par exemple, une attaque sur des systèmes de contrôle industriel IoT pourrait perturber la production, causer des dommages matériels, ou même déclencher des incidents de sécurité graves. De plus, les véhicules connectés sont également vulnérables à la manipulation à distance.

Ces nombreuses vulnérabilités sont la conséquence de la négligence de l'aspect sécurité des dispositifs IoT lors de leur conception. De nombreux dispositifs sont commercialisés avec des configurations de sécurité par défaut qui sont faciles à exploiter. Par exemple, l'utilisation de mots de passe par défaut ou la non-application de mises à jour de sécurité expose ces dispositifs à des risques accrus. En outre, les dispositifs IoT sont souvent conçus pour être simples à utiliser, ce qui peut entraîner des compromis sur la sécurité. Les fabricants de dispositifs IoT doivent adopter des pratiques de conception plus sécurisée pour minimiser les vulnérabilités.

Cependant, étant donné que les dispositifs IoT sont souvent contraints par des capacités de calcul et de stockage limitées, il est difficile d'embarquer certains mécanismes de sécurité tels que les schémas de chiffrement très sophistiqués. Pour surmonter les limitations inhérentes des dispositifs IoT, il est crucial de mettre en place des solutions de sécurité innovantes basées sur les technologies émergentes telles que l'intelligence artificielle, notamment le machine learning et le deep learning. Ces technologies offrent des capacités avancées permettant de détecter les attaques les plus sophistiquées, capables de s'adapter et de réagir proactivement aux menaces. En utilisant des modèles d'apprentissage automatique pour analyser le trafic réseau et identifier les comportements anormaux, il est possible de compenser les faiblesses des dispositifs en matière de sécurité intégrée.

Objectifs

Dans cette thèse, l'objectif est d'élaborer une solution innovante qui vise à renforcer la sécurité dans les réseaux IoT. La mise en place d'une telle solution sera basée sur l'utilisation des techniques évoluées de l'intelligence artificielle, notamment l'apprentissage automatique et l'apprentissage profond.

Ainsi, à travers une combinaison harmonieuse des algorithmes d'apprentissage supervisé et des algorithmes d'apprentissage non supervisés, nous voulons développer un Framework intelligent pour la sécurité des réseaux IoT. Ce dernier devra analyser les messages échangés entre les objets pour détecter toute anomalie ou malveillance dans le trafic réseau. Il devra être capable de détecter, non seulement les attaques connues, mais aussi les attaques nouvelles dites "zero-day attacks", en anglais. En plus, les attaques connues, après leur détection, doivent être identifiées. Grâce cette double action, le Framework pourra contenir les attaques, empêcher leur propagation et limiter les dégâts.

Contributions

Les contributions de cette thèse sont structurées autour des trois points principaux qui mettent en évidence l'effectivité et l'efficacité des solutions apportées par l'apprentissage automatique et l'apprentissage profond dans le domaine de la sécurité des réseaux IoT.

- Le premier point est développé dans le chapitre 3. Ce dernier contribue de manière significative à la littérature en explorant six modèles de classification pour la détection de trafic malveillant au sein d'un réseau IoT. Pour chaque modèle, une version binaire et une version multi-classe sont implémentées. Ces modèles sont évalués et optimisés pour offrir les meilleures performances possibles dans divers environnements IoT. En outre, une analyse comparative est réalisée aussi bien entre les modèles mais aussi entre les approches binaire et multi-classe . Cette dernière met en évidence les forces et les faiblesses de chaque modèle, permettant de choisir le modèle optimal en fonction des exigences spécifiques de l'application et des caractéristiques des données. Les informations fournies par cette analyse sont précieuses pour les implémentations réelles(Diallo et al. 2022).

- La deuxième contribution se focalise sur la détection d'anomalie dans les réseaux IoT,

plus précisément dans un trafic MQTT. Pour cela, nous avons misé sur l'utilisation des autoencodeurs, en tant que modèles d'apprentissage non supervisé très réputés dans la détection d'anomalie, mais surtout dans la détection des attaques nouvelles dites « zero-day attacks ». La proposition de modèles d'autoencodeurs capables de reconstruire les données normales avec une précision élevée facilite la détection de tout écart par rapport au comportement normal, donc des anomalies. Trois modèles d'autoencodeurs sont implémentés. Chaque modèle est évalué avec cinq seuils de détection différents.Ces modèles sont conçus pour s'adapter aux spécificités des flux de données IoT, offrant une détection précise et fiable. Par ailleurs, une analyse détaillée de l'impact des différents seuils de décision sur les performances de détection fournit des lignes directrices pour optimiser ces paramètres dans des déploiements réels(Diallo et al. 2023a).

- La troisième et principale contribution est un Framework intelligent pour la sécurité des réseaux IoT. Ce Framework propose un système de détection d'intrusion à trois niveaux basé sur l'apprentissage profond. Il intègre des approches d'apprentissage supervisées et des approches d'apprentissage non supervisées pour offrir une solution de sécurité complète et robuste. Au premier niveau, un auto-encodeur est déployé pour détecter la présence d'intrusion. En passant au deuxième niveau, un autre auto-encodeur est utilisé pour classer l'intrusion comme une attaque connue ou nouvelle lorsqu'une intrusion se produit. En cas d'attaque reconnue, le troisième niveau entre en jeu. À ce stade, un perceptron multicouche (MLP) sert de classificateur multi-classe, permettant une identification précise du type d'attaque en cours. Des tests complets sont effectués à chacun des trois niveaux. Les résultats obtenus démontrent une précision de 99%, 99,60% et presque 100% respectivement au niveau 1, niveau 2 et niveau 3. Ce Framework combinant des techniques de détection d'anomalies et de classification multi-classe pour offrir une protection complète contre les menaces constitue une innovation importante. Cette approche hybride permet de bénéficier des avantages des deux types d'apprentissage, garantissant une sécurité accrue pour les réseaux IoT en perpétuelle évolution (Diallo et al. 2023b).

Organisation de la thèse

En dehors de l'introduction générale et la conclusion générale, cette thèse est organisée en cinq chapitres principaux, chacun traitant d'un aspect crucial de la sécurité des réseaux

IoT. Chaque chapitre est structuré pour fournir une compréhension approfondie du sujet, suivie d'une analyse des résultats et des discussions sur les implications et les futures recherches.

- Le premier chapitre fournit une vue d'ensemble de l'Internet des Objets, en expliquant ses origines, ses principes de fonctionnement, et son impact sur divers secteurs comme la sante, les villes intelligentes, l'industrie, l'agriculture, et l'énergie. Il explore les différentes architectures et technologies de communication utilisées dans les systèmes IoT, mettant en évidence les protocoles et les méthodes de connexion. Le chapitre aborde également les vulnérabilités de sécurité inhérentes a l'IoT et décrit les types d'attaques courantes, soulignant la nécessite de mesures de protection robustes pour assurer la sécurité et la fiabilité des dispositifs connectés.

- Le deuxième chapitre explore l'utilisation de l'intelligence artificielle (IA) dans le domaine de la cybersécurité, en se concentrant sur son évolution et ses applications, plus particulièrement dans la sécurité des réseaux IoT. Après une introduction a l'IA, décrivant ses capacités a simuler les processus intellectuels humains, le chapitre retrace l'historique de l'IA, des débuts théoriques aux avancées récentes en apprentissage automatique et apprentissage profond. La section suivante examine les techniques d'apprentissage automatique, y compris supervise, non supervise et par renforcement. L'accent est mis sur les réseaux de neurones profonds, sur les principales architectures telles que les réseaux de neurones convolutifs(CNN), les réseaux de neurones récurrents (RNN), les auto-encodeurs(AE), etc. Le chapitre conclut en mettant en lumière l'utilisation de l'IA dans la détection d'anomalies, la détection d'intrusion et la classification d'attaques, l'identification des dispositifs IoT, démontrant ainsi comment l'IA transforme les stratégies de défense contre les cybermenaces, notamment dans les environnements IoT.

- Dans le troisième chapitre, nous présentons des modèles de classification du trafic dans les réseaux IoT à l'aide des techniques d'apprentissage automatique et d'apprentissage profond. Après une introduction a la classification, six algorithmes sont présentés : arbre de décision, forêt aléatoire, AdaBoost, XGBoost, perceptron multicouche (MLP), et LSTM. Après l'entrainement, les performances des modèles sont évaluées pour la classification binaire (détection d'attaques) et multi-classe (identification des types d'attaque), utilisant des métriques telles que précision, rappel, F-score, et justesse. Une méthode novatrice de transformation des matrices de confusion permet

de comparer équitablement les modèles binaires et multi-classes, démontrant que les modèles de machine learning et de deep learning peuvent efficacement sécuriser les réseaux IoT contre les intrusions.

- Le quatrième chapitre présente une méthode de détection d'anomalie basée sur les auto-encodeurs. Le chapitre commence par une introduction qui campe le décor. La section qui suit l'introduction fait une description détaillée du jeu de donnée utilisée. Ensuite, nous faisons une brève présentation des auto-encodeurs, notamment dans le contexte de la détection d'anomalie. Il s'en suit une description de l'implémentation effective d'un modèle d'auto-encodeur. A cet effet, trois modèles d'auto-encodeurs sont implémentés et évalués avec cinq seuil de décision différents. Dans la conclusion de ce chapitre, une analyse comparative des résultats des trois modèles effectuée.

- Le cinquième et dernier chapitre aborde la mise en œuvre d'un Framework intelligent pour la sécurité des réseaux IoT. Ce dernier est base sur une approche de détection d'intrusion a trois niveaux utilisant des techniques de deep learning. Après une brève introduction, nous présentons quelques travaux connexes. Ensuite, nous effectuons une description détaillée de la méthodologie utilisée. Dans cette descriptions, nous présentons le jeu de données expérimental, le Framework proposé ainsi que les modèles implémentés. La section qui suit montrent les résultats. Le Framework est évalué par niveau. Pour chaque niveau, les performances sont évaluées. Ce chapitre se termine par une conclusion qui fait la synthèse du Framework proposé ainsi que celle des résultats obtenus.

Chapitre 1

Généralités sur l'IoT

1.1 Introduction 10
1.2 Les domaines d'applications IoT 11
1.3 Architecture des systèmes IoT 13
1.3.1 Architecture à trois couches 13
1.3.2 Architecture à cinq couches 14
1.3.3 Architecture à sept Couches 15
1.4 Technologies et protocoles de communication IoT 17
1.4.1 Réseaux à courte portée 17
1.4.2 Réseaux à longue portée (LPWAN) 19
1.4.3 Technologies cellulaires 20
1.4.4 Protocoles applicatifs pour pour l'IoT 23
1.5 Les vulnérabilités de sécurité dans l'IoT 26
1.6 Les attaques courantes contre l'IoT 29
1.7 Conclusion 32

1.1 Introduction

L'Internet des Objets (IoT) représente une avancée technologique majeure, intégrant le monde physique et le monde numérique de manière sans précédent. Cette technologie repose sur des dispositifs interconnectés qui collectent, échangent et analysent des données pour automatiser et améliorer les processus dans divers secteurs. L'IoT se distingue par sa capacité à fournir des solutions innovantes qui transcendent les limitations des systèmes traditionnels, offrant ainsi des opportunités inédites pour accroître l'efficacité, la sécurité et la durabilité. Le concept d'Internet des Objets (IoT), introduit par Kevin Ashton en 1998, désigne une infrastructure réseau globale reliant des objets physiques et virtuels. Selon la CASAGRAS [1], l'IoT est une extension de l'internet actuel qui intègre les objets via des technologies de connexion et de collecte de données, offrant une identification unique et des capacités de communication avancées pour des services et applications coopératifs. D'autre part, le SAP [2] voit l'IoT comme un monde où les objets physiques participent activement aux processus métiers, interagissant avec l'internet pour modifier leur état et les informations associées tout en respectant la confidentialité et la sécurité. En essence, l'IoT s'articule autour de deux définitions : un réseau mondial qui est une extension de l'internet pour de nouveaux appareils, et un paradigme qui fusionne le numérique avec le physique, permettant une coopération étendue entre utilisateurs et objets pour atteindre des objectifs divers (Serbanati, Medaglia et Ceipidor 2011).

Ce chapitre explore les fondements de l'IoT, en mettant en lumière ses applications variées, son architecture complexe et les technologies de communication qui sous-tendent son fonctionnement. En comprenant les domaines d'application de l'IoT, nous découvrons comment cette technologie transforme des secteurs aussi divers que la santé, l'industrie, l'agriculture, et les villes intelligentes. Nous examinerons également les différentes architectures proposées pour les systèmes IoT, ainsi que les protocoles de communication qui permettent la transmission de données entre les dispositifs.La diversité des technologies et protocoles IoT requiert une compréhension approfondie pour concevoir des systèmes robustes et efficaces. En abordant les vulnérabilités et les menaces spécifiques aux environnements IoT, ce chapitre vise à fournir une base solide pour appréhender les défis de sécurité inhérents à cette technologie émergente. Ainsi, nous pouvons mieux protéger et optimiser les infrastructures IoT, en assurant leur résilience et leur efficacité face aux cybermenaces croissantes.

1. Coordination and Support Action (CSA) for Global RFID-related Activities and Standardizations
2. Systems, Applications, and Products in Data

1.2 Les domaines d'applications IoT

L'Internet des Objets (IoT) marque une transformation profonde dans de nombreux secteurs, en redéfinissant radicalement les processus de collecte, de surveillance et d'analyse des données. Ce phénomène révolutionne la prise de décision en temps réel, avec un impact significatif sur la productivité, la sécurité et la durabilité. Cette section explore divers domaines où l'IoT est déjà en train de remodeler les pratiques traditionnelles, en introduisant de nouvelles méthodes pour optimiser et rendre plus efficaces les services et les opérations industrielles. En mettant en lumière ces transformations, nous soulignerons comment l'IoT s'impose comme un élément incontournable de l'évolution technologique et industrielle contemporaine(Asghari, Rahmani et Javadi 2019).

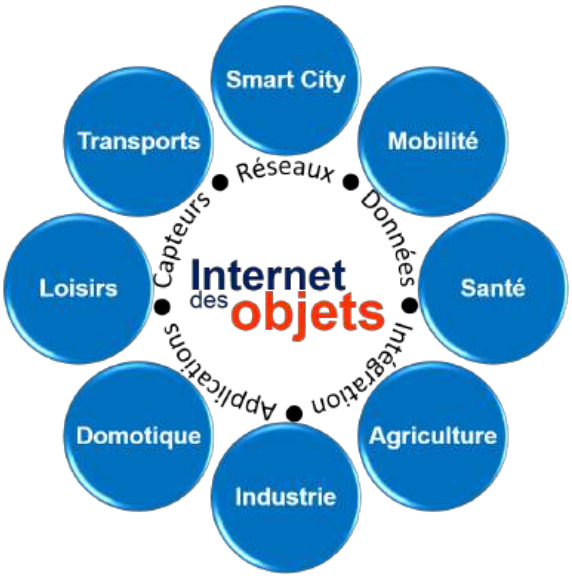


Figure 1.1 – Applications IoT. Naxoo 2022

— **Santé connectée**
L'IoT a transformé le secteur de la santé en facilitant la télésurveillance, ce qui permet aux médecins de suivre les conditions des patients à distance. Des dispositifs comme les bracelets intelligents et les moniteurs d'activité collectent des données vitales en temps réel, permettant une intervention médicale proactive. Ces technologies améliorent également la gestion des stocks de médicaments et la logistique hospitalière, rendant les traitements plus efficaces et réduisant les coûts. Cela inclut des innovations telles que les pilules connectées qui signalent leur ingestion et les appareils de surveillance à domicile qui alertent automatiquement les services d'urgence en cas d'anomalie critique (Baker, Xiang et Atkinson 2017).

— **Villes intelligentes**

Dans l'urbanisme, l'IoT facilite la création de villes plus intelligentes et durables en optimisant les services tels que l'éclairage public et la gestion des déchets, ce qui rend les villes plus écologiques et économiquement viables. Les capteurs IoT intégrés dans les infrastructures urbaines permettent une gestion intelligente du trafic, réduisant les embouteillages et la pollution. Les systèmes de gestion de l'eau intelligents détectent les fuites en temps réel, économisant des ressources précieuses. Ces systèmes sont essentiels pour répondre aux défis posés par l'urbanisation rapide et le changement climatique (Priya Dharshini et Ramya 2022).

— **Industrie 4.0**
L'IoT est au cœur de la quatrième révolution industrielle, permettant une automatisation avancée et une production plus flexible grâce à des usines intelligentes. Les capteurs et les machines connectés améliorent la chaîne de production en facilitant une maintenance prédictive, réduisant ainsi les temps d'arrêt et augmentant la productivité. L'intégration de l'IoT dans les processus industriels permet également une personnalisation en masse des produits, répondant ainsi mieux aux attentes des consommateurs (Okano 2017 ; Lampropoulos, Siakas et Anastasiadis 2019).

— **Agriculture intelligente**
L'IoT transforme l'agriculture en augmentant l'efficacité des exploitations agricoles grâce à une gestion précise des ressources. Les capteurs de sol et les drones agricoles fournissent des données précieuses sur l'état des cultures et les conditions environnementales, permettant une agriculture de précision. Cela aide à maximiser les rendements tout en minimisant l'utilisation d'eau, de fertilisants et de pesticides, ce qui est bénéfique pour l'environnement et économiquement avantageux pour les agriculteurs(Gondchawar, Kawitkar et al. 2016 ; Kassim 2020).

— **Gestion de l'énergie**
L'IoT joue un rôle crucial dans le secteur de l'énergie, notamment dans le développement et la gestion des réseaux électriques intelligents. Les appareils IoT aident à équilibrer la demande et l'offre d'énergie, à intégrer efficacement les énergies renouvelables et à améliorer la fiabilité du réseau en détectant et en répondant automatiquement aux pannes ou aux fluctuations de tension (Abir et al. 2021 ; Ahmad et Zhang 2021).

— **Transport intelligent**
L'intégration de l'IoT dans le secteur des transports conduit à des systèmes de transport plus intelligents et plus sûrs. Les véhicules connectés communiquent avec les infrastructures routières pour optimiser les itinéraires, réduire les accidents et améliorer l'efficacité énergétique. Les systèmes de transport public utilisent l'IoT pour le suivi en temps réel des véhicules et pour fournir aux passagers des informations pré-

cises sur les horaires et les itinéraires (Saarika, Sandhya et Sudha 2017 ; Oladimeji et al. 2023).

— **Maisons intelligentes**
L'IoT permet aux résidents de contrôler à distance et d'automatiser les appareils ménagers et les systèmes de sécurité, ce qui augmente le confort et l'efficacité énergétique. Les thermostats intelligents ajustent le chauffage et la climatisation en fonction des habitudes des utilisateurs et des conditions météorologiques, tandis que les réfrigérateurs connectés peuvent surveiller la fraîcheur des aliments et suggérer des recettes en fonction du contenu (Stojkoska et Trivodaliev 2017 ; Khan et al. 2018). Chacune de ces applications illustre comment l'IoT connecte le monde physique au réseau numérique, facilitant ainsi des innovations qui rendent nos environnements plus intelligents, plus sûrs, et plus efficaces. Ces exemples montrent l'impact profond et étendu de l'IoT à travers divers secteurs vitaux.

1.3 Architecture des systèmes IoT

L'architecture des systèmes IoT est complexe et se compose généralement de capteurs et d'actuateurs, de réseaux de communication, de systèmes de traitement des données et d'applications utilisateur. Nous discuterons de ces composants en détail, en soulignant leur rôle et leur interaction dans les systèmes IoT. Comme pour la définition, il n'existe pas pour l'IoT une seule architecture qui soit universellement admise. Beaucoup d'architectures proposées par différents auteurs figurent dans la littérature.

1.3.1 Architecture à trois couches

Elle a été introduite au tout début des recherches dans le domaine de l'IoT. Ces trois couches sont notamment la couche perception (perception layer), la couche réseau (network layer) et la couche application (application layer).

— **Couche de perception**
Cette couche se compose principalement de capteurs et d'actionneurs, responsables de la collecte de données du monde physique. Elle utilise des protocoles tels que RFID et NFC pour capturer et transmettre les informations. Cette couche est la fondation de l'interaction IoT, car elle établit le premier point de contact entre les dispositifs physiques et le système numérique (Banu et C 2017).

— **Couche de communication**

Cette couche est essentielle pour la transmission des données collectées à des systèmes plus avancés. Elle utilise divers protocoles de communication comme ZigBee, Bluetooth, Wi-Fi, et LTE pour assurer un transfert sécurisé et efficace des données vers des plates-formes de traitement ou directement aux utilisateurs (Jabraeil Jamali et al. 2020). Cette couche joue un rôle de pont et est critique pour maintenir l'intégrité et la disponibilité des données en transit.

— **Couche d'application**
La dernière couche utilise les données traitées pour fournir des services adaptés aux besoins spécifiques des utilisateurs. Les applications peuvent varier de la gestion de la maison intelligente à des solutions industrielles complexes, souvent soutenues par des plateformes cloud comme AWS IoT et Microsoft Azure. Cette couche est la face visible de l'IoT pour la plupart des utilisateurs, offrant des interfaces et des fonctionnalités directement exploitables (Banu et C 2017).

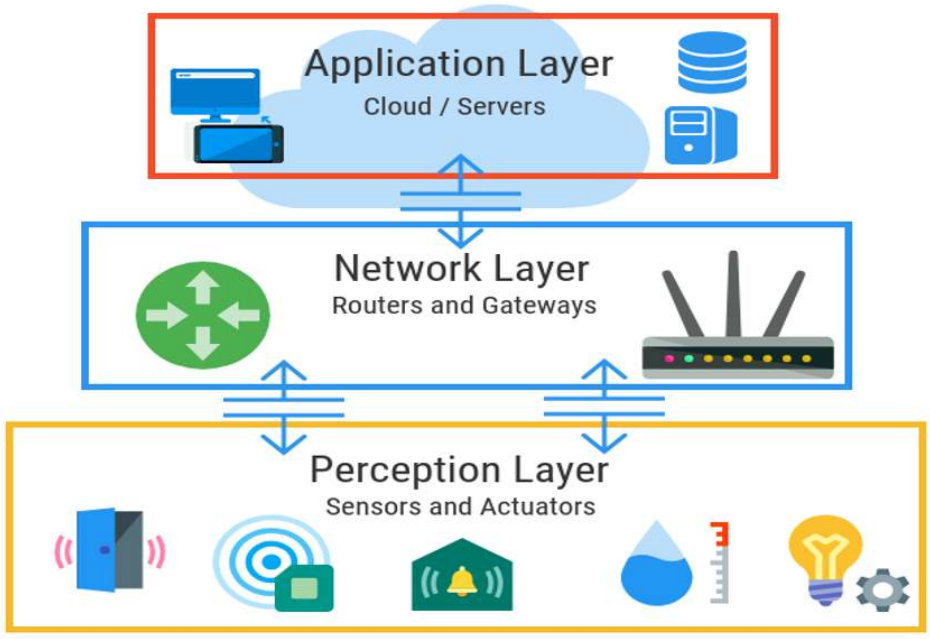


Figure 1.2 – Architecture IoT à 3 couches. Marian-Vladut 2021

1.3.2 Architecture à cinq couches

— **Couche de Perception et de Transport**
Elle étend la couche de perception traditionnelle pour inclure le transport des données, utilisant des protocoles plus robustes comme MQTT et CoAP, qui sont cruciaux pour une communication IoT efficace.

— **Couche de traitement (middleware)**

Elle agit comme une couche intermédiaire qui traite et analyse les données collectées, en utilisant des technologies de cloud computing et de big data. Cette couche joue un rôle clé dans la transformation des données brutes en informations exploitables, facilitant ainsi des décisions rapides et éclairées (El Hakim 2018).

— **Couche d'application et couche business**
Non seulement elle fournit des applications interactives basées sur les données traitées, mais cette couche gère également les aspects commerciaux et stratégiques des applications IoT. Cette double fonctionnalité est essentielle pour intégrer les services IoT dans les stratégies d'entreprise globales et pour exploiter les données à des fins analytiques et de prise de décision (El Hakim 2018).

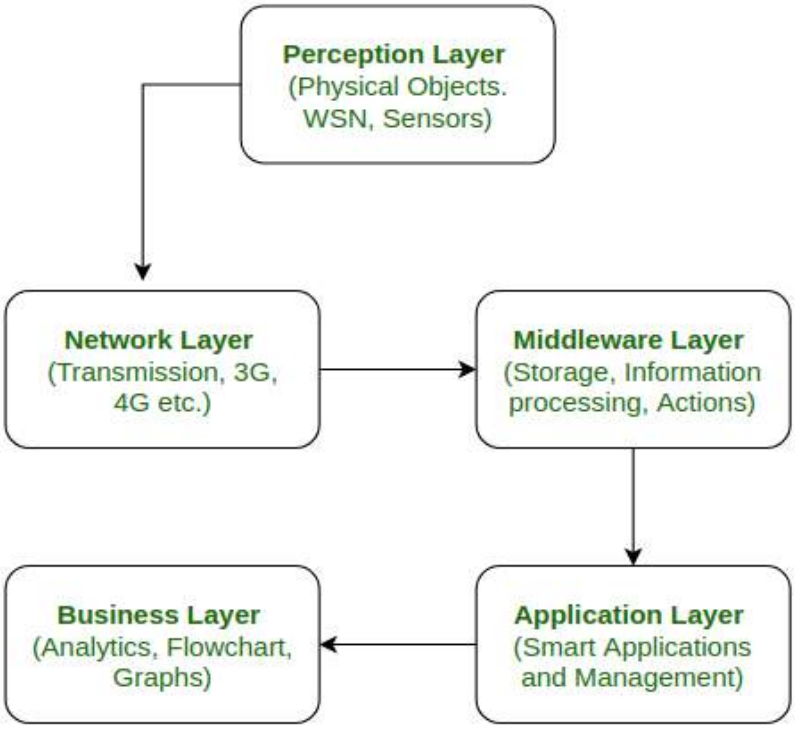


Figure 1.3 – Architecture IoT à 5 couches. GeeksforGeeks 2022

1.3.3 Architecture à sept Couches

— **Couche physique (Physical layer)**
Elle intègre les équipements physiques et les contrôleurs, incluant les dispositifs qui contrôlent ou sont contrôlés au sein de l'IoT, tels que les capteurs environnementaux, les actionneurs, et les contrôleurs industriels.

— **Couche de connectivité (Connectivity layer)**

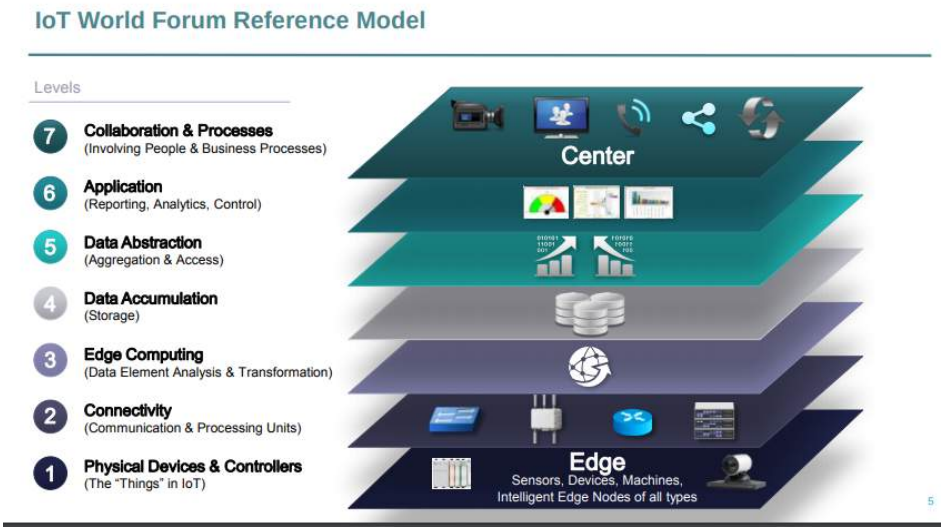


Figure 1.4 – Architecture IoT à 7 couches : modèle de référence du Forum Mondial de l'IoT. Bauer et al. 2013

Permet la communication entre les dispositifs IoT à travers divers équipements de réseau comme les routeurs, les commutateurs, et les passerelles. *Exemple :* Utilisation de technologies telles que Wi-Fi, Ethernet, et Bluetooth pour connecter les dispositifs IoT aux réseaux locaux ou étendus.

- **Couche d'informatique en périphérie (Edge computing L layer)**
  Cette couche Convertit les données réseau en informations utilisables pour le stockage et le traitement avancé, optimisant la gestion des données à la périphérie du réseau. *Exemple :* Traitement local des données pour réduire la latence et la bande passante utilisée vers les serveurs centraux.
- **Couche d'accumulation des données (Data accumulation layer)**
  Cette couche stocke les données collectées par les dispositifs IoT, facilitant l'accès et la gestion ultérieure des données. *Exemple :* Bases de données et solutions de stockage en cloud qui reçoivent et conservent les données traitées par la couche d'informatique de périphérie.
- **Couche d'abstraction des données (Data abstraction layer)**
  Elle agrège et formate les données pour les rendre accessibles aux applications de manière structurée et efficace. *Exemple :* Middleware qui normalise les données de formats multiples pour une utilisation cohérente dans les applications IoT.
- **Couche d'application (application layer)**
  Elle comprend les applications qui utilisent les données IoT pour fournir des fonctionnalités spécifiques, comme dans les domaines de la santé, de la domotique, etc. *Exemple :* Systèmes de gestion de l'énergie à domicile, plateformes de surveillance de la santé en temps réel.

— **Couche de collaboration et de processus (collaboration and processes layer)**
Elle facilite la collaboration entre différentes applications IoT pour améliorer la fonctionnalité et l'utilité des données collectées. *Exemple :* Intégration de systèmes de gestion de bâtiments intelligents avec des services de sécurité et d'urgence pour une réponse coordonnée en cas d'incident.

Cette architecture en sept couches illustre la complexité et la modularité nécessaires pour soutenir un large éventail d'applications IoT, en offrant une structure qui facilite non seulement la communication et le traitement des données, mais aussi l'intégration inter-applicative et la collaboration.

Les architectures à trois, cinq et sept couches présentent des avantages variés en termes de flexibilité, complexité, coût, et sécurité, adaptées à différents besoins d'application. Le choix de l'architecture appropriée doit être guidé par les exigences spécifiques de l'application envisagée, y compris les besoins de traitement des données, la réactivité nécessaire et les considérations de sécurité. Une compréhension approfondie des différentes architectures et des protocoles qu'elles utilisent permet de concevoir des systèmes IoT robustes et sécurisés, adaptés à une large gamme d'applications (Drăgulinescu et al. 2020).

1.4 Technologies et protocoles de communication IoT

L'Internet des Objets (IoT) repose sur une diversité de protocoles et de technologies de communication, chacun étant choisi en fonction des exigences spécifiques du réseau et du domaine d'application. Cette diversité permet d'adapter la solution technologique aux besoins en termes de portée, consommation énergétique, capacité de données et sécurité (Herrero 2022 ; Stiller, Schiller et Schmitt 2021).

1.4.1 Réseaux à courte portée

Il existe une pléthore de technologies IoT utilisées dans ces réseaux. Ces dernières sont surtout caractérisées par leur faible rayon de couverture avec quelques différences. Ces technologies sont cruciales pour les applications où les dispositifs sont physiquement proches les uns des autres, comme dans les environnements domestiques ou industriels (Stiller et al. 2020 ; Sarawi et al. 2017).

— **NFC (Near field communication)**
NFC est essentiellement utilisé pour des interactions sécurisées telles que les paie-

ments mobiles et l'appairage rapide d'appareils, grâce à une communication rapide et sécurisée qui requiert simplement de rapprocher deux appareils. Cette proximité nécessaire, typiquement limitée à quelques centimètres, rend le NFC idéal pour les transactions sécurisées, minimisant ainsi les risques de détournement à distance. Opérant sur la fréquence de 13.56 MHz, NFC peut fonctionner dans des modes actif et passif, ce qui permet d'économiser de l'énergie dans les applications IoT et d'augmenter l'efficacité des appareils.

— **Bluetooth**
La dernière évolution, Bluetooth 5.0, inclut des améliorations significatives en termes de portée et de débit, permettant des connexions plus stables sur de plus grandes distances et la gestion d'un plus grand volume de données. Ces caractéristiques rendent Bluetooth 5.0 parfait pour les applications qui nécessitent une faible consommation énergétique et un transfert de données continu, telles que les appareils de santé connectée ou les accessoires portables. Sa capacité à gérer efficacement un grand volume de données est particulièrement essentielle pour les appareils IoT dans les environnements domestiques et médicaux, où la fiabilité et la performance sont cruciales.

— **ZigBee**
ZigBee est particulièrement adapté pour les réseaux maillés qui nécessitent une communication fiable entre un grand nombre d'appareils à faible consommation. Cette technologie est utilisée pour créer des réseaux complexes où chaque nœud peut non seulement transmettre mais aussi relayer des informations, ce qui est crucial pour maintenir l'intégrité du réseau même avec de nombreux nœuds. Sa faible consommation énergétique combinée avec sa capacité à supporter un grand nombre de nœuds rend ZigBee idéal pour les applications de domotique et de surveillance industrielle, où des réseaux fiables et efficaces sont nécessaires pour une gestion optimale et sécurisée des environnements.

— **Z-Wave**
Similaire à ZigBee en termes de formation de réseaux maillés, Z-Wave utilise cependant une bande de fréquence de 900 MHz, qui est moins susceptible aux interférences comparée aux technologies opérant dans la bande de 2.4 GHz. Cette caractéristique rend Z-Wave particulièrement fiable pour les applications domestiques automatisées où la fiabilité est cruciale. De plus, Z-Wave est hautement modulaire et bénéficie d'une large interopérabilité entre différents appareils et fabricants, ce qui le rend idéal pour une intégration dans divers environnements de domotique.

— **Wi-Fi**
Le WiFi, standardisé par l'IEEE 802.11, est essentiel pour les réseaux locaux sans fil (WLAN), offrant des connexions à large bande pour les appareils vers Internet avec

une couverture étendue et des débits élevés. Depuis son introduction, le WiFi a évolué pour répondre aux besoins croissants de débit et de capacité. Les premières normes, IEEE 802.11a et 802.11b (1999), permettaient des débits allant jusqu'à 54 Mbps et 11 Mbps respectivement. En 2003, l'IEEE 802.11g a atteint 54 Mbps en 2.4 GHz. Face à la demande croissante, de nouvelles générations, IEEE 802.11n (2008) et IEEE 802.11ac (2014), ont été développées, offrant des débits jusqu'à 600 Mbps et 7 Gbps grâce à des modulations denses et la technologie MIMO. En 2017, l'IEEE 802.11ah (WiFi HaLow) a été introduit pour l'IoT, avec une couverture étendue, faible consommation d'énergie, et un débit maximal de 300 Mbps sur des bandes passantes de 1 MHz à 16 MHz.

— **6LoWPAN**
Le Wi-Fi est souvent utilisé pour les applications nécessitant la transmission de grandes quantités de données, telles que les caméras de sécurité IoT, en raison de sa haute capacité de débit et de sa portée étendue par rapport à d'autres protocoles de réseaux à courte portée. Bien qu'il consomme plus d'énergie, le Wi-Fi offre une portée et des débits supérieurs, rendant cette technologie particulièrement adaptée pour les applications qui demandent un transfert de données volumineux et continu, comme la surveillance vidéo.

Tableau 1.1 – Tableau comparatif des technologies de communication IoT à courte portée

Technologie	Norme	Frequence	Réseau	Puissance	Débit	Portée	Securité
NFC	ISO/IEC 14443, JIS X-6319-4	13.56 MHz	P2P	50 mA	424 kbps	10 cm	RSA, AES
Bluetooth	IEEE 802.15.1	2.4 GHz	WPAN	30 mA	1 Mbps	15-30 m	E0, AES-128
ZigBee	IEEE 802.15.4	2.4 GHz	WPAN	30 mA	250 kbps	10-100 m	AES
Z-Wave	Z-Wave	900 MHz	WPAN	2.5 mA	100 kbps	30-100 m	AES-128
Wi-Fi	802.11n	2.4 GHz and 5 GHz	WLAN	High consumption	1 Mbps	20-100 m	WPA, WEP
6LoWPAN	IEEE 802.15.4	868 MHz (EU), 915 MHz (USA), 2.4 GHz (Global)	WPAN	Low consumption	250 kbps	10-100 m	AES

1.4.2 Réseaux à longue portée (LPWAN)

Ces technologies sont conçues pour des communications à faible fréquence sur de longues distances, optimales pour le suivi des actifs et les applications agricoles. Dans cette catégorie, nous retrouvons entre autre deux technologies d'origine française bien connues du monde de l'IoT que sont SIGFOX et LoRaWAN (Low Range Wide Area Network). Il s'agit de 2 technologies à longue portée et faible consommation (Drăgulinescu et al. 2020).

— **LoRaWAN**
LoRaWAN est conçu pour la communication à longue portée avec une faible consommation d'énergie, ce qui le rend idéal pour les applications agricoles et de surveillance

environnementale où les dispositifs doivent fonctionner sur batterie pendant de longues périodes. Distingué par sa portée étendue et sa faible consommation énergétique, LoRaWAN supporte également bien les environnements urbains denses grâce à son immunité aux interférences. Cette technologie est spécialement adaptée pour les applications nécessitant une communication bidirectionnelle sur de longues distances sans la nécessité d'une grande bande passante.

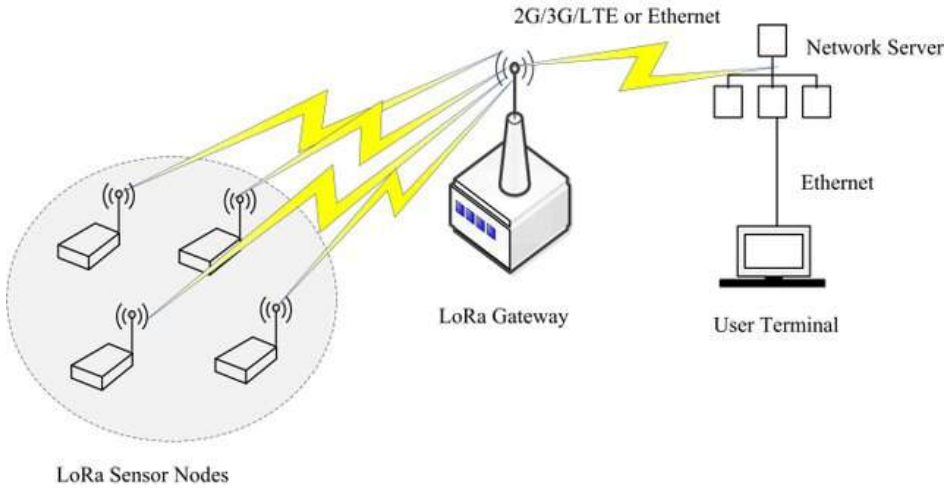


Figure 1.5 – Architecture réseau de LoraWan. Akpakwu et al. 2018

— **SigFox**
SigFox offre une couverture globale avec une consommation très faible d'énergie, ce qui en fait une excellente solution pour les applications nécessitant une communication de données minimes sur de vastes zones géographiques, telles que les applications de suivi des actifs à travers de longues distances. Cependant, sa capacité de transmission de données est limitée, ce qui peut rendre SigFox moins idéal pour les applications nécessitant des échanges de données fréquents ou volumineux. Malgré ces limitations, SigFox reste une option privilégiée pour les contextes où la simplicité et l'efficacité de la couverture sont prioritaires.

1.4.3 Technologies cellulaires

A ces technologies que nous venons de voir, on pourrait ajouter les réseaux cellulaires GSM (2G, 3G, 4G, 5G) qui peuvent constituer une solution avec des débits plus importants. Mais le besoin en énergie de ces réseaux rend leur utilisation très limitée dans le domaine de l'IoT. Cependant il faut signaler que les opérateurs de télécommunication sont en train de s'aligner pour intégrer le domaine de l'IoT en proposant des technologies plus adaptées comme le LTE-M et le NB-IoT. Ces deux technologies fonctionnent sur les bandes de fréquences 4G déjà attribués aux opérateurs. Ces technologies continuent d'évoluer pour

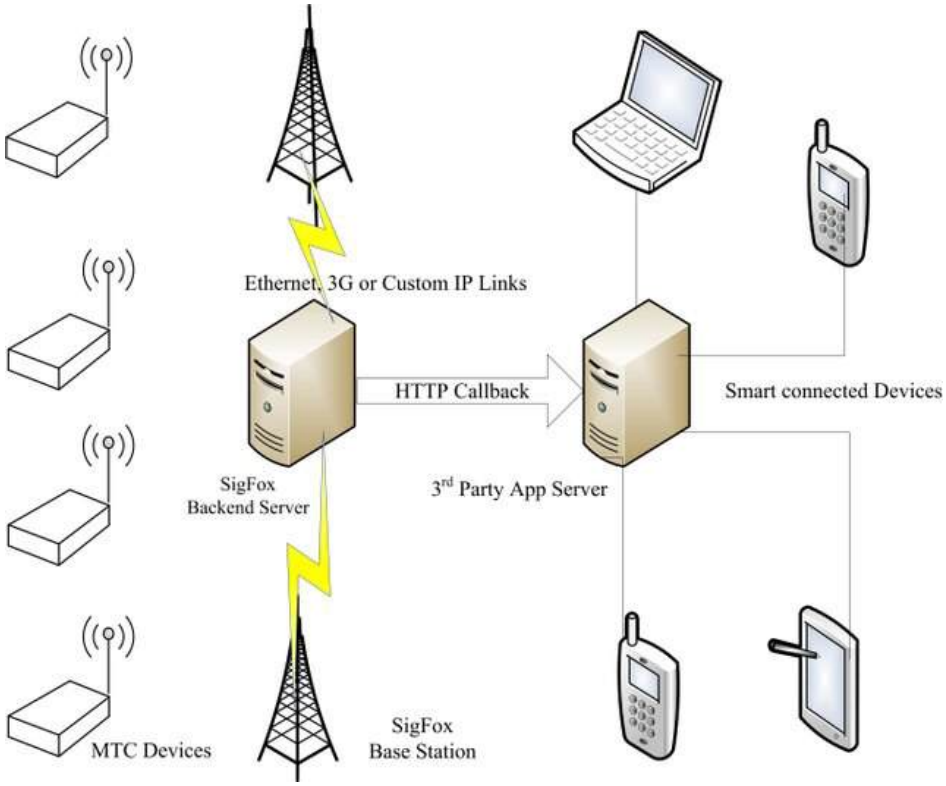


Figure 1.6 – Architecture réseau Sigfox. Akpakwu et al. 2018

mieux répondre aux exigences des applications IoT en offrant un équilibre entre portée, bande passante et consommation d'énergie.

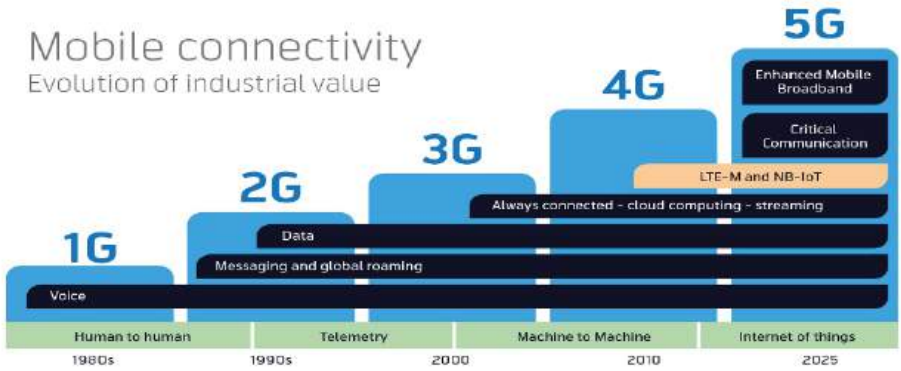


Figure 1.7 – Evolution de la connectivité mobile. Digital Matter 2022

— **LTE-M**

Le LTE-M, une des technologies cellulaires avancées, est apprécié pour sa faible consommation d'énergie et sa capacité à fournir une communication efficace sur de longues distances. Contrairement aux systèmes cellulaires traditionnels, le LTE-M améliore la pénétration du signal à l'intérieur des bâtiments et soutient une grande variété

d'applications IoT sans nécessiter une grande quantité de données. LTE-M dispose des avantages des technologies de la téléphonie mobile (Connectivité en temps réel, itinéraire à l'international). Le débit important, jusqu'à 10Mbps, permet d'envisager cette solution pour le transport d'image ou de la voix avec l'inconvénient d'être plus énergivore. Cette technologie est idéale pour les applications telles que les capteurs d'environnement, les dispositifs de suivi des actifs et la surveillance à distance.

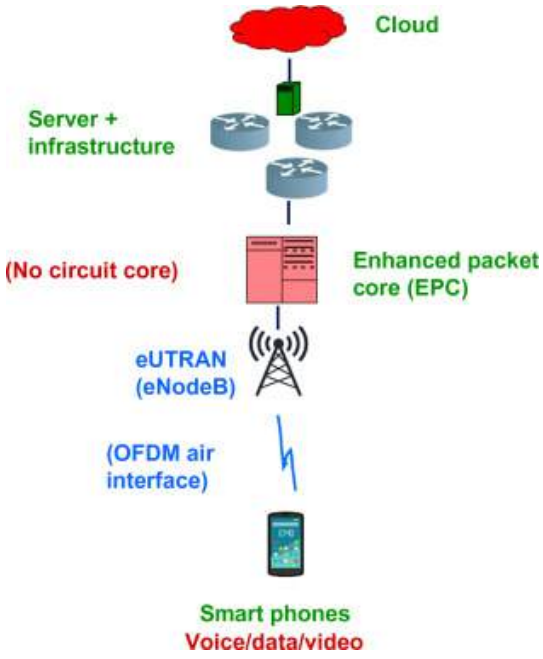


Figure 1.8 – Architecture réseau de LTE-M. Borkar 2020

— **NB-IoT**

D'autre part, le NB-IoT, spécialisé dans les connexions de faible puissance, est reconnu pour sa profonde couverture et son efficacité opérationnelle dans des zones densément peuplées. Le NB-IoT est optimisé pour des applications qui transmettent de petites quantités de données de manière intermittente. NB-IoT se rapproche plus des protocoles SigFox et Lora en termes d'autonomie. Son débit est limité à 250Kbit/s avec une portée de 1km en ville et de l'ordre de 10km en milieu rural. En raison de sa bande étroite, il assure une consommation d'énergie extrêmement réduite, ce qui prolonge la durée de vie des batteries des dispositifs IoT implantés dans des environnements où il est difficile de remplacer ou de recharger les batteries, comme dans les systèmes de gestion des déchets, l'agriculture intelligente, et les applications de surveillance de l'eau.

La poursuite de l'innovation dans les technologies cellulaires pour l'IoT promet de transformer radicalement la manière dont les industries fonctionnent, en permettant une meilleure

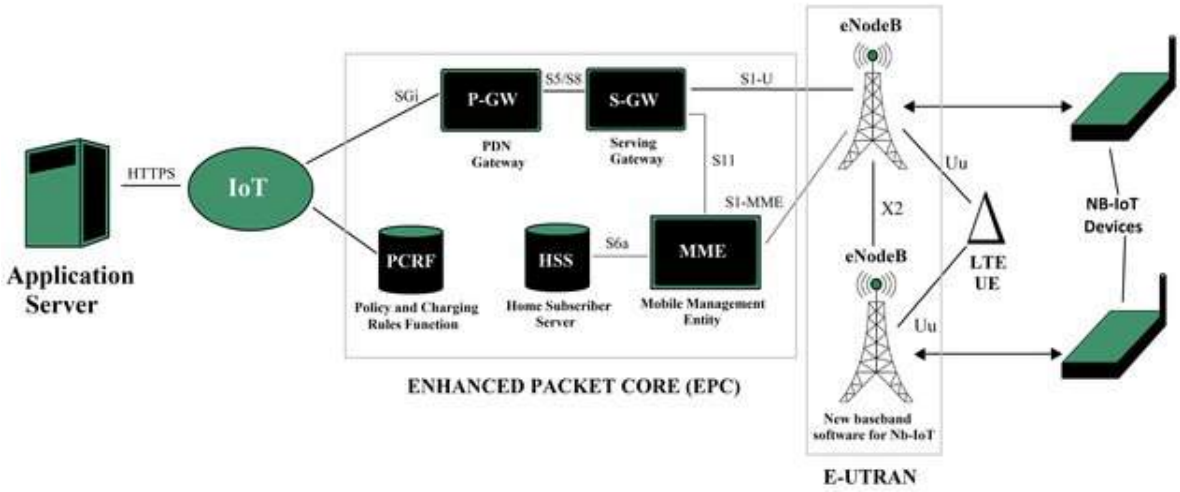


Figure 1.9 – Architecture réseau de NB-IoT. Iqbal, Abdullah et Shabnam 2020

collecte de données et en améliorant significativement la connectivité des dispositifs. À mesure que ces technologies continuent de mûrir, nous pouvons nous attendre à voir une adoption accrue dans des domaines divers allant de l'agriculture intelligente à la ville intelligente, en passant par la gestion de l'énergie et la santé publique.

1.4.4 Protocoles applicatifs pour pour l'IoT

La couche application est responsable du formatage et de la présentation des données. Elle est généralement basée sur HTTP. Cependant, HTTP ne convient pas toujours dans les environnements à ressources limitées car il est de nature assez verbeuse et entraîne donc une grande analyse aérienne. De nombreux protocoles alternatifs ont été développés pour les environnements IoT tels que CoAP, MQTT, AMQP, XMPP... Ces protocoles sont conçus pour s'exécuter sur des réseaux utilisant TCP, à l'exception de CoAP qui utilise UDP comme transport sous-jacent. MQTT et AMQP implémentent un modèle de publication / abonnement, tandis que CoAP implémente un modèle d'interaction requête / réponse (Hamid et Alisa 2021 ; Karagiannis et al. 2015).

— **HyperText transfer protocol (HTTP)**
Le protocole HTTP est largement utilisé pour les systèmes où la compatibilité avec le web est cruciale. Cependant, il est considéré comme trop verbeux pour certains dispositifs IoT en raison de la surcharge qu'il introduit. HTTP utilise un modèle de requête/réponse entre les clients et les serveurs, ce qui peut introduire une surcharge significative, surtout lorsque les dispositifs communiquent des messages fréquents et de petite taille. Toutefois, son ubiquité assure une intégration facile avec l'infrastruc-

ture Internet existante, facilitant les interactions entre les dispositifs IoT et les applications web. Les améliorations de sécurité sont généralement gérées par TLS/SSL, ajoutant une couche de communication sécurisée dans les écosystèmes IoT.

— **Constrained application protocol (CoAP)**
CoAP est spécialement conçu pour les dispositifs à ressources limitées typiques des configurations IoT. Il opère sur UDP pour minimiser la surcharge et prend en charge les méthodes similaires à HTTP (GET, POST, PUT, DELETE) pour faciliter l'intégration web tout en répondant aux besoins des dispositifs à faible puissance. CoAP inclut un support pour la découverte et les capacités multicast, ce qui le rend idéal pour les interactions dispositif-à-dispositif localisées dans des scénarios comme les maisons intelligentes ou l'automatisation industrielle. De plus, il offre un support intégré pour les échanges de messages asynchrones et observe la fiabilité par des messages simples confirmables et non-confirmables.

— **Message queuing telemetry transport (MQTT)**
MQTT est un protocole de publication/abonnement optimisé pour les réseaux à faible bande passante et haute latence typiques de nombreux environnements IoT. Il est conçu pour offrir une distribution de messages fiable et efficace dans la communication machine-à-machine (M2M), avec une consommation minimale de batterie et de bande passante. MQTT est hautement évolutif, prenant en charge des milliers de dispositifs dans un seul réseau. Il utilise trois niveaux de qualité de service (QoS) pour garantir la livraison des messages selon les besoins de l'application.

— **Extensible messaging and presence protocol (XMPP)**
XMPP offre une solution complète pour la communication en temps réel, avec des mécanismes pour la présence, la découverte et le chat de groupe. Initialement conçu pour les services de messagerie, son extension pour l'IoT prend en charge des besoins de messagerie complexes et des interactions multiparties. L'architecture XMPP facilite l'extensibilité et l'intégration avec d'autres systèmes, bien qu'elle puisse imposer une surcharge plus élevée en raison de son format XML verbeux.

— **Data distribution service (DDS)**
DDS est un protocole de middleware pour les systèmes en temps réel qui nécessitent des échanges de données à haute performance. Il est construit pour supporter de grandes architectures de publication-abonnement avec des exigences strictes en matière de timing et de débit, telles que celles trouvées dans les secteurs automobile, aérospatial et de la défense.

— **Advanced message queuing protocol (AMQP)**
AMQP prend en charge la messagerie fiable et sécurisée avec une architecture de routage flexible, ce qui le rend adapté pour les applications IoT de niveau entreprise

où l'interopérabilité et la robustesse sont cruciales.

— **Secure MQTT (SMQTT)**
SMQTT est une extension de MQTT avec des fonctionnalités de sécurité améliorées conçues pour fournir des échanges de messages cryptés dans des environnements susceptibles d'être écoutés et où la confidentialité des données est une préoccupation.

— **WebSocket**
WebSocket fournit des canaux de communication full-duplex sur une seule connexion longue durée, permettant aux dispositifs IoT de faciliter des interactions en temps réel. Il réduit la surcharge, minimise la latence par rapport au polling HTTP et est particulièrement efficace dans des scénarios nécessitant un échange de données constant (Yassein, Shatnawi et al. 2016).

— **Architecture RESTful** L'architecture RESTful utilise des méthodes HTTP standard et est connue pour sa simplicité et son efficacité pour s'intégrer dans l'environnement web. Bien qu'elle prenne en charge la communication sans état, qui s'aligne bien avec la nature sans état d'internet, elle peut ne pas être optimale pour toutes les applications IoT en raison de la surcharge associée à ses transactions HTTP sans état.

Tableau 1.2 – Protocoles applicatifs pour l'IoT : Comparaison détaillée (Saritha et Sarasvathi 2017)

Protocole	Transport	QoS	Architecture	Sécurité
HTTP	TCP	Non	Requête/Réponse	TLS/SSL
CoAP	UDP	Oui	Requête/Réponse	DTLS
MQTT	TCP	Oui	Publication/abonnement	TLS/SSL
SMQTT	TCP	Oui	Publication/abonnement	TLS/SSL
XMPP	TCP	Non	Requête/Réponse, Publication/abonnement	TLS/SSL
AMQP	TCP	Oui	Publication/abonnement	TLS/SSL
WebSocket	TCP	Oui	Client/Serveur	TLS/SSL
RESTful	HTTP	Non	Requête/Réponse	HTTPS

L'IoT continue d'évoluer, intégrant de nouvelles technologies et améliorant les protocoles existants pour mieux répondre aux besoins de connectivité, d'efficacité énergétique et de sécurité des données. En comprenant les caractéristiques et les applications optimales de chaque technologie, les développeurs et les chercheurs peuvent mieux concevoir des solutions IoT qui améliorent les services dans des domaines aussi divers que la santé, l'industrie et le domestique.

1.5 Les vulnérabilités de sécurité dans l'IoT

Cette section traitera des défis de sécurité auxquels sont confrontés les systèmes IoT, notamment la gestion des identités et l'accès, la protection des données et la résilience des dispositifs face aux attaques. Contrairement à l'environnement informatique traditionnel où les systèmes sont dotés d'une sécurité physique appropriée, les éléments de l'IoT sont souvent exposés sans surveillance. Avoir un accès physique à un appareil permet à un attaquant de modifier les paramètres de configuration dans les cas d'émission d'une nouvelle demande d'appariement d'appareil, de réinitialisation de l'appareil aux paramètres d'usine, de génération d'un nouveau mot de passe ou d'installation de certificats SSL (Secure Sockets Layer) personnalisés pour rediriger le trafic à un autre serveur qui leur appartient. De plus, l'énorme variété des appareils IoT et l'hétérogénéité des technologies utilisées pose un sérieux problème d'interopérabilité. L'absence de norme commune IoT entre les différents constructeurs est un obstacle pour la sécurité IoT (Jurcut et al. 2020).

En sécurité informatique, une vulnérabilité est une faiblesse ou une faille du système qui pourraient être exploitées par un individu malintentionné pour violer la politique de sécurité et mener des actions compromettantes. Les vulnérabilités peuvent être trouvées à différents niveaux des systèmes IoT. Les vulnérabilités peuvent être identifiées dans le matériel et les logiciels du système. Elles peuvent se trouver également dans les politiques et procédures de sécurité des systèmes et réseaux, mais aussi au niveau des utilisateurs (Kizza 2009). La multiplication des appareils IoT a révélé beaucoup de vulnérabilités de sécurité et a entrainé des défis de sécurité importants dans ce domaine.

L'Open Web Application Security Project (OWASP) a identifié 10 principales vulnérabilités susceptibles de compromettre la sécurité des appareils et des écosystèmes IoT. La liste des 10 principales vulnérabilités de l'OWASP IoT a été élaborée et présentée dans un document intitulés OWASP IoT Top 10 2018 grâce à une méthodologie complète qui a analysé les incidents de sécurité IoT dans le monde réel (Infosec 2023) . Ces différentes vulnérabilités sont décrites dans le tableau qui suit. Ces 10 principales vulnérabilités de l'OWASP IoT fournissent des informations précieuses sur les failles de sécurité potentielles qui doivent être corrigées.

Sans pour remettre en cause les travaux de l'OWASP IoT top 10 2018, une équipe indienne dénommée Payatu a publié, à son tour, un top 10 des vulnérabilité dans l'IoT. Ils s'agit des informations sensibles codées en dur , des ports de débogage matériel activés , du firmware non sécurisé , du stockage de données non sécurisé, d'une authentification insuffisante, d'une communication non sécurisée, d'une configuration non sécurisée, d'un

Tableau 1.3 – Liste des vulnérabilités et leurs descriptions selon OWASP IoT Top 10 2018

Vulnérabilités	Description
Mot de passe faibles, devinables ou codés en dur	Des mots de passe et des noms d'utilisateurs sont souvent codés en dur avec des informations d'identification. (Facile à deviner par force brute)
Services réseau non sécurisés	Les services réseau non sécurisés font référence aux vulnérabilités des protocoles, services ou configurations réseau et comprennent généralement des protocoles de communication non chiffrés, des configurations de sécurité réseau médiocres et l'utilisation de logiciels obsolètes ou vulnérables.
Interfaces écosystémiques non sécurisées	Faille dans l'application web, l'API, le service cloud ou l'interface mobile de l'écosystème qui permettrait de compromettre l'objet ou ses composants reliés. Problèmes fréquents : • Manque d'authentification / autorisation • Manque ou chiffrement faible • Manque dans la validation des entrées et sorties.
Absence de mécanisme de mise à jour sécurisé	Impossible d'appliquer des mises à jour de façon sécurisée. Par exemple, impossibilité de valider le micrologiciel (firmware) sur l'appareil, mise à jour téléchargée à partir d'un protocole non chiffré, possibilité de revenir à une version précédente du micrologiciel et manque de notifications sur les mises à jour de sécurité.
Utilisation de composants non sécurisés ou obsolètes	Utilisation de bibliothèques logicielles désuètes ou non sécurisés pouvant mener à une compromission de l'appareil. Cela inclut des configurations non sécurisés et l'utilisation de logiciel ou matériel compromis provenant d'une tierce partie.
Protection insuffisante de la vie privée	Les données personnelles stockées sur l'appareil ou dans l'écosystème sont utilisées de façons non sécurisées, indument ou sans permission.
Transfert et stockage de données non sécurisés	L'absence de cryptage lors du traitement des données sensibles, que ce soit pendant la transmission, le traitement ou au repos, est une opportunité pour les pirates informatiques de voler et d'exposer des données. Le cryptage est indispensable partout où le transfert de données est impliqué.
Manque de gestion des appareils	Il s'agit de l'incapacité à sécuriser efficacement tous les appareils du réseau. Il expose le système à de nombreuses menaces. Cela inclut la gestion des actifs, la gestion des mises à jour, la mise hors service sécurisée, la surveillance des systèmes et la capacité à répondre aux incidents. Quel que soit le nombre d'appareils concernés ou leur taille, chacun d'entre eux doit être protégé contre les violations de données.
Paramètres par défaut non sécurisés	Les vulnérabilités existantes dans les paramètres par défaut exposent le système à un éventail de problèmes de sécurité. Ils d'appareils ou systèmes livrés avec des paramètres par défaut non sécurisés ou qui n'ont pas la capacité d'être sécurisés.
Manque de durcissement physique	L'absence de mesures de renforcement physique peut permettre aux attaquants d'obtenir des informations sensibles qui pourraient les aider lors d'une future attaque à distance à prendre le contrôle local de l'appareil.

filtrage insuffisant des entrées de données, d'une interface mobile non sécurisée et d'une interface Cloud/Web non sécurisée. Cette listes de vulnérabilités est essentiellement basée sur l'expérience et sur les problèmes publiés sur internet (Payatu 2018).

L'Internet des objets (IoT) est particulièrement vulnérable en raison de la diversité de ses dispositifs et de leur interconnectivité accrue. Parmi les vulnérabilités courantes, on trouve l'empoisonnement du cache ARP, l'usurpation DNS, le détournement de session, le détournement SSL, et l'utilisation de certificats SSL auto-signés non vérifiés (Cekerevac et al. 2017). Les dispositifs IoT sont également sujets à des attaques telles que le Cross-site scripting (XSS), les problèmes de confidentialité des données, et les vulnérabilités des protocoles de base tels que IP, TCP, et HTTP (Šimec 2017).

Les faiblesses dans les mécanismes d'authentification et les vulnérabilités des environnements applicatifs sont également préoccupantes. Ces faiblesses permettent des accès non autorisés et sont souvent dues à une gestion inappropriée des données volumineuses (Gamundani 2015). De plus, les technologies de connexion comme RFID, WiFi et les réseaux de capteurs sans fil présentent des vulnérabilités intrinsèques (Andrea, Chrysostomou et Hadjichristofi 2015). Les mises à jour de firmware non sécurisées et l'utilisation de mots

de passe faibles ou par défaut accentuent encore davantage les risques (Mishra et Pandya 2021 ; Anand et al. 2020).

Les dispositifs IoT souffrent également de faiblesses liées aux systèmes d'exploitation obsolètes et aux informations de canal latéral, qui peuvent être exploitées par les attaquants pour découvrir des clés secrètes ou altérer des dispositifs (Anand et al. 2020). L'absence de chiffrement des données et le manque de protection physique des appareils augmentent les risques d'interception et de manipulation des communications (Cekerevac et al. 2017 ; G. Uganya et Vijayaraj 2021).

Abomhara et al. répartit les vulnérabilités en trois parties : les vulnérabilités matérielles, les vulnérabilités logicielles et les vulnérabilités techniques. Il souligne la difficulté d'identifier ces failles, notamment les vulnérabilités physiques. Même si elles sont trouvées, la correction des vulnérabilités physiques n'est pas facile à corriger en raison des problèmes de compatibilité et d'interopérabilité. Quant aux vulnérabilités logicielles, elles peuvent résider au niveau du système d'exploitation, dans les logiciels applicatifs et les logiciels de contrôle tels que les protocole de communication (Abomhara, Køien et Alghamdi 2021).

Neshenko et al. effectuent une étude détaillées sur les vulnérabilités de sécurité dans les systèmes IoT. En s'appuyant sur les travaux d'OWASP PAYATU, ils distinguent neuf classes de vulnérabilité, allant de la sécurité physique déficiente au mécanisme d'audit insuffisant, en passant par la récupération d'énergie insuffisante, une authentification inadéquate, le cryptage incorrect et les ports ouverts inutilement. A cela s'ajoutent le contrôle d'accès insuffisant, les capacité de gestion des correctifs inappropriées et des pratiques de programmation faibles. Ainsi, ils considèrent l'architecture à trois couche de l'IoT et dans chaque couche, il essaie de localiser les vulnérabilités correspondantes (Neshenko et al. 2019).

Selon Eshtiak et al., l'IoT fait face à nombreuses vulnérabilités dont certaines sont reparties dans les trois couches de son architecture. La couche perception souffre des problèmes de confidentialité, d'authentification des nœuds et du manque de cryptage des données après la capture. La couche réseau est victime d'attaques opérées durant le routage des données d'une couche à une autre. Il s'agit notamment des attaques de types DDoS et MiTM. Quant à la couche application, ces principales failles sont liées à la confidentialité des données, le contrôle d'accès et la divulgation d'informations. A cela s'ajoute l'interface non sécurisée des appareils IoT. Mais les vulnérabilités des systèmes IoT ne se limitent pas au niveau de ces couches de base. Il y a d'autres aspects qu'il faut prendre en compte. A ce propos, il rappelle les vulnérabilités identifiées par l'OWASP (Ahmed et al. 2020). En comprenant ces vulnérabilités et en mettant en œuvre des stratégies d'atténuation efficaces, les organisations et les individus peuvent mettre en place une posture de sécurité robuste qui

Tableau 1.4 – Les vulnérabilités de sécurité dans l'IoT

Vulnérabilité	Couche	Description
Capture de nœud	Physique	Capture ou remplacement d'un nœud par un nœud malveillant, compromettant ainsi la sécurité du réseau IoT.
Injection de code malveillant	Physique	Injection de code malveillant dans le firmware ou le logiciel des dispositifs IoT.
Jamming électromagnétique	Physique	Création d'interférences radio pour perturber la communication entre les dispositifs IoT.
Épuisement des ressources énergétiques	Physique	Attaque visant à épuiser les ressources énergétiques des dispositifs IoT, les rendant inopérants.
Absence de chiffrement des données	Réseau	Transmission de données sensibles sans chiffrement, permettant aux attaquants d'intercepter et de lire les communications.
Mots de passe faibles	Réseau	Utilisation de mots de passe par défaut ou faibles, faciles à deviner ou à forcer.
Ports ouverts	Réseau	Les ports réseau ouverts permettent aux attaquants d'accéder au réseau et de lancer des attaques.
Absence d'authentification	Réseau	L'absence de mécanismes d'authentification permet aux attaquants d'usurper l'identité des nœuds légitimes.
Vulnérabilités des protocoles de base	Réseau	Les protocoles Internet de base, tels que IP, TCP et HTTP, sont vulnérables aux attaques comme le DDoS et l'écoute clandestine.
Détournement de session	Transport	Interception des sessions utilisateur pour usurper l'identité des utilisateurs légitimes.
Détournement SSL	Transport	Les attaquants peuvent intercepter les communications sécurisées en utilisant des certificats SSL falsifiés.
Exploitation des protocoles de routage	Réseau	Les attaques exploitent les faiblesses des protocoles de routage pour créer des boucles de routage ou détourner le trafic.
Failles d'injection de commande	Application	Les attaquants peuvent exécuter du code malveillant en exploitant les failles d'injection de commande dans les dispositifs IoT.
Problèmes de confidentialité des données	Application	La gestion et le partage des données peuvent compromettre la confidentialité des utilisateurs.
Problèmes d'authentification	Application	Les mécanismes d'authentification insuffisants permettent des accès non autorisés.
Mécanismes de mise à jour non sécurisés	Application	L'absence de mécanismes de mise à jour sécurisée permet l'injection de code malveillant dans les dispositifs.

protège leurs écosystèmes, leurs données et leur confidentialité IoT.

1.6 Les attaques courantes contre l'IoT

Avec l'essor de l'internet des objets (IoT), les dispositifs connectés deviennent de plus en plus présents dans nos vies quotidiennes. Cependant, cette interconnexion accrue entraîne également une augmentation des vulnérabilités et des risques de sécurité. Les systèmes IoT sont susceptibles d'être attaqués à divers niveaux de leur architecture, rendant essentiel une compréhension approfondie des types d'attaques courantes pour mieux sécuriser ces

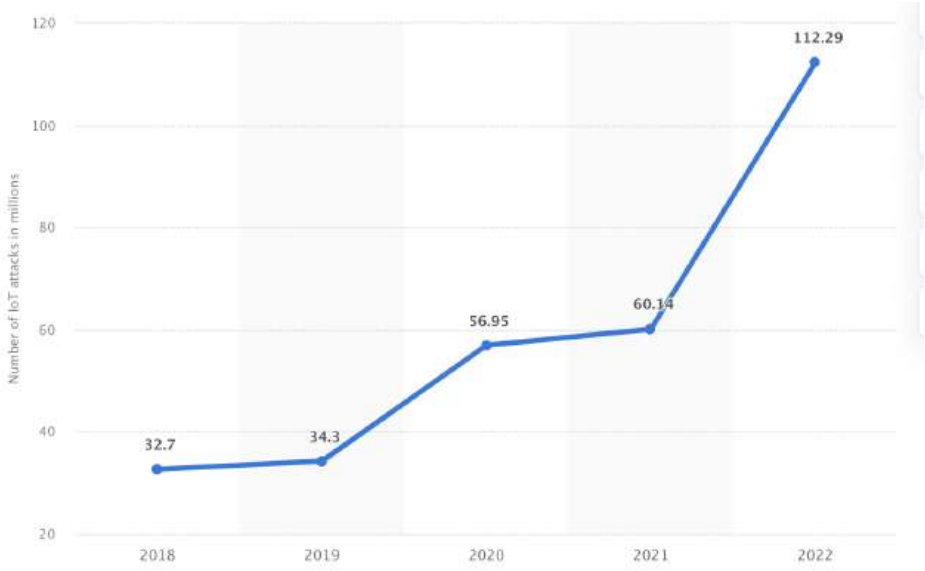


Figure 1.10 – Nombre annuel d'attaques de logiciels malveillants liés à l'Internet des objets (IoT) dans le monde de 2018 à 2022 (Lemeš 2023)

dispositifs.

Les attaques contre les systèmes IoT sont variées et peuvent cibler différentes couches de l'architecture IoT. Parmi les attaques les plus courantes, on trouve les attaques par déni de service (DoS) et déni de service distribué (DDoS) qui saturent les ressources réseau pour rendre les services indisponibles (Sonar et Upadhyay 2014 ; Mishra et Pandya 2021). Ces attaques peuvent paralyser les services IoT critiques, entraînant des perturbations importantes, comme dans les infrastructures de santé ou les systèmes industriels.

Les attaques de l'homme du milieu (MITM) interceptent et modifient les communications entre deux parties, compromettant ainsi la confidentialité et l'intégrité des données (Cekerevac et al. 2017 ; Nawir et al. 2016). Par exemple, en interceptant les communications entre un capteur IoT et son contrôleur, un attaquant peut manipuler les données envoyées, ce qui peut avoir des conséquences désastreuses, notamment dans les applications de contrôle industriel ou de gestion de l'énergie.

Les attaques par relecture (Replay) consistent à répéter des transmissions légitimes pour tromper les dispositifs IoT (Gamundani 2015 ; Nawir et al. 2016). Ces attaques peuvent perturber le fonctionnement des dispositifs en réutilisant des commandes légitimes, créant ainsi des dysfonctionnements ou des comportements non désirés. De plus, les attaques Sybil impliquent qu'un seul nœud se fasse passer pour plusieurs identités pour fausser les données de routage, perturbant ainsi le réseau et rendant difficile la distinction entre les nœuds légitimes et malveillants (Nawir et al. 2016).

Les attaques par injection de code malveillant compromettent les dispositifs IoT en introduisant des logiciels malveillants dans leurs applications (Andrea, Chrysostomou et Hadjichristofi 2015 ; Mishra et Pandya 2021). Ces logiciels peuvent permettre à un attaquant de prendre le contrôle du dispositif, de voler des données sensibles ou de lancer d'autres attaques à partir du dispositif compromis.

Les attaques de flooding génèrent un trafic excessif pour épuiser les ressources des nœuds, tandis que les attaques de surcharge de table de routage exploitent les messages DAO falsifiés pour surcharger les tables de routage et provoquer des débordements de mémoire (Mayzaud, Badonnel et Chrisment 2016). Ces attaques peuvent gravement affecter la disponibilité des réseaux IoT en rendant les routes de communication inefficaces.

Les attaques par écoute clandestine (Eavesdropping) interceptent les communications pour voler des informations sensibles (G. Uganya et Vijayaraj 2021). Par exemple, un attaquant peut intercepter des données de santé transmises par un dispositif médical IoT, compromettant ainsi la confidentialité des patients. Les attaques par usurpation d'identité permettent aux attaquants de se faire passer pour des dispositifs légitimes pour accéder au réseau, ce qui peut conduire à la compromission de l'ensemble du système IoT (G. Uganya et Vijayaraj 2021).

Les attaques exploitant les vulnérabilités de mise à jour de firmware et les systèmes d'exploitation obsolètes permettent l'injection de mises à jour malveillantes, compromettant ainsi les dispositifs IoT (Anand et al. 2020). En manipulant le processus de mise à jour, un attaquant peut installer un firmware malveillant qui contrôle le dispositif ou l'utilise pour lancer d'autres attaques.

Les attaques par ransomware verrouillent les dispositifs IoT et demandent une rançon pour déverrouiller les dispositifs ou restituer les données (Rizvi et al. 2020 ; G. Uganya et Vijayaraj 2021). Ces attaques peuvent paralyser des dispositifs critiques et exiger des paiements substantiels pour restaurer les services, ce qui représente une menace importante pour les organisations utilisant des réseaux IoT étendus.

La diversité et la sophistication croissantes des attaques contre les systèmes IoT nécessitent une approche de sécurité multidimensionnelle. La mise en œuvre de mécanismes de sécurité robustes, tels que le chiffrement des communications, l'authentification forte, et des mises à jour régulières et sécurisées, est essentielle pour protéger les dispositifs IoT contre ces menaces. En outre, la sensibilisation et la formation des développeurs et des utilisateurs sur les meilleures pratiques de sécurité peuvent considérablement améliorer la résilience des systèmes IoT face aux cyberattaques.

Tableau 1.5 – Typologie des attaques l'IoT

Attaque	Couche	Description
Attaque par déni de service (DoS)	Réseau	Inondation du réseau avec des paquets pour rendre les services indisponibles.
Attaque de l'homme du milieu (MITM)	Réseau/Transport	Interception et modification des communications entre deux parties sans qu'elles ne s'en aperçoivent.
Usurpation d'identité (Spoofing)	Réseau	Les attaquants se font passer pour des dispositifs légitimes pour accéder au réseau.
Sniffing de paquets	Réseau	Les attaquants capturent et analysent les paquets de données en transit pour extraire des informations sensibles.
Routing Table Overload	Réseau	Les attaquants surchargent les tables de routage avec des entrées fausses, empêchant la création de nouvelles routes légitimes.
Flooding Attack	Réseau	Les attaquants génèrent un grand volume de trafic pour rendre les nœuds et les liens indisponibles.
Détournement de session (Session Hijacking)	Transport	Les attaquants interceptent les sessions utilisateur pour usurper l'identité des utilisateurs légitimes.
SSL Hijacking	Transport	Les attaquants interceptent les communications sécurisées en utilisant des certificats SSL falsifiés.
Injection de code malveillant	Application	Les attaquants introduisent du code malveillant dans des dispositifs ou des applications IoT pour compromettre leur fonctionnement.
Ransomware	Application	Les attaquants verrouillent les dispositifs IoT et demandent une rançon pour les déverrouiller ou restituer les données.
Attaque par relecture (Replay)	Application	Les attaquants réutilisent des messages interceptés pour tromper les dispositifs IoT.
Écoute clandestine (Eavesdropping)	Réseau	Les attaquants interceptent passivement les communications pour voler des informations sensibles.

Tableau 1.6 – Classification des attaques en fonction des catégories

Attaques physiques	Attaques Réseaux	Attaques logicielles	Attaques de chiffrement
Capture de nœud	Attaque par déni de service (DoS)	Injection de code malveillant	SSL Hijacking
Jamming	Attaque de l'homme du milieu (MITM)	Ransomware	Détournement SSL
Exploitation des protocoles de routage	Usurpation d'identité (Spoofing)	Injection de paquets frauduleux	
Écoute clandestine (Eavesdropping)	Sniffing de paquets	Attaque par relecture (Replay)	
Flooding Attack	Routing Table Overload		

1.7 Conclusion

L'Internet des Objets est en train de redéfinir les paradigmes technologiques et industriels, en offrant des solutions innovantes et en ouvrant de nouvelles perspectives pour l'automatisation et l'efficacité opérationnelle. Ce chapitre a permis de dresser un panorama complet des différentes facettes de l'IoT, de ses applications pratiques à son architecture complexe, en passant par les technologies et protocoles de communication qui en assurent le fonctionnement. L'exploration des domaines d'application de l'IoT a révélé l'ampleur de son impact sur des secteurs clés comme la santé, l'industrie, les villes intelligentes, l'agriculture, la gestion de l'énergie, et le transport. Chaque application démontre comment l'IoT

peut optimiser les processus, améliorer la prise de décision en temps réel et contribuer à une utilisation plus efficiente des ressources. Cependant, l'essor de l'IoT s'accompagne de défis significatifs, notamment en matière de sécurité. Les vulnérabilités et les attaques potentielles sur les systèmes IoT soulignent la nécessité d'adopter des mesures de protection rigoureuses et de développer des stratégies de sécurité robustes. Comprendre les menaces et mettre en place des solutions adaptées est crucial pour garantir la fiabilité et la sécurité des dispositifs IoT. En conclusion, l'IoT est une technologie transformatrice avec un potentiel immense pour révolutionner divers secteurs. Toutefois, son adoption doit être accompagnée d'une vigilance accrue en matière de sécurité pour prévenir les risques et assurer une intégration réussie et durable. La compréhension approfondie des concepts, des technologies et des défis de l'IoT est essentielle pour tirer pleinement parti de ses avantages tout en minimisant les risques associés.

Chapitre 2

IA et Cybersécurité

2.1 Introduction 37
2.2 Historique et évolution de l'IA 38
2.3 Apprentissage automatique 39
2.3.1 Apprentissage supervisé 39
2.3.2 Apprentissage non supervisé 41
2.3.3 Apprentissage par renforcement 43
2.3.4 Exemple d'algorithmes d'apprentissage automatique . . . 44
2.4 Apprentissage profond 46
2.4.1 Architecture des réseaux de neurones profonds 47
2.4.2 Algorithmes et techniques avancées 52
2.5 Les principes fondamentaux de la sécurité informatique 57
2.5.1 Définition 57
2.5.2 Principes de base de la sécurité informatique 57
2.5.3 Défis de la sécurité dans le domaine de l'IoT 58
2.6 Applications de l'IA en cybersécurité 59
2.6.1 Détection d'anomalies 59
2.6.2 Détection d'intrusions et classification d'attaques 63
2.6.3 Identification des dispositifs IoT 67
2.6.4 Discussion 69
2.7 Conclusion 70

2.1 Introduction

L'intelligence artificielle (IA), depuis sa conception, a progressivement transformé de nombreux domaines d'application grâce à sa capacité à simuler, reproduire et même surpasser les processus intellectuels humains dans certaines tâches. En tant que domaine interdisciplinaire, l'IA a non seulement avancé dans des tâches autonomes complexes mais a également intégré des secteurs nécessitant des analyses et des décisions critiques, notamment la cybersécurité.

Marvin Lee Minsky, un précurseur de l'IA, la définit comme "la construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau". Cette définition met l'accent sur la capacité des programmes informatiques à effectuer des tâches intellectuellement complexes. Yann Le Cun, un pionnier de l'apprentissage profond, propose une perspective similaire, définissant l'IA comme "un ensemble de techniques permettant à des machines d'accomplir des tâches et de résoudre des problèmes normalement réservés aux humains et à certains animaux". Cette définition souligne la capacité des machines à traiter des problèmes autrefois considérés comme exclusivement humains.

À l'ère numérique actuelle, la cybersécurité est devenue une préoccupation centrale pour les organisations et les gouvernements du monde entier. Avec l'augmentation exponentielle des données générées et des interactions en ligne, les systèmes traditionnels de sécurité informatique peinent à suivre le rythme des nouvelles menaces sophistiquées. Cette croissance rapide est due en grande partie à l'avènement de l'internet des objets (IoT). Selon Statista, le nombre d' appareils Internet des objets (IoT) dans le monde devrait presque doubler, passant de 15,1 milliards en 2020 à plus de 29 milliards d'appareils IoT en 2030. Ces nouveaux types d'appareils viennent avec un lots de vulnérabilités agrandissant la surface d'attaque pour les cyber-attaquants. C'est ici que l'IA intervient, offrant des capacités avancées de détection d'anomalies, de prédiction des attaques et de surveillance du trafic réseau, transformant ainsi radicalement la façon dont les entités se défendent contre les cyberattaques.

Ce chapitre explorera le concept d'intelligence artificielle, son évolution, ses différentes branches telles que l'apprentissage automatique et l'apprentissage profond. Une attention particulière sera accordée aux applications de l'IA dans le domaine de la cybersécurité, notamment de la sécurité dans l'IoT.

2.2 Historique et évolution de l'IA

L'intelligence artificielle (IA) a évolué de manière remarquable depuis ses débuts, marquée par des phases d'innovation intense et des périodes de stagnation, façonnant ainsi le paysage actuel de cette technologie (B, C et U 2020).

Les débuts de l'IA, dans les années 1950 et 1960, ont été une période d'effervescence intellectuelle où les fondements théoriques ont été établis. Alan Turing, considéré comme le père de l'informatique théorique, a introduit des concepts clés avec son article de 1950 sur les machines pensantes et le Test de Turing (TURING 1950). La conférence de Dartmouth en 1956, où le terme "intelligence artificielle" a été formalisé par John McCarthy et Marvin Minsky, marque le début officiel de l'IA comme domaine académique (McCarthy et al. 2006). Cette période a également vu le développement du premier programme de jeu de dames et l'introduction des réseaux neuronaux avec le perceptron (Rosenblatt 1957).

Cependant, les années 1970 ont marqué une période de désillusion, connue sous le nom de "premier hiver de l'IA". Le manque de progrès tangibles et les limites des systèmes de l'époque ont conduit à une réduction drastique des financements. Des rapports critiques, tels que le rapport Lighthill, ont accentué cette tendance en soulignant les limitations techniques et théoriques des approches existantes, ralentissant ainsi les recherches dans le domaine (Lighthill 1973).

Dans les décennies suivantes, l'IA a évolué vers une approche symbolique, centrée sur la manipulation de symboles et de règles formelles pour résoudre des problèmes. Cette période a vu l'émergence des systèmes experts comme MYCIN et SHRDLU, démontrant la capacité des ordinateurs à manipuler des connaissances structurées et à raisonner sur des problèmes spécifiques. Ces développements ont préparé le terrain pour des applications plus larges de l'IA (Shortliffe 1976 ; Winograd 1972).

Avec l'avènement de l'apprentissage automatique et de l'apprentissage profond dans les années 1990, l'IA a connu un renouveau spectaculaire. Les capacités de calcul et de stockage accrues ont permis le développement d'algorithmes capables d'apprendre à partir de grandes quantités de données. Des réalisations notables comme la victoire de Deep Blue contre Garry Kasparov, les avancées en reconnaissance de motifs et en traitement du langage naturel ont marqué cette ère (Campbell, Hoane Jr et Hsu 2002). Au début des années 2010, l'explosion de la disponibilité des données et l'utilisation avancée des GPU ont révolutionné l'apprentissage automatique. Ces avancées ont conduit à des succès significatifs, tels que la victoire de Watson d'IBM à "Jeopardy !" en 2011, la reconnaissance de chats dans des vidéos par Google X en 2012, et les victoires d'AlphaGo de Google sur les champions du

monde de Go en 2016 (Baker 2011 ; Chen 2016 ; Hölldobler, Möhle et Tigunova 2017). Ces développements continuent d'impacter divers secteurs, y compris la cybersécurité, où l'IA joue un rôle crucial dans la détection et la prévention des menaces.

2.3 Apprentissage automatique

L'apprentissage automatique, une sous-discipline de l'intelligence artificielle, est une discipline en constante évolution qui a captivé l'intérêt des chercheurs depuis des décennies. Comme l'a souligné Arthur Samuel dès 1959, il s'agit d'un domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés (Samuel 1959) (Painuli et al. 2021). Cette définition reste pertinente aujourd'hui, alors que les systèmes d'apprentissage automatique continuent d'occuper une place centrale dans de nombreux domaines. Ils sont utilisés pour filtrer des contenus indésirables, classer des résultats de recherche, faire des recommandations personnalisées et bien plus encore, comme l'ont souligné LeCun et al. dans leur rapport sur l'intelligence artificielle (*Y. LeCun, Qu'est-ce que l'intelligence artificielle* s. d.). L'apprentissage automatique permet aux ordinateurs de découvrir des schémas et des relations dans les données, et d'ajuster leur comportement en conséquence. En exploitant des modèles mathématiques, les algorithmes d'apprentissage automatique peuvent extraire des informations significatives des ensembles de données d'entraînement, comme le souligne également la CNIL. Ces informations sont ensuite utilisées pour améliorer les performances des modèles dans l'accomplissement de tâches spécifiques. Dans cette section, nous explorerons les principaux aspects de l'apprentissage automatique, en commençant par les bases théoriques de l'apprentissage supervisé et non supervisé, jusqu'aux techniques avancées telles que l'apprentissage par renforcement. Nous étudierons également quelques-uns des algorithmes les plus couramment utilisés, tels que les arbres de décision, les forêts aléatoires, les machines à vecteurs de support et les réseaux neuronaux, ainsi que leurs applications dans divers domaines de la vie réelle.

2.3.1 Apprentissage supervisé

L'apprentissage supervisé est une approche fondamentale de l'apprentissage automatique où les algorithmes sont entrainés sur un ensemble de données étiquetées. Ces données étiquetées consistent en des paires d'entrée-sortie, où l'entrée est une observation et la sortie est la réponse attendue. En d'autres termes, l'apprentissage supervisé consiste à apprendre une fonction qui mappe une entrée à une sortie en se basant sur des exemples d'entrée-sortie.

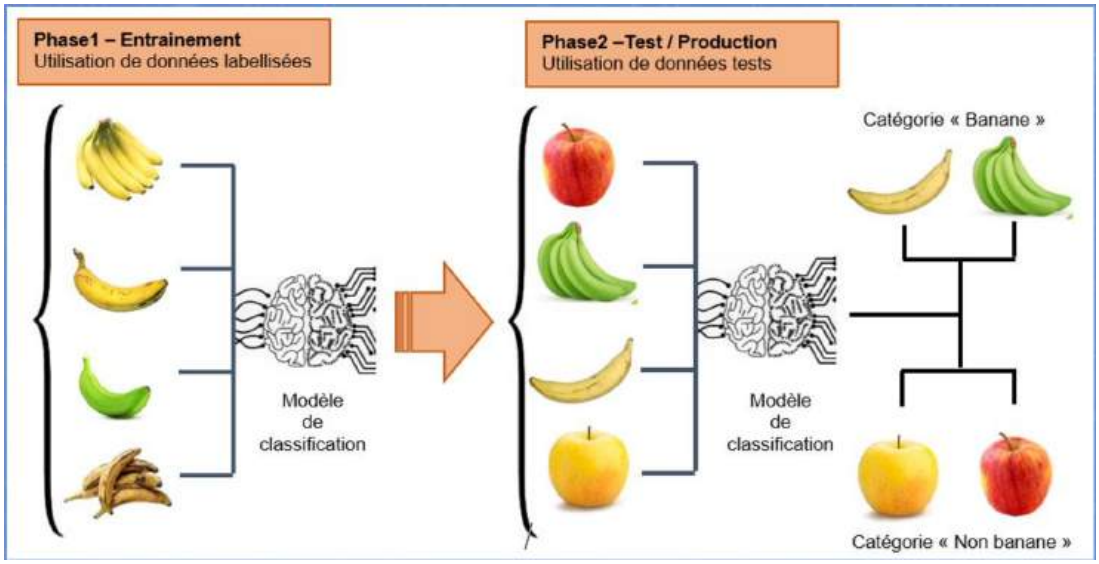


Figure 2.1 – Exemple de déroulement d'un algorithme d'apprentissage supervisé. Le modèle est entraîné pour la reconnaissance des bananes et des pommes

Les algorithmes d'apprentissage supervisé nécessitent une assistance externe et sont généralement divisés en ensembles de données d'entraînement et de test. L'ensemble de données d'entraînement est utilisé pour former le modèle, tandis que l'ensemble de données de test est utilisé pour évaluer sa performance. Les algorithmes apprennent des modèles à partir de l'ensemble d'entraînement et les appliquent ensuite à l'ensemble de test pour la prédiction ou la classification (LeCun, Bengio et Hinton 2015). Dans l'apprentissage supervisé, comme le souligne le rapport de Carleo et al. (Carleo et al. 2019), l'objectif est de trouver une fonction f qui, lorsqu'elle est présentée avec de nouvelles données sans étiquette, produit une sortie qui se rapproche au mieux de l'étiquette attendue. Pour cela, on utilise généralement une fonction de perte qui mesure l'erreur entre la sortie prédite par le modèle et l'étiquette réelle. Le but est de minimiser cette fonction de perte en ajustant les paramètres du modèle, souvent appelés poids. En pratique, plusieurs algorithmes sont utilisés pour minimiser la fonction de perte et ajuster les poids du modèle. Parmi eux, l'algorithme de descente de gradient stochastique (SGD) est largement utilisé. Il consiste à ajuster les poids du modèle dans la direction opposée du gradient de la fonction de perte, comme décrit dans (Carleo et al. 2019). Ce processus est répété sur de petits ensembles de données appelés mini-lots jusqu'à ce que la fonction de perte moyenne cesse de diminuer. En définitive, nous pouvons retenir que l'apprentissage supervisé est une méthode puissante qui permet aux algorithmes d'apprendre à prédire des sorties à partir d'entrées en se basant sur des exemples étiquetés. Cette approche est largement utilisée dans divers domaines, notamment la classification, la régression et la prédiction.

Regression

La régression est une technique d'apprentissage supervisé utilisée pour prédire des valeurs numériques continues. Elle est largement utilisée dans divers domaines, tels que l'économie, la finance, la météorologie et bien d'autres. Les modèles de régression tentent de modéliser la relation entre une variable dépendante et une ou plusieurs variables indépendantes en ajustant une ligne ou une courbe aux données d'entraînement.

Classification

La classification en apprentissage automatique est une tâche d'apprentissage supervisé où le but est de prédire la catégorie ou l'étiquette d'une observation donnée, basée sur des données d'entrée. Par exemple, elle peut être utilisée pour classifier les e-mails en spam ou non spam, les images en différentes catégories, les patients en différentes conditions médicales, etc. Formulée formellement, la classification peut être définie comme suit : Soit $\mathcal{X}$ l'espace des données d'entrée, et $\mathcal{Y}$ l'ensemble fini des classes possibles, avec $Y = \{y_1, y_2, \ldots, y_k\}$. Un modèle de classification apprend une fonction $f : \mathcal{X} \to \mathcal{Y}$ à partir d'un ensemble d'apprentissage $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, où chaque $x_i \in \mathcal{X}$ est un vecteur de caractéristiques et chaque $y_i \in \mathcal{Y}$ est la classe associée. L'objectif est de trouver une fonction f telle que, pour une nouvelle observation $x \in \mathcal{X}$, $f(x)$ prédise correctement la classe $y \in \mathcal{Y}$. Cela se fait généralement en minimisant une fonction de perte $L(y, f(x))$, qui mesure la différence entre la classe prédite $f(x)$ et la classe réelle y. La performance du modèle est souvent évaluée à l'aide de métriques telles que la précision, le rappel, la score-f1. Ces métriques seront définies dans le chapitre 3, plus précisément dans les sous-sous-sections 3.4.1 et 3.5.1.

2.3.2 Apprentissage non supervisé

L'apprentissage non supervisé est une approche de l'apprentissage automatique qui diffère de l'apprentissage supervisé en ce sens qu'aucune étiquette n'est fournie pour l'entraînement des algorithmes. Contrairement à l'apprentissage supervisé, où les modèles sont formés sur des données étiquetées, les algorithmes d'apprentissage non supervisé tentent de découvrir des structures intrinsèques ou des modèles intéressants dans les données sans aucune supervision externe.

Selon Mahesh Batta (Mahesh 2019), ces techniques sont appelées "apprentissage non supervisé" car il n'y a pas de réponses correctes ni de superviseur pour guider le proces-

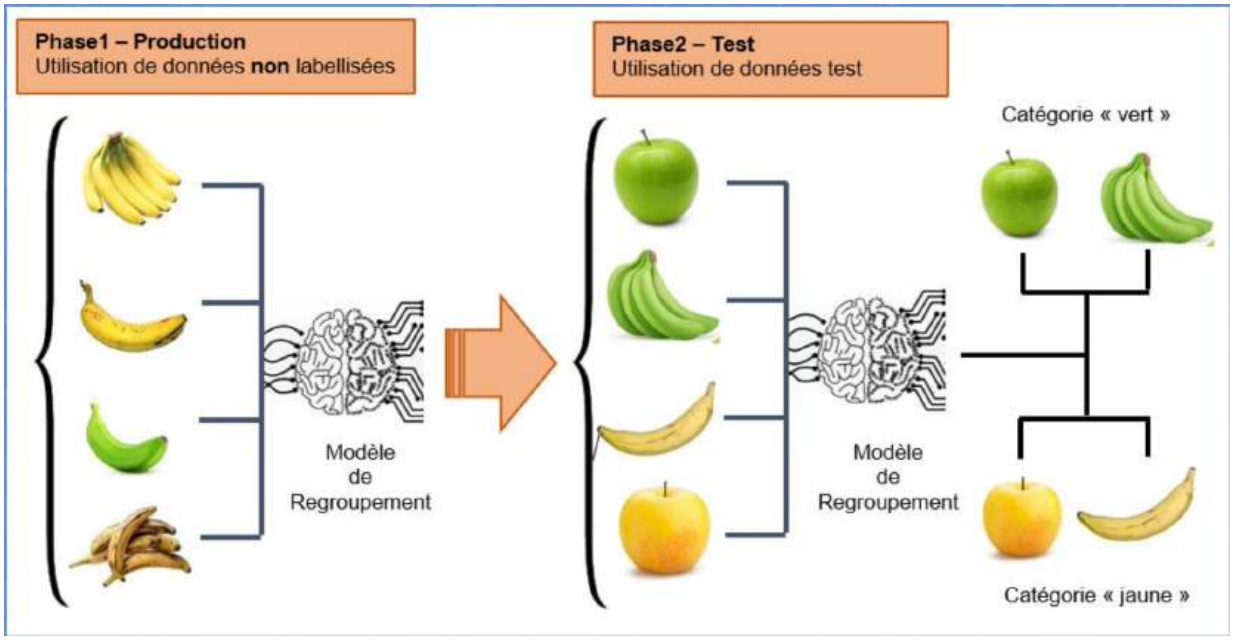


Figure 2.2 – Exemple de déroulement d'un algorithme d'apprentissage non supervisé. Le modèle est entraîné sur des données non étiquetées. Il apprend lui même à trouver des similarité pour former des groupes. Dans cet exemple le modèle s'est basé sur la couleur pour créer les groupes. Ainsi, on a le groupe des "vert" et celui des "jaune".

sus d'apprentissage. Les algorithmes sont laissés à eux-mêmes pour découvrir et présenter la structure intéressante dans les données. En utilisant les données disponibles, les algorithmes non supervisés apprennent à identifier des caractéristiques pertinentes et à les utiliser pour regrouper les données ou réduire leur dimensionnalité.

Comme le mentionne Carleo et al. (Carleo et al. 2019), l'apprentissage non supervisé comprend diverses techniques, telles que le clustering, où les données sont regroupées en clusters ou en groupes en fonction de similitudes intrinsèques, et la réduction de dimension, où la dimensionnalité des données est réduite tout en préservant les informations importantes. L'objectif principal de l'apprentissage non supervisé est de découvrir une structure sous-jacente dans les données sans aucune indication préalable sur leur nature.

Dans l'apprentissage non supervisé, une tâche courante est la modélisation générative, où l'on cherche à apprendre une distribution de probabilité qui génère des échantillons statistiquement similaires aux données observées. Des méthodes telles que les machines de Boltzmann et les réseaux antagonistes génératifs (GANs) sont utilisées pour modéliser ces distributions et générer de nouvelles données à partir de celles-ci.

Pour résumer, l'apprentissage non supervisé est une approche puissante qui permet d'explorer et de découvrir des structures intrinsèques dans les données sans aucune supervision externe. Ces techniques sont largement utilisées dans divers domaines, notamment la

segmentation de données, l'exploration de données et la génération de nouvelles données.

Clustering

Le clustering est une technique d'apprentissage non supervisé utilisée pour regrouper des données similaires en ensembles distincts appelés clusters. Les algorithmes de clustering tentent de maximiser la similarité intra-cluster tout en minimisant la similarité inter-cluster. Cette technique est largement utilisée dans des domaines tels que la segmentation de marché, la détection de fraude, l'analyse de données biologiques, etc.

Reduction de dimension

La réduction de dimension vise à réduire le nombre de variables ou de caractéristiques dans un ensemble de données tout en préservant autant que possible les informations importantes. Cela peut aider à simplifier la modélisation et à éviter les problèmes de surajustement (overfitting). Les techniques de réduction de dimension sont souvent utilisées pour la visualisation des données, la compression de données et l'extraction de caractéristiques.

2.3.3 Apprentissage par renforcement

L'apprentissage par renforcement est une méthode d'apprentissage automatique qui concerne la manière dont des agents logiciels doivent prendre des actions dans un environnement afin de maximiser une notion de récompense cumulative. Il s'agit de l'un des trois paradigmes fondamentaux de l'apprentissage automatique, aux côtés de l'apprentissage supervisé et de l'apprentissage non supervisé. Cette approche implique qu'un agent interagisse avec un environnement dynamique en effectuant des actions et en observant les réponses de cet environnement. Comme mentionné par Carleo et al. (2019), l'agent apprend progressivement à prendre des décisions afin de maximiser une récompense cumulative à long terme. Le processus d'apprentissage par renforcement repose sur la notion de rétroaction de l'environnement en réponse aux actions de l'agent.

Dans l'apprentissage par renforcement, l'agent prend des décisions séquentielles en fonction de son état actuel et des récompenses attendues à long terme. Comme l'indique Sutton et Barto (Sutton et Barto 2018), l'agent ajuste ses stratégies d'action en fonction des observations de l'environnement et des récompenses perçues, en recherchant un équilibre entre l'exploitation des stratégies actuelles et l'exploration de nouvelles stratégies pour maximiser la récompense finale. Une méthode couramment utilisée en apprentissage par

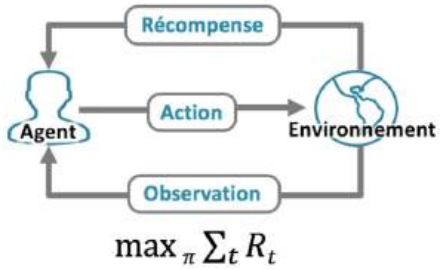


Figure 2.3 – Le principe de l'apprentissage par renforcement

renforcement est le Q-learning, comme le soulignent Carleo et al. (2019). Le Q-learning repose sur une matrice de valeurs Q qui évalue la qualité des actions dans un état donné de l'environnement. Cette matrice est itérativement mise à jour à mesure que l'agent interagit avec l'environnement et apprend de ses expériences. Dans les applications avancées, telles que celles utilisées dans AlphaGo, les matrices de valeur peuvent être représentées de manière compacte à l'aide de réseaux neuronaux profonds, donnant ainsi naissance à une technique appelée deep Q-learning.

En somme, l'apprentissage par renforcement est une méthode puissante pour entraîner des agents à prendre des décisions séquentielles dans des environnements dynamiques afin de maximiser des récompenses cumulatives. Cette sous-section explorera les concepts fondamentaux de l'apprentissage par renforcement ainsi que certains algorithmes populaires utilisés dans ce domaine.

2.3.4 Exemple d'algorithmes d'apprentissage automatique

Dans cette sous-section, nous explorerons quelques exemples d'algorithmes d'apprentissage automatique couramment utilisés dans la pratique (Somvanshi et al. 2016). Nous examinerons en détail les arbres de décision, les forêts aléatoires, les machines à vecteurs de support, et les réseaux neuronaux, ainsi que leurs applications et leurs avantages et inconvénients respectifs.

Arbres de décision

Les arbres de décision sont des modèles d'apprentissage automatique utilisés pour la classification et la régression. Ils fonctionnent en divisant récursivement l'ensemble de données en sous-ensembles plus petits basés sur des caractéristiques spécifiques, de manière à ce que les instances similaires se regroupent ensemble. Chaque division est choisie en fonction d'un critère qui vise à maximiser la pureté des sous-ensembles résultants. Les avantages

des arbres de décision incluent leur facilité d'interprétation, leur capacité à gérer à la fois des données numériques et catégorielles, ainsi que leur résistance aux valeurs aberrantes. Cependant, ils ont tendance à être sensibles au surajustement, en particulier lorsqu'ils sont non contraints (Navada et al. 2011) (Charbuty et Abdulazeez 2021).

Forêt aléatoire

Une forêt aléatoire est une technique d'apprentissage automatique qui combine plusieurs arbres de décision pour créer un modèle plus robuste et performant (Liu, Wang et Zhang 2012). Chaque arbre dans la forêt est entraîné sur un sous-ensemble aléatoire de données et utilise une sélection aléatoire de caractéristiques pour chaque division. Ensuite, les prédictions sont agrégées à partir de tous les arbres pour obtenir une prédiction finale. Les forêts aléatoires sont moins susceptibles de surajuster par rapport à un seul arbre de décision et peuvent capturer des relations plus complexes dans les données. Elles sont largement utilisées dans les applications où la précision est primordiale, comme la classification d'images et la prédiction de maladies (Kulkarni et Sinha 2013) (Rigatti 2017).

Machine à vecteur de support

Les machines à vecteur de support (SVM) sont des modèles d'apprentissage automatique utilisés pour la classification et la régression (Pisner et Schnyer 2020). Ils fonctionnent en trouvant l'hyperplan qui sépare de manière optimale les différentes classes dans l'espace des caractéristiques. L'objectif est de maximiser la marge, c'est-à-dire la distance entre l'hyperplan et les instances les plus proches de chaque classe. Les SVM sont efficaces dans les espaces de grande dimension et sont robustes face aux données bruitées. Cependant, ils peuvent être sensibles au choix du noyau et sont moins adaptés aux ensembles de données très volumineux en raison de leur coût computationnel élevé (Suthaharan 2016).

Réseaux de neurones artificiels (RNA)

Les réseaux de neurones artificiels sont des modèles d'apprentissage automatique qui s'inspirent de la structure et du fonctionnement des réseaux neuronaux biologiques. Tout comme les réseaux neuronaux biologiques, les RNA sont composés de plusieurs couches de neurones interconnectés, chaque neurone traitant l'information et transmettant des signaux à travers le réseau (Mitchell 1997). Ces réseaux sont capables d'apprendre des représentations complexes des données et sont efficaces pour capturer des relations non linéaires,

ce qui les rend largement utilisés dans divers domaines tels que la vision par ordinateur, le traitement du langage naturel et la prédiction de séries temporelles. Un RNA comprend trois composants essentiels : les caractéristiques des nœuds, la topologie du réseau et les règles d'apprentissage. Les caractéristiques des nœuds déterminent comment les signaux sont traités par chaque nœud, y compris le nombre d'entrées et de sorties, les poids associés à chaque entrée et sortie, ainsi que la fonction d'activation. La topologie du réseau décrit la manière dont les nœuds sont organisés et connectés entre eux. Enfin, les règles d'apprentissage déterminent comment les poids sont initialisés et ajustés pendant l'entraînement du réseau. Lors de son fonctionnement, chaque nœud reçoit des signaux d'entrée provenant d'autres nœuds via des connexions pondérées, et les traite en fonction de la somme pondérée de ces entrées. Si cette somme dépasse un certain seuil, le nœud s'active et transmet le signal à travers une fonction de transfert vers les nœuds voisins. La fonction de transfert peut prendre différentes formes, telles que la fonction sigmoïde, permettant de rendre les RNA plus flexibles et adaptés à la capture de motifs complexes dans les données. La topologie du réseau détermine la manière dont les nœuds sont organisés en couches, avec généralement des couches d'entrée, des couches de sortie et éventuellement des couches cachées. La conception de la topologie implique de déterminer le nombre de nœuds dans chaque couche, le nombre de couches dans le réseau et les connexions entre les nœuds (Zou, Han et So 2009).

En ce qui concerne l'apprentissage, les RNA utilisent généralement deux types de processus : l'apprentissage supervisé et l'apprentissage non supervisé. Dans l'apprentissage supervisé, un ensemble de données d'entraînement est fourni, avec des exemples d'entrées et de sorties cibles correspondantes, et les poids du réseau sont ajustés pour minimiser l'erreur entre la sortie du réseau et la sortie correcte. En revanche, l'apprentissage non supervisé ne nécessite pas de valeurs cibles dans l'ensemble de données, et le réseau tente de découvrir les motifs sous-jacents dans les données d'entrée seul. Malgré leur puissance et leur polyvalence, les réseaux de neurones artificiels peuvent être sensibles au surapprentissage, en particulier lorsque les ensembles de données sont petits ou mal équilibrés, et leur entraînement nécessite souvent des ensembles de données volumineux ainsi que des ressources computationnelles considérables (Chen et al. 2019).

2.4 Apprentissage profond

Les débuts de l'IA ont été marqués par des succès dans la résolution de problèmes intellectuellement ardues mais formellement définis, tels que les échecs, où les ordinateurs pouvaient surpasser les humains en raison de règles simples et bien définies. Cependant, le

véritable défi de l'IA réside dans la résolution de tâches intuitives pour les humains mais difficiles à décrire formellement, comme la reconnaissance d'objets sur des images ou la compréhension du langage parlé (LeCun, Bengio et Hinton 2015). L'apprentissage profond émerge comme une solution à ces défis.

L'apprentissage profond est une sous-catégorie de l'apprentissage automatique qui utilise des réseaux de neurones artificiels pour modéliser et comprendre des structures de données complexes. Ces réseaux de neurones profonds sont composés de multiples couches hiérarchiques, où chaque couche apprend à partir des sorties des couches précédentes pour extraire des caractéristiques de plus en plus abstraites des données. L'apprentissage profond permet de traiter des données brutes de manière non supervisée ou supervisée, et il est particulièrement efficace pour les tâches telles que la reconnaissance d'images, le traitement du langage naturel, et la détection d'anomalies (Bengio, Léonard et Courville 2013 ; LeCun, Bengio et Hinton 2015).

Les progrès récents de l'apprentissage profond ont été rendus possibles par l'avènement de grandes bases de données et de puissantes unités de traitement graphique (GPU), ainsi que par le développement de logiciels flexibles et d'outils d'optimisation efficaces. Ces avancées ont permis aux réseaux de neurones profonds de révolutionner de nombreux domaines, de la reconnaissance vocale à la découverte de médicaments, ouvrant la voie à des perspectives nouvelles et passionnantes dans le domaine de l'IA (Goodfellow, Bengio et Courville 2016).

2.4.1 Architecture des réseaux de neurones profonds

Les réseaux de neurones profonds ou DNN (Deep neural network) en anglais, sont des architectures sophistiquées de réseaux de neurones artificiels qui ont révolutionné de nombreux domaines de l'apprentissage automatique. Inspirés par la structure et le fonctionnement des réseaux neuronaux biologiques, les DNN sont organisés en différentes couches hiérarchiques, chacune jouant un rôle crucial dans le traitement et l'apprentissage des données. Cette section explore les composantes clés des DNN, notamment leur structure en couches, les fonctions d'activation et les processus d'apprentissage (Sewak, Sahay et Rathore 2020).

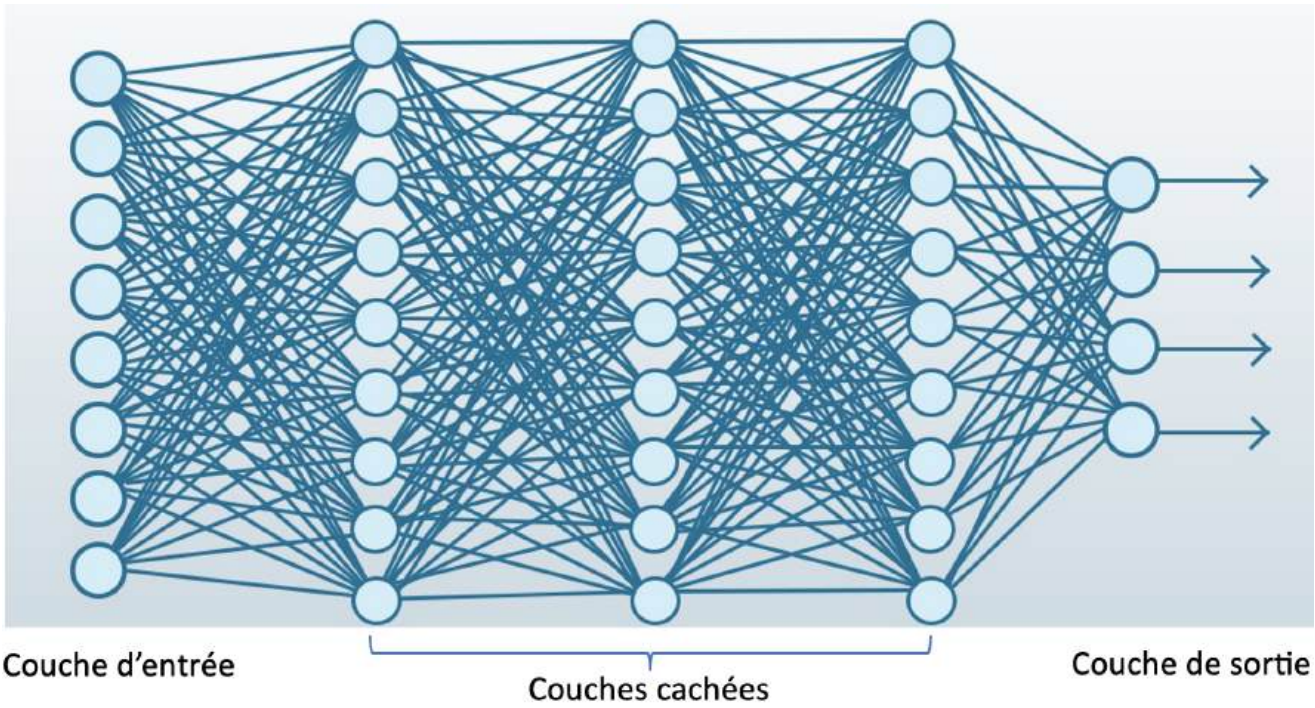


Figure 2.4 – Exemple d'un réseau de neurones avec trois couches cachées.

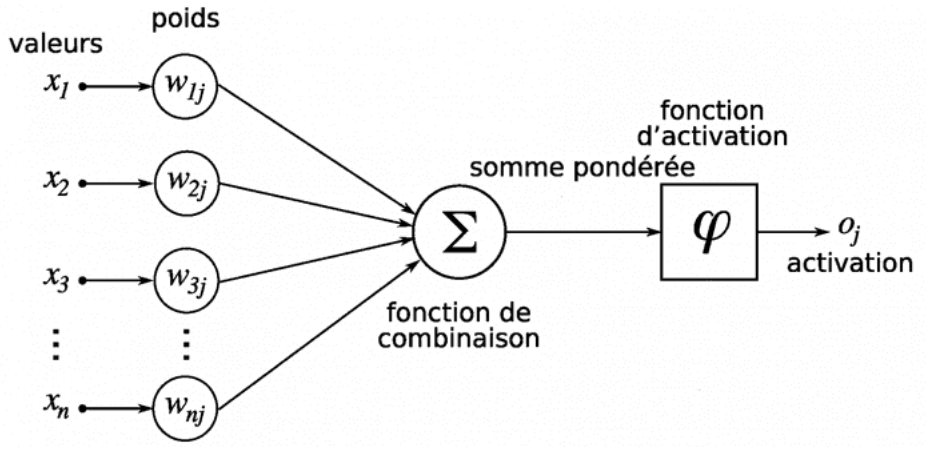


Figure 2.5 – Un neurone simple avec les différentes opérations

Structure en couches

Un réseau de neurones profond se compose généralement de plusieurs couches hiérarchiques, comprenant une couche d'entrée, des couches cachées et une couche de sortie. Contrairement aux réseaux peu profonds, les DNN peuvent avoir un grand nombre de couches cachées, ce qui leur permet de capturer des représentations complexes des données. La détermination du nombre de couches cachées et de neurones dans chaque couche est souvent guidée par des heuristiques basées sur la complexité des données et des expériences antérieures.

Fonctions d'activation

Les fonctions d'activation jouent un rôle crucial dans la capacité des DNN à modéliser des relations non linéaires. Différentes fonctions d'activation, telles que les fonctions Softmax, Sigmoid, Tanh, ReLU et ELU, sont utilisées dans différentes parties du réseau pour introduire de la non-linéarité et faciliter l'apprentissage. Par exemple, la fonction Softmax est couramment utilisée dans la couche de sortie pour fournir des probabilités de classe, tandis que ReLU est préférée dans les couches cachées en raison de sa capacité à atténuer le problème du gradient disparaissant. Les fonctions d'activation Sigmoid et Tanh étaient populaires dans les réseaux de neurones traditionnels, mais peuvent être sujettes au problème du gradient disparaissant ou explosif dans les DNN en raison de leur nature saturante. Cependant, elles sont encore utilisées dans certaines configurations spécifiques.

— **Softmax** :

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.1}$$

— **Sigmoid** :

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

— **Tanh** :

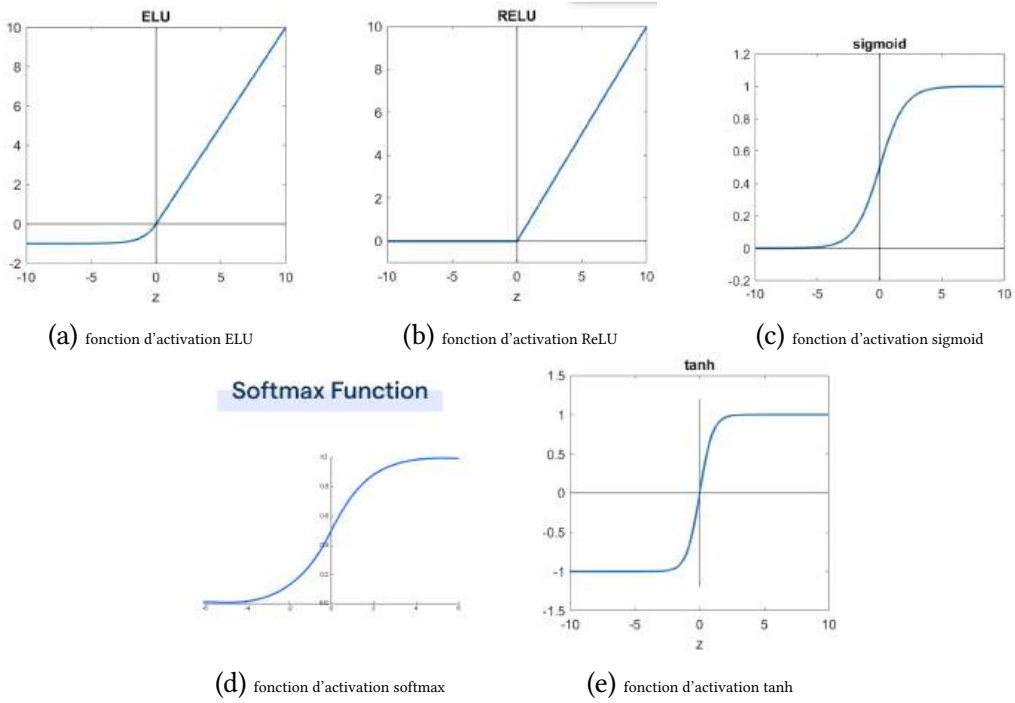
$$\text{Tanh}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.3}$$

— **ReLU (Rectified Linear Unit)** :

$$\text{ReLU}(x) = \max(0, x) \tag{2.4}$$

— **ELU (Exponential Linear Unit)** :

$$\text{ELU}(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha(e^x - 1) & \text{sinon} \end{cases} \tag{2.5}$$



(a) fonction d'activation ELU (b) fonction d'activation ReLU (c) fonction d'activation sigmoid

(d) fonction d'activation softmax (e) fonction d'activation tanh

Figure 2.6 – Les fonctions d'activation les plus utilisées

Fonctions de perte

Les fonctions de perte sont essentielles en apprentissage automatique et en apprentissage profond, servant à évaluer la disparité entre les prédictions d'un modèle et les valeurs attendues. La mesure de cette différence est cruciale pour guider l'optimisation des modèles.La fonction de perte, également désignée parfois sous le nom de fonction de coût, permet d'évaluer la performance du modèle en quantifiant l'erreur entre les prédictions et les valeurs réelles. Si la valeur de la fonction de perte est faible, le modèle fonctionne bien ; autrement, des ajustements sont nécessaires pour minimiser cette perte. La confusion peut survenir entre les termes "fonction de perte" et "fonction de coût". En général, la fonction de perte est calculée sur un seul point de données d'entraînement, tandis que la fonction de coût est la somme ou la moyenne des pertes sur un ensemble de données.

Fonctions de perte pour la régression : Dans les tâches de régression, où le modèle prédit des valeurs continues, différentes fonctions de perte sont utilisées pour évaluer la performance du modèle.

- **Erreur Quadratique Moyenne (MSE) :** Calculant la moyenne des écarts au carré entre les valeurs réelles et les prédictions du modèle, le MSE est simple à interpréter mais sensible aux valeurs aberrantes.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{\text{réelle},i} - y_{\text{prédite},i})^2 \tag{2.6}$$

- **Erreur Absolue Moyenne (MAE) :** Cette fonction calcule la moyenne des écarts absolus entre les valeurs réelles et les prédictions. Bien qu'elle soit robuste aux valeurs aberrantes, elle n'est pas différentiable à zéro.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{\text{réelle},i} - y_{\text{prédite},i}| \tag{2.7}$$

- **Perte de Huber :** Cette fonction combine des caractéristiques du MSE et du MAE, offrant une approche hybride pour gérer les valeurs aberrantes.

Fonctions de perte pour la classification : Pour les problèmes de classification, différentes fonctions de perte sont utilisées pour évaluer la capacité du modèle à classer correctement les entrées.

- **Entropie Croisée Binaire :** Utilisée dans la classification binaire, cette fonction mesure la divergence entre les probabilités prédites et les valeurs réelles.

$$Binary\,Cross{-}Entropy = -\frac{1}{n}\sum_{i=1}^{n}(y_{\text{réelle},i}\,log(y_{\text{prédite},i}){+}(1{-}y_{\text{réelle},i})\,log(1{-}y_{\text{prédite},i})) \tag{2.8}$$

- **Entropie Croisée Catégorique :** Adaptée à la classification multi-classes, elle quantifie la similarité entre les distributions de probabilités prédites et les valeurs réelles.

$$Categorical\,Cross-Entropy = -\sum_{i=1}^{n}\sum_{j=1}^{k} y_{\text{réelle},i,j}\;log(y_{\text{prédite},i,j}) \tag{2.9}$$

- **Entropie Croisée Catégorique Éparses :** Similaire à l'entropie croisée catégorique,

cette fonction est utilisée lorsque le codage one-hot des valeurs cibles devient inefficace en termes de mémoire.

$$Sparse\ Categorical\ Cross - Entropy = -\sum_{i=1}^{n} y_{\text{réelle},i}\ log(y_{\text{prédite},i}) \tag{2.10}$$

Ces fonctions de perte jouent un rôle crucial dans l'entrainement et l'optimisation des modèles d'apprentissage automatique et profond. Elles permettent d'évaluer la performance des modèles et de guider leur amélioration continue.

Entrainement et optimiseurs

Les DNN utilisent généralement des processus d'apprentissage supervisés ou non supervisés pour ajuster les poids du réseau et minimiser la perte entre la prédiction et la sortie attendue. Les optimiseurs, tels que ADAM, ADAGRAD et ADADELTA, sont ensuite utilisés pour minimiser cette perte en ajustant les poids du réseau via des algorithmes de descente de gradient stochastique. De plus, lors de l'entraînement d'un DNN, plusieurs paramètres doivent être pris en compte, notamment la taille des lots (batch size), le nombre d'époques (epochs) et la régularisation de l'abandon (dropout regularization) pour prévenir le surapprentissage.

2.4.2 Algorithmes et techniques avancées

Réseaux de neurones convolutifs (CNN)

Les réseaux de neurones convolutifs (CNN) sont devenus l'un des algorithmes les plus influents dans le domaine de l'apprentissage profond, trouvant des applications dans divers domaines tels que la vision par ordinateur, le traitement de la parole, la reconnaissance faciale, et bien d'autres. La structure typique d'un CNN comprend plusieurs couches, chacune jouant un rôle spécifique dans le traitement des données. Les éléments clés d'un CNN sont les couches de convolution, les couches de sous-échantillonnage (pooling) et les couches dites entièrement connectées ou de classification (Alom et al. 2018).

Couches CNN

1. **Couche de convolution :** Cette couche est fondamentale dans un CNN. Elle utilise des filtres de convolution pour analyser les caractéristiques de l'image d'entrée.

Chaque filtre est appliqué à des parties de l'image, permettant ainsi d'extraire des informations importantes.

2. **Couche de sous-échantillonnage (pooling) :** Après la convolution, les cartes de caractéristiques sont souvent réduites en dimension pour améliorer l'efficacité du modèle. Cela se fait généralement via des opérations de sous-échantillonnage comme le max-pooling ou l'average pooling.
3. **Couche entièrement connectée (FC) :** Les dernières couches d'un CNN sont des couches entièrement connectées qui effectuent la classification finale. Elles rassemblent les informations extraites par les couches précédentes pour produire des scores de classification.

Entraînement des CNN

L'entraînement d'un CNN implique plusieurs étapes, y compris l'initialisation des poids, la propagation avant, le calcul de la perte, la rétropropagation et la mise à jour des poids. Ces étapes sont répétées sur de nombreux mini-lots d'entraînement jusqu'à ce que le modèle atteigne une convergence satisfaisante.

Applications des CNN

Les CNN sont largement utilisés dans des domaines tels que la classification d'images, la détection d'objets, la reconnaissance faciale, le traitement d'images médicales et le traitement du langage naturel. Leur capacité à apprendre des caractéristiques directement à partir des données en fait un outil puissant pour de nombreuses tâches.

Quelques célèbres architectures CNN

AlexNet : Introduit en 2012, c'était l'une des architectures CNN profondes pionnières, présentant une profondeur accrue par rapport aux modèles précédents comme LeNet. AlexNet utilisait des GPU pour l'entraînement et introduisait des innovations telles que l'activation ReLU, le dropout et des techniques d'augmentation des données pour améliorer la généralisation et les performances.

Network-in-Network : Cette architecture a introduit le concept de convolutions 1x1 pour améliorer la profondeur du réseau et réduire le nombre de paramètres du modèle. Elle utilisait également le pooling moyen global (GAP) pour réduire le surajustement et améliorer l'efficacité computationnelle.

ZefNet (DeconvNet) : Introduit en 2013, ZefNet se concentrait sur la visualisation des activations du réseau pour comprendre le comportement du réseau et optimiser

les performances. Il utilisait des réseaux neuronaux déconvolutifs pour visualiser les caractéristiques apprises à chaque couche.

Visual Geometry Group (VGG) : VGG proposait une architecture CNN simple et efficace avec des réseaux profonds (19 couches) composés de petits filtres de convolution 3x3. Il obtenait des résultats significatifs dans les tâches de classification d'images, bien qu'il présentait une complexité computationnelle élevée en raison de son grand nombre de paramètres.

GoogLeNet (Inception-V1) : Lauréat de la compétition ILSVRC 2014, GoogLeNet introduisait le concept de modules d'inception, qui permettaient la combinaison efficace de filtres de convolution de différentes tailles dans la même couche. Il visait à améliorer l'efficacité computationnelle tout en maintenant une haute précision.

Highway Network : Introduit en 2015, Highway Network répondait au défi de l'entraînement de réseaux très profonds en incorporant des unités de portes inspirées des réseaux LSTM. Il permettait une meilleure circulation de l'information entre les couches et facilitait l'entraînement de réseaux avec des centaines de couches.

ResNet : ResNet (Residual Network) introduisait des connexions résiduelles pour résoudre le problème du gradient qui disparaît dans les réseaux profonds. Il permettait l'entraînement de réseaux très profonds (jusqu'à 152 couches) en permettant l'écoulement direct des gradients à travers des connexions de raccourci.

Inception-ResNet et Inception-V3/4 : Ces architectures combinaient les forces des modules Inception avec des connexions résiduelles pour améliorer l'efficacité et la généralisation de l'entraînement. Elles utilisaient des convolutions asymétriques et des couches de bottleneck pour réduire le coût computationnel.

DenseNet : DenseNet résolvait le problème des gradients qui disparaissent en connectant chaque couche à toutes les autres couches de manière feed-forward. Cela facilitait la réutilisation des caractéristiques et améliorait la circulation de l'information à travers le réseau, bien que cela augmente la complexité computationnelle.

ResNext : ResNext améliorait l'architecture Inception en introduisant la cardinalité, ce qui permettait une utilisation plus efficace des ressources du réseau. Il utilisait des transformations multiples dans les blocs pour améliorer les performances.

WideResNet : WideResNet résolvait le problème de la réutilisation des caractéristiques dans les réseaux profonds en augmentant la largeur du réseau au lieu de la profondeur. Il utilisait des blocs résiduels plus larges et une régularisation dropout pour améliorer l'efficacité de l'entraînement (Alzubaidi et al. 2021).

Réseaux de neurones récurents (RNN)

- **RNN :** Les réseaux neuronaux récurrents (RNN) ont été développés dans les années 1980 pour modéliser les séquences temporelles. Contrairement aux réseaux neuronaux classiques, les RNN peuvent prendre en compte des dépendances à long terme dans les données en traitant les séquences itérativement à l'aide de boucles de rétroaction (Graves 2013).
- **LSTM :** Les mémoires à court et long terme (LSTM) sont une variante avancée des RNN, introduite pour résoudre le problème du gradient qui disparaît. Les LSTM sont dotés de portes d'entrée, de sortie et d'oubli, qui leur permettent de maintenir et d'oublier sélectivement les informations au fil du temps, ce qui les rend particulièrement efficaces pour capturer les dépendances à long terme dans les données séquentielles (Lindemann et al. 2021).
- **GRU :** Les unités récurrentes à portes (GRU) sont une alternative plus légère aux LSTM, tout en conservant une performance similaire dans de nombreuses tâches. Les GRU combinent les portes d'oubli et d'entrée en une seule "porte de mise à jour", simplifiant ainsi l'architecture tout en maintenant une capacité de modélisation efficace des dépendances à long terme (Chung et al. 2014).
- **Applications des RNN :** Les RNN, y compris les LSTM et les GRU, sont largement utilisés dans divers domaines. Parmi leurs applications figurent le traitement du langage naturel, la modélisation de séries temporelles, la reconnaissance d'actions dans les vidéos, la classification d'images, la prédiction de séquences et bien d'autres encore. Leur capacité à capturer les relations séquentielles en fait un outil polyvalent pour une gamme étendue de problèmes (Hochreiter et Schmidhuber 1997).

Auto-Encodeurs (AE)

Les auto-encodeurs (AE) sont des réseaux neuronaux qui visent à reproduire leur entrée à la couche de sortie. Ils se composent d'un encodeur et d'un décodeur, qui cartographient l'entrée vers une représentation latente, puis reconstruisent l'entrée à partir de cette représentation, respectivement. La structure AE de base comprend une couche d'entrée, une couche cachée pour l'encodage et une couche de sortie qui reflète l'entrée. Une fonction de perte, telle que l'erreur quadratique moyenne (MSE), est minimisée pendant l'entraînement pour garantir une reconstruction précise (Hinton et Salakhutdinov 2006)(Chen et Guo 2023).

Variantes des AE :

- **Auto-encodeur épars (Sparse Autoencoder) :** Introduit des contraintes de parcimonie sur les unités cachées pour apprendre des représentations plus informatives (Ranzato et al. 2006).
- **Auto-encodeur débruiteur (Denoising Autoencoder) :** Entraîné pour reconstruire des données propres à partir de versions corrompues, améliorant la robustesse et l'extraction de caractéristiques (Vincent et al. 2010).
- **Auto-encodeur variationnel (Variatinal Autoencoder) :** Ajoute des contraintes sur les représentations encodées, permettant la modélisation probabiliste et la génération de nouveaux échantillons de données (Kingma et Welling 2022).

Applications des auto-encodeurs : Les auto-encodeurs sont largement utilisés pour la réduction de dimensionnalité, l'apprentissage de caractéristiques et les tâches de modélisation générative. Ils ont des applications dans le débruitage d'images, la détection d'anomalies, la compression de données et la génération d'échantillons de données réalistes (Berahmand et al. 2024).

Réseaux génératifs antagonistes (GAN)

Les réseaux génératifs antagonistes se sont imposés comme des modèles puissants pour diverses applications, notamment la génération d'images, la prédiction vidéo et l'augmentation de données. Le concept fondamental des GAN implique l'entraînement simultané de deux réseaux neuronaux : un générateur et un discriminateur. Le générateur vise à produire des échantillons réalistes à partir de bruit aléatoire, tandis que l'objectif du discriminateur est de distinguer entre les échantillons faux et réels. Grâce à l'entraînement antagoniste, le générateur apprend à tromper le discriminateur, générant ainsi des échantillons de haute qualité qui ressemblent étroitement à des données réelles (Goodfellow et al., 2014) (Goodfellow et al. 2014) (Zhao, Mathieu et LeCun 2016). Plusieurs variantes et extensions des GAN ont été proposées pour relever des défis spécifiques et améliorer le contrôle sur le processus de génération. Par exemple, le GAN à classificateur auxiliaire (ACGAN) incorpore une couche de classification supplémentaire dans le discriminateur pour guider le processus de génération en fonction d'étiquettes conditionnelles (Lee et Seok 2019). Malgré leur efficacité, contrôler l'aléatoire inhérent aux échantillons générés par les GAN reste un défi. Les GAN ont trouvé de nombreuses applications dans des domaines variés, tels que la médecine, la réponse aux pandémies, le traitement d'images, la détection de visages, le transfert de texture et le contrôle du trafic (Aggarwal, Mittal et Battineni 2021). Dans le

domaine de l'imagerie médicale, les GAN jouent un rôle crucial dans des tâches telles que la reconstruction d'images, la synthèse, la segmentation, la classification, la détection d'anomalies et l'enregistrement (Yi, Walia et Babyn 2019) (Iqbal et Ali 2018). Ces applications illustrent la polyvalence et l'impact potentiel des GAN dans divers scénarios réels.

2.5 Les principes fondamentaux de la sécurité informatique

2.5.1 Définition

La sécurité informatique se définit comme l'ensemble des mesures techniques, organisationnelles et humaines visant à protéger les systèmes d'information contre toute forme d'attaque, de vol, de destruction ou d'altération des données. Elle s'articule autour de trois objectifs principaux : la confidentialité, l'intégrité et la disponibilité des informations et des systèmes. La sécurité informatique ne se limite pas à la protection des données mais englobe également la protection des ressources et des infrastructures numériques. Elle repose sur la mise en œuvre de stratégies et de politiques de sécurité adaptées à chaque environnement technologique et organisationnel .

2.5.2 Principes de base de la sécurité informatique

Les principes de base de la sécurité informatique sont ancrés dans trois concepts clés : la confidentialité, l'intégrité et la disponibilité.

- **Confidentialité** La confidentialité vise à assurer que les informations sensibles ne sont accessibles qu'aux personnes autorisées. Elle implique l'utilisation de techniques de cryptage, de contrôle d'accès, et de gestion des identités et des accès (IAM). La confidentialité est cruciale pour prévenir les fuites de données et protéger la vie privée des utilisateurs.
- **Intégrité**
  Intégrité : L'intégrité garantit que les données et les systèmes sont exacts et complets, et qu'ils n'ont pas été modifiés de manière non autorisée. Les mécanismes de contrôle de l'intégrité comprennent l'utilisation de hachages cryptographiques, de signatures numériques, et de journaux d'audit. L'intégrité des données est essentielle pour maintenir la fiabilité et la crédibilité des informations.

— **Disponibilité**
Disponibilité : La disponibilité assure que les systèmes et les données sont accessibles et utilisables par les utilisateurs autorisés lorsqu'ils en ont besoin. Cela implique la mise en œuvre de plans de reprise après sinistre, de sauvegardes régulières, et de systèmes redondants pour prévenir les interruptions de service. La disponibilité est critique pour garantir la continuité des opérations et des services

Ces principes sont réalisés par la mise en œuvre de fonctions de sécurité telles que le contrôle d'accès, l'authentification, la traçabilité, et la gestion des incidents. Les stratégies de sécurité doivent également inclure des politiques de sauvegarde et des procédures pour gérer les risques et assurer la continuité des opérations en cas d'incident .

2.5.3 Défis de la sécurité dans le domaine de l'IoT

L'internet des objets (IoT) présente des défis uniques en matière de sécurité en raison de la diversité des dispositifs connectés et de leur intégration dans les réseaux. Voici quelques-uns des principaux défis :

— **Hétérogénéité des dispositifs** : Les dispositifs IoT varient largement en termes de capacité de traitement, de mémoire et de puissance, rendant difficile la mise en place de mesures de sécurité uniformes. Chaque dispositif peut avoir des exigences et des vulnérabilités spécifiques, ce qui complique la gestion de la sécurité.
— **Scalabilité** : Le nombre massif de dispositifs IoT nécessite des solutions de sécurité qui peuvent s'adapter à grande échelle sans compromettre les performances. La gestion de la sécurité doit pouvoir évoluer pour protéger un nombre croissant de dispositifs et de connexions.
— **Mises à jour et gestion des correctifs** : Beaucoup de dispositifs IoT sont difficiles à mettre à jour, ce qui pose des risques importants si des vulnérabilités sont découvertes. L'absence de mécanismes de mise à jour automatiques ou faciles à déployer peut laisser des dispositifs exposés aux attaques pendant des périodes prolongées.
— **Sécurité physique** : Les dispositifs IoT sont souvent déployés dans des environnements non sécurisés, augmentant le risque de manipulation physique et de compromission. La protection physique des dispositifs est essentielle pour prévenir les attaques matérielles et les altérations non autorisées.

La sécurité informatique est un domaine complexe et en constante évolution, nécessitant une approche holistique intégrant des aspects techniques, organisationnels et humains. La protection des systèmes d'information repose sur des principes fondamentaux tels que la confidentialité, l'intégrité et la disponibilité. Dans le contexte de l'IoT, ces prin-

cipes doivent être adaptés pour faire face à des défis spécifiques liés à l'hétérogénéité et à la scalabilité des dispositifs. Une compréhension approfondie de ces principes et des défis permet de développer des stratégies de sécurité efficaces pour protéger les infrastructures numériques contre une variété de menaces. Les méthodes d'apprentissage automatique et profond peuvent jouer un grand rôle dans la mise en oeuvre des solutions de sécurité dans le domaine de l'IoT.

2.6 Applications de l'IA en cybersécurité

L'adoption croissante de l'intelligence artificielle dans divers secteurs a inévitablement étendu son influence à la cybersécurité, un domaine où la rapidité et la précision sont primordiales. En combinant des capacités avancées de traitement de données et d'apprentissage automatique, l'IA offre des perspectives révolutionnaires pour renforcer les mécanismes de défense contre les cyberattaques toujours plus sophistiquées. Cette section explore comment l'IA est utilisée pour renforcer la cybersécurité à travers trois applications clés : la détection d'anomalies, la prédiction des attaques et la surveillance du trafic réseau.

2.6.1 Détection d'anomalies

La détection d'anomalies est cruciale pour identifier les comportements inhabituels ou suspects dans les systèmes informatiques. L'IA, grâce à son habileté à apprendre et à reconnaître des schémas complexes, peut rapidement détecter des écarts par rapport à la norme qui pourraient indiquer une activité malveillante. Cette approche est fondamentale pour non seulement reconnaître les menaces connues mais aussi pour anticiper les attaques inédites, offrant ainsi une protection proactive contre les cybermenaces. Les techniques d'apprentissage non supervisé sont les plus utilisées dans ce domaine. Les travaux exposés dans (Usama et al. 2017) se concentrent sur l'utilisation de l'apprentissage non supervisé en matière de réseaux informatiques, abordant les techniques, applications, et défis de recherche associés. Il décrit comment l'apprentissage non supervisé peut aider à gérer les réseaux de manière plus efficace en utilisant des données non structurées pour améliorer la performance du réseau, notamment dans la gestion du trafic, la détection d'anomalies, la classification du trafic internet, et l'optimisation de la qualité de service.

Ahmad et al. ont exploré l'utilisation des réseaux de neurones profonds (DNN) pour améliorer la détection des anomalies dans les architectures IoT, démontrant une amélioration notable par rapport aux modèles d'apprentissage automatique traditionnels (Ahmad

et al. 2021). Naseer et al. ont développé des modèles basés sur différents types de DNN, y compris les réseaux convolutifs (CNN) et les réseaux récurrents (LSTM), pour la détection d'anomalies dans les réseaux, enregistrant une performance accrue par rapport aux techniques classiques (Naseer et al. 2018). Lin et al. proposent un système de détection dynamique des anomalies réseau utilisant des techniques d'apprentissage profond. Les auteurs ont développé un modèle basé sur des réseaux de neurones récurrents à longue et courte mémoire (LSTM) avec un mécanisme d'attention pour améliorer les performances de détection. Ils ont également utilisé l'algorithme SMOTE pour gérer le problème de déséquilibre des classes dans le dataset CSE-CIC-IDS2018. Le système montre une précision de classification de 96.2%, surpassant d'autres algorithmes d'apprentissage machine traditionnels (Lin, Ye et Xu 2019).Le modèle utilise LSTM pour traiter les séquences de données de trafic réseau, ce qui est essentiel pour comprendre les dépendances temporelles dans les activités de réseau. Le mécanisme d'attention aide à concentrer le modèle sur les aspects les plus pertinents des données, ce qui améliore l'exactitude de la détection.L'utilisation de SMOTE aide à surmonter le déséquilibre des classes dans les données d'entraînement, permettant au modèle d'être plus efficace dans la détection de divers types d'anomalies.

Les spécificités des réseaux IoT nécessitent des approches adaptées en matière de sécurité, notamment des algorithmes d'apprentissage automatique conçus pour fonctionner avec des ressources limitées. C'est ainsi que Huc et al. explorent les défis et les solutions pour la détection d'anomalies dans les réseaux IoT, en mettant l'accent sur l'importance de l'architecture des systèmes et de la transparence des processus d'apprentissage automatique (Huc et Trcek 2021). Les auteurs discutent de la nécessité d'architectures légères et adaptées aux contraintes spécifiques des réseaux IoT, telles que la limitation des ressources de calcul et d'énergie. Ils proposent une architecture en couches pour la détection d'anomalies et soulignent l'importance de la transparence fonctionnelle et des données pour améliorer la compréhension et l'explicabilité des modèles d'apprentissage automatique. Les réseaux IoT sont extrêmement vulnérables à des attaques en raison de leur interconnectivité et de leur ouverture. Pour adresser cette vulnérabilité, les auteurs ont développé un système de détection d'intrusions basé sur l'apprentissage profond, en utilisant spécifiquement une architecture d'auto-encodeur LSTM. Ce système apprend à identifier les activités suspectes ou anormales en analysant le trafic des réseaux IoT. Il a été testé avec succès sur deux ensembles de données de référence, atteignant une précision de plus de 99% pour le dataset de piratage de voitures et de 98% pour le dataset UNSW-NB15 (Saba et al. 2022).

Les techniques de détection d'anomalies sont cruciales pour identifier les comportements qui dévient des normes établies dans des ensembles de données complexes. La classification et les méthodes basées sur les plus proches voisins sont couramment utilisées

pour les anomalies pointues, tandis que le clustering et les approches spectrales sont efficaces pour détecter des groupes d'anomalies. Toshniwal et al. ont présenté une vue d'ensemble des méthodes de détection d'anomalies appliquées dans l'apprentissage machine, mettant en lumière les diverses techniques comme la classification, le clustering, et les approches statistiques pour identifier les comportements anormaux (Toshniwal, Mahesh et Jayashree 2020). Ces méthodes sont essentielles pour traiter les ensembles de données complexes rencontrés dans les réseaux de communication et la surveillance de la santé.

Dawoud et al. abordent la question de la détection d'intrusions dans les réseaux de l'Internet des Objets (IoT) en utilisant des méthodes d'apprentissage profond. Les auteurs proposent un cadre de détection d'anomalies basé sur des algorithmes d'apprentissage profond non supervisés, notamment les machines de Boltzmann restreintes (RBMs) et les auto-encodeurs (AEs). L'objectif est d'identifier les menaces de réseau en analysant les données massives générées par les dispositifs IoT. Leur étude met en avant l'efficacité des algorithmes proposés à travers des simulations qui montrent une précision de détection d'environ 99%, surpassant les travaux connexes (Dawoud et al. 2020a).

L'utilisation des techniques d'apprentissage non supervisé pour la détection d'anomalies a été étudiée par Kabir et Luo, qui ont examiné des approches comme le K-Means et les Cartes Auto-Organisatrices pour évaluer leur efficacité dans la détection des anomalies basées sur les flux de réseau (Kabir et Luo 2020).L'étude évalue ces algorithmes sur deux jeux de données de référence, en se concentrant sur leur efficacité à détecter des attaques réseau connues et nouvelles (non vues). Les critères d'évaluation incluent les taux de détection et les taux de faux positifs. Dans leurs travaux, Hwang et al. présentent un modèle d'apprentissage profond non supervisé, nommé D-PACK, pour la détection précoce des anomalies dans le trafic réseau. Ce modèle utilise une combinaison de réseaux de neurones convolutionnels (CNN) et d'un auto-encodeur pour profiler automatiquement les patterns de trafic et filtrer les activités anormales. L'approche est conçue pour inspecter seulement les premiers paquets de chaque flux, permettant une détection rapide et efficace avec une précision proche de 100% et un taux très bas de fausses alertes(Hwang et al. 2020).

Meira et al. ont analysé l'efficacité de six algorithmes d'apprentissage non supervisé, dont l'auto-encodeur et l'Isolation Forest, le plus proche voisin pour détecter des attaques inconnues et des anomalies sans entrainement sur des exemples spécifiques d'attaques (Meira et al. 2020). Les auto-encodeurs ont également été utilisés par Legrand et al. pour détecter des anomalies dans les bâtiments connectés, où ils ont pu analyser efficacement les données provenant de nombreux capteurs (Legrand et al. 2018a).Ils mettent en lumière comment ces réseaux peuvent être utilisés pour analyser les données de haute dimensionnalité issues des nombreux capteurs présents dans les bâtiments modernes. Les auteurs

examinent plusieurs types d'auto-encodeurs, y compris ceux récurrents et convolutionnels, pour leur capacité à détecter des comportements anormaux de manière supervisée ou non supervisée, étant donné l'absence fréquente de données d'anomalies étiquetées. Dans (Guo et al. 2020), est proposée une méthode de détection d'anomalies non supervisée pour les systèmes IoT dans les villes intelligentes, utilisant une combinaison d'Unités Récurrentes à Portes (GRU) et d'auto-encodeurs Variationnels à Mélange Gaussien (GGM-VAE). L'approche est centrée sur la gestion de données temporelles multidimensionnelles, utilisant les GRU pour capter les corrélations temporelles et les mélanges gaussiens pour caractériser la multimodalité des données. Des simulations étendues sur quatre ensembles de données démontrent que cette approche dépasse les performances des méthodes de détection d'anomalies existantes, en améliorant les scores F1 jusqu'à 47.88% en moyenne.

En outre, Apostol et al. ont abordé la détection des anomalies des botnets IoT en utilisant des techniques d'apprentissage profond non supervisé, démontrant la capacité de ces méthodes à identifier les comportements de botnets dans les réseaux IoT (Apostol et al. 2021a).Une évaluation empirique est menée sur des ensembles de données équilibrés et déséquilibrés pour évaluer la capacité de détection des menaces de la méthode proposée. Les résultats expérimentaux révèlent la performance de la méthode de détection proposée et incluent une comparaison avec d'autres approches d'apprentissage non supervisé. Abordant dans le même sens, les travaux dans (Nõmm et Bahsi 2018) se concentrent sur la détection de botnets dans les réseaux de l'Internet des objets (IoT) en utilisant des méthodes d'apprentissage automatique non supervisées, avec un accent particulier sur la sélection de caractéristiques. En raison de la croissance rapide de la technologie IoT, le nombre de dispositifs vulnérables susceptibles de faire partie d'un botnet a considérablement augmenté. Les auteurs ont exploré l'utilisation de modèles d'apprentissage non supervisés avec des ensembles de caractéristiques réduits pour minimiser les ressources computationnelles requises, tout en maintenant une haute précision dans la détection du trafic malveillant.

Dans(Tien et al. 2021) les auteurs abordent l'utilisation des auto-encodeurs pour la détection d'anomalies et l'apprentissage par transfert dans les environnements IoT, en particulier dans les usines intelligentes. Les auteurs proposent une méthode d'apprentissage automatique pour identifier les types d'appareils via l'analyse du trafic réseau, utilisant d'abord l'apprentissage supervisé pour apprendre les caractéristiques des différents types de dispositifs. Ensuite, des méthodes d'apprentissage non supervisé telles que l'auto-encodeur sont employées pour la réduction de dimensionnalité et la détection d'anomalies. L'efficacité des auto-encodeurs dans l'apprentissage des caractéristiques essentielles pour la détection d'anomalies est démontrée par des expériences sur des données réelles.L'intégration des auto-encodeurs permet une détection précise des anomalies en apprenant les com-

portements normaux des dispositifs, avec un score F1 atteignant jusqu'à 98.36% dans certaines configurations. Cette méthode montre également un potentiel dans l'apprentissage par transfert, permettant l'application des modèles d'apprentissage machine à différents sites sans étiquetage manuel laborieux. Les expériences montrent que les auto-encodeurs surpassent d'autres méthodes telles que SVM à classe unique et la forêt d'isolation dans les tâches de détection d'anomalies, soulignant leur adaptabilité et efficacité dans des scénarios industriels complexes.

En conclusion, les techniques d'apprentissage automatique et profond, grâce notamment aux approches basées sur l'apprentissage non supervisé, montrent un potentiel significatif pour améliorer la sécurité des réseaux modernes, et IoT en particulier, où la capacité à détecter des anomalies précisément et rapidement peut significativement contribuer à la protection contre les cybermenaces. Les recherches continuent de pousser les limites de ce qui est possible, affinant constamment la précision et l'efficacité de ces systèmes de détection.

2.6.2 Détection d'intrusions et classification d'attaques

Un système de détection d'intrusion (IDS) est un dispositif ou un logiciel utilisé pour surveiller un réseau ou des systèmes à la recherche d'activités malveillantes ou de violations des politiques de sécurité. Les IDS analysent le trafic réseau pour détecter des comportements anormaux, identifier des attaques potentielles et émettre des alertes pour permettre une réponse rapide. Les systèmes IoT, avec leurs ressources de calcul et de stockage limitées, ne peuvent pas supporter efficacement les logiciels de sécurité traditionnels. Les systèmes de détection d'intrusion (IDS) habituels sont souvent inadaptés pour les environnements IoT en raison de la diversité des dispositifs, des comportements normaux variables, et de la vulnérabilité croissante due à l'expansion rapide des appareils connectés. Un changement d'approche est donc nécessaire pour assurer leur sécurité. Les deux méthodes principales utilisées dans les IDS pour IoT sont les techniques basées sur les signatures, qui utilisent des connaissances existantes sur les menaces pour identifier le trafic malveillant, et les approches basées sur les anomalies, qui détectent les activités suspectes en analysant les modèles de trafic. Tandis que les techniques basées sur les signatures sont efficaces contre les menaces connues, elles ne peuvent pas détecter les nouvelles attaques, notamment les attaques zero-day, en raison de leur dépendance à une base de données qui doit être constamment mise à jour. En revanche, L'un des avantages de l'utilisation de systèmes basés sur les anomalies est qu'ils sont efficaces dans la détection des attaques zero-day et inconnues.

Dans ce contexte, les travaux de Lewandowska mettent en lumière l'évolution des méthodologies d'IDS, incluant à la fois les approches basées sur les signatures et celles basées sur les anomalies, ainsi que les systèmes de prévention d'intrusions (IPS) qui bloquent activement les activités malveillantes détectées (Lewandowska 2024). Un nombre croissant de recherches se concentre sur l'utilisation de l'apprentissage profond pour renforcer les IDS dans les réseaux IoT. Par exemple, Nahida et al. ont examiné l'efficacité des systèmes IDS basés sur l'apprentissage automatique et l' apprentissage profond, en démontrant que les approches basées sur le DL, notamment les LSTM et les réseaux LSTM empilés, surpassent les méthodes traditionnelles d'apprentissage automatique en termes de précision, rappel et score F1 (Nahida Islam 2021). Pour leur part, Saba et al. mettent en place un CNN qui permet de traiter efficacement les grandes quantités de données générées par les dispositifs IoT, offrant une solution viable pour sécuriser ces réseaux contre des menaces croissantes. Leur article présente une méthode d'IDS (Intrusion Detection System) basée sur les anomalies pour les réseaux IoT, en utilisant un modèle d'apprentissage profond. Le système utilise un réseau de neurones convolutifs (CNN) pour analyser le trafic des réseaux IoT et détecter des comportements anormaux, potentiellement malveillants. Le modèle est évalué avec deux ensembles de données, NID et BoT-IoT, atteignant une précision de 99.51% et 92.85% respectivement (Saba et al. 2022).

Dans la même lancée l'utilisation de l'apprentissage profond pour développer un système de détection d'intrusion réseau adaptable et résilient a été examiné par (Ashiku et Dagli 2021). Les auteurs se concentrent sur la manière dont les réseaux de neurones profonds peuvent offrir un système de détection d'intrusion flexible, capable d'apprendre et de détecter des caractéristiques comportementales de réseau reconnues ainsi que nouvelles ou de type zero-day. Pour valider l'efficacité de leur modèle, ils utilisent le jeu de données UNSW-NB15, qui reflète le comportement de communication réseau moderne avec des activités d'attaque générées synthétiquement. Dans (Malgwi, Ibrahim et Bamanga Ahmad 2022) est proposé un modèle de système de détection d'intrusion (IDS) utilisant des réseaux de neurones artificiels (ANN) pour améliorer la sécurité des réseaux informatiques. Les auteurs utilisent un dataset réaliste, CICIDS2017, pour entrainer et tester leur modèle. Le modèle ANN développé vise à détecter des intrusions en identifiant des comportements anormaux dans le trafic réseau, avec une attention particulière aux attaques zero-day et autres menaces émergentes. Le modèle ANN a montré une performance exceptionnelle dans la détection des intrusions, atteignant une aire sous la courbe (AUC) de 99.999% sur la courbe caractéristique de fonctionnement du récepteur (ROC), ce qui indique une très haute capacité de discrimination entre les activités bénignes et malicieuses.

Pour améliorer la détection dans les environnements IoT, qui présentent des défis uniques

en raison de la nature éphémère et de la connectivité intermittente des dispositifs, Yoon propose une méthode innovante utilisant les réseaux de neurones artificiels pour analyser les paquets réseau. Cette méthode adapte les nœuds neuronaux pour identifier spécifiquement les paquets potentiellement vulnérables, démontrant une nette amélioration par rapport aux méthodes traditionnelles (Yoon 2019). Ces travaux soulignent l'importance de l'adaptabilité des systèmes de sécurité IoT pour gérer les caractéristiques uniques des dispositifs IoT, telles que leur variabilité et leur connectivité intermittente. Les résultats montrent que les approches basées sur l'apprentissage profond peuvent offrir une meilleure détection des menaces en analysant les paquets de données et en identifiant des modèles qui pourraient indiquer des tentatives d'attaque. Les recherches d'Islam et al. illustrent l'efficacité de l'apprentissage profond dans l'amélioration de la détection d'intrusions pour les réseaux IoT. En utilisant des ensembles de données comme NSL-KDD et IoT datasets, ils montrent que les modèles d'apprentissage profond, notamment les LSTM et les réseaux neuronaux empilés, surpassent les approches classiques en offrant une meilleure précision, rappel, et score F1 (Nahida Islam 2021).

Par ailleurs, Otoum et al. développent un cadre nommé DL-IDS basé sur l'apprentissage profond pour la détection d'intrusions dans les environnements IoT. Ils utilisent l'algorithme d'optimisation Spider Monkey pour la sélection de caractéristiques et un réseau de polynômes profonds empilés pour la classification, aboutissant à des performances significativement améliorées pour la détection de divers types d'attaques (Otoum Ph.D P.Eng, Liu et Nayak 2022). Les auteurs mettent en avant l'efficacité de cette approche pour identifier divers types d'attaques comme les attaques DoS, U2R, de sondage et R2L, en démontrant une amélioration significative de la précision, du rappel et du score F par rapport aux méthodes traditionnelles. Le système DL-IDS combine efficacement des techniques d'optimisation et d'apprentissage profond pour améliorer la détection d'intrusions, en gérant les données incertaines ou manquantes et en apprenant les caractéristiques de manière optimale pour améliorer la précision de détection. Des analyses détaillées montrent que le DL-IDS réalise de meilleures performances en termes de précision, de précision, de rappel et de score F, indiquant une capacité robuste à détecter des attaques dans des environnements IoT variés. Abu Al-Haija propose une architecture d'apprentissage machine pour l'identification et la classification des cyberattaques dans les réseaux de communication IoT. L'auteur introduit une approche top-down pour développer un système capable de détecter les mutations de cyberattaques communes utilisant des méthodes non traditionnelles d'apprentissage machine. Le système est conçu avec trois sous-systèmes principaux : l'ingénierie des caractéristiques, l'apprentissage des caractéristiques, et la détection et classification, utilisant des ensembles de données comme le CICIDS et le MQTT. L'architecture tire parti des réseaux neuronaux profonds et convolutifs pour améliorer l'exactitude de détection/classification

des attaques IoT (Abu Al-Haija 2022).

En outre, Ashraf et al. passent en revue les méthodes d'IDS basées sur le ML et le DL dans les réseaux IoT, soulignant l'efficacité de ces techniques pour identifier les cyberattaques et discutant des défis persistants pour sécuriser ces dispositifs et réseaux connectés (Ashraf et al. 2020). L'étude couvre les aspects de l'architecture IoT, souligne les défis posés par les vulnérabilités des dispositifs IoT, et discute des protocoles impliqués. Les différentes techniques de ML et DL sont examinées en fonction de leur efficacité à identifier les cyberattaques, mettant en lumière les progrès réalisés ainsi que les défis persistants dans la sécurisation des dispositifs et réseaux IoT. La détection des attaques minoritaires, qui sont souvent sous-représentées et donc négligées dans les systèmes IDS, est abordée par Sharma et Mukherjee. Ils proposent une approche multicouche combinant des classificateurs bayésiens naïfs et des arbres bayésiens pour améliorer le taux de détection des attaques R2L et U2R, optimisant ainsi la précision sans compromettre la performance globale (Sharma et Mukherjee 2012).

L'article de Shone et al. met en avant un modèle de NIDS basé sur un auto-encodeur profond non symétrique pour l'apprentissage non supervisé des caractéristiques, combiné à un système de classification qui utilise des auto-encodeurs empilés et l'algorithme de forêt aléatoire. Ce système a été testé sur les ensembles de données KDD Cup99 et NSL-KDD, montrant des améliorations significatives par rapport aux méthodes traditionnelles (Shone et al. 2018). Le même jeu de données a été utilisé dans (Gurung, Ghose et Subedi 2019) pour l'entraînement et la validation des modèles. Les auteurs mettent en œuvre un auto-encodeur profond non symétrique (NDAE) pour l'apprentissage non supervisé des caractéristiques, suivi d'une classification par régression logistique. L'objectif est de détecter à la fois les attaques connues et inconnues par le biais d'une analyse de comportement des flux de trafic réseau. Chaabouni et al. examinent les systèmes de détection d'intrusions réseau (NIDS) déployés pour sécuriser les systèmes de l'Internet des Objets (IoT) en utilisant des techniques d'apprentissage automatique. Ils abordent la croissance omniprésente de l'IoT et les défis de sécurité que cela pose, en soulignant la nécessité d'approches avancées pour détecter et prévenir les cyberattaques. L'article propose une classification des menaces IoT, évalue les techniques de défense existantes, et examine l'application des techniques d'apprentissage pour améliorer les NIDS dans le contexte de l'IoT (Chaabouni et al. 2019).

En définitive, l'avancement des techniques d'apprentissage profond et leur application dans les systèmes IDS modernes ont considérablement renforcé la capacité de détecter et de classifier efficacement les attaques réseau, s'adaptant ainsi aux défis posés par l'évolution rapide des environnements numériques et des tactiques d'attaque.

2.6.3 Identification des dispositifs IoT

La classification des dispositifs IoT est devenue une composante essentielle pour la gestion efficace des réseaux intelligents. L'Internet des objets (IoT) connecte une multitude de dispositifs hétérogènes, allant des capteurs simples aux appareils plus complexes comme les caméras de surveillance et les thermostats intelligents. Cette diversité entraîne une complexité accrue dans la gestion des réseaux, rendant nécessaire le développement de techniques avancées pour identifier et classifier les dispositifs en fonction de leur trafic réseau. Les méthodes traditionnelles de gestion et de sécurité des réseaux ne suffisent plus à répondre aux besoins spécifiques des environnements IoT, où les dispositifs génèrent des flux de données variés et continus. L'apprentissage automatique (ML) et l'apprentissage profond (DL) se révèlent être des solutions puissantes pour analyser ces flux de données et extraire des caractéristiques pertinentes pour la classification des dispositifs IoT. Ces approches permettent non seulement d'améliorer la précision de la classification, mais aussi de détecter des comportements anormaux, renforçant ainsi la sécurité et la fiabilité des réseaux IoT. Dans cette section, nous explorerons les différentes approches de classification des dispositifs IoT basées sur le ML et le DL, en mettant en lumière leurs forces et leurs limites.

Bassene et Gueye ont proposé une approche de classification appelée GBC-IoT, utilisant six algorithmes d'apprentissage automatique (ANN, k-NN, DT, GBN, RF, SVM) pour analyser les caractéristiques du trafic bidirectionnel. Cette méthode a atteint une précision globale de 99,98% grâce à une sélection rigoureuse des algorithmes les plus performants pour le modèle final, tout en réduisant la surcharge de traitement en utilisant moins de caractéristiques(Bassene et Gueye 2021). De même, Bai et al. (2018) ont proposé une méthode de classification automatique des dispositifs IoT en utilisant un modèle cascade LSTM-CNN. Ce modèle capture les corrélations temporelles globales et locales du trafic réseau, atteignant une précision de 74,8% pour la classification de nouveaux dispositifs(Bai et al. 2018). Jmila et al. (2022) ont examiné les défis et les méthodologies de classification des dispositifs IoT dans les maisons intelligentes. Leur cadre implique l'acquisition de données de trafic réseau, l'extraction de caractéristiques pertinentes, et l'utilisation d'algorithmes d'apprentissage automatique pour classifier les dispositifs, comparant différentes méthodes d'extraction de caractéristiques et algorithmes pour améliorer la précision de classification(Jmila et al. 2022).

Resende et al. ont développé un modèle pour identifier les dispositifs IoT et les PC dans les réseaux domestiques en utilisant des algorithmes d'apprentissage automatique supervisés. Leur étude a évalué l'impact de l'entraînement avec différents types de dispositifs et la

capacité à identifier de nouveaux types de dispositifs IoT, montrant des résultats significatifs pour l'identification précise du trafic IoT et non-IoT(De Resende et al. 2022). Shahid et al. (2018) ont proposé une approche utilisant l'apprentissage automatique pour reconnaître les types de dispositifs IoT en analysant les flux de paquets. En utilisant la visualisation t-SNE et six classificateurs différents, ils ont atteint une précision de 99,9% avec le classificateur Random Forest, bien que cette étude soit limitée à un réseau domestique expérimental avec seulement quatre dispositifs IoT(Shahid et al. 2018). Aksoy et Gunes (2019) ont développé SysID, un système de classification automatisée des dispositifs IoT basé sur des algorithmes d'apprentissage automatique et un algorithme génétique pour la sélection des caractéristiques. SysID a atteint une précision de classification de plus de 95% pour un seul paquet, démontrant une haute performance sans intervention humaine pour la sélection des caractéristiques(Aksoy et Gunes 2019). Lopez-Martin et al. (2017) ont proposé une technique de classification du trafic réseau en combinant des réseaux de neurones récurrents (RNN) avec des réseaux de neurones convolutionnels (CNN). Cette méthode a atteint une précision de 96,32% et un score F1 de 95,74%, en s'appuyant uniquement sur les en-têtes de paquets sans nécessiter de feature engineering(Lopez-Martin et al. 2017). Ullah et Mahmoud (2021) ont développé une technique de classification basée sur l'analyse des flux de trafic réseau pour identifier les dispositifs IoT connectés. En utilisant des fichiers Pcap de l'ensemble de données IoT23, ils ont construit un modèle qui atteint une précision de 100% pour l'identification des dispositifs en utilisant des ensembles de caractéristiques complètes, réduites et basées sur les flux(Ullah et Mahmoud 2021b).

Les approches basées sur l'apprentissage automatique et l' apprentissage profond pour la classification des dispositifs IoT offrent des résultats prometteurs en termes de précision et d'adaptabilité. Cependant, elles présentent également des défis, notamment la dépendance aux données, la complexité computationnelle, et la nécessité d'une infrastructure adéquate pour la collecte et l'analyse des données en temps réel. La robustesse et la généralisation des modèles doivent être améliorées pour qu'ils puissent fonctionner efficacement dans des environnements réels variés et dynamiques. Il est crucial de développer des techniques de classification capables de s'adapter aux évolutions rapides des technologies IoT et aux nouvelles menaces de sécurité. Par ailleurs, l'intégration de ces techniques dans des solutions pratiques et évolutives est essentielle pour répondre aux besoins croissants de gestion et de sécurité des réseaux IoT. Les recherches futures devraient également explorer l'application de ces méthodes dans des contextes industriels et commerciaux pour valider leur efficacité à grande échelle. La collaboration entre les chercheurs, les fabricants de dispositifs IoT, et les professionnels de la sécurité réseau sera déterminante pour le développement de solutions holistiques et résilientes. La classification des dispositifs IoT est essentielle pour la gestion et la sécurité des réseaux intelligents. Les approches basées sur l'apprentissage

automatique et l'apprentissage profond, bien que prometteuses, nécessitent des améliorations continues pour surmonter les défis actuels. L'intégration de ces techniques dans des solutions pratiques et évolutives contribuera à une meilleure gestion des dispositifs IoT et à une protection renforcée contre les menaces émergentes. En fin de compte, l'objectif est de créer des réseaux IoT plus sûrs, plus efficaces et capables de s'adapter aux évolutions technologiques rapides. Pour cela, une coopération étroite entre les chercheurs et l'industrie est nécessaire pour mettre en place des systèmes robustes et évolutifs, capables de répondre aux défis futurs du monde connecté.

2.6.4 Discussion

L'étude comparative des solutions de sécurité mentionnées dans cette section met en lumière à la fois leurs forces et leurs faiblesses. Les approches de détection d'anomalies basées sur l'apprentissage non supervisé, comme les auto-encodeurs, excellent dans l'identification des anomalies sans nécessiter de données étiquetées. Cependant, elles peuvent souffrir de taux de faux positifs élevés, et leur interprétabilité reste un défi. D'autre part, les techniques supervisées, telles que les réseaux convolutifs (CNN) et les réseaux de neurones à mémoire à long terme (LSTM), offrent des performances de classification élevées avec des données étiquetées mais sont limitées dans leur capacité à détecter des attaques inconnues.La nécessité de combiner les algorithmes d'apprentissage supervisé et non supervisé devient alors évidente. En intégrant des modèles non supervisés pour la détection initiale d'anomalies avec des modèles supervisés pour la classification précise des attaques, il est possible de tirer parti des avantages de chaque approche. Cette combinaison permet de réduire les taux de faux positifs, d'améliorer la précision de la détection et d'offrir une meilleure capacité à identifier des attaques zero-day.

Les travaux de (Lin, Ye et Xu 2019) illustrent cette synergie en utilisant des réseaux LSTM avec un mécanisme d'attention pour détecter des anomalies et améliorer les performances de détection. De même, l'utilisation d'auto-encodeurs pour la réduction de dimensionnalité suivie par des classificateurs supervisés, comme démontré par (Tien et al. 2021), montre comment ces approches combinées peuvent améliorer la sécurité des réseaux IoT. La transparence et l'explicabilité des modèles doivent également être prises en compte pour gagner la confiance des utilisateurs et des administrateurs de systèmes. En définitive, une approche hybride qui combine les forces des algorithmes d'apprentissage supervisé et non supervisé offre une solution de sécurité plus robuste et efficace pour les réseaux IoT. Cette méthode permet non seulement de détecter et de classifier des attaques connues avec une grande précision mais aussi d'identifier de nouvelles menaces de manière proactive, offrant

ainsi une protection globale renforcée contre les cybermenaces.

2.7 Conclusion

En concluant ce chapitre sur l'application de l'intelligence artificielle en cybersécurité, particulièrement dans le contexte des systèmes de l'Internet des Objets (IoT), il est évident que l'IA représente un levier transformationnel pour la sécurité des réseaux interconnectés et des dispositifs omniprésents dans notre quotidien. L'IoT, avec ses caractéristiques uniques de connectivité étendue et de diversité des dispositifs, présente des défis de sécurité spécifiques qui sont efficacement adressés par les technologies d'intelligence artificielle.Les capacités d'apprentissage automatique et d'apprentissage profond, en particulier, jouent un rôle crucial dans la détection rapide et précise d'anomalies et dans la classification proactive des menaces potentielles. Ces technologies permettent non seulement d'analyser le vaste flux de données généré par les appareils IoT mais aussi d'identifier des comportements anormaux qui pourraient indiquer des intrusions ou des dysfonctionnements.

En outre, la protection des données sensibles traitées par ces dispositifs contre les manipulations malveillantes est une préoccupation majeure. Malgré les nombreux défis, l'avenir de l'IA en cybersécurité IoT est prometteur. L'innovation continue dans le domaine de l'apprentissage non supervisé et des réseaux neuronaux ouvre la voie à des solutions encore plus efficaces et adaptatives. Par ailleurs, l'intégration de l'IA avec d'autres technologies émergentes comme la blockchain pourrait renforcer la sécurité et la transparence des systèmes IoT.

La synergie entre l'intelligence humaine et artificielle reste essentielle pour naviguer dans le paysage complexe de la cybersécurité IoT. En combinant les intuitions humaines avec les capacités analytiques avancées de l'IA, il est possible de développer des systèmes de défense robustes qui non seulement préviennent les attaques mais aussi apprennent et s'adaptent à de nouvelles menaces. En somme, l'intégration de l'IA dans la cybersécurité des systèmes IoT marque le début d'une nouvelle ère où la sécurité devient non seulement proactive mais aussi intrinsèquement intelligente, capable de répondre dynamiquement aux évolutions de l'environnement de menace.

Chapitre 3

Classification du trafic réseau IoT à l'aide d'apprentissage automatique

3.1 Introduction . . . 73
3.2 Brève présentation des modèles implémentés . . . 74
3.2.1 Arbre de décision . . . 75
3.2.2 Forêt aléatoire . . . 75
3.2.3 AdaBoost . . . 76
3.2.4 XGBoost . . . 77
3.2.5 Perceptron multicouche . . . 77
3.2.6 LSTM . . . 78
3.3 Le jeu de données expérimental . . . 79
3.3.1 Description du jeu de données . . . 79
3.3.2 Prétraitement des données . . . 81
3.4 Classification binaire . . . 83
3.4.1 Définition et métriques . . . 83
3.4.2 Résultats du modèle d'arbre de de décision binaire . . . 85
3.4.3 Résultats du modèle de forêt aléatoire binaire . . . 86
3.4.4 Résultats du modèle d'AdaBoost binaire . . . 87
3.4.5 Résultats du modèle de XGBoost binaire . . . 88
3.4.6 Résultats du modèle MLP binaire . . . 89
3.4.7 Résultats du modèle LSTM binaire . . . 90
3.5 Classification multi-classe . . . 93
3.5.1 Définition et métriques . . . 93
3.5.2 Résultats du modèle d'arbre de décision multi-classe . . . 95
3.5.3 Résultats du modèle de forêt aléatoire multi-classe . . . 96
3.5.4 Résultats du modèle d'AdaBoost multi-classe . . . 97
3.5.5 Résultats du modèle de XGBoost multi-classe . . . 98
3.5.6 Résultats du modèle MLP multi-classe . . . 99
3.5.7 Résultats du modèle LSTM multi-classe . . . 100
3.6 Comparaison des techniques de classification binaire et multi-classe 103
3.7 Conclusion . . . 109

3.1 Introduction

La classification en apprentissage automatique est une tâche fondamentale qui consiste à attribuer des étiquettes à des exemples en fonction de leurs caractéristiques. Cette technique est largement utilisée dans divers domaines, allant de la reconnaissance d'images à la détection de spams, en passant par le diagnostic médical. L'objectif de la classification est de créer un modèle capable de prédire l'étiquette de nouvelles données, en se basant sur un ensemble d'exemples étiquetés(Neal 2006 ; Hastie, Tibshirani et Friedman 2009 ; Murphy 2012). La classification peut être binaire, où les données sont classées en deux catégories, ou multi-classe, où les données peuvent appartenir à plusieurs classes. Ces méthodes permettent de traiter des problèmes variés et de prendre des décisions automatiques basées sur les données (Czarnowski et Jędrzejowicz 2021).

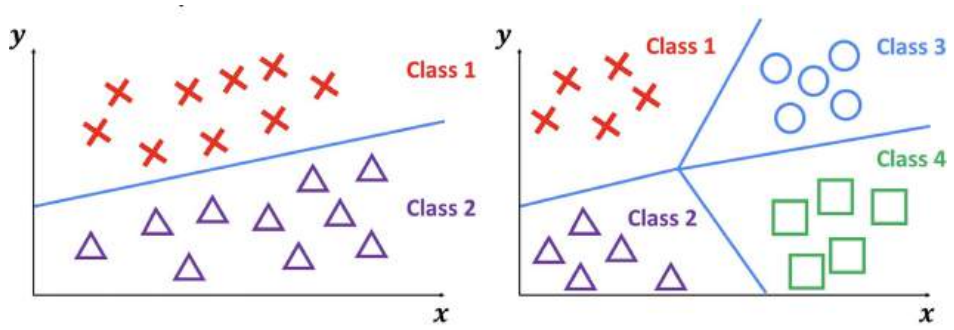


Figure 3.1 – Exemples de classification

La détection du spam dans le courrier électronique est un exemple courant de classification. Le courrier reçu est classé comme spam ou non spam. La détection d'intrusion peut être formalisée comme un problème de classification binaire. Dans le contexte du trafic réseau, par exemple, il peut s'agir de séparer le trafic normal du trafic malveillant (anomalie). Cependant, dans le cadre de la cybersécurité, il existe plusieurs types d'intrusions ou d'attaques. Parfois, il est nécessaire non seulement de détecter l'intrusion mais aussi de l'identifier. Dans ce cas, on a plusieurs classes puisque chaque attaque constitue alors une classe différente. C'est dans ce contexte qu'intervient la classification multi-classe. Elle permettra ainsi de classer, en d'autres termes, d'identifier les différents types d'attaques. Nous débutons ce chapitre par une brève présentation des algorithmes d'apprentissage automatique et d'apprentissage profond que nous avons rigoureusement implémentés. Cette première partie sert de fondation à notre analyse, offrant un aperçu des approches que nous avons exploitées pour la détection des cyber-attaques dans les réseaux IoT. Ensuite, nous plongeons dans la description du jeu de données utilisé, élément fondamental qui a façonné le cadre expérimental de notre étude. Cette section expose les caractéristiques essentielles de notre

ensemble de données. La classification binaire constitue la pierre angulaire de nos investigations, et c'est dans la troisième partie que nous présentons en détail les résultats obtenus par les six modèles binaires déployés. Chacun de ces algorithmes fait l'objet d'une analyse minutieuse, mettant en évidence leurs performances individuelles dans la distinction entre trafic normal et malicieux. La quatrième partie, axée sur la classification multi-classe, offre une extension de notre étude en considérant la capacité des modèles à catégoriser plusieurs types d'attaques. Les résultats spécifiques à chaque algorithme sont exposés, fournissant un panorama complet de leur adaptabilité face à une variété de scénarios. Enfin, la cinquième partie se consacre à la comparaison des techniques de classification binaire et multi-classe (Jha, Dave et Madan 2019). Nous analysons de près les résultats, dévoilant les nuances significatives dans les performances des modèles selon le contexte de classification. Cette comparaison approfondie nous guide vers une compréhension plus fine des forces et des limites de chaque approche dans le domaine complexe de la sécurité IoT. Les travaux présentés dans ce chapitre ont fait l'objet d'une publication [1].

3.2 Brève présentation des modèles implémentés

La réussite d'une étude en apprentissage automatique repose en grande partie sur la sélection judicieuse des algorithmes, chaque choix reflétant un compromis unique entre complexité, précision et adaptabilité au contexte particulier. Dans cette perspective, cette section inaugure notre exploration en fournissant une brève présentation des algorithmes d'apprentissage automatique et d'apprentissage profond que nous avons rigoureusement sélectionnés et mis en œuvre.

Les six modèles déployés dans notre étude représentent une variété d'approches computationnelles, chacune apportant sa propre perspective et méthodologie à la détection des cyber-attaques dans les réseaux IoT. De l'arbre de décision, réputé pour sa simplicité et son interprétabilité, au LSTM, réputé pour sa capacité à traiter des séquences complexes, chaque algorithme incarne une solution distincte à des problèmes spécifiques.

L'objectif de cette partie est d'offrir un aperçu concis de ces modèles, en mettant en évidence leurs principes fondamentaux, leurs caractéristiques uniques et leurs domaines d'application préférentiels. En comprenant les forces et les limitations de chaque algorithme, nous jetterons les bases nécessaires pour une évaluation approfondie de leurs performances ultérieures dans la classification des cyber-attaques.

1. **A. Diallo**, L. Affognon, C. Diallo and E. C. Ezin, *"Deep learning Based Binary and Multi-class Classification Comparison for Anomaly Detection,"* 2022 International Conference on Engineering and Emerging Technologies (ICEET), Kuala Lumpur, Malaysia, 2022, pp. 1-6, doi : 10.1109/ICEET56468.2022.10007171

3.2.1 Arbre de décision

Les arbres de décision sont des modèles prédictifs utilisés en apprentissage supervisé. Ils représentent à la fois des classificateurs et des modèles de régression (Yang 2019). Leur principe repose sur une structure hiérarchique sous forme d'arbre, avec un nœud racine et des nœuds internes (ou de test) qui partitionnent l'espace des instances en fonction d'attributs d'entrée. Les feuilles de l'arbre représentent les classes ou les valeurs cibles. Le processus de classification implique la navigation depuis le nœud racine jusqu'à une feuille, en suivant les résultats des tests le long du chemin. Ces modèles peuvent intégrer à la fois des attributs nominaux et numériques, offrant une interprétation géométrique dans le cas des attributs numériques. Les caractéristiques des arbres de décision comprennent une partition récursive de l'espace des instances, des nœuds internes qui effectuent des tests sur les attributs, et des feuilles qui représentent les classes ou les valeurs prédites. Le processus d'apprentissage se fait généralement de manière top-down et de façon récursive, en sélectionnant les meilleurs attributs pour diviser le jeu de données à chaque étape. Les critères d'arrêt, ou de terminaison, sont des conditions qui déterminent quand l'arbre ne doit plus se subdiviser. Ces critères incluent, entre autres, le fait que toutes les instances dans le nœud appartiennent à une seule valeur de y, l'atteinte de la profondeur maximale de l'arbre, et la vérification du nombre minimum d'instances dans un nœud (Maimon et Rokach 2014).

3.2.2 Forêt aléatoire

Les Forêts Aléatoires constituent un ensemble d'arbres de décision individuels, non élagués, créés par la randomisation des divisions à chaque nœud. L'idée sous-jacente est d'utiliser plusieurs arbres approximatifs pour améliorer la précision globale. Le processus de construction des arbres individuels suit un schéma particulier. Le paramètre N, représentant le nombre de variables d'entrée à chaque nœud, est crucial et doit être considérablement inférieur au nombre total d'attributs dans l'ensemble d'entraînement. L'algorithme utilise un échantillonnage aléatoire avec remplacement pour construire les arbres, répétant ce processus jusqu'à un nombre prédéterminé d'itérations (T). La décision à chaque nœud est basée sur un vote majoritaire.

Différentes approches introduisent l'élément aléatoire. Une méthode consiste à utiliser un sous-échantillon d'instances pour déterminer les points de division optimaux. Une autre utilise des histogrammes, rendant les caractéristiques discrètes et choisissant le point de division de manière aléatoire dans un intervalle autour de la meilleure limite(frontière) des classes. Une caractéristique clé des Forêts Aléatoires est leur capacité à gérer efficace-

ment un grand nombre d'attributs d'entrée. Leur rapidité d'exécution en fait une solution attrayante pour des applications pratiques où l'efficacité est cruciale. Bien que leur définition initiale soit associée aux arbres de décision, les Forêts Aléatoires peuvent être étendues à divers types de classifieurs, les rendant polyvalentes et applicables à un large éventail de problèmes d'apprentissage automatique (Maimon et Rokach 2014).

3.2.3 AdaBoost

Adaboost est une méthode d'apprentissage ensembliste dépendante qui tire parti de la sélection d'instances guidée par le modèle. Son introduction en 1996 par Freund et Schapire en fait une approche significative pour améliorer les performances des modèles faibles, tels que les règles de classification ou les arbres de décision. L'idée fondamentale d'Adaboost est d'entraîner de manière itérative des modèles faibles sur différentes distributions des données d'entraînement, puis de les combiner pour former un modèle fort. Le processus implique l'attribution de poids à chaque exemple dans l'ensemble d'entraînement. Initialement, tous les poids sont égaux, mais à chaque itération, les poids des instances mal classées augmentent, tandis que ceux des instances correctement classées diminuent. Cela force le modèle faible à se concentrer sur les instances difficiles, créant ainsi une série de modèles qui se complètent mutuellement. L'algorithme Adaboost commence par initialiser les poids des instances, puis procède à l'entraînement itératif de modèles faibles sur des ensembles d'entraînement pondérés. Ces modèles faibles sont ensuite combinés en un modèle fort à l'aide d'un vote pondéré. Les poids sont ajustés pour l'itération suivante, mettant l'accent sur les instances mal classées. Ce processus est répété pour un nombre d'itérations prédéfini.

Adaboost présente des caractéristiques notables. Il peut gérer la classification binaire, bien que des extensions soient disponibles pour les problèmes multiclasse. Cependant, il nécessite des modèles faibles capables de traiter des instances pondérées. L'algorithme adresse le problème du surajustement en combinant des hypothèses avec de petites erreurs, et il génère un modèle final avec une variance réduite par rapport aux modèles faibles individuels. Il est connu pour approximer des classificateurs à marge large tels que les machines à vecteurs de support (SVM). Le mode de fonctionnement d'Adaboost implique une classification binaire basée sur un vote pondéré. Il cherche à améliorer itérativement les performances en ajustant les poids des instances. L'algorithme fonctionne en combinant plusieurs modèles faibles pour créer un modèle ensemble robuste. Cependant, Adaboost n'est pas sans inconvénients. Un nombre excessif d'itérations peut conduire au surajustement, et l'ensemble résultant peut être difficile à interpréter en raison de sa complexité. Malgré ces limitations,

Adaboost demeure un outil puissant pour construire des classificateurs précis, exploitant les forces des modèles faibles tout en s'adaptant à la complexité des données d'entraînement.

3.2.4 XGBoost

XGBoost, abréviation de Extreme Gradient Boosting, est un modèle d'ensemble d'arbres de décision. Il combine les résultats de K arbres pour donner la classe prédite. Cette méthode d'ensemble est basée sur le concept de boosting, qui est un procédé pour créer des ensembles de nombreux modèles d'apprentissage automatique, similaires au concept de bagging utilisé dans les forêts aléatoires. Cependant, la différence clé entre le bagging et le boosting réside dans le fait que, dans le boosting, chaque nouvel estimateur ajouté à l'ensemble dépend de tous les estimateurs ajoutés avant lui. Il se distingue par plusieurs caractéristiques clés. Tout d'abord, son implémentation est parallèle, accélérant le temps d'entraînement par rapport au gradient boosting traditionnel. De plus, XGBoost offre une flexibilité en permettant l'utilisation d'algorithmes faibles autres que des arbres de régression, tels que des modèles linéaires.

Dans le processus d'apprentissage, XGBoost utilise à la fois la première et la seconde dérivée de la fonction de perte, ce qui contribue à une estimation plus précise. Le modèle intègre également une régularisation, avec un terme lié au nombre de nœuds feuilles pour réduire la variance et prévenir le surajustement. XGBoost optimise sa fonction objective de manière similaire à la descente de gradient, utilisant des informations sur le gradient de la fonction de perte pour former de nouveaux arbres. Pour éviter le surajustement, il introduit un seuil lors de la division des arbres. Dans l'ensemble, XGBoost est une méthode d'ensemble puissante largement utilisée en raison de ses performances supérieures, que ce soit dans des compétitions d'apprentissage automatique ou dans divers domaines d'application (Zhou, Song et Zhou 2021).

3.2.5 Perceptron multicouche

Le Perceptron Multicouche ou Multi-Layer Perceptron (MLP), est un type spécifique d'architecture dans les Réseaux de Neurones Artificiels (RNA). Il tire son inspiration du fonctionnement des neurones dans le cerveau humain. Le cerveau humain fonctionne de manière parallèle pour la reconnaissance, le raisonnement et la récupération après des dommages. Les RNA, en raison de leur capacité à traiter de tels processus, trouvent des applications allant des tâches simples aux algorithmes complexes de reconnaissance de motifs.

Dans la structure de base d'un RNA, on a trois couches : l'entrée (input), la sortie (out-

put) et une couche cachée (hidden layer). Chaque couche contient des neurones, et chaque neurone a des poids associés pour le traitement ultérieur. Les neurones sont interconnectés à d'autres neurones dans le réseau par le biais de liens appelés synapses. Chaque neurone peut recevoir de multiples entrées soit d'autres neurones, soit directement sous forme de données d'origine. Chaque neurone a également une valeur de seuil. L'activation du neurone, ou le potentiel post-synaptique (PSP), est calculée comme la somme pondérée des entrées soustraite du seuil du neurone. Le réseau génère sa sortie en faisant passer le signal d'activation à travers une fonction d'activation ou de transfert.

Un réseau Feed-Forward, comme le MLP, a des neurones disposés en une topologie de couche distincte. La couche d'entrée introduit les valeurs des variables d'entrée. Les couches cachées et de sortie ont des neurones connectés aux neurones de leurs couches précédentes. La connexion dans ce type de topologie peut être entièrement connectée ou partiellement connectée. Dans le MLP, chaque unité effectue une somme pondérée biaisée des entrées et fait passer ce niveau d'activation à travers une fonction de transfert pour générer une sortie. Les fonctions d'activation les plus courantes dans le MLP sont les fonctions sigmoïdes et tangentes hyperboliques (Singh et Sachan 2014).

3.2.6 LSTM

Le Long Short-Term Memory (LSTM) est une architecture de réseau de neurones récurrents (RNN) spécifiquement conçue pour surmonter les problèmes tels que les gradients qui disparaissent, les gradients qui explosent et les difficultés liées aux dépendances à long terme dans les séquences. Lorsqu'on parle généralement d'une unité RNN aujourd'hui, cela fait souvent référence à une unité LSTM en raison de sa popularité. L'idée fondamentale derrière le LSTM est de doter une cellule RNN d'une mémoire persistante à long terme sous la forme de son état. Cet état peut conserver ses valeurs d'une entrée à l'autre indéfiniment. En plus de la mémoire à long terme, la cellule dispose également d'une mémoire à court terme sous la forme des sorties des neurones, qui existent uniquement pendant le traitement d'une nouvelle entrée et sont remplacées par de nouvelles valeurs à l'arrivée d'une nouvelle entrée.

Le LSTM vise à prendre des valeurs à court terme fugaces et à leur donner une durée de vie plus longue, leur permettant de contribuer à des calculs futurs. Ainsi, la mémoire à court terme est prolongée, d'où le nom de "long short-term memory" (mémoire à court terme longue). Cette conception peut également être interprétée comme une "persistent short-term memory" (mémoire à court terme persistante). Le fonctionnement interne détaillé de l'architecture LSTM implique des mécanismes tels que des portes (gates) qui contrôlent le

flux d'informations. Ces portes, utilisées à des fins d'oubli, de mémorisation et de sélection, sont également appelées portes d'entrée et de sortie. Les opérations des portes sont régies par des mécanismes de multiplication et d'addition contrôlés par des valeurs de porte, restreintes à la plage de 0 à 1.

En résumé, le LSTM utilise des portes pour gérer sa mémoire interne, permettant un contrôle fin sur ce qui doit être mémorisé, oublié ou sélectionné à chaque étape du traitement des séquences. Cette architecture évite les problèmes de gradients qui disparaissent ou explosent en interne et permet de gérer efficacement les dépendances à long terme dans les séquences. Les poids des neurones internes du LSTM sont ajustés pendant l'entraînement grâce à la rétropropagation du gradient, permettant au LSTM d'apprendre à mémoriser et oublier les données de manière appropriée.

3.3 Le jeu de données expérimental

Le jeu de données utilisé dans ces travaux porte le nom de IoT Botnet dataset (Ullah 2020). Il est l'un des plus récents, si ce n'est le plus récent dans ce domaine. Élaboré à l'Université d'Ontario (Ontario Tech University) au Canada, il se distingue principalement en tant que l'un des rares jeux de données basés sur le trafic réseau IoT. Il présente des caractéristiques réseau plus générales et basées sur les flux. Les auteurs se sont inspirés d'un autre jeu de données, portant un nom assez similaire. Il s'agit de BoT-IoT créé en 2018 et publié en 2019 par l'UNSW Canberra (l'University of New South Wales), Australie (Koroniotis et al. 2018). Les auteurs, ont envisagé plusiseurs scénarios de botnet pour capturer le trafic réseau malicieux et le trafic normal mais BoT-IoT comporte un nombre limité de caractéristiques générales basées sur le trafic.

3.3.1 Description du jeu de données

Le jeu de données IoT Botnet contient des caractéristiques réseau plus générales et davantage de caractéristiques réseau basées sur les trafic.Le jeu de données IoT Bonet a été créé pour pallier les limites des ensembles de données existants, qui manquent souvent de données récentes sur les attaques et ne sont pas spécifiquement adaptés aux réseaux IoT. Le projet se concentre sur la détection de botnets dans les environnements IoT et nécessite un ensemble de données fournissant des informations suffisantes sur les traces IoT. Le banc d'essai du jeu de données comprend des services IoT simulés, des plates-formes réseau et l'application dénommée ISCXFlowmeter (Habibi Lashkari et al. 2017). ISCXFlowMeter

est un générateur et analyseur de flux de trafic Ethernet pour la détection d'anomalies. Les plates-formes réseau comprennent des machines virtuelles normales et attaquantes, connectées à LAN et WAN, et reliées à Internet via la machine PFSense. Un pare-feu de filtrage de paquets est utilisé pour garantir la validité de l'étiquetage du jeu de données.Le réseau de machines virtuelles est constitué d'une machine Ubuntu Tap, d'une machine Metasploitable, d'une machine Ubuntu Mobile, d'une machine Windows 7 et de quatre machines Kali Linux. Les services IoT sont implémentés sur le serveur Ubuntu pour simuler un véritable réseau IoT, et les machines Kali Linux ont été utilisées comme systèmes d'attaque.

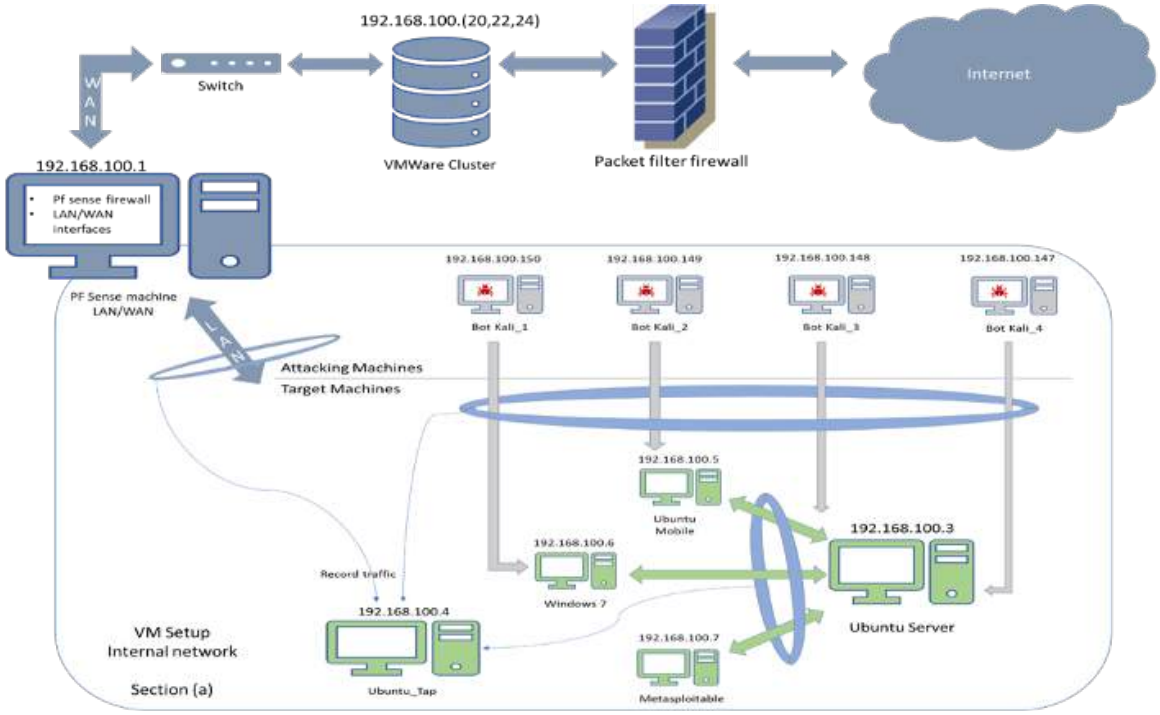


Figure 3.2 – Banc d'essai de Botnet IoT dataset. Source image, Koroniotis et al. 2018

La Figure 3.2 montre le banc d'essai dans lequel BoT-IoT a été créé. Les auteurs de IoT Botnet se sont inspirés de ce dernier pour mettre en place un banc d'essai similaire avec les mêmes équipements et la même architecture. Le trafic réseau normal est généré via l'outil Ostinato, tandis que les services IoT simulés, tels que lumières activées par le mouvement, le réfrigérateur intelligent, le thermostat intelligent, la porte de garage activée à distance et la station météo, sont implémentés sur le serveur Ubuntu. Les services IoT sont connectés au courtier MQTT dans le serveur Ubuntu ainsi qu'au hub IoT AWS, créant un environnement domestique intelligent typique avec cinq périphériques IoT connectés localement et au cloud via l'outil Node-red [2]. Les machines Kali génèrent le trafic malicieux. Elles effectuent diverses attaques, notamment le balayage de ports et les attaques DDoS, sur d'autres

2. https://nodered.org/,https://noderedguide.com/

VM. La Figure 3.3 présente la taxonomie des attaques du jeu de données proposé.

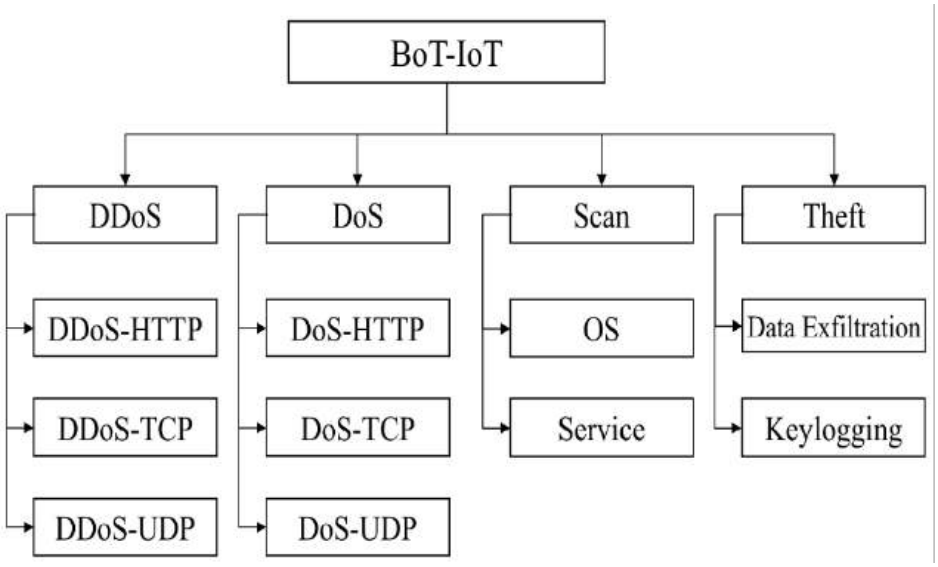


Figure 3.3 – Taxonomie des attaques dans Botnet IoT dataset

La conversion des fichiers Pcap en fichiers CSV est effectuée à l'aide de l'application ISCXFlowmeter, extrayant 80 caractéristiques de statistiques réseau. Le jeu de données IoT Botnet propose 83 caractéristiques réseau et trois caractéristiques d'étiquetage (Label, catégorie et sous-catégorie), soit un total de 86 caractéristiques. Ce jeu de données fournit une base pour le développement d'un modèle de détection d'intrusion basé sur les flux pour les réseaux IoT. Pour aller plus loin, un prétraitement du jeu de données doit être effectué pour rendre les données compatibles avec les algorithmes d'apprentissage automatique, utilisant des techniques de normalisation des colonnes et des algorithmes d'apprentissage supervisés.

3.3.2 Prétraitement des données

L'ensemble de données IoT Botnet a été divisé en deux parties distinctes pour l'entraînement et le test des modèles de détection d'intrusion. Cette division suit un ratio de 70% pour l'ensemble d'entraînement et 30% pour l'ensemble de test. Les 70% des données ont été consacrés à l'entraînement des modèles, permettant ainsi aux algorithmes d'apprentissage automatique d'ajuster leurs paramètres en se basant sur cette portion substantielle du jeu de données. Le choix d'allouer 30% des données à l'ensemble de test vise à évaluer la performance et la généralisation des modèles sur des données non vues pendant l'entraînement. Cela permet de simuler des conditions réelles où le modèle doit faire face à de nouvelles instances et démontrer sa capacité à généraliser au-delà des données d'entraînement. Cette approche de division des données en ensembles d'entraînement et de test est couramment

utilisée en apprentissage automatique pour évaluer la robustesse des modèles. Elle contribue à identifier la capacité d'un modèle à généraliser et à prendre des décisions précises sur des données inconnues.

Après le partitionnement de notre jeu de données IoT Botnet, un prétraitement a été effectué pour garantir la pertinence des caractéristiques et traiter les valeurs manquantes ou infinies. Certaines caractéristiques jugées non pertinentes pour le contexte de généralisation ont été supprimées du jeu de données. Les caractéristiques éliminées sont : *Flow_ID, Src_IP, Dst_IP, Timestamp, Label, Cat, et Sub_Cat*, contribuant ainsi à simplifier le jeu de données et à éliminer des informations redondantes. En outre, les valeurs infinies et les valeurs manquantes ont été gérées pour assurer la stabilité des calculs ultérieurs. Les valeurs infinies et les valeurs manquantes ont été remplacées par zéro.

La normalisation ou mise à l'échelle des caractéristiques est une étape cruciale dans le processus de modélisation des algorithmes avec des ensembles de données. En effet, les données recueillies peuvent comporter des caractéristiques de dimensions et d'échelles diverses. La présence de caractéristiques à différentes échelles peut avoir un impact négatif sur la modélisation d'un ensemble de données, entraînant des résultats biaisés en termes d'erreurs de classification et de taux de précision. Afin de remédier à ces problèmes potentiels, il devient impératif de procéder à la mise à l'échelle des données avant d'appliquer les modèles. L'objectif principal de la mise à l'échelle est d'assurer que toutes les caractéristiques du jeu de données sont exprimées de manière homogène, éliminant ainsi les disparités dues aux différentes unités de mesure ou échelles. Cette homogénéisation facilite le processus d'apprentissage automatique, améliore la convergence des algorithmes et contribue à obtenir des résultats plus fiables lors de l'évaluation des performances des modèles. Ainsi, la mise à l'échelle des caractéristiques joue un rôle crucial dans l'optimisation de la modélisation, contribuant à garantir la robustesse et l'efficacité des algorithmes appliqués sur des données hétérogènes.

Pour cette tâche, nous avons choisi la méthode de normalisation standard (Standard Scaler), qui permet de centrer les données en supprimant la moyenne et de les mettre à l'échelle pour avoir une variance unitaire. Chaque caractéristique est transformée indépendamment, garantissant que les statistiques pertinentes sont calculées sur l'ensemble d'apprentissage. Le score standard d'un échantillon est obtenu en soustrayant la moyenne et en divisant par l'écart type comme indiqué sur l'équation suivante.

$$X_{\text{normalisé}} = \frac{X - m}{\sigma} \tag{3.1}$$

Cette approche est essentielle car de nombreux algorithmes d'apprentissage automatique supposent que les caractéristiques sont centrées autour de zéro et ont une variance comparable. Cependant, il est important de noter que la méthode Standard Scaler est sensible aux valeurs aberrantes, et la présence de telles valeurs peut influencer négativement la standardisation. En présence de valeurs aberrantes, les caractéristiques peuvent évoluer différemment les unes des autres, impactant la performance des modèles. C'est pourquoi nous avons éliminé les valeurs aberrantes avant de procéder à la normalisation.

3.4 Classification binaire

Après avoir dressé le cadre avec la description du jeu de données expérimental et la revue théorique des modèles développés, cette section vise à présenter les résultats obtenus dans le cadre de la classification binaire. L'objectif principal est d'évaluer les performances des différents modèles dans la détection des cyber-attaques au sein du réseau IoT.

3.4.1 Définition et métriques

Définition

La classification binaire consiste à attribuer une observation à l'une des deux classes possibles. Mathématiquement, soit $\mathcal{X}$ l'espace des caractéristiques et $\mathcal{Y} = \{0, 1\}$ l'espace des étiquettes. Une fonction de classification binaire cherche à apprendre un mapping $f : \mathcal{X} \to \mathcal{Y}$. Si nous avons un ensemble de données d'entraînement $\{(x_i, y_i)\}_{i=1}^{n}$ où $x_i \in \mathcal{X}$ et $y_i \in \mathcal{Y}$, alors $f(x)$ est une règle qui prédit l'étiquette y d'une nouvelle observation x. Un exemple typique de classification binaire est l'analyse médicale. Les symptômes du patient sont recueillis et le classificateur binaire prédit s'il est en bonne santé (0) ou potentiellement malade (1).

Métriques

Les performances d'un classificateur binaire sont mesurées avec les métriques suivantes : la matrice de confusion, la précision, le rappel, le F-score, et la justesse. Ces métriques sont calculées à partir des vrais positifs(TP), vrais négatifs(TN), faux positifs(FP)et les faux négatifs(FN).

Vrais Positifs (True Positives, TP)

- **Définition** : Les vrais positifs sont les instances correctement prédites comme positives par le modèle. Cela signifie que l'instance est réellement positive et que le modèle l'a prédite correctement comme telle.
- **Formule** :

$$\text{TP} = \sum \text{instances prédites positives et réellement positives}$$

Vrais Négatifs (True Negatives, TN)

- **Définition** : Les vrais négatifs sont les instances correctement prédites comme négatives par le modèle. Cela signifie que l'instance est réellement négative et que le modèle l'a prédite correctement comme telle.
- **Formule** :

$$\text{TN} = \sum \text{instances prédites négatives et réellement négatives}$$

Faux Positifs (False Positives, FP)

- **Définition** : Les faux positifs sont les instances incorrectement prédites comme positives par le modèle. Cela signifie que l'instance est réellement négative, mais le modèle l'a prédite comme positive.
- **Formule** :

$$\text{FP} = \sum \text{instances prédites positives mais réellement négatives}$$

Faux Négatifs (False Negatives, FN)

- **Définition** : Les faux négatifs sont les instances incorrectement prédites comme négatives par le modèle. Cela signifie que l'instance est réellement positive, mais le modèle l'a prédite comme négative.
- **Formule** :

$$\text{FN} = \sum \text{instances prédites négatives mais réellement positives}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

$$Rappel = \frac{TP}{TP + FN} \tag{3.3}$$

$$F1 - Score = 2 * \frac{Prcision * Rappel}{Prcision + Rappel} \tag{3.4}$$

$$Justesse = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.5}$$

L'analyse des résultats sera accompagnée d'une discussion approfondie, soulignant les forces et les faiblesses de chaque modèle dans le contexte de la détection des cyber-attaques dans les réseaux IoT. Cette évaluation critique guidera la sélection du modèle le plus approprié pour les besoins spécifiques de la détection d'intrusion dans l'environnement expérimental considéré.

3.4.2 Résultats du modèle d'arbre de de décision binaire

Le modèle arbre de décision a été défini en utilisant la bibliothèque scikit-learn en Python. Le processus d'entraînement a commencé en initialisant le modèle avec les hyperparamètres par défaut. Ces hyperparamètres déterminent la structure et le comportement de l'arbre de décision.Le modèle est entraîné sur l'ensemble d'entraînement, x_train et y_train, où x_train représente les caractéristiques et y_train les étiquettes correspondantes. Le temps d'entraînement, mesuré en secondes, a été enregistré pour évaluer l'efficacité du processus d'apprentissage. Dans notre cas, le temps d'entraînement était de 305 secondes, démontrant la rapidité de convergence de l'algorithme d'arbre de décision sur notre ensemble de données. Une fois le modèle entraîné, nous avons procédé à sa phase de test en utilisant l'ensemble de test, x_test, et en réalisant des prédictions avec le modèle. Le temps de test a également été mesuré pour évaluer la rapidité des prédictions. Dans notre cas, le temps de test était de 2.13 secondes. Les résultats de l'évaluation du modèles sont indiqués dans la figure suivante.

La Figure 3.4 la matrice de confusion et le rapport de classification. Sur les matrice de confusion, on peut voir que les faux positifs et les faux négatifs sont négligeables par rapport au vrais positifs et aux vrais négatifs. Cela indique une bonne capacité de classification du modèle. Cette bonne performance du modèle est traduite encore par le rapport de classification, à travers les métriques telles que la précision, le rappel, le f1-score et la justesse (accuracy). Toutes ces métriques affichent des taux qui avoisinent les 100% et qui

A

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	11713
Anomaly	1.00	1.00	1.00	1852161
accuracy			1.00	1863874
macro avg	1.00	1.00	1.00	1863874
weighted avg	1.00	1.00	1.00	1863874

B

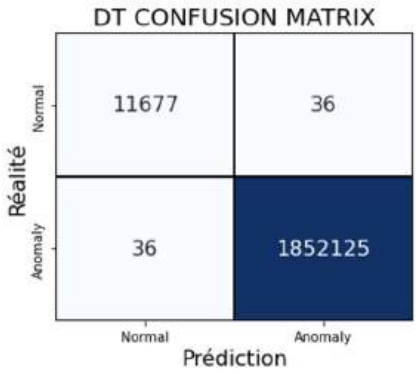


Figure 3.4 – Performances du modèle d'arbre de décision binaire avec : A) le rapport de classification et B) la matrice de confusion

ont d'ailleurs été arrondis à 100%. Ces mesures parfaites indiquent une classification précise des anomalies et des instances normales. On peut dire que le modèle d'arbre de décision binaire a montré une grande capacité à discerner les attaques du trafic normal, comme le suggèrent le rapport de classification.

3.4.3 Résultats du modèle de forêt aléatoire binaire

Avec la bibliothèque scikit-learn, nous avons mis en place un modèle de Forêt Aléatoire composé de 100 arbres de décision chacun utilisant l'indice de Gini comme critère de division. Aucune limite de profondeur spécifique n'a été définie pour les arbres, et des paramètres tels que le nombre minimal d'échantillons pour la division et le nombre minimal d'échantillons par feuille ont été configurés à 2 et 1 respectivement. L'entrainement et le test du modèle ont duré respectivement 2241 secondes et 23 secondes. Lors de l'évaluation sur l'ensemble de test, les performances du modèle se sont révélées remarquables. La matrice de confusion révèle un nombre élevé de vrais positifs, indiquant que le modèle identifie efficacement les anomalies, avec un nombre minime de faux positifs, soulignant la capacité du modèle à éviter les erreurs de classification des instances normales en tant qu'anomalies. Le rapport de classification présente des mesures de performance idéales, avec une précision, un rappel et un F1-score de 100% pour les deux classes (Anomalie et Normal). La classe des anomalies affiche un rappel élevé de 99%, mettant en évidence la capacité du modèle à capturer la grande majorité des anomalies dans l'ensemble de test.

La justesse globale du modèle atteint 100%, suggérant que toutes les prédictions sont correctes. Cependant, dans des problèmes de déséquilibre de classe, il est crucial d'utiliser des métriques spécifiques pour évaluer les performances. En définitif, ce modèle de Forêt Aléatoire se révèle robuste, généralise bien aux données inconnues, ce qui témoigne de sa

A

	precision	recall	f1-score	support
Normal	1.00	0.99	1.00	11713
Anomaly	1.00	1.00	1.00	1852161
accuracy			1.00	1863874
macro avg	1.00	1.00	1.00	1863874
weighted avg	1.00	1.00	1.00	1863874

B

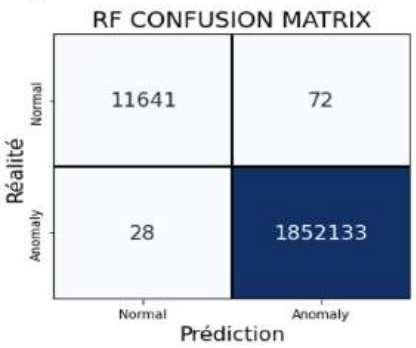


Figure 3.5 – Performances du modèle de forêt aléatoire binaire avec : A) le rapport de classification et B) la matrice de confusion

capacité dans la détection d'anomalies.

3.4.4 Résultats du modèle d'AdaBoost binaire

Nous avons développé un modèle de classification AdaBoost en utilisant la classe AdaBoostClassifier. Le modèle a été entraîné sur les données d'entraînement, avec un temps d'entraînement de 1174 secondes. L'implémentation d'AdaBoost a utilisé les paramètres par défaut, où l'algorithme de boosting utilisé était 'SAMME.R', le taux d'apprentissage était de 1.0, le nombre d'estimateurs (n_estimators) était fixé à 50. Le modèle a ensuite été testé, montrant un temps de test de 40 secondes.

Le modèle AdaBoost a fourni des résultats solides lors de l'évaluation. La matrice de confusion révèle un nombre élevé de vrais positifs (TP), indiquant que le modèle a correctement identifié un grand nombre d'anomalies, bien que quelques faux positifs (FP) aient été enregistrés. Cela signifie que le modèle a tendance à classer certaines instances normales comme des anomalies. Cependant, le nombre de faux négatifs (FN) est relativement faible, montrant une bonne capacité du modèle à éviter de manquer des anomalies.

Le rapport de classification confirme ces observations, avec une précision de 95% et un rappel de 93% pour la classe 0 (Normal), et une précision, un rappel et un F1-score parfaits de 100% pour la classe 1 (Anomaly). L'exactitude globale atteint également 100%, soulignant la performance globale du modèle.

En résumé, bien que le modèle AdaBoost ait montré des résultats solides, il présente une légère tendance à classer certaines instances normales comme des anomalies. Cependant, cela est compensé par une faible quantité de faux négatifs. La justesse élevée et les performances parfaites pour la classe des anomalies suggèrent que le modèle est globalement

A

	precision	recall	f1-score	support
0	0.95	0.93	0.94	11713
1	1.00	1.00	1.00	1852161
accuracy			1.00	1863874
macro avg	0.97	0.96	0.97	1863874
weighted avg	1.00	1.00	1.00	1863874

B

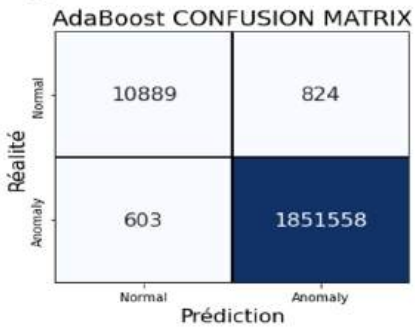


Figure 3.6 – Performances du modèle AdaBoost binaire avec : A) le rapport de classification et B) la matrice de confusion

robuste. Pour des améliorations potentielles, il peut être utile d'explorer l'ajustement des hyperparamètres ou l'utilisation d'un ensemble d'apprentissage plus équilibré en termes de classes.

3.4.5 Résultats du modèle de XGBoost binaire

Nous avons construit un modèle XGBoost en utilisant la bibliothèque xgboost de Python pour résoudre notre problème de classification. Le modèle a été entraîné sur les données d'entraînement avec un temps d'entraînement de 366 secondes et testé en 2.25 secondes. Les principaux paramètres utilisés pour construire le modèle sont les suivants : n_estimators (nombre d'estimateurs) à 100, learning_rate (taux d'apprentissage) à 0.3, max_depth (profondeur maximale de l'arbre) à 6, et l'objective défini sur 'binary :logistic' indiquant que le modèle effectue une classification binaire. Le modèle XGBoost a produit des résultats remarquables lors de l'évaluation sur l'ensemble de test. La matrice de confusion montre un nombre élevé de vrais positifs (TP) et de vrais négatifs (TN), avec un très faible nombre de faux positifs (FP) et de faux négatifs (FN). Cela suggère que le modèle a bien capturé les instances positives et négatives de manière précise.

Le rapport de classification confirme ces observations avec des scores de précision, de rappel et de F1 égaux à 100% pour les deux classes (0 et 1). L'exactitude globale du modèle atteint également 100%, indiquant que toutes les prédictions sont correctes. Ces résultats sont caractéristiques de la qualité du modèle, avec une bonne capacité à généraliser efficacement aux données non vues. De façon générale, le modèle XGBoost a démontré une puissance de prédiction impressionnante sur l'ensemble de test, avec une classification parfaite pour les deux classes. Les paramètres choisis semblent être bien adaptés à notre problème de classification binaire.

A

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11713
1	1.00	1.00	1.00	1852161
accuracy			1.00	1863874
macro avg	1.00	1.00	1.00	1863874
weighted avg	1.00	1.00	1.00	1863874

B

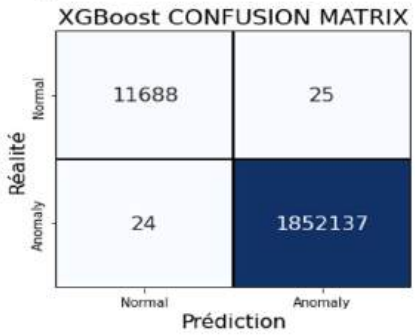


Figure 3.7 – Performances du modèle XGBoost binaire avec : A) le rapport de classification et B) la matrice de confusion

3.4.6 Résultats du modèle MLP binaire

Nous avons construit un modèle MLP (Perceptron Multicouche) en utilisant la classe MLPClassifier de la bibliothèque scikit-learn pour résoudre notre problème de classification. Le modèle est configuré avec une architecture de réseau de neurones composée de trois couches cachées de 50, 100 et 50 neurones respectivement. Le modèle MLP (Perceptron Multicouche) a été configuré avec divers paramètres clés lors de sa construction. Le nombre maximum d'itérations pendant l'entraînement (max_iter) a été fixé à 1000, déterminant ainsi la limite d'itérations pour la convergence du modèle. La fonction d'activation utilisée dans les couches cachées a été définie comme 'relu' (activation='relu'), et l'optimiseur Adam (solver='adam') a été choisi pour minimiser la fonction de perte. Le taux d'apprentissage (learning_rate)) a été fixé à 0.001. Ces paramètres jouent un rôle crucial dans la configuration du réseau de neurones, influençant sa capacité à apprendre à partir des données d'entraînement. Le temps d'entraînement du modèle a duré 1612 secondes.

Le modèle a été évalué sur l'ensemble de test, générant des prédictions en 30 secondes. Les résultats d'évaluation montrent un nombre élevé de vrais positifs (TP) et de vrais négatifs (TN), avec un faible nombre de faux positifs (FP) et de faux négatifs (FN). Le rapport de classification confirme ces résultats avec des scores de précision, de rappel et de F1 proches de 100% pour les deux classes (Normal et Anomaly).

Le modèle MLP a réalisé un très bon score sur l'ensemble de test, avec une classification presque parfaite pour les deux classes. Les paramètres du modèle semblent bien adaptés à la tâche de classification, et l'architecture multicouche a réussi à capturer efficacement les motifs complexes dans les données.

La justesse globale, la précision, le rappel et le F1-score proches de 100% indiquent une capacité élevée du modèle à généraliser et à discriminer entre les classes. De façon globale,

A

	precision	recall	f1-score	support
Normal	0.99	0.99	0.99	11713
Anomaly	1.00	1.00	1.00	1852161
accuracy			1.00	1863874
macro avg	1.00	0.99	1.00	1863874
weighted avg	1.00	1.00	1.00	1863874

B

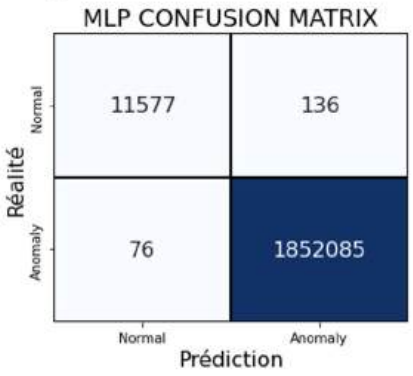


Figure 3.8 – Résultats du modèle mlp binaire avec : A) le rapport de classification et B) la matrice de confusion

le modèle MLP a réussi à fournir de bons résultats sur cette tâche de classification, mettant en évidence la puissance des réseaux de neurones multicouches pour capturer des motifs complexes dans les données.

3.4.7 Résultats du modèle LSTM binaire

Le modèle LSTM a été construit en utilisant la bibliothèque Keras avec une architecture spécifique. Le réseau est composé de deux couches LSTM successives, avec 100 et 150 unités respectivement, suivies de deux couches denses de 100 et 79 unités, activées par la fonction tangente hyperbolique ('tanh'). La dernière couche dense a une seule unité activée par la fonction sigmoïde pour la classification binaire. Le modèle a été compilé avec une fonction de perte binaire ('binary_crossentropy') et l'optimiseur Adam. L'entraînement a été effectué sur un ensemble de données avec 50 époques et une taille de lot de 1000, avec un temps d'entraînement de 1160 secondes. Lors de l'évaluation sur l'ensemble de test, le modèle LSTM a présenté de très bons résultats. La matrice de confusion montre un nombre élevé de vrais positifs (TP) et de vrais négatifs (TN), avec un nombre négligeable de faux positifs (FP) et de faux négatifs (FN). Cela suggère une capacité élevée du modèle à discriminer entre les classes positives et négatives.

Le rapport de classification confirme ces résultats avec des scores de précision, de rappel et de F1 proches de 100% pour les deux classes (0 et 1). L'exactitude globale du modèle atteint également 100%, indiquant une classification parfaite sur l'ensemble de test.

En conclusion, le modèle LSTM a réussi à capturer efficacement les motifs séquentiels dans les données, réalisant des performances compétitives en termes de précision et de rappel. Ces résultats soulignent la puissance des réseaux LSTM dans la modélisation de

A

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3904
1	1.00	1.00	1.00	617386
accuracy			1.00	621290
macro avg	1.00	1.00	1.00	621290
weighted avg	1.00	1.00	1.00	621290

B

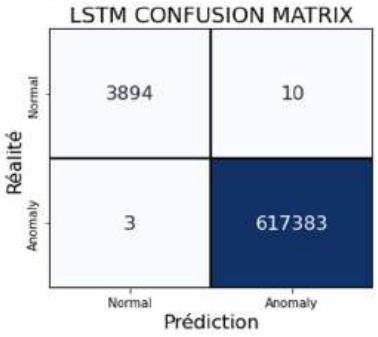


Figure 3.9 – Performances du modèle lstm binaire avec : A) le rapport de classification et B) la matrice de confusion

séquences complexes, faisant de ce modèle une option robuste pour la tâche de classification binaire.

La Figure 3.10 présente une comparaison des performances des six modèles de classification binaire pour la détection des cyber-attaques dans les réseaux IoT, en se basant sur le F1-Score, le temps d'entrainement et le temps de test. Une analyse approfondie et comparative des résultats, nous permet d'avoir une vision globale sur les performances des différents modèles. Les modèles basés sur des techniques d'arbre de décision, tels que l'Arbre de Décision et la Forêt Aléatoire, se distinguent aussi avec une précision, un rappel et un F-score trés élevés. Ces performances suggèrent une capacité efficace à apprendre les motifs sous-jacents dans le jeu de données. Cependant, l'examen plus approfondi des résultats montre des variations subtiles entre les modèles. Bien qu'AdaBoost présente des performances solides, avec une bonne précision pour la classe Anomaly, une légère diminution dans la classification de la classe Normale peut indiquer une sensibilité accrue aux instances normales, conduisant à quelques fausses prédictions. XGBoost, en tant qu'algorithme de boosting, se distingue en maintenant des performances parfaites pour les deux classes. Cette robustesse souligne la capacité de XGBoost à généraliser sur des données nouvelles et à maintenir une haute précision. Les modèles basés sur les réseaux de neurones, tels que MLP et LSTM, affichent également des performances élevées. MLP maintient une précision élevée pour la classe Anomaly, bien qu'avec une légère baisse pour la classe Normal. En revanche, LSTM se démarque en atteignant des performances parfaites, soulignant son aptitude à capturer des motifs temporels complexes.

En conclusion, la visualisation comparative met en lumière les forces et les faiblesses spécifiques de chaque modèle. Les modèles basés sur des arbres se démarquent par leur rapidité et leur interprétabilité, tandis que les réseaux de neurones, en particulier LSTM, se distingue dans la capture de motifs complexes malgré des temps d'entraînement et de

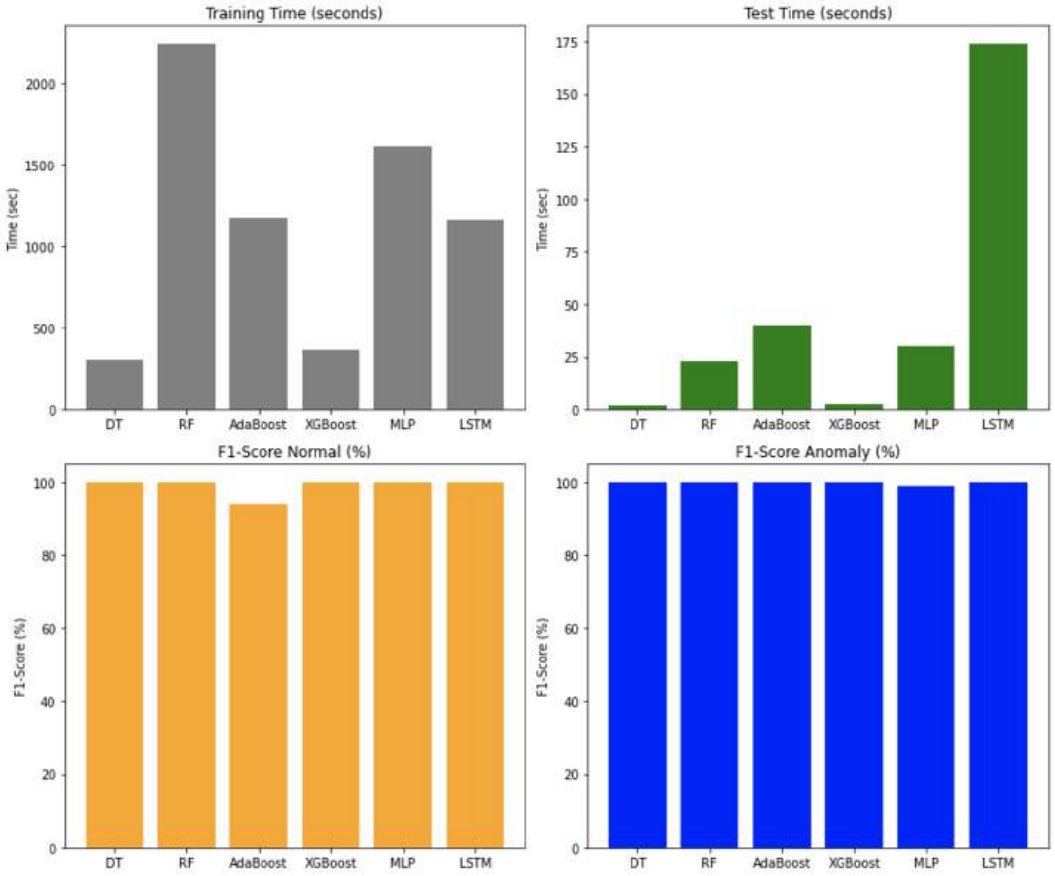


Figure 3.10 – Analyse comparative des performances du modèles binaires en terme de f1-score et de durée d'entrainement et de test

test plus longs. Le choix du modèle optimal dépendra des exigences spécifiques de l'application et des caractéristiques du jeu de données. Ces résultats constituent une base solide, mais une validation supplémentaire sur des données en temps réel et une exploration plus approfondie des performances renforcera la fiabilité des conclusions.

3.5 Classification multi-classe

Dans cette phase de notre analyse de sécurité des réseaux IoT, nous élargissons notre perspective en passant de la classification binaire à la classification multi-classe. Après avoir réussi à discriminer efficacement entre le trafic normal et malicieux avec les modèles Decision Tree (DT), Random Forest (RF), AdaBoost, XGBoost, MLP, et LSTM, notre objectif actuel est d'aller au-delà de la simple détection et d'identifier spécifiquement le type d'attaques en cours. Cette tâche complexe de classification multi-classe implique cinq catégories distinctes : DDoS, DoS, Normal, Reconnaissance, et Theft. En abordant cette étape, nous cherchons à renforcer notre capacité à réagir de manière spécifique aux diverses menaces potentielles qui peuvent affecter les réseaux IoT, en allant au-delà de la simple délimitation entre normalité et malveillance. Dans ce contexte, les modèles préalablement entraînés sur la classification binaire sont réutilisés et adaptés pour traiter ces cinq classes distinctes. L'objectif est d'établir une compréhension approfondie de la nature des attaques détectées et d'attribuer de manière précise chaque instance de trafic malicieux à sa catégorie respective. Cette évolution vers la classification multi-classe constitue une étape cruciale dans le renforcement de la sécurité des réseaux IoT, fournissant des informations plus fines sur la nature et la variété des attaques potentielles.

3.5.1 Définition et métriques

Définition

La classification multi-classe est une généralisation de la classification binaire où une observation peut appartenir à plus de deux classes. Soit $\mathcal{X}$ l'espace des caractéristiques et $\mathcal{Y} = \{1, 2, \ldots, K\}$ l'espace des étiquettes avec $K > 2$. Une fonction de classification multi-classe cherche à apprendre un mapping $f : \mathcal{X} \to \mathcal{Y}$. Pour un ensemble de données d'entraînement $\{(x_i, y_i)\}_{i=1}^{n}$ où $x_i \in \mathcal{X}$ et $y_i \in \mathcal{Y}$, la tâche consiste à prédire l'étiquette y pour une nouvelle observation x. Contrairement à la classification binaire, dans la classification multi-classe, le nombre de classes est supérieur ou égal à trois.

Métriques

Pour chaque classe C_i dans un problème à k classes :
Vrais Positifs (True Positives, TP) :

- **Définition** : Pour une classe donnée C_i, les vrais positifs sont les instances de C_i qui sont correctement prédites comme appartenant à C_i.
- **Formule** :

$$\text{TP}_{C_i} = \sum \text{instances prédites comme } C_i \text{ et réellement } C_i$$

Vrais Négatifs (True Negatives, TN) :

- **Définition** : Pour une classe donnée C_i, les vrais négatifs sont les instances qui n'appartiennent pas à C_i et qui sont correctement prédites comme n'appartenant pas à C_i, les faux positifs sont les instances qui n'appartiennent pas à C_i mais qui sont incorrectement prédites comme appartenant à C_i, .
- **Formule** :

$$\text{TN}_{C_i} = \sum \text{instances prédites comme non-}C_i \text{ et réellement non-}C_i$$

Faux Positifs (False Positives, FP) :

- **Définition** : Pour une classe donnée C_i, les faux positifs sont les instances qui n'appartiennent pas à C_i mais qui sont incorrectement prédites comme appartenant à C_i.
- **Formule** :

$$\text{FP}_{C_i} = \sum \text{instances prédites comme } C_i \text{ mais réellement non-}C_i$$

Faux Négatifs (False Negatives, FN)

- **Définition** : Pour une classe donnée C_i, les faux négatifs sont les instances qui appartiennent à C_i mais qui sont incorrectement prédites comme n'appartenant pas à C_i.
- **Formule** :

$$\text{FN}_{C_i} = \sum \text{instances prédites comme non-}C_i \text{ mais réellement } C_i$$

Les performances d'un classificateur sont souvent évaluées à l'aide des métriques suivantes :

Précision (Precision) pour une classe C_i

- **Définition** : La précision est le rapport du nombre de vrais positifs au nombre total d'instances prédites comme positives.
- **Formule** :

$$\text{Precision}_{C_i} = \frac{\text{TP}_{C_i}}{\text{TP}_{C_i} + \text{FP}_{C_i}} \tag{3.6}$$

Rappel (Recall) pour une classe C_i

- **Définition** : Le rappel est le rapport du nombre de vrais positifs au nombre total d'instances réellement positives.
- **Formule** :

$$\text{Recall}_{C_i} = \frac{\text{TP}_{C_i}}{\text{TP}_{C_i} + \text{FN}_{C_i}} \tag{3.7}$$

Score-F1 (F1-Score) pour une classe C_i

- **Définition** : Le score-F1 est la moyenne harmonique de la précision et du rappel.
- **Formule** :

$$\text{F1-Score}_{C_i} = 2 \times \frac{\text{Precision}_{C_i} \times \text{Recall}_{C_i}}{\text{Precision}_{C_i} + \text{Recall}_{C_i}} \tag{3.8}$$

Justesse (Accuracy) pour une classe C_i

- **Définition** : La justesse est le rapport du nombre de prédictions correctes (vrais positifs et vrais négatifs) au nombre total d'instances.
- **Formule** :

$$\text{Justesse}_{C_i} = \frac{\text{TP}_{C_i} + \text{TN}_{C_i}}{\text{TP}_{C_i} + \text{FP}_{C_i} + \text{TN}_{C_i} + \text{FN}_{C_i}} \tag{3.9}$$

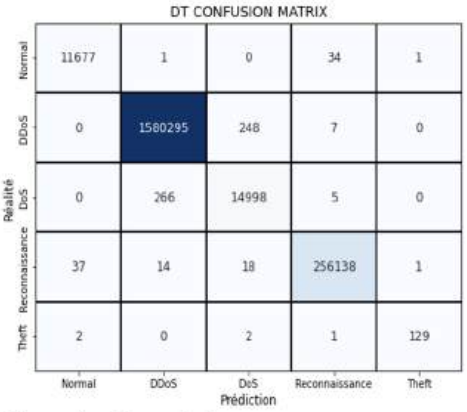
3.5.2 Résultats du modèle d'arbre de décision multi-classe

Le modèle d'Arbre de Décision, adapté à la classification multi-classe, a été construit avec succès en utilisant les données d'entraînement. Ce modèle a été entraîné pour identifier cinq catégories distinctes : DDoS, DoS, Normal, Reconnaissance, et Theft. Lors du test du modèle, le temps d'entraînement s'est établi à 171.95 secondes, tandis que le temps de test a été notablement rapide, avec seulement 0.87 seconde. Les résultats obtenus sont prometteurs, illustrés par la matrice de confusion qui met en évidence la capacité du modèle

à discriminer efficacement entre les différentes classes. Le rapport de classification fournit des détails sur la précision, le rappel et le F1-score pour chaque classe. Notamment, le modèle présente une précision, un rappel et un F1-score de 100% pour la classe Normal, indiquant une capacité parfaite à identifier le trafic normal. Les performances pour les autres classes, telles que DDoS, DoS, Reconnaissance, et Theft, sont également remarquables, avec des scores élevés pour la plupart des métriques évaluées

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	11713
DDoS	1.00	1.00	1.00	1580550
DoS	0.98	0.98	0.98	15269
Reconnaissance	1.00	1.00	1.00	256208
Theft	0.98	0.96	0.97	134
accuracy			1.00	1863874
macro avg	0.99	0.99	0.99	1863874
weighted avg	1.00	1.00	1.00	1863874

a) rapport de classification



b) matrice de confusion

Figure 3.11 – Performances du modèle d'arbre de décision multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b)

En conclusion, le modèle d'Arbre de Décision pour la classification multi-classe démontre une forte aptitude à caractériser et à différencier les différentes attaques potentielles dans les réseaux IoT. Ces résultats prometteurs constituent une base solide pour poursuivre l'évaluation des autres modèles sur cette tâche de classification multi-classe.

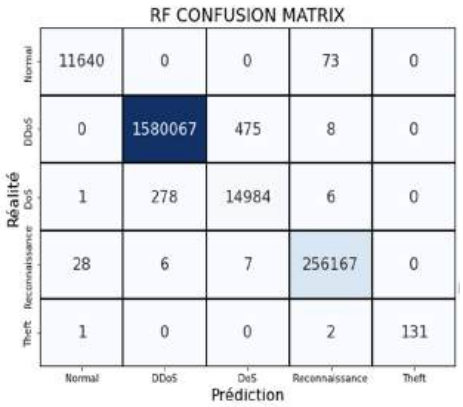
3.5.3 Résultats du modèle de forêt aléatoire multi-classe

Le modèle de Forêt Aléatoire (Random Forest) pour la classification multi-classe a été construit avec succès en utilisant les données d'entraînement. Le temps d'entraînement s'est établi à 1420.18 secondes, montrant une certaine complexité computationnelle, mais les résultats prometteurs lors du test justifient cet investissement en temps. Le temps de test s'est avéré être de 16.76 secondes, confirmant l'efficacité du modèle dans la phase d'évaluation. La matrice de confusion illustre la capacité du modèle à classifier les instances dans les différentes catégories. Chaque cellule de la matrice représente le nombre d'instances prédites pour chaque classe par rapport à la véritable classe. Le rapport de classification fournit ensuite une évaluation détaillée des performances du modèle pour chaque classe, en mettant en avant la précision, le rappel et le F1-score. Notamment, le modèle de Forêt Aléatoire atteint une précision, un rappel et un F1-score de 100% pour la classe Normal, montrant

une grande capacité à identifier le trafic normal. Les performances pour les autres classes, telles que DDoS, DoS, Reconnaissance, et Theft, sont également très élevées, avec des scores équilibrés pour la plupart des métriques évaluées.

	precision	recall	f1-score	support
Normal	1.00	0.99	1.00	11713
DDoS	1.00	1.00	1.00	1580550
DoS	0.97	0.98	0.98	15269
Reconnaissance	1.00	1.00	1.00	256208
Theft	1.00	0.98	0.99	134
accuracy			1.00	1863874
macro avg	0.99	0.99	0.99	1863874
weighted avg	1.00	1.00	1.00	1863874

a) rapport de classification



b) matrice de confusion

Figure 3.12 – Performances du modèle de forêt aléatoire multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b)

En résumé, le modèle de Forêt Aléatoire pour la classification multi-classe présente des performances robustes, confirmant sa capacité à gérer efficacement la complexité de la tâche de différenciation entre différentes catégories d'attaques dans les réseaux IoT. Ces résultats renforcent la crédibilité de l'approche de classification multi-classe et fournissent une base solide pour la comparaison avec d'autres modèles.

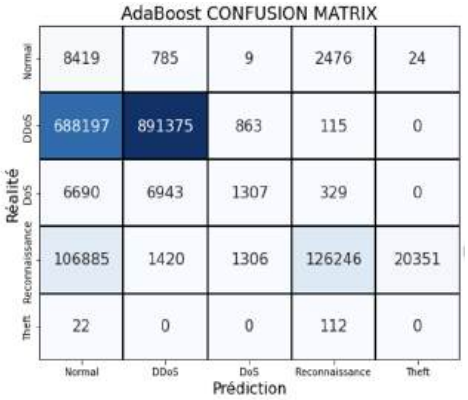
3.5.4 Résultats du modèle d'AdaBoost multi-classe

Le modèle AdaBoost pour la classification multi-classe a été construit avec succès en utilisant les données d'entraînement. Le temps d'entraînement s'est établi à 1201.99 secondes, indiquant un certain coût computationnel. L'évaluation du modèle lors du test a nécessité 42.41 secondes. Cependant, les résultats obtenus révèlent des performances mitigées par rapport aux autres modèles évalués jusqu'à présent. La matrice de confusion met en évidence des difficultés dans la classification des différentes catégories, avec des prédictions significatives pour certaines classes mais des erreurs notables pour d'autres. Le rapport de classification confirme ces observations, indiquant une précision, un rappel et un F1-score variés pour chaque classe. Notamment, le modèle affiche des difficultés dans la classification des classes 1 et 4, avec des scores particulièrement bas pour la précision, le rappel et le F1-score. Ces résultats soulignent des défis spécifiques rencontrés par le modèle AdaBoost dans la gestion de certaines catégories.

En résumé, bien que le modèle AdaBoost ait montré des performances solides dans

	precision	recall	f1-score	support
Normal	0.01	0.72	0.02	11713
DDoS	0.99	0.56	0.72	1580550
DoS	0.38	0.09	0.14	15269
Reconnaissance	0.98	0.49	0.65	256208
Theft	0.00	0.00	0.00	134
accuracy			0.55	1863874
macro avg	0.47	0.37	0.31	1863874
weighted avg	0.98	0.55	0.70	1863874

a) rapport de classification



b) matrice de confusion

Figure 3.13 – Performances du modèle d'AdaBoost multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b)

d'autres contextes, ses résultats sur la tâche de classification multi-classe pour les attaques dans les réseaux IoT indiquent des défis spécifiques. Ces résultats soulignent l'importance de comparer les performances des différents modèles dans des scénarios spécifiques et d'ajuster les approches en conséquence.

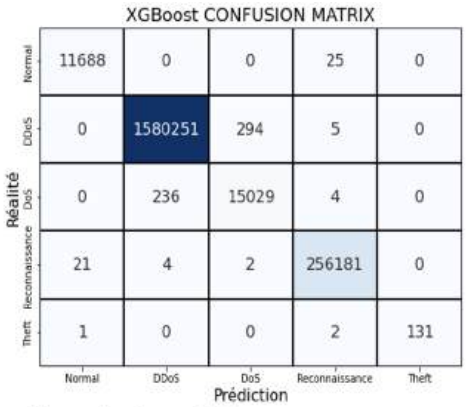
3.5.5 Résultats du modèle de XGBoost multi-classe

Le modèle XGBoost pour la classification multi-classe a été construit avec succès en utilisant les données d'entraînement. Le temps d'entraînement s'est établi à 1832.30 secondes, indiquant un certain coût computationnel, mais les résultats élevés lors du test justifient cet investissement en temps. Le temps de test s'est avéré être de 6.58 secondes, confirmant l'efficacité du modèle dans la phase d'évaluation. La matrice de confusion illustre la capacité du modèle à classifier les instances dans les différentes catégories. Chaque cellule de la matrice représente le nombre d'instances prédites pour chaque classe par rapport à la véritable classe. Le rapport de classification fournit ensuite une évaluation détaillée des performances du modèle pour chaque classe, en mettant en avant la précision, le rappel et le F1-score. Notamment, le modèle XGBoost atteint une précision, un rappel et un F1-score de 100% pour la classe Normal. Le résultat prouve la capacité de ce modèle à identifier le trafic normal. Les performances pour les autres classes, telles que DDoS, DoS, Reconnaissance, et Theft, sont également très élevées, avec des scores équilibrés pour la plupart des métriques évaluées.

En résumé, le modèle XGBoost pour la classification multi-classe présente des performances robustes, confirmant sa capacité à gérer efficacement la complexité de la tâche de différenciation entre différentes catégories d'attaques dans les réseaux IoT. Ces résultats

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	11713
DDoS	1.00	1.00	1.00	1580550
DoS	0.98	0.98	0.98	15269
Reconnaissance	1.00	1.00	1.00	256208
Theft	1.00	0.98	0.99	134
accuracy			1.00	1863874
macro avg	1.00	0.99	0.99	1863874
weighted avg	1.00	1.00	1.00	1863874

a) rapport de classification



b) matrice de confusion

Figure 3.14 – Performances du modèle de XGBoost multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b)

renforcent la crédibilité de l'approche de classification multi-classe et fournissent une base solide pour la comparaison avec d'autres modèles.

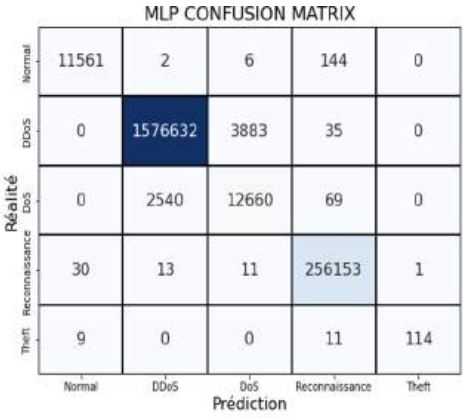
3.5.6 Résultats du modèle MLP multi-classe

Le modèle MLP (Réseau de Neurones Artificiels) pour la classification multi-classe a été construit en utilisant les données d'entraînement. Le temps d'entraînement s'est avéré être de 2489.42 secondes, indiquant un coût computationnel plus élevé par rapport à d'autres modèles. Cependant, les résultats élevés lors du test justifient cet investissement en temps. Le temps de test s'est établi à 28.60 secondes, soulignant l'efficacité du modèle dans la phase d'évaluation. La matrice de confusion illustre la capacité du modèle à classifier les instances dans les différentes catégories. Chaque cellule de la matrice représente le nombre d'instances prédites pour chaque classe par rapport à la véritable classe. Le rapport de classification fournit ensuite une évaluation détaillée des performances du modèle pour chaque classe, en mettant en avant la précision, le rappel et le F1-score. Notamment, le modèle MLP atteint une précision, un rappel et un F1-score de 100% pour la classe Normale, montrant sa très bonne capacité à identifier le trafic normal. Les performances pour les autres classes, telles que DDoS, DoS, Reconnaissance, et Theft, sont également très élevées, bien que légèrement inférieures par rapport au modèle XGBoost, en particulier en termes de rappel et de F1-score pour certaines classes.

En résumé, le modèle MLP pour la classification multi-classe présente des performances solides, confirmant sa capacité à gérer efficacement la complexité de la tâche de différenciation entre différentes catégories d'attaques dans les réseaux IoT. Ces résultats fournissent une base solide pour la comparaison avec d'autres modèles.

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	1580550
DDoS	0.76	0.83	0.80	15269
DoS	1.00	0.99	0.99	11713
Reconnaissance	1.00	1.00	1.00	256208
Theft	0.99	0.85	0.92	134
accuracy			1.00	1863874
macro avg	0.95	0.93	0.94	1863874
weighted avg	1.00	1.00	1.00	1863874

a) rapport de classification



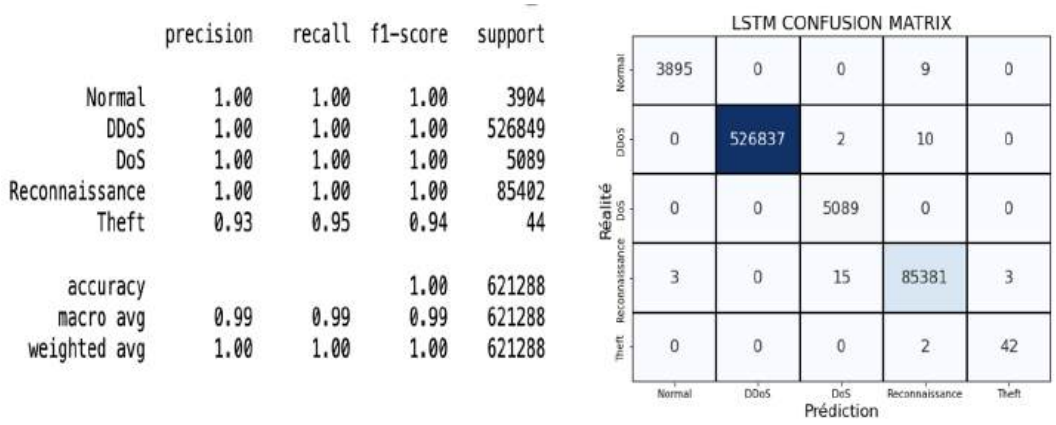
b) matrice de confusion

Figure 3.15 – Performances du modèle MLP multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b)

3.5.7 Résultats du modèle LSTM multi-classe

Le modèle LSTM (Réseau de Neurones Récursif à Longue Mémoire) pour la classification multi-classe a été construit en utilisant des séquences temporelles homogènes extraites des données d'entraînement. Le modèle utilise deux couches LSTM, suivies de couches denses avec une activation tanh pour la non-linéarité, et la dernière couche utilise une activation softmax pour la classification des classes. Le modèle est compilé avec une perte de 'sparse_categorical_crossentropy' et l'optimiseur 'adam'. Le temps d'entraînement du modèle LSTM a été de 1865.34 secondes, soulignant le coût computationnel élevé de l'entraînement de réseaux de neurones récurrents. Cependant, ce modèle a également montré sa capacité à capturer les motifs temporels complexes dans les données. La matrice de confusion met en lumière la capacité du modèle à classer correctement les instances dans les différentes catégories. Chaque cellule de la matrice représente le nombre d'instances prédites pour chaque classe par rapport à la véritable classe. Le rapport de classification fournit ensuite une évaluation détaillée des performances du modèle pour chaque classe, mettant en avant la précision, le rappel et le F1-score.

Le modèle LSTM atteint une précision, un rappel et un F1-score de 100% pour la plupart des classes, montrant ainsi son aptitude à identifier et classifier les différentes attaques dans les réseaux IoT. Même pour la classe avec un nombre réduit d'instances (Theft), le modèle parvient à atteindre un F1-score de 94%, indiquant une bonne capacité de généralisation. En résumé, le modèle LSTM pour la classification multi-classe excelle dans la capture des motifs temporels complexes et offre des performances solides pour la détection et la classification des différentes attaques dans les réseaux IoT. Ces résultats confirment l'efficacité des réseaux de neurones récurrents dans le contexte de la classification multi-classe pour



a) rapport de classification b) matrice de confusion

Figure 3.16 – Performances du modèle LSTM multi-classe illustrées par le rapport de classification en (a) et la matrice de confusion en (b)

la sécurité des réseaux IoT.

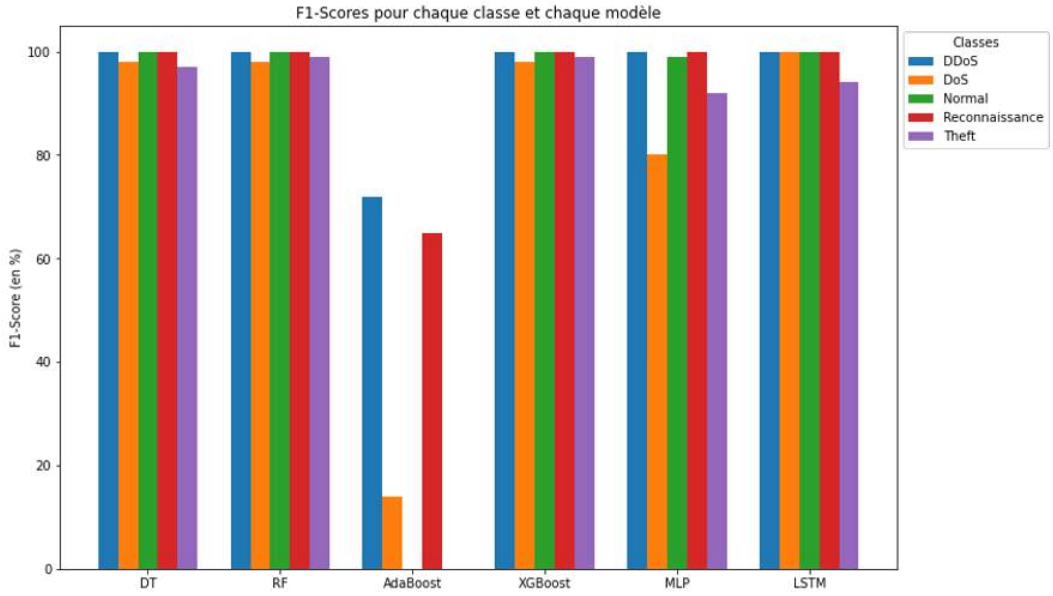


Figure 3.17 – Performances comparatives des six modèles multi-classes avec la métrique f1-score pour chacune des 5 classes

La Figure 3.17 et la Figure 3.18 montrent les performances comparatives des modèles pour chacune des 5 classes mais aussi leur complexité en temps d'entrainement et de test.

L'analyse des performances des six modèles (DT, RF, AdaBoost, XGBoost, MLP et LSTM) pour la classification multi-classe révèle des tendances distinctes. L'Arbre de Décision (DT) et la Forêt Aléatoire (RF) se distinguent par leur précision globale parfaite, démontrant leur capacité à généraliser efficacement sur toutes les classes. AdaBoost, bien que figurant parmi les modèles les plus rapides en termes de temps d'entrainement et de test, présente

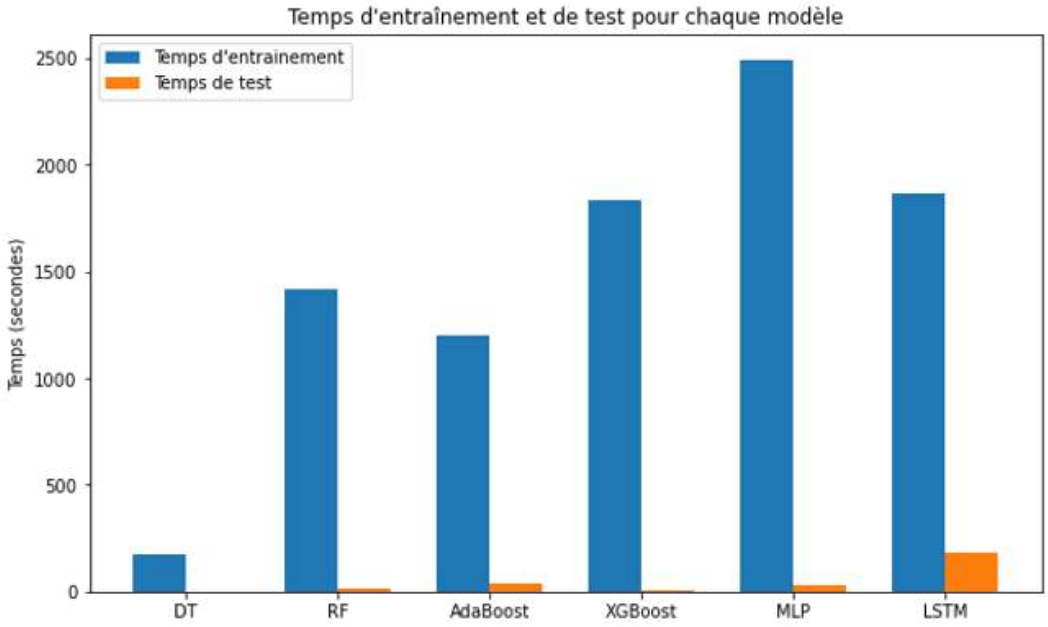


Figure 3.18 – Comparaison des temps d'entrainement et de test des six modèles multi-classes

des précision très faibles et passe complétement à côté pour les classes Normale et Theft . XGBoost maintient une bonne précision générale et est compétitif en termes de temps de test. Les modèles basés sur les réseaux de neurones affichent des performances élevées bien que le MLP des enregistre des précisions légèrement inférieures à celles des modèles basés sur des arbres. Comparés aux autres modèles, le MLP présente une légère baisse de précision pour les classes DoS et Theft, et dans une moindre mesure pour la classe Normal. D'autre part, LSTM, spécialement conçu pour les séquences temporelles, excelle dans la capture de motifs complexes avec une précision parfaite pour toutes les classes. Cependant, il nécessite un temps d'entraînement significatif.

En conclusion, les modèles basés sur des arbres offrent simplicité et interprétabilité, tandis que les réseaux de neurones, en particulier LSTM, démontrent une capacité accrue à traiter des données temporelles complexes. Le choix du modèle optimal dépendra des compromis entre la précision, le temps d'entraînement et les caractéristiques spécifiques du jeu de données. Ces résultats soulignent l'importance d'adapter le choix du modèle en fonction des exigences de la tâche de classification multi-classe dans le contexte de la détection des cyber-attaques dans les réseaux IoT.

3.6 Comparaison des techniques de classification binaire et multi-classe

La détection d'intrusion dans les réseaux IoT représente un enjeu majeur en matière de sécurité informatique, nécessitant des modèles de classification robustes. Dans cette section, nous entreprenons une évaluation approfondie en comparant les résultats obtenus par les modèles binaires, axés sur la discrimination entre trafic normal et malveillant, et les modèles multi-classes, qui s'étendent à la classification des différentes catégories d'attaques. Les deux figures suivantes montrent respectivement les performances des modèles binaires et des modèles multi-classes.

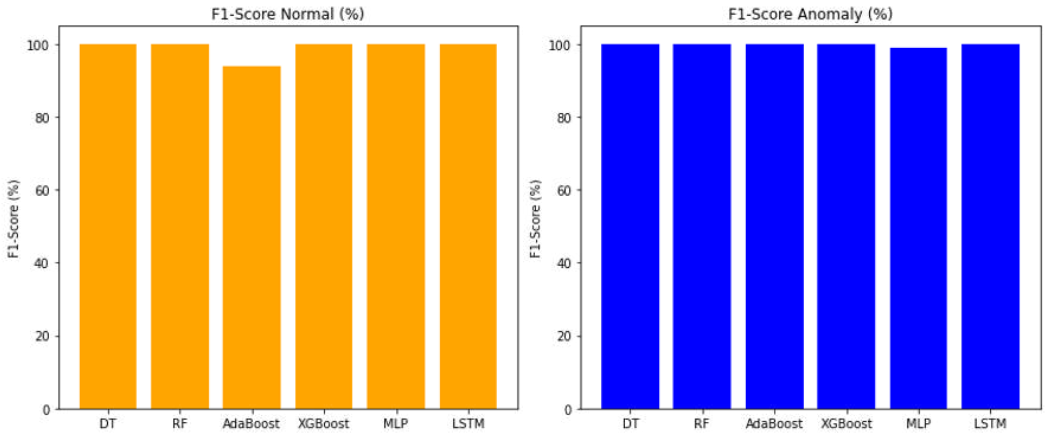


Figure 3.19 – Performances des modèles binaires

On peut constater avec ces deux figures la difficulté à comparer les performances des modèles binaires et les modèles multi-classes. La complexité de la tâche varie considérablement entre ces deux approches, ce qui soulève des questions sur la pertinence de la comparaison directe des performances. Afin de surmonter cette différence fondamentale, nous explorons également la transformation des matrices de confusion des modèles multi-classes, offrant ainsi une base plus équitable pour l'analyse comparative. Soient A une matrice de confusion du modèle de classificateur multi-classe et n le nombre de classes. La $classe\ 0$ correspond à la $classe\ Normal$, les classes de 1 à $n-1$ représentent les catégories d'attaques. La matrice A de dimension $n \times n$ est illustrée par la figure suivante.

On va tenter de transformer la matrice de confusion du modèle multi-classe en une matrice de dimension 2×2 comparables aux matrices de confusion des modèles binaires. La matrice de confusion transformée doit ressembler à celle présentée à la figure suivante.

Les éléments de cette matrices sont déterminés par les formules qui suivent. TN (Vrais

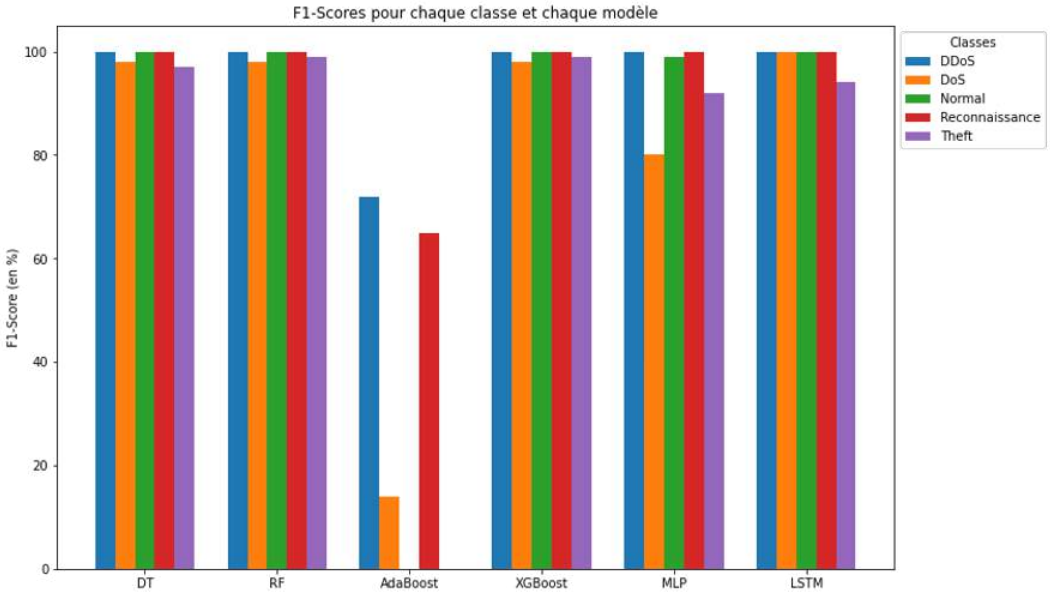


Figure 3.20 – Performances des modèles multi-classes

	0	1	$\cdots$	$n-1$
0	$A_{0,0}$	$A_{0,1}$	$\cdots$	$A_{0,n-1}$
1	$A_{1,0}$	$A_{1,1}$	$\cdots$	$A_{1,n-1}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$n-1$	$A_{n-1,0}$	$A_{n-1,1}$	$\cdots$	$A_{n-1,n-1}$

Figure 3.21 – Format de la matrice des classificateurs multi-classes avec n classes

		Prédiction	
		0	**1**
Réalité	**0**	**TN**	**FN**
	1	**FP**	**TP**

Figure 3.22 – Format de la matrice des classificateurs binaires

Négatifs) : Correspond à l'élément A_{00} de la matrice d'origine, qui représente le nombre d'instances correctement classées comme 'Normal'.

$$TN = A_{00} \tag{3.10}$$

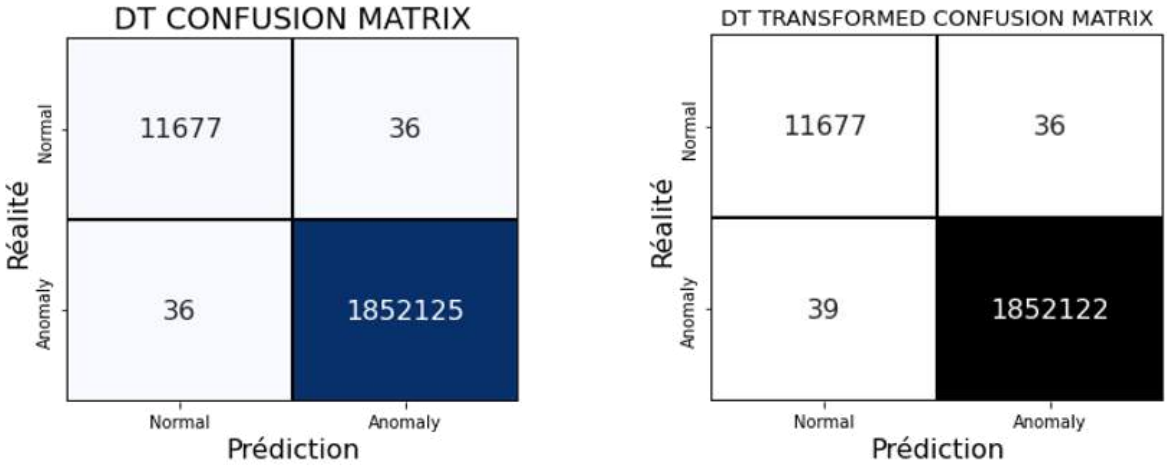
$$TP = \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} A_{ij} \tag{3.11}$$

$$FN = \sum_{i=1}^{n-1} A_{i0} \tag{3.12}$$

$$FP = \sum_{j=1}^{n-1} A_{0j} \tag{3.13}$$

TP (Vrais Positifs) : Correspond à la somme de tous les éléments de la matrice d'origine sauf la première ligne et la première colonne, représentant les instances correctement classées comme des catégories d'attaques. FN (Faux Négatifs) : Correspond à la somme des éléments de la première colonne de la matrice d'origine, représentant les instances de la classe 'Normal' mal classées comme des catégories d'attaques. FP (Faux Positifs) : Correspond à la somme des éléments de la première ligne de la matrice d'origine, représentant les instances de catégories d'attaques mal classées comme la classe 'Normal'. La méthodologie mentionnée a été mise en œuvre sur les matrices de confusion générées par les six modèles multi-classes. En résultat, six nouvelles matrices de confusion ont été obtenues suite à cette transformation. Cette approche de transformation permet une comparaison directe avec les matrices de confusion issues des modèles binaires précédemment évoqués. Les résultats de cette comparaison sont présentés de manière visuelle dans les figures ci-dessous.

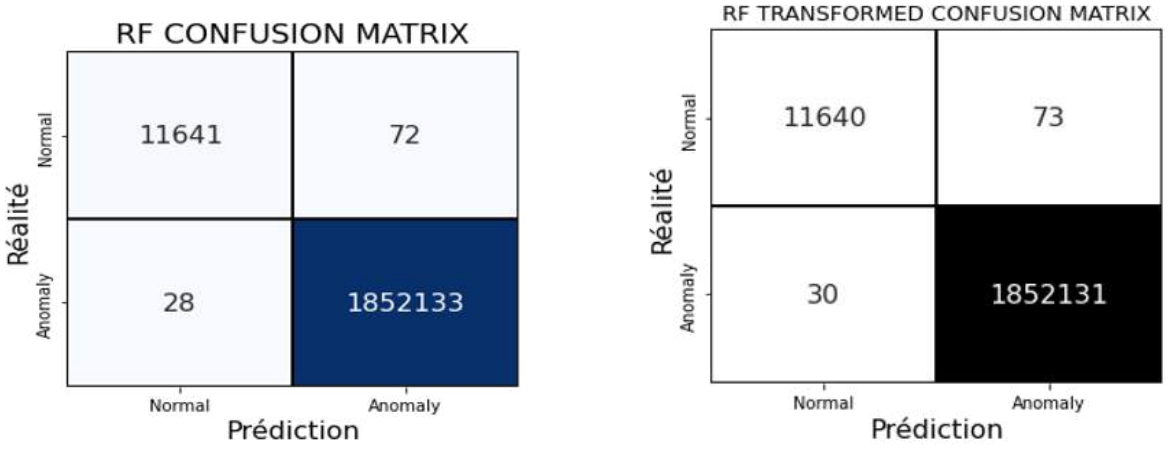
En résumé, les modèles binaires ont tendance à montrer de très bons scores, avec des TP élevés et des erreurs minimales. Les versions multi-classes, bien que globalement performantes, peuvent présenter des dégradations légères en raison de la complexité accrue de la tâche.Cette nouvelle méthodologie de transformation des matrices de confusion que nous avons introduite ci-dessus, permet d'établir une comparaison significative entre les modèles binaires et les modèles multi-classes. En attribuant une classe particulière à la classe Normal et en consolidant les résultats des différentes catégories d'attaques, nous simplifions la comparaison tout en maintenant l'objectif principal de détection d'anomalies. Cette approche permet d'analyser les performances des modèles dans un contexte plus harmonisé, facilitant ainsi une évaluation plus équilibrée de leur efficacité dans la sécurisation des réseaux



a) matrice de confusion du modèle binaire

b) matrice de confusion multi-classe transformée

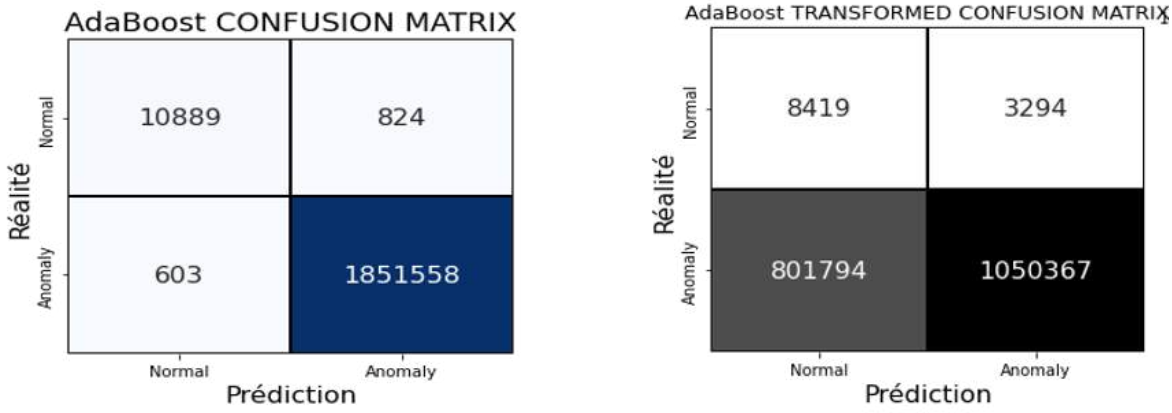
Figure 3.23 – Comparaison des matrices de confusion des modèles arbres de décision binaire et multi-classe aprés transformation. Dans la version binaire, l'arbre de décision présente un nombre de TP élevé et un nombre très faible de FP et FN. La version multi-classe montre une performance similaire avec des chiffres proches, indiquant une bonne généralisation aux classes multiples.



a) matrice de confusion du modèle binaire

b) matrice de confusion multi-classe transformée

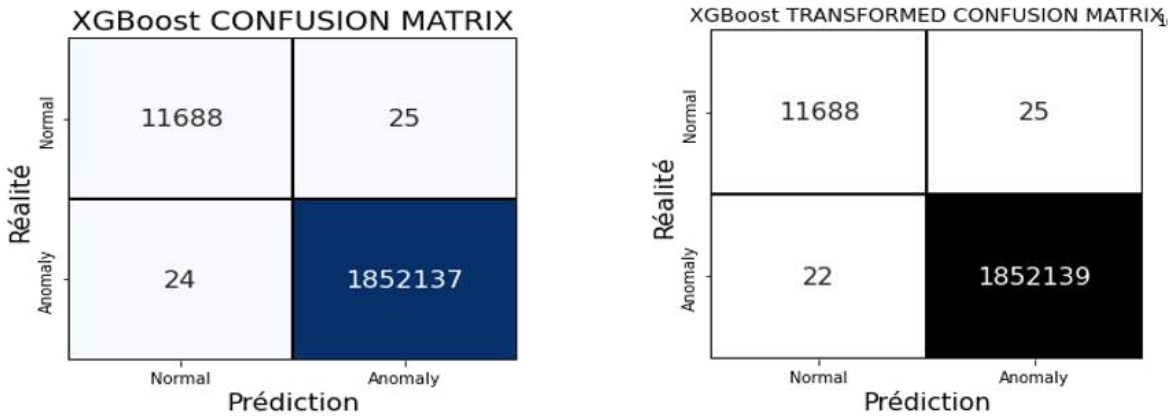
Figure 3.24 – Comparaison des matrices de confusion des modèles de forêts aléatoires binaire et multi-classe aprés transformation. Le modèle binaire présente une performance satisfaisante avec un nombre élevé de TP et un faible nombre de FP et FN. La version multi-classe démontre également de bons résultats, bien que certains indicateurs montrent une légère dégradation par rapport au modèle binaire.



a) matrice de confusion du modèle binaire

b) matrice de confusion multi-classe transformée

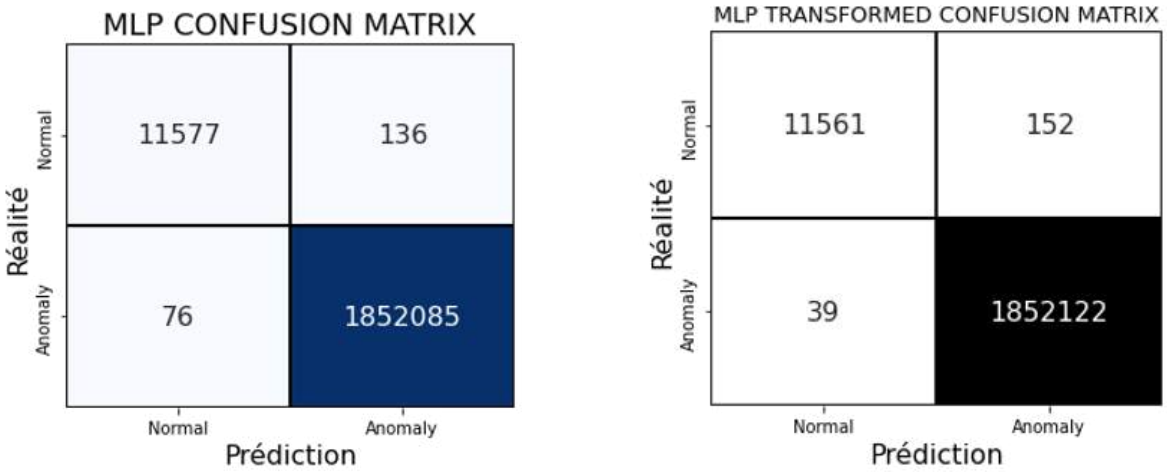
Figure 3.25 – Comparaison des matrices de confusion des modèles Adaboost binaire et multi-classe aprés transformation. Dans le modèle binaire, AdaBoost a montré une précision élevée avec un nombre limité de FP et FN. La version multi-classe a des résultats contrastés, avec une diminution de la précision due à une augmentation du nombre de FN et FP.



a) matrice de confusion du modèle binaire

b) matrice de confusion multi-classe transformée

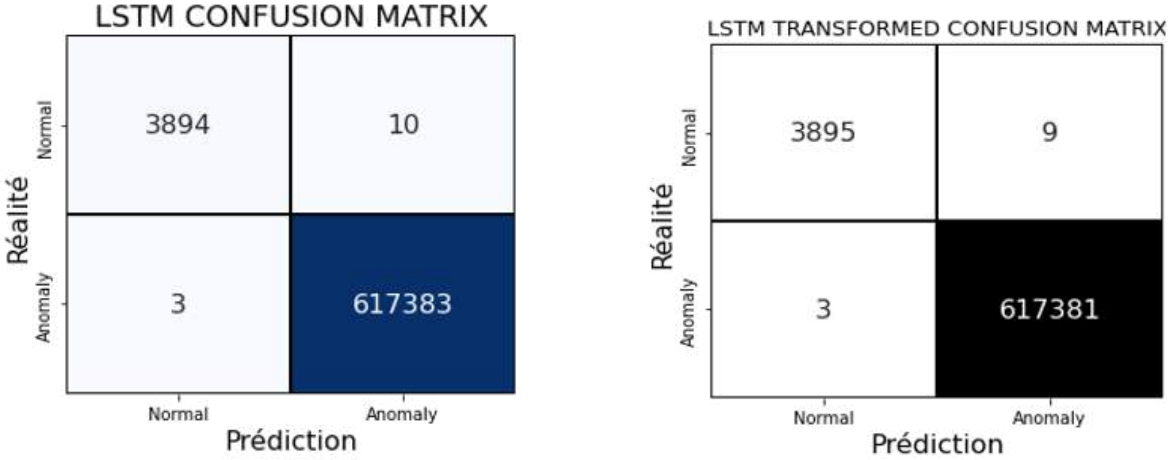
Figure 3.26 – Comparaison des matrices de confusion des modèles XGBoost binaire et multi-classe aprés transformation. Les performances du modèle binaire sont bonnes, avec un TP élevé et un nombre très faible de FP et FN. La version multi-classe conserve une performance élevée, montrant une bonne capacité à gérer plusieurs classes.



a) matrice de confusion du modèle binaire

b) matrice de confusion multi-classe transformée

Figure 3.27 – Comparaison des matrices de confusion des modèles MLP binaire et multi-classe aprés transformation. Le modèle binaire MLP affiche une performance solide, avec un TP élevé et un faible nombre de FP et FN. Dans la version multi-classe, bien que la précision reste élevée, cependant les FP sont légèrement plus élevés.



a) matrice de confusion du modèle binaire

b) matrice de confusion multi-classe transformée

Figure 3.28 – Comparaison des matrices de confusion des modèles LSTM binaire et multi-classe aprés transformation. Le modèle binaire LSTM présente des résultats remarquables, avec un TP élevé et un nombre très faible de FP et FN. La version multi-classe conserve une performance élevée, indiquant une bonne capacité à traiter plusieurs classes tout en maintenant une précision élevée.

IoT contre les intrusions.

3.7 Conclusion

Cette section a abordé la classification des cyber-attaques dans un réseau IoT en utilisant des techniques d'apprentissage automatique et d'apprentissage profond. La classification a été définie comme le processus de prédiction de la classe de points de données, avec des exemples de classification binaire et multi-classe. Une brève présentation des algorithmes d'apprentissage automatique et d'apprentissage profond, tels que l'arbre de décision, la forêt aléatoire, AdaBoost, XGBoost, le perceptron multicouche et le LSTM, a été fournie. Ces algorithmes ont été rigoureusement sélectionnés pour leurs compromis uniques entre complexité, précision et adaptabilité.

Le jeu de données expérimental, l'IoT Botnet dataset, a été présenté, soulignant sa pertinence en tant que l'un des plus récents et complets dans le domaine. Le prétraitement du jeu de données a été abordé, avec une division en ensembles d'entraînement et de test, suivie de la mise en œuvre et de l'évaluation des modèles de classification binaire et multi-classe.

Les résultats des différents modèles ont été détaillés, mettant en évidence leurs performances dans la détection des cyber-attaques. Les métriques telles que la matrice de confusion, la précision, le rappel, le F-score et la justesse ont été utilisées pour évaluer la capacité des modèles à classer correctement les instances positives et négatives.

Enfin, la section a introduit une approche novatrice de comparaison des performances entre les modèles binaires et multi-classes. Cette méthode de transformation des matrices de confusion a permis une évaluation plus équitable des modèles, facilitant ainsi une comparaison significative de leur efficacité dans la sécurisation des réseaux IoT contre les intrusions. Cette étude approfondie fournit une base solide pour l'analyse et le choix approprié des modèles de classification dans le contexte spécifique des cyber-attaques sur les réseaux IoT.

Chapitre 4

Détection d'anomalie au sein d'un réseau IoT à l'aide des auto-encodeurs

4.1 Introduction 112
4.2 Le jeu de données expérimental 113
4.3 Auto-encodeurs et détection d'anomalie 115
4.4 Implémentation d'un auto-encodeur 119
4.4.1 Auto-encodeur 1 121
4.4.2 Auto-encodeur 2 123
4.4.3 Auto-encodeur 3 125
4.5 Conclusion 128

4.1 Introduction

La détection d'anomalies dans les réseaux IoT émerge comme un enjeu crucial pour garantir l'intégrité et la fiabilité des systèmes connectés. De nombreux chercheurs se penchent sur les systèmes de détection d'intrusion pour les réseaux IoT, proposant une pléthore d'approches basées sur l'apprentissage automatique et l'apprentissage profond. Les algorithmes classiques tels que la machine à vecteurs de support (SVM) ou le plus proche voisin (KNN) sont efficaces lorsque les volumes de données sont modestes, mais leur efficacité diminue considérablement avec l'augmentation du volume de données. Dans le domaine de l'IoT, où les volumes de données collectées sont massifs, les modèles d'apprentissage profond sont plus adaptés, car ils deviennent plus efficaces à mesure que le volume de données augmente. Parmi ces modèles, on distingue ceux basés sur l'apprentissage supervisé, précis dans la détection des intrusions connues et largement utilisés dans la classification du trafic pour identifier les attaques. Toutefois, ces modèles supervisés ne parviennent pas à détecter les attaques nouvelles, connues sous le nom de "zero-day attacks" en anglais. Pour combler cette lacune, les modèles d'apprentissage non supervisés se révèlent plus adaptés, une approche de plus en plus préconisée dans la recherche, notamment dans le domaine de la cybersécurité où l'émergence de nouvelles attaques est fréquente. Dans (Khan et al. 2021), un système de détection d'intrusion basé sur l'apprentissage profond est mis en œuvre. Les auteurs proposent un réseau neuronal profond (DNN) pour la détection d'intrusions dans le protocole basé sur MQTT. Leur modèle est entraîné et testé sur deux ensembles de données, dont MQTT-IoT-IDS2020. Ils effectuent une classification binaire et multiclasse. Les résultats montrent une précision élevée dans la détection de différents types d'attaques. Dans (Elsayed et al. 2021), N. Elsayed et al. proposent un système de détection d'intrusions pour la maison intelligente. Ils utilisent un modèle hybride d'apprentissage profond nommé BiLSTM+CNN pour surveiller, détecter et déclencher des actions en fonction des anomalies du trafic réseau.

Dans (Cultice, Ionel et Thapliyal 2020), les auteurs proposent une détection d'anomalies basée sur l'apprentissage profond dans un système domotique. Ils utilisent un réseau neuronal auto-encodeur pour identifier les anomalies à partir de données collectées à partir d'une station météorologique placée dans le jardin d'une maison intelligente. Dans cet article (Apostol et al. 2021b), I. Apostol et al. propose une solution de détection basée sur les anomalies qui utilise des techniques d'apprentissage profond non supervisé pour identifier les activités de botnet IoT. Ils testent spécifiquement un modèle d'auto-encodeur profond dans différents scénarios.

E. Ciklabakkal et al. (Ciklabakkal et al. 2019) présentent un système de détection

d'intrusions basé sur les anomalies. Ils utilisent la bibliothèque $PyOD$ pour implémenter plusieurs modèles d'apprentissage automatique. Ils se concentrent sur les attaques liées à MQTT et obtiennent de bons résultats dans la prédiction des données normales et des attaques. Dans leur travail (Legrand et al. 2018b), A. Legrand et al. étudient la capacité de différents types d'auto-encodeurs dans la détection d'anomalies pour les bâtiments connectés. Les auto-encodeurs récurrents se révèlent les plus efficaces pour détecter les anomalies. A. Dawood et al. (Dawoud et al. 2020b) déploient un cadre de détection d'anomalies basé sur des algorithmes d'apprentissage profond non supervisés. Ils ont réalisé une étude comparative entre deux algorithmes d'apprentissage profond non supervisés, à savoir l'Auto-encodeur (AE) et la machine de Boltzmann restreinte (RBM). Le modèle AE est meilleur que le RBM en termes de performances.

Dans ce chapitre, nous présentons une méthode basée sur l'apprentissage non supervisé, à savoir la détection d'anomalies. Cette approche vise à caractériser le fonctionnement normal du système, toute divergence par rapport à cette norme étant considérée comme une anomalie. Nous proposons ainsi d'explorer l'utilisation des auto-encodeurs, des modèles d'apprentissage profond, pour cette tâche de détection d'anomalies dans les réseaux IoT. Les contributions de chapitre soumis pour comparaître dans le proceedings du CNRIA 2023 [1].

4.2 Le jeu de données expérimental

Pour entrainer et évaluer nos modèles de détection d'anomalies, nous utilisons un jeu de données expérimental représentatif des flux de données dans un réseau IoT. Il s'agit du jeu de données intitulé MQTT-IoT-IDS2020 (Hindy et al. 2020 ; Hindy et al. 2021). Ce jeu de données, minutieusement sélectionné, permet de simuler des scénarios réalistes et variés pour tester l'efficacité de nos modèles.Le dataset MQTT-IoT-IDS2020 est une collection de données de trafic réseau générées en simulant des capteurs MQTT. Il est publié et accessible sur IEEE DataPort. Ce dataset comprend cinq scénarios enregistrés, dont un de fonctionnement normal et quatre scénarios d'attaques. Les types d'attaques comprennent une analyse agressive (Scan_A), une analyse du protocole de datagramme utilisateur (UDP) (Scan_sU), une attaque brute-force SSH Sparta (Sparta), et une attaque brute-force MQTT (MQTT BF). Chaque scénario d'attaque est enregistré indépendamment.

La collecte des données est effectuée à l'aide de tcpdump en enregistrant le trafic Ether-

1. **A. Diallo**, L. Affognon, C. Diallo, and E.C. Ezin, *"Towards the implementation of a dynamic IDS for IoT : Anomaly detection in MQTT traffic."* In : Research in Computer Science and Its Applications : 13th Conference on Research in Computer Science and its Applications, CNRIA 2023. May 2023

net et en exportant les données vers des fichiers pcap. Des machines virtuelles sont utilisées pour simuler les dispositifs du réseau. Nmap est employé pour les attaques de balayage, VLC simule le flux de la caméra, et MQTT-PWN est utilisé pour l'attaque brute-force MQTT. Les composants du réseau comprennent 12 capteurs MQTT, un courtier, une machine simulant un flux de caméra et un attaquant.

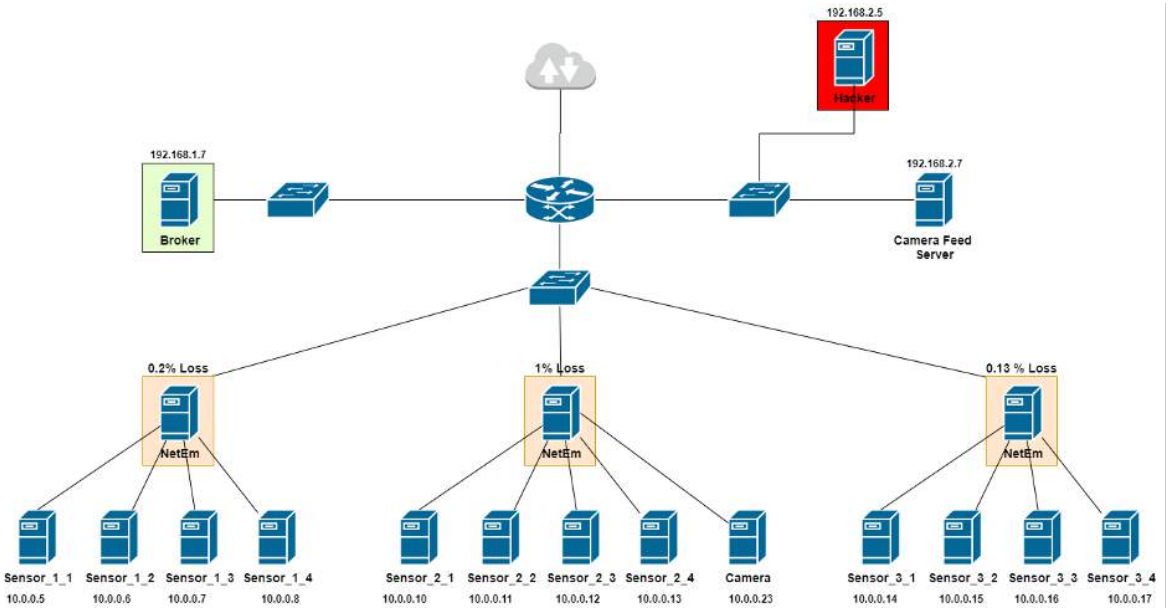


Figure 4.1 – Le banc d'essai du jeu de données MQTT-IoT-IDS2020 (Hindy et al. 2020)

En situation normale, les 12 capteurs envoient des messages randomisés en utilisant la commande MQTT "Publish". Les messages ont des longueurs différentes entre les capteurs pour simuler divers scénarios d'utilisation. Le flux de la caméra est simulé en utilisant VLC avec un flux UDP. Les émulateurs réseau laissent tomber des paquets à des taux de 0,2%, 1%, et 0,13% pour simuler des scénarios réalistes. Pendant les scénarios d'attaque, le fonctionnement normal en arrière-plan continue. Les systèmes d'exploitation des différents dispositifs sont Tiny Core Linux pour les capteurs, Ubuntu pour la caméra et le serveur de flux, et enfin, Kali Linux pour l'attaquant. L'importance de ce dataset est quadruple : il simule un réseau MQTT IoT réaliste en scénario d'exploitation normale, inclut des attaques génériques de balayage réseau ainsi que des attaques brute-force MQTT. Les chercheurs peuvent l'utiliser pour développer et évaluer des systèmes de détection d'intrusion IoT. De plus, il s'agit du premier dataset à inclure des scénarios et des données d'attaques MQTT. La version de MQTT-IoT-IDS2020 utilisée dans ce travail est une version améliorée,adaptée. Elle est présentée dans (Ullah et Mahmoud 2021a) et accessible sur : https://sites.google.com/view/iotdataset1 . Ce jeu de données IoT propose 83 caractéristiques réseau et deux caractéristiques d'étiquetage (Label, Cat), soit un total de 85 caractéristiques. Label est composé de deux classes à savoir Normal

(pour le trafic bénin) et Anomaly (pour le trafic malicieux). Quant à la caractéristique Cat (catégorie), elle est composé des catégories suivantes : Normal(fonctionnement normal),MQTT_bruteforce (attaque par force brute MQTT),Sparta (force brute Sparta SSH) ,Scan_sU(scan UDP) , Scan_A (scan aggressive).La figure suivante montre la catégorisation des attaques dans l'ensemble de données MQTT-IoTIDS2020.

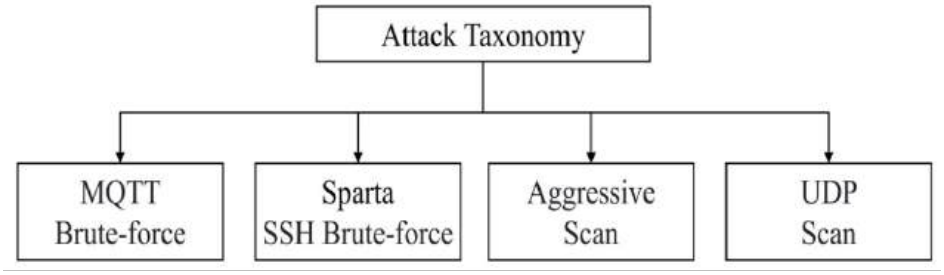


Figure 4.2 – Taxonomie des attaques dans MQTT-IoT-IDS2020

Ce dataset représente une ressource précieuse pour les chercheurs travaillant sur la sécurité IoT, les systèmes de détection d'intrusion, et l'intelligence artificielle, leur permettant de développer et évaluer des modèles dans un environnement réaliste de réseau MQTT IoT.

4.3 Auto-encodeurs et détection d'anomalie

Les auto-encodeurs (Li, Pei et Li 2023) sont des modèles d'apprentissage profond qui offrent une méthode efficace pour la détection d'anomalies dans les données. Comme nous l'avons mentionné précédemment, ce choix n'est pas fortuit. Les auto-encodeurs se sont révélés très efficaces dans la détection d'attaques zero-day. Leur capacité à apprendre des représentations utiles des données en comprimant puis reconstruisant ces dernières permet de mettre en évidence les écarts significatifs par rapport à la normale. Ce sont des modèles d'apprentissage profond non supervisés représentés par un réseau neuronal artificiel. Ils ont exactement le même nombre de neurones sur leur couche d'entrée et leur couche de sortie. Cela signifie que les dimensions de l'entrée et de la sortie sont les mêmes. Mais la grande particularité de ce modèle est que les couches cachées sont symétriques par rapport la couche centrale qu'on appelle code . Le code a la dimension la plus petite par rapport aux autres couches. Le but de l'auto-encodeur est d'obtenir une sortie aussi proche que possible de l'entrée. La structure d'un auto-encodeur comprend trois composantes : l'encodeur, le code (espace latent) et le décodeur.

Dans la phase d'encodage, les échantillons d'entrée sont généralement mappés dans l'espace de caractéristiques de dimension inférieure avec une représentation de caractéristiques restrictive. Cette approche peut être répétée jusqu'à ce que l'espace dimensionnel des

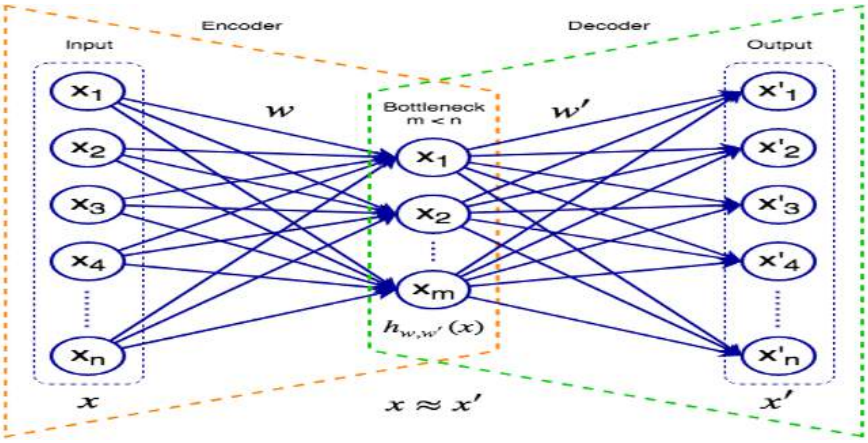
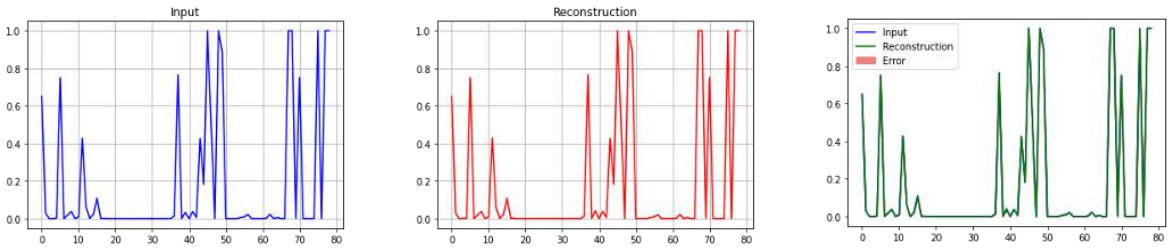


Figure 4.3 – Exemple d'auto-encodeur avec ces trois parties mise en évidence

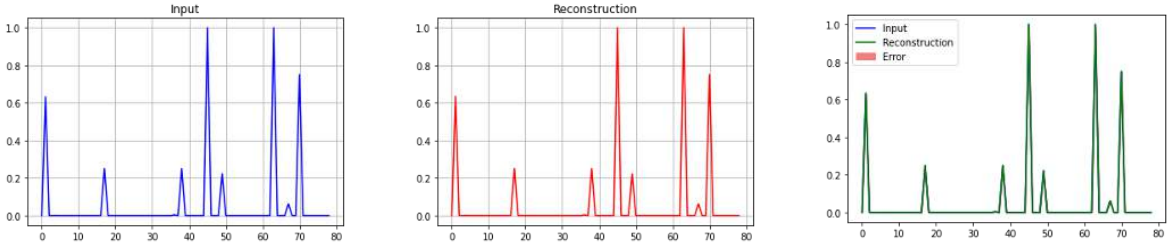
caractéristiques soit atteint. Dans la phase de décodage, nous régénérons les vraies caractéristiques à partir des caractéristiques de dimension inférieure (les caractéristiques codées) avec un traitement inverse. Le modèle reconstruit l'entrée en sa sortie. Ainsi, le modèle est entraîné en minimisant les erreurs de reconstruction, définies comme une fonction de perte (L) comme suit :

$$L(x, x') = \|x - x'\|^2 \tag{4.1}$$



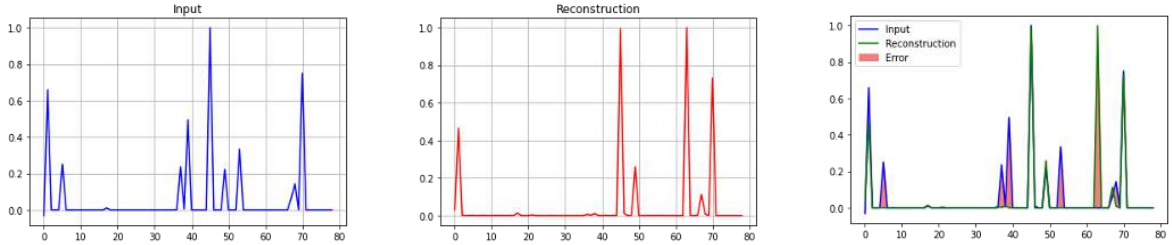
(a) x : une entrée normale (b) x' : reconstruction de x (c) x $_{vs}$ x'

Figure 4.4 – Exemple 1 d'une entrée bénigne avec sa reconstruction



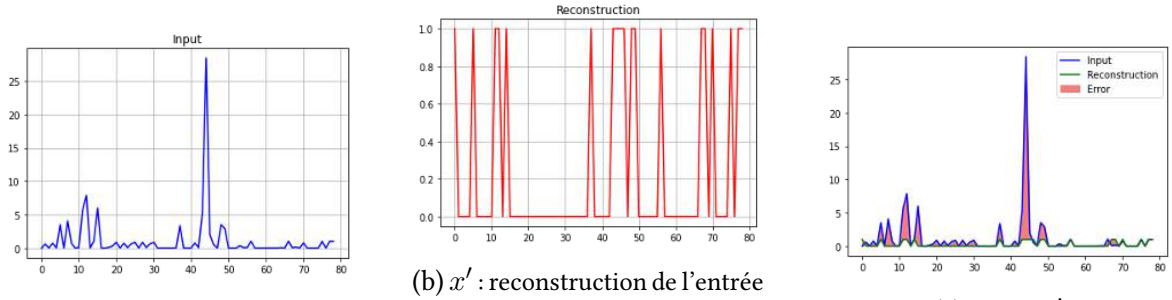
(a) x : une entrée normale (b) x' : reconstruction de x (c) x $_{vs}$ x'

Figure 4.5 – Exemple 2 d'une entrée bénigne avec sa reconstruction



(a) x : une entrée malicieuse (b) x' : reconstruction de x (c) $x \ _{vs} \ x'$

FIGURE 4.6 – Exemple 1 d'une entrée malicieuse avec sa reconstruction



(a) x : une entrée malicieuse (b) x' : reconstruction de l'entrée malicieuse x (c) $x \ _{vs} \ x'$

FIGURE 4.7 – Exemple 2 d'une entrée malicieuse x avec sa reconstruction x'

Figure 4.4, Figure 4.5, Figure 4.6 et Figure 4.7 montrent une entrée (x) pour un auto-encodeur et la sortie ou reconstruction (x') correspondante. Les deux premières figures, à savoir Figure 4.4 et Figure 4.5, correspondent à des entrées bénignes (Normal) alors que les deux dernières, Figure 4.6 et Figure 4.7, illustrent des entrées malicieuses (Anomaly). On peut déjà constater que les entrée bénignes qui correspondent à des traces de trafic normal sont identiques à leur sorties respectives, ces sorties n'étant rien d'autres que leur reconstruction. A l'inverse, les deux dernières entrées sont très différentes de leurs sorties respectives. Les reconstructions présentent des erreurs (en rose) très importantes. L'approche de détection d'anomalies basée sur les auto-encodeurs repose sur le principe que les données normales seront bien reconstruites avec peu d'erreur, tandis que les données anormales, qui diffèrent significativement de ce que le modèle a appris, entraîneront des reconstructions de moindre qualité, avec des erreurs importantes. En d'autres termes, l'auto-encodeur apprend à représenter et à reconstruire les caractéristiques dominantes des données normales, et toute déviation notable par rapport à ces caractéristiques est interprétée comme une anomalie.

Les auto-encodeurs, un type de réseau de neurones artificiels, sont souvent exploités dans le domaine de la détection d'anomalies pour leur capacité à apprendre des représentations compactes des données et à reconstruire fidèlement des entrées similaires à celles qu'ils ont apprises. Dans ce contexte, les auto-encodeurs sont généralement entraînés sur

des données considérées comme normales, c'est-à-dire des exemples de comportement typique ou sain dans un système. L'avantage des auto-encodeurs dans ce contexte réside dans leur capacité à apprendre des représentations significatives des données sans nécessiter de supervision explicite pour les anomalies. En effet, le modèle apprend implicitement à représenter les données normales de manière compacte dans l'espace latent, ce qui rend les anomalies plus faciles à détecter en raison de leur éloignement de cette représentation apprise.

Dans le processus d'entraînement, les auto-encodeurs tentent de minimiser une fonction de perte qui mesure la différence entre l'entrée et la reconstruction produite par le modèle. Une fois entraînés, ces auto-encodeurs peuvent être utilisés pour reconstruire de nouvelles entrées et évaluer la qualité de la reconstruction en calculant une mesure d'erreur, généralement basée sur une distance ou une divergence entre les entrées originales et leurs reconstructions. Pour déterminer si une entrée est normale ou anormale, un seuil d'erreur est souvent défini. Les entrées dont l'erreur de reconstruction dépasse ce seuil, sont classées comme anormales, tandis que celles dont l'erreur reste en dessous sont considérées comme normales. L'équation suivante montre comment le modèle d'auto-encodeur procéde pour prédire la nature d'une entrée.

$$\text{predictor}(x) = \begin{cases} \text{Normal}, & \text{if } x_{\text{error}} < \text{Threshold} \\ \text{Anomaly}, & \text{if } x_{\text{error}} \geq \text{Threshold} \end{cases} \tag{4.2}$$

où x est un vecteur de caractéristiques donné en entrée au modèle et x_{error} est une erreur de reconstruction de x. L'algorithme suivant donne plus de détails sur la prédiction des anomalies par les auto-encodeurs.

Algorithm 1 Algorithme de prediction des anomalies

Fonction `prédicteur`$(x : feature\ vector, threshold : real, AE : autoencoder)$ **:**
  $x' \leftarrow AE.predict(x)$
  $x_error \leftarrow$ `calculer_erreur`(x, x')
  if $x_error < threshold$ **then**
    return "Normal"
  else
    return "Anomalie"
  end

Fonction `calculer_erreur`$(x : feature\ vector, x' : feature\ vector)$ **:**
  $x_error \leftarrow \|x - x'\|^2$ **return** x_error

Le choix approprié du seuil d'erreur est crucial et peut être influencé par divers facteurs, tels que la nature spécifique du problème de détection d'anomalies, la distribution des données, et les exigences en matière de performance et de tolérance aux faux positifs et faux négatifs. En pratique, cette approche de détection d'anomalies avec des auto-encodeurs peut être efficace dans de nombreux domaines, notamment la cybersécurité, mais aussi la détection de fraudes, la maintenance prédictive et la surveillance des systèmes industriels. Il convient de noter que le succès de la détection d'anomalies par auto-encodeur dépend de plusieurs facteurs, notamment la qualité et la représentativité des données d'entraînement, le choix approprié de l'architecture et des hyperparamètres du modèle, ainsi que la sélection judicieuse du seuil d'erreur. De plus, il est souvent nécessaire de réévaluer et d'ajuster périodiquement ces paramètres pour maintenir la performance du modèle dans des environnements dynamiques ou face à des attaques sophistiquées.

4.4 Implémentation d'un auto-encodeur

Le processus de construction du modèle d'auto-encodeur a été méticuleux et stratégique, impliquant plusieurs étapes clés pour parvenir à une configuration optimale du modèle. Dans un premier temps, nous avons entrepris une recherche exhaustive des meilleurs hyperparamètres pour la fonction de perte et l'optimiseur. Cette phase initiale a été cruciale pour déterminer les paramètres fondamentaux qui influencent directement les résultats du modèle. Pour cette recherche, nous avons utilisé $GridSearchCV$ (*GridSearchCV in scikit-learn: A Comprehensive Guide* s. d. ; *Hyperparameter Tuning with GridSearchCV* s. d.), une méthode robuste permettant d'explorer un large éventail de combinaisons de paramètres. La grille de paramètres que nous avons testée comprenait différentes fonctions de perte telles que l'erreur absolue moyenne (MAE) et l'erreur quadratique moyenne (MSE), ainsi que plusieurs optimiseurs populaires tels que $Adam, SGD, RMSprop, Adagrad$ et $Adadelta$. Après avoir identifié la fonction de perte MSE et l'optimiseur $adam$ comme étant les meilleurs choix, nous avons poursuivi le processus en cherchant à déterminer la structure optimale de l'auto-encodeur. Cette phase de recherche a impliqué l'exploration de différentes configurations pour le nombre de neurones dans chaque couche de l'auto-encodeur, ainsi que la dimension de l'espace codé. Encore une fois, nous avons utilisé $GridSearchCV$ (Regulagadda et al. 2023) pour parcourir systématiquement les différentes combinaisons de paramètres et identifier la configuration optimale.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - x_i')^2 \tag{4.3}$$

couche1	couche2	couche3	code
80	64	48	32
72	56	40	24
64	48	32	16

Tableau 4.1 – Paramètres de la grille de recherche du nombre de neurones pour les différentes couches cachées symétriques

loss	optimizer
(mae)	(adam)
(MSE)	(sgd)
	(rmsprop)
	(adagrad)
	(adadelta)

Tableau 4.2 – Paramètres de la grille de recherche de la meilleure combinaison de fonction perte et d'optimiseur

Où x_i est la i^{me} composante du vecteur d'entrée x et $x_i^{'}$ est la i^{me} composante du vecteur de sortie $x^{'}$, x_i' étant donc la reconstruction de x_i.

Après avoir analysé les résultats de cette recherche approfondie, nous avons identifié une configuration optimale de l'auto-encodeur avec 80 neurones dans la première couche, 64 neurones dans la deuxième couche, 48 neurones dans la troisième couche, et un espace codé de dimension 24. Ces paramètres ont été sélectionnés en tenant compte de la complexité du jeu de données et de la capacité de généralisation du modèle.

Enfin, avec les meilleurs hyperparamètres identifiés, nous avons procédé à la construction du modèle d'auto-encodeur en utilisant la bibliothèque Keras. Chaque couche a été configurée en fonction du nombre de neurones déterminé lors de la recherche des hyperparamètres, et les fonctions d'activation $ReLU$ et $sigmode$ ont été choisies en fonction de la nature des données et des objectifs de modélisation. Compilé avec l'optimiseur "adam" et la fonction de perte MSE, le modèle est prêt à être entraîné.

Toujours dans la quête d'un modèle optimal,à partir du modèle générique initial, nous avons développé trois variantes d'auto-encodeurs : $auto-encodeur\ 1$, $auto-encodeur\ 2$ et $auto-encodeur\ 3$. Dans le cadre de cette démarche, l'$auto-encodeur\ 1$ et l'$auto-encodeur\ 2$ ont été entraînés sur une version du jeu de données comprenant 79 caractéristiques. Cependant, pour l'$auto-encodeur\ 1$, le jeu de données original a été utilisé tel quel. En revanche, pour l'$auto-encodeur\ 2$, une méthode d'oversampling a été appliquée afin d'équilibrer la taille des exemples normaux et celle des exemples d'attaques. Cette stratégie visait à garantir que le modèle soit suffisamment exposé à toutes les classes de données afin de mieux généraliser et détecter les anomalies de manière plus précise. D'autre part, l'$auto-encodeur\ 3$ a été entraîné sur une version du jeu de données comprenant seulement 50 caractéristiques. Pour sélectionner ces caractéristiques, nous avons utilisé la méthode $SelectKBest$ de la bibliothèque $sklearn$, qui identifie les 50 meilleures caractéristiques en fonction de leur importance pour la tâche de détection d'anomalies. Cette approche de sélection de caractéristiques vise à réduire la dimensionnalité de l'espace des données tout en

préservant les informations les plus discriminantes pour la tâche de détection d'anomalies. En combinant une sélection judicieuse des caractéristiques, des techniques d'équilibrage des données et une exploration approfondie des hyperparamètres, chaque auto-encodeur a été conçu pour capturer au mieux les particularités des données et détecter les anomalies avec une précision optimale.

4.4.1 Auto-encodeur 1

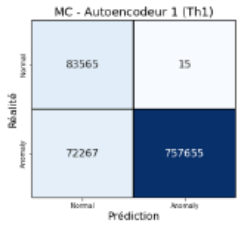
Nous commençons par la présentation et l'évaluation du premier modèle, à savoir l'$auto-encodeur$ 1. Ce modèle est entraîné sur les données originales, comprenant un ensemble de 79 caractéristiques, dans le but de détecter les anomalies présentes dans ces données brutes. Pour préparer les données d'entraînement, nous avons d'abord chargé le jeu de données MQTT-IoT-IDS2020 à l'aide de la bibliothèque pandas. Ce jeu de données contient des informations sur le trafic IoT, classé en anomalies et données normales. Nous avons ensuite divisé les données en deux parties : les données normales et les données d'anomalies. Cela nous a permis de nous concentrer sur les données normales pour l'entraînement de l'auto-encodeur, tandis que les données d'anomalies sont réservées pour l'évaluation ultérieure.

Les données ont été prétraitées en remplissant les valeurs manquantes et en normalisant les caractéristiques à l'aide de la fonction $MinMaxScaler$ de $scikit-learn$. En utilisant $Keras$, nous avons construit le modèle de l'auto-encodeur, composé de plusieurs couches denses avec des fonctions d'activation $ReLU$ pour les couches cachées et une fonction d'activation sigmoïde pour la couche de sortie. Nous avons ensuite entraîné le modèle en utilisant la méthode $fit()$, en spécifiant les données d'entraînement et de validation, ainsi que des $callbacks$ tels que $EarlyStopping$ pour arrêter l'entraînement lorsque la perte de validation cesse de diminuer.

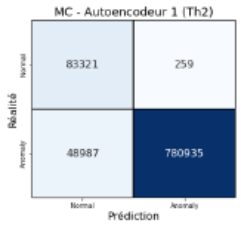
La durée totale de l'entraînement du modèle a été enregistrée pour évaluer son efficacité et son temps d'exécution. Cette démarche nous a permis de créer et d'entraîner le premier modèle d'auto-encodeur, qui sera évalué par la suite pour sa capacité à détecter les anomalies dans les données IoT. Après avoir entraîné l'auto-encodeur et calculé la perte sur l'ensemble de validation, nous avons entrepris d'évaluer le modèle en ajustant le seuil de détection des anomalies. Nous avons testé cinq seuils différents $(Th1, Th2, Th3, Th4, Th5)$ pour déterminer celui qui offre la meilleure précision dans la détection des anomalies.

Les matrices de confusion ci-dessous nous permettent de visualiser le score du modèle en termes de vrais positifs (TP), de faux positifs (FP), de vrais négatifs (TN) et de faux

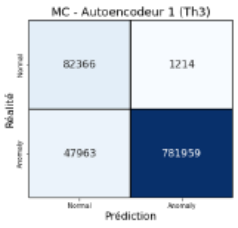
négatifs (FN) à chaque seuil.En haut à gauche, nous avons les vrais négatifs (TN), qui représentent les données normales correctement identifiées comme telles par le modèle.En haut à droite, nous avons les faux positifs (FP), qui représentent les données normales incorrectement identifiées comme des anomalies par le modèle.En bas à gauche, nous avons les faux négatifs (FN), qui représentent les anomalies incorrectement identifiées comme des données normales par le modèle.En bas à droite, nous avons les vrais positifs (TP), qui représentent les anomalies correctement identifiées comme telles par le modèle.



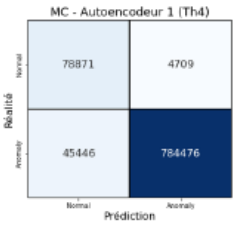
(a) matrice de confusion avec Th1



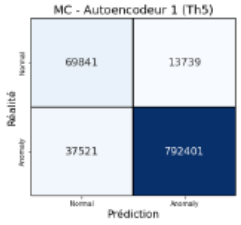
(b) matrice de confusion avec Th2



(c) matrice de confusion avec Th3



(d) matrice de confusion avec Th4



(e) matrice de confusion avec Th5

Figure 4.8 – Les performances de l'$auto - encodeur\ 1 avec les diff rents seuils$

En analysant ces matrices de confusion, nous pouvons évaluer le modèle à différents seuils. Un seuil plus élevé (comme pour $Th1$) conduit à une augmentation des vrais positifs et des faux positifs, ce qui indique une sensibilité accrue mais une spécificité plus faible.Un seuil plus bas (comme pour $Th5$) conduit à une augmentation des vrais négatifs et des faux négatifs, ce qui indique une spécificité accrue mais une sensibilité plus faible. En choisissant un seuil intermédiaire (comme pour $Th3$), nous cherchons un équilibre entre sensibilité et spécificité pour obtenir la meilleure performance globale du modèle. En analysant ces différentes matrices de confusion, nous pouvons ajuster le seuil en fonction des besoins spécifiques de notre application, en optimisant la détection des anomalies tout en minimisant les faux positifs et les faux négatifs.

Pour $Th1$, le seuil a été fixé à la moyenne de la perte d'entraînement plus 25 fois l'écart type de la perte d'entraînement. Le seuil obtenu était de 0.0354. L'analyse des résultats a révélé un taux de faux positifs (FP) très faible, avec seulement 15 erreurs de classification, mais un taux de faux négatifs (FN) plus élevé de 72 267. La précision globale obtenue était de 92.09%, avec une précision normale de 99.98% et une précision d'anomalie de 91.29%. Pour Th2, le seuil a été ajusté à la moyenne de la perte d'entraînement plus 6 fois l'écart

type de la perte d'entraînement. Le seuil obtenu était de 0.0087. Les résultats ont montré une amélioration significative par rapport à $Th1$, avec un taux de faux positifs plus élevé de 259, mais un nombre beaucoup plus faible de faux négatifs de 48 987. La précision globale a augmenté à 94.61%, avec une précision normale de 99.69% et une précision d'anomalie de 94.10%. En utilisant $Th3$, le seuil a été fixé à la moyenne de la perte d'entraînement plus un écart type de la perte d'entraînement. Le seuil obtenu était de 0.0017. Les résultats ont montré une légère amélioration de la précision globale à 94.62%, avec une précision normale de 98.55% et une précision d'anomalie de 94.22%. $Th4$ a été calculé en fixant le seuil à 25% de la moyenne de la perte d'entraînement plus 25% de l'écart type de la perte d'entraînement. Le seuil obtenu était de 0.00042. Les performances du modèle ont été maintenues avec une précision globale de 94.51%, une précision normale de 94.37% et une précision d'anomalie de 94.52%. Enfin, avec $Th5$, le seuil a été fixé à 12.5% de la moyenne de la perte d'entraînement plus 12.5% de l'écart type de la perte d'entraînement. Le seuil obtenu était de 0.00021. Les performances du modèle ont été légèrement affectées, avec une précision globale de 94.39%, une précision normale de 83.56% et une précision d'anomalie de 95.48%.

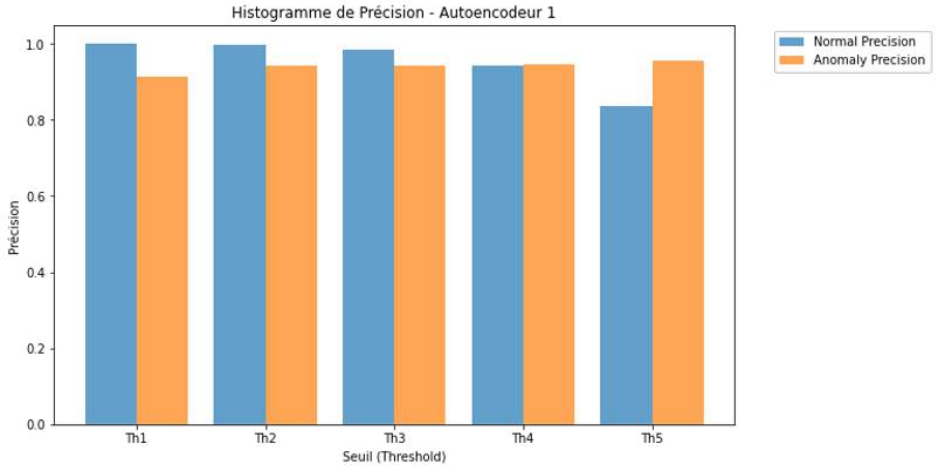


Figure 4.9 – Variation des performances de l'$auto-encodeur$ 1 en fonction du seuil

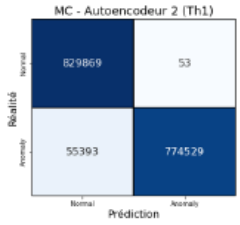
Ces résultats détaillés fournissent un aperçu précis du comportement du modèle à différents seuils de détection des anomalies, permettant ainsi de choisir le seuil optimal en fonction des besoins spécifiques de détection d'anomalies dans les réseaux IoT.

4.4.2 Auto-encodeur 2

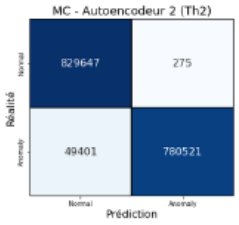
L'$auto-encodeur$ 2 que nous avons développé a été entraîné sur des données auxquelles nous avons appliqué une méthode d'oversampling avec SMOTE. Ces données ont conservé les mêmes 79 caractéristiques que les données originales. Comme avec l'$auto-encodeur$ 1, quatre autres caractéristiques ont été éliminées car elles ne sont pas déter-

minantes pour la détection d'anomalies dans un cadre général. Ces caractéristiques sont : $Flow_ID$, Src_IP, Dst_IP et $Timestamp$. Après avoir chargé le jeu de données, nous avons effectué une conversion des étiquettes en utilisant des valeurs numériques pour $Normal$ et $Anomaly$ ($0 pour Normal$ et $1 pour Anomaly$). Ensuite, nous avons appliqué la méthode SMOTE pour équilibrer les classes en augmentant le nombre d'exemples de la classe minoritaire ($Normal$). Le processus d'entraînement a été programmé pour s'arrêter prématurément si la perte sur les données de validation n'a pas diminué pendant plusieurs époques consécutives, afin de prévenir le surapprentissage. La durée totale de l'entraînement a été enregistrée pour évaluer l'efficacité du processus. Une fois l'entraînement terminé, nous avons obtenu un modèle d'$auto-encodeur$ 2 prêt à être évalué sur les données de test.

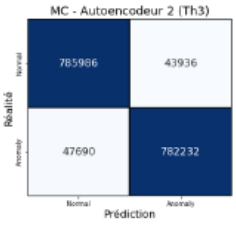
Dans la phase test, nous avons utilisé différents seuils pour mesurer les différentes métriques. L'analyse des résultats obtenus en variant les seuils $(Th1, Th2, Th3, Th4, Th5)$ pour l'évaluation de l'$auto-encodeur$ 2 permet de mieux comprendre l'impact de ce paramètre sur la détection des anomalies. Les matrices de confusion suivantes montrent les résultats du modèle avec les différents seuils.



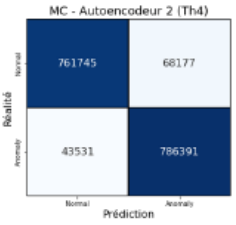
(a) matrice de confusion avec Th1



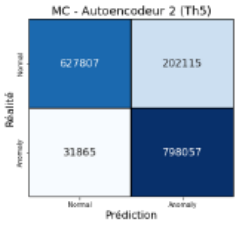
(b) matrice de confusion avec Th2



(c) matrice de confusion avec Th3



(d) matrice de confusion avec Th4



(e) matrice de confusion avec Th5

Figure 4.10 – Les performances de l'$auto-encodeur$ 2 avec les différents seuils

En utilisant le seuil $Th1$, défini à partir de la moyenne des pertes de test augmentée de 40 fois l'écart type, nous obtenons une très haute précision pour les données normales (99,99%) mais une précision légèrement moins élevée pour les anomalies (93,33%). Le choix de ce seuil privilégie ainsi la détection précise des données normales au détriment d'un risque accru de faux négatifs.En revanche, en utilisant le seuil $Th2$, calculé à partir de la moyenne des pertes de test augmentée de 25 fois l'écart type, nous parvenons à un meilleur équilibre entre la précision des données normales (99,69%) et celle des anomalies (94,05%).

Ce seuil offre ainsi un score global plus équilibrée avec une précision globale de 97,01%.Le seuil $Th3$, défini à partir de la moyenne des pertes de test augmentée d'un écart type, maintient une précision élevée pour les anomalies (94,25%) tout en réduisant légèrement la précision pour les données normales (94,71%). La précision globale reste cependant équilibrée à 94,48%. À l'inverse, l'utilisation du seuil $Th4$, calculé à partir de la moyenne des pertes de test augmentée de 0,05 fois l'écart type, réduit davantage la précision pour les données normales (91,79%) tout en conservant une précision élevée pour les anomalies (94,76%). Cela entraîne une diminution significative de la précision globale à 93,27%. Enfin, le seuil $Th5$, défini à partir de 0,5 fois la moyenne des pertes de test ajoutée à 0,05 fois l'écart type, se concentre principalement sur la détection des anomalies avec une précision très élevée (96,16%). Cependant, la précision pour les données normales diminue considérablement (75,65%), entraînant une baisse significative de la précision globale à 85,90%.

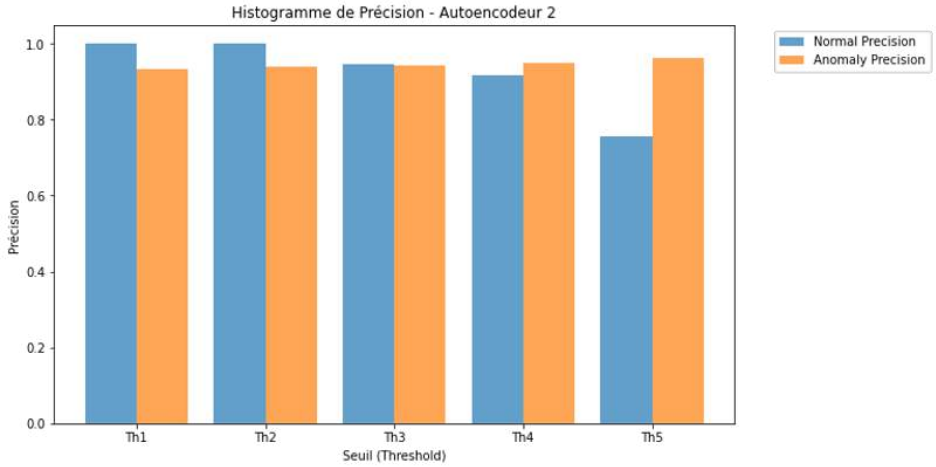


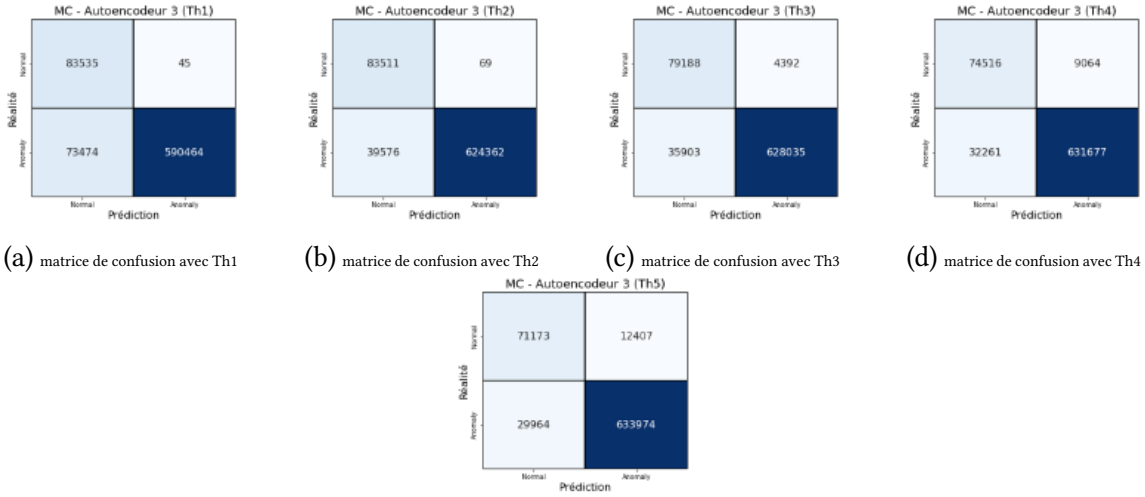
Figure 4.11 – Variation des performances de l'$auto-encodeur$ 2 en fonction du seuil

Ajuster le seuil est donc crucial pour obtenir le bon équilibre entre la détection précise des données normales et des anomalies. La figure ci-dessus illustre la variation des taux de détection des données normales et des données malicieuses. Ces résultats montrent à suffisance l'importance de mettre en place un bon seuil.Dans ce cas, le seuil $Th2$ semble être le plus optimal car il offre le meilleur compromis entre la précision des deux classes et atteint la précision globale la plus élevée.

4.4.3 Auto-encodeur 3

L'$auto-encodeur3$ a été spécialement conçu pour exploiter les 50 meilleures caractéristiques sélectionnées à l'aide de la méthode $SelectKBest$ (*Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning* s. d.). Cette approche vise à choisir les caractéristiques les plus pertinentes pour la tâche de détection d'anomalies,

ce qui permet de réduire la dimensionnalité des données et de focaliser l'apprentissage sur les aspects les plus discriminants. Après le chargement du jeu de données MQTT-IoT-IDS2020, les caractéristiques sont prétraitées et les étiquettes sont converties en valeurs numériques ($Normal- > 0$, $Anomaly- > 1$). Ensuite, la méthode $SelectKBest$ est appliquée pour sélectionner les 50 meilleures caractéristiques. Ces dernières sont ensuite utilisées pour créer un nouveau jeu de données, tout en éliminant également les colonnes $Flow_ID', Src_IP, Dst_IP, Timestamp$. Les données sont ensuite divisées en ensembles d'entraînement et de test, puis normalisées pour faciliter la convergence de l'algorithme d'apprentissage. Le modèle d'auto-encodeur est construit avec une architecture similaire aux précédents modèles, composé de huit couches, avec une couche de sortie de 50 neurones pour correspondre au nombre de caractéristiques sélectionnées. Le processus d'entraînement est surveillé à l'aide du callback $EarlyStopping$, qui arrête l'entraînement lorsque la perte de validation cesse de diminuer. L'auto-encodeur est entraîné sur 100 epochs avec un batch size de 50. Après avoir entraîné l'$auto - encodeur3$, pour évaluer sa performance, nous avons entrepris une série de tests en variant le seuil de décision afin de déterminer le seuil optimal pour la détection des anomalies. Cinq seuils différents, désignés Th1 à $Th5$, ont été utilisés, et les résultats ont été enregistrés pour chacun d'eux. Pour chacun des seuils $Th1$ à $Th5$, les matrices de confusion correspondantes ont été générées afin d'analyser en détail la variation de la performance de l'$auto-encodeur$ 3 en fonction du seuil de décision.



(a) matrice de confusion avec Th1 (b) matrice de confusion avec Th2 (c) matrice de confusion avec Th3 (d) matrice de confusion avec Th4

(e) matrice de confusion avec Th5

Figure 4.12 – Les performances de l'auto-encodeur 3 avec les différents seuils

La variation des matrices de confusion en fonction du seuil de décision met en évidence l'impact significatif de ce paramètre sur la performance de l'$auto - encodeur$ 3 dans la détection d'anomalies. Les seuils plus bas tendent à favoriser la détection des anomalies, tandis que les seuils plus élevés privilégient la précision des données normales. L'évaluation de l'$auto - encodeur$ 3 à travers une série de seuils différents ($Th1Th5$) a fourni des

informations précieuses sur sa capacité à détecter les anomalies dans les données. À travers ces tests, nous avons pu observer comment la variation du seuil de décision affecte la performance globale du modèle.

Pour aller plus loin, la précision ou le taux de détection a été généré aussi bien pour les données que pour les données d'anomalies. En commençant par le seuil $Th1$, relativement élevé (0.0927), nous avons constaté que le modèle tendait à classer davantage de données comme normales, ce qui se traduisait par une précision élevée pour les données normales (99,94 %) mais une précision inférieure pour les données anormales (89,33%), conduisant à une précision globale de 90,16 %. En abaissant le seuil à $Th2(0.0202)$, nous avons observé une amélioration significative de la performance, avec une précision globale de 94,70 %. Cela était principalement dû à une précision élevée pour les données normales (99,92 %) et une précision légèrement améliorée pour les données anormales (94,04 %). En poursuivant avec Th3 (0.00058), nous avons constaté une performance équilibrée entre la précision pour les données normales (94,75 %) et celle pour les données anormales (94,59 %), aboutissant à une précision globale de 94,61 %. $Th4(0.00038)$ a montré des performances similaires à Th3 en termes de précision globale (94,47 %), avec une précision légèrement inférieure pour les données normales (89,16 %) mais une précision améliorée pour les données anormales (95,14 %). Enfin, le seuil $Th5(0.00035)$ a présenté une précision globale de 94,33 %, avec une précision pour les données normales légèrement inférieure (85,16 %) mais une précision élevée pour les données anormales (95,49 %). Ces résultats sont illustrés sur la figure suit.

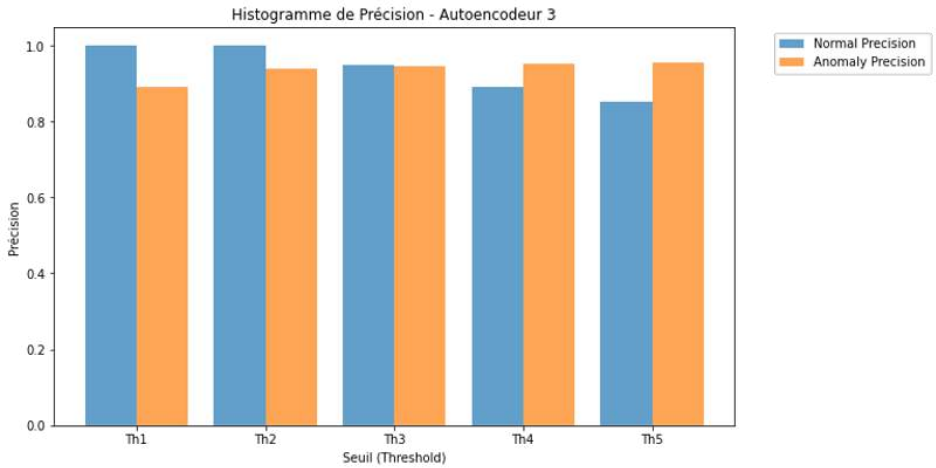


Figure 4.13 – Variation des performances de l'$auto-encodeur$ 3 en fonction du seuil

L'analyse de la variation des résultats du modèle en fonction du seuil met en lumière l'importance de choisir judicieusement ce paramètre pour obtenir la meilleure performance dans la détection d'anomalies. En ajustant le seuil, il est possible d'équilibrer la précision pour les données normales et anormales afin d'optimiser le score global du modèle en fonc-

tion des exigences spécifiques du cas d'utilisation. Bien que $Th2$ semble offrir la meilleure précision globale dans cette analyse, d'autres facteurs et considérations spécifiques au cas d'utilisation doivent être pris en compte pour choisir le seuil optimal.

4.5 Conclusion

En examinant les résultats fournis par les trois auto-encodeurs, nous pouvons avoir un aperçu global de la pertinence de leur utilisation dans la cybersécurité, notamment dans la détection d'anomalie. Tout d'abord, pour l'$auto-encodeur\ 1$, en utilisant les seuils $Th1$ à $Th5$, nous avons constaté une variation significative de la précision pour les données normales et anormales. Par exemple, avec $Th1$ (0.0354), l'$AE1$ a obtenu une haute précision pour les données normales (99,98 %), mais une précision inférieure pour les données anormales (91,29 %), conduisant à une précision globale de 90,16 %. En revanche, $Th2(0.0087)$ a montré une augmentation du score global avec une précision de 94,70 %. Pour l'$auto-encodeur\ 2$, nous avons observé des performances similaires, avec des variations dans la précision en fonction du seuil. Par exemple, avec $Th1(0.0321)$, l'$AE2$ a obtenu une précision élevée pour les données normales (99,99 %) mais une précision inférieure pour les données anormales (93,33 %), donnant une précision globale de 96,66 %. Cependant, $Th2(0.0202)$ a montré une amélioration des résultats avec une précision globale de 97,01 %. Enfin, l'$auto-encodeur\ 3$ a également montré des performances similaires, avec des variations dans la précision en fonction du seuil. Par exemple, avec $Th1(0.0927)$, l'$AE3$ a obtenu une haute précision pour les données normales (99,95 %) mais une précision inférieure pour les données anormales (88,93 %), donnant une précision globale de 90,16 %. Cependant, $Th2(0.0226)$ a montré une amélioration de la précision globale qui passe à 94,70 %.

En ce qui concerne la détection d'anomalies dans le contexte de la cybersécurité, les résultats montrent que les auto-encodeurs sont des outils puissants pour cette tâche. Ils sont capables de détecter des schémas anormaux dans les données qui pourraient indiquer une activité malveillante. Les variations des seuils de décision permettent d'ajuster la sensibilité de la détection en fonction des besoins spécifiques du système de sécurité. Par exemple, un seuil plus bas peut être utilisé pour détecter un large éventail d'anomalies, tandis qu'un seuil plus élevé peut être utilisé pour minimiser les faux positifs. En définitive, les auto-encodeurs sont des outils efficaces pour la détection d'anomalies et d'intrusions dans le domaine de la cybersécurité. Ils offrent une flexibilité dans la détection en permettant d'ajuster les seuils de décision en fonction des besoins spécifiques du système. Cependant, le choix du seuil optimal dépend des exigences de précision et de sensibilité du système,

ainsi que des caractéristiques spécifiques des données et des menaces potentielles.

Chapitre 5

Un Framework intelligent pour la sécurité des réseaux IoT : un système de détection d'intrusions à trois niveaux basé sur l'apprentissage profond

5.1 Introduction 132
5.2 Travaux connexes 133
5.3 Méthodologie 134
5.3.1 Jeu de données expérimental 134
5.3.2 Framework proposé 135
5.3.3 Les modèles implémentés 137
5.4 Résultats 141
5.4.1 Résultats du niveau 1 141
5.4.2 Résultats du niveau 2 144
5.4.3 Résultats du niveau 3 147
5.5 Conclusion 148

5.1 Introduction

Les systèmes de détection d'intrusion (IDS) sont devenus des composants essentiels dans la protection des réseaux IoT, offrant une première ligne de défense contre les attaques en surveillant en permanence le trafic réseau à la recherche de comportements suspects ou malveillants. Cependant, la plupart des systèmes de détection d'intrusion existants se limitent à un seul niveau de détection, ce qui présente des limitations dans leur capacité à identifier et à caractériser les intrusions (Samrin et Vasumathi 2017). Dans cette optique, deux approches principales se démarquent dans la littérature pour la détection d'intrusion dans les réseaux IoT : la détection d'anomalies et la classification d'attaques spécifiques (Dawoud et al. 2020c ; Diallo et al. 2022).

La première approche, basée principalement sur l'apprentissage non supervisé, vise à construire un profil du fonctionnement normal du système pour détecter toute activité anormale. Cette approche, détaillée dans le chapitre 7, repose sur des modèles tels que les auto-encodeurs, qui sont entraînés sur des données normales pour reconnaître les caractéristiques dominantes et signaler toute déviation notable comme une anomalie. Bien que cette approche offre l'avantage de détecter des attaques nouvelles et inconnues, elle présente des lacunes dans la caractérisation précise des intrusions, ne permettant pas d'identifier l'attaque détectée. En revanche, la seconde approche, reposant sur l'apprentissage supervisé, permet d'identifier les intrusions et les types d'attaques spécifiques en s'appuyant sur des ensembles de données étiquettées. Comme décrit dans le chapitre 6, cette approche a donné lieu à plusieurs modèles d'apprentissage automatique, tant binaires que multi-classes, offrant des performances remarquables dans l'identification des attaques connues. Cependant, ces modèles présentent l'inconvénient majeur de ne pas être efficaces contre les attaques de type zero-day, limitant ainsi leur utilité dans un environnement en constante évolution.

Face à ces défis, certains chercheurs ont proposé des systèmes hybrides de détection d'intrusion, combinant les avantages des approches non supervisées et supervisées. Cependant, ces systèmes hybrides (Anthi et al. 2019 ; Ibrahim, Badr et Shaheen 2012) se heurtent à des difficultés comme la complexité, compromettant leur efficacité opérationnelle. Pour surmonter ces lacunes, ce chapitre propose une approche novatrice pour la détection d'intrusion dans les réseaux IoT, en utilisant un système à trois niveaux basé sur le deep learning.

En combinant des modèles non supervisés, tels que les auto-encodeurs, avec des modèles supervisés comme le perceptron multi-couches (MLP), ce système offre une solution

robuste et complète pour renforcer la sécurité des systèmes IoT. Dans ce chapitre, nous examinerons en détail cette approche à trois niveaux, en analysant les travaux connexes, aussi en présentant sa méthodologie, son architecture et ses performances à l'aide de l'ensemble de données MQTTSet. En outre, nous discuterons des implications de cette recherche et des pistes pour les travaux futurs dans le domaine de la sécurité des systèmes IoT. Les travaux exposés dans ce chapitre ont été publiés dans les proceedings de l'ICECCE 2023 [1].

5.2 Travaux connexes

Pour renforcer la sécurité des réseaux informatiques en général, et des systèmes IoT en particulier, de nombreuses recherches sont menées sur les systèmes de détection d'intrusion fondés sur les techniques d'apprentissage automatique et d'apprentissage profond. Dans cette section, nous examinerons quelques travaux connexes traitant de la détection d'intrusion, en mettant l'accent sur les approches basées sur l'apprentissage profond ainsi que sur les systèmes à plusieurs niveaux.

Anthi et ses collaborateurs ont développé un système de détection d'intrusion à trois niveaux basé sur l'apprentissage supervisé pour les maisons connectées (smart home) (Anthi et al. 2019). Leur approche comporte une première couche chargée d'analyser le réseau pour identifier les périphériques connectés en se basant sur leur adresse MAC et leur comportement, une deuxième couche pour classer le trafic réseau comme normal ou anormal, et une troisième couche pour identifier le type d'attaque. Ils ont évalué leurs modèles avec un score F1 de 96,2%, 90,0% et 98,0% respectivement pour le profilage des appareils, la classification du trafic et l'identification des attaques. Cependant, leur système suppose que les attaques potentielles sont connues à l'avance, limitant ainsi sa capacité à détecter de nouvelles attaques.

Dans une autre étude (Ge et al. 2019), des chercheurs ont proposé un système de détection d'intrusion basé sur le deep learning pour les réseaux IoT. Ils ont développé des modèles de réseaux de neurones pour la classification binaire et multi-classes des attaques. Bien que leur système soit capable de classifier les attaques en plusieurs catégories, il ne peut pas détecter les attaques inconnues, ce qui reste une limitation majeure.

Un système de détection d'intrusion à deux niveaux a été présenté dans une étude antérieure (Ibrahim, Badr et Shaheen 2012). Le premier niveau, appelé étape 1, consiste en la détection d'anomalies pour classer le trafic entrant comme normal ou malveillant. Dans

1. **A. Diallo**, L. Affognon, C. Diallo and E. C. Ezin, *"A Three-Level Deep Learning Intrusion Detection System for IoT Network,"* (ICECCE 2023), Dubai,2023, pp. 1-6, doi : 10.1109/ICECCE61019.2023.10442798

le cas d'un trafic malveillant, le système passe à l'étape 2, qui comporte quatre couches séquentielles correspondant à différents types d'attaques. Chaque couche effectue une classification binaire pour détecter et identifier l'attaque correspondante. Cependant, si aucune attaque connue n'est identifiée après avoir traversé les quatre couches, l'attaque est déclarée inconnue. Cette approche se révèle efficace pour détecter et identifier les attaques connues, mais elle peut être limitée dans sa capacité à détecter les attaques inconnues.

En comparaison avec ces travaux, notre approche propose un système de détection d'intrusion à trois niveaux basés sur le deep learning, combinant des modèles non supervisés et supervisés pour détecter avec précision les attaques connues et inconnues dans les réseaux IoT. En utilisant une architecture à plusieurs niveaux, notre système offre une protection avancée contre les menaces potentielles, tout en minimisant les faux positifs et faux négatifs, ce qui en fait une contribution significative à la sécurité des réseaux IoT.

5.3 Méthodologie

5.3.1 Jeu de données expérimental

Il existe une variété d'ensembles de données utilisés dans la littérature par les chercheurs pour entraîner des modèles d'apprentissage automatique et profond destinés aux systèmes de détection d'intrusion. Dans le cadre de cette étude, nous avons opté pour l'utilisation de l'ensemble de données MQTTSet (Vaccari et al. 2020). Cette sélection découle du caractère récent et peu exploité de cet ensemble de données, ainsi que de sa pertinence dans le contexte des communications IoT (Internet des objets) grâce à son utilisation du protocole MQTT (Message Queuing Telemetry Transport) (Nikolov 2020).

MQTTSet se démarque notamment par la diversité des attaques qu'il contient, telles que les attaques par déni de service (DoS) et par force brute, lesquelles sont malheureusement courantes dans les environnements IoT. Comparé à d'autres ensembles de données, comme MQTT-IoT-IDS2020, MQTTSet offre une gamme plus large de scénarios d'attaques, ce qui en fait un choix plus intéressant pour notre étude.

Le scénario simulé dans MQTTSet reflète un environnement domestique intelligent composé de divers capteurs, tels que des capteurs de température, d'intensité lumineuse, d'humidité, de CO-Gaz, de mouvement, de fumée, ainsi qu'un contrôleur de vitesse du ventilateur. Ces capteurs, typiques des environnements IoT domestiques, génèrent un trafic capturé par le courtier MQTT.

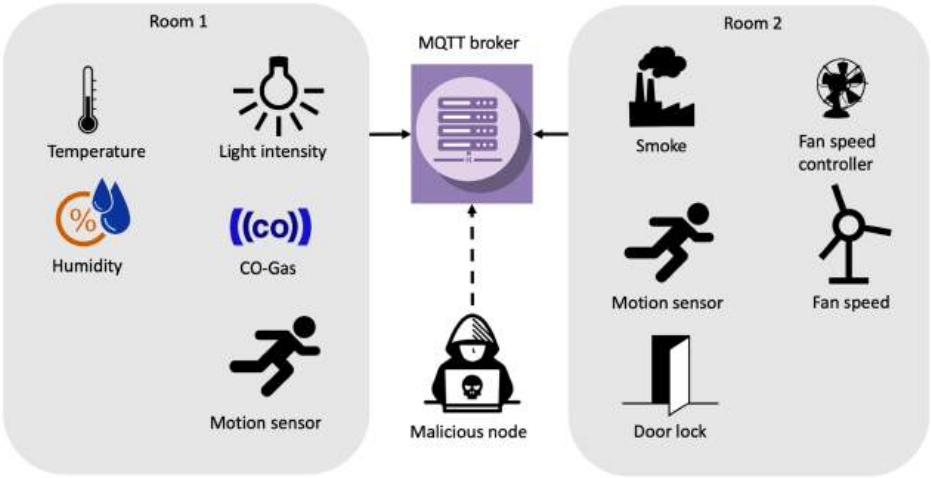


Figure 5.1 – Le banc d'essai du jeu de données MQTTSet

Durant les phases d'attaque simulées dans MQTTSet, un nœud malveillant est directement connecté au courtier, lui permettant d'exécuter diverses cyberattaques. Ces attaques incluent des formes variées telles que les attaques par déni de service (DoS), les inondations MQTT Publish, SlowITe, les données malformées, ainsi que l'authentification par force brute.

Tableau 5.1 – Proportions des types de données dans MQTTSet

Catégories	Legitimate	Bruteforce	DoS	Flood	Malformed	Slowite
Occurrences	11 915 716	14 501	130 233	613	10 924	9 202
Pourcentage	98.71%	0.12%	1.08%	0.01%	0.09%	0.08%

Le Tableau 5.1 présente les proportions relatives des différentes catégories de données dans l'ensemble MQTTSet, mettant en évidence la diversité des scénarios d'attaques couverts par cet ensemble de données.

5.3.2 Framework proposé

Dans cette section, il est fourni une description détaillée du cadre proposé qui est un système de détection d'intrusion à trois niveaux pour l'IoT.

— **Niveau 1 :**
Le niveau 1 représente l'entrée principale de notre système de détection d'intrusion. À ce stade, les paquets entrants sont interceptés et les données correspondantes sont soumises à un processus de prétraitement minutieux. Une fois ce prétraitement effectué, les données sont acheminées vers le $MODEL\ 1$, notre premier détecteur d'anomalies basé sur un auto-encodeur spécialement conçu pour analyser le trafic

réseau provenant de l'ensemble de données MQTTSet. $MODEL\ 1$ est entrainé à partir d'exemples de trafic normal (légitime) extraits de MQTTSet, lui permettant d'internaliser les schémas de communication typiques entre les dispositifs IoT. Cette caractéristique lui confère la capacité de distinguer avec précision les flux de trafic normaux des comportements malveillants. Une force majeure de ce modèle réside dans sa capacité à détecter à la fois les attaques connues et les attaques émergentes, souvent désignées sous le terme de "zero-day attacks" en anglais.

— **Niveau 2 :**
Lorsque $MODEL\ 1$ identifie une anomalie dans le trafic, il est crucial de discriminer entre les cas où le trafic est normal et ceux où il est malveillant. Dans le premier cas, aucune action n'est requise, tandis que dans le second cas, il est impératif de déterminer si l'attaque détectée est connue ou nouvelle. C'est là que le niveau 2 entre en jeu. Le niveau 2 est chargé de déterminer la nature des attaques détectées au niveau 1, en distinguant les attaques connues des attaques émergentes. À cette fin, un autre auto-encodeur, nommé $MODEL\ 2$, est déployé. $MODEL\ 2$ est entraîné avec des exemples de quatre types d'attaques spécifiques inclus dans MQTTSet : Slowite, Flood, Malformed et Brute Force. Ces attaques sont considérées comme étant familières au système, tandis que les données correspondant à l'attaque DDoS sont réservées à la phase de test, simulant ainsi une attaque inconnue.

— **Niveau 3 :**
Lorsqu'une attaque nouvelle est détectée, elle est enregistrée pour une analyse ultérieure. En revanche, si l'attaque appartient à la catégorie des attaques connues, elle doit être identifiée pour permettre la prise de mesures appropriées. Cette tâche d'identification précise est confiée au niveau 3. Le niveau 3 joue un rôle essentiel dans l'identification et la classification des attaques connues détectées par le niveau 2. À cette fin, $MODEL\ 3$, un classificateur multi-classes basé sur un réseau de neurones artificiels (MLP), est utilisé. Les classes du classificateur correspondent aux différentes attaques connues, telles que celles utilisées pour entraîner $MODEL\ 2$. Une fois entraîné, $MODEL\ 3$ est capable de reconnaître et de classifier avec une précision élevée les attaques détectées. Lorsqu'une attaque est identifiée, $MODEL\ 3$ transmet l'information à l'administrateur du système, lui permettant de prendre des mesures correctives appropriées pour contrer l'attaque et préserver l'intégrité du réseau IoT.

Algorithm 2 Algorithme de détection d'intrusion à trois niveaux pour l'IoT

Fonction `prétraiter_données(`$trafic_entrant$`)` :
| **return** données

Fonction `MODEL_1_détecter(`$donnes$`)` :
| $anomalie \leftarrow autoencoder1.predict(donnes)$
| **return** anomalie

Fonction `MODEL_2_classifier(`$anomalie$`)` :
| $type_attaque \leftarrow autoencoder2.predict(anomalie)$
| **return** type_attaque

Fonction `MODEL_3_identifier(`$type_attaque$`)` :
| $classe_attaque \leftarrow MLP.predict(type_attaque)$
| **return** classe_attaque

Fonction `notifier_administrateur(`$nouvelle_attaque$`)` :
| **return** "anomalie inconnue" ;

Algorithme :
| $donnes \leftarrow$ `prétraiter_données(`*trafic_entrant*`)`
| $anomalie \leftarrow$ `MODEL_1_détecter(`*données*`)`
| **if** $anomalie$ **then**
| | $type_attaque \leftarrow$ `MODEL_2_classifier(`*anomalie*`)`
| | **if** $type_attaque$ *est connu* **then**
| | | $classe_attaque \leftarrow$ `MODEL_3_identifier(`*type_attaque*`)` **return** "Attaque identifiée : " + classe_attaque
| | **else**
| | | `notifier_administrateur(`*type_attaque*`)` **return** "Anomalie inconnue détectée"
| | **end**
| **else**
| | **return** "Trafic normal"
| **end**

5.3.3 Les modèles implémentés

Dans chacun des trois niveaux de notre framework, nous avons implémenté des modèles de réseaux de neurones artificiels spécifiquement conçus pour détecter et classifier les attaques dans le trafic réseau IoT. Aux deux premiers niveaux, nous avons développé

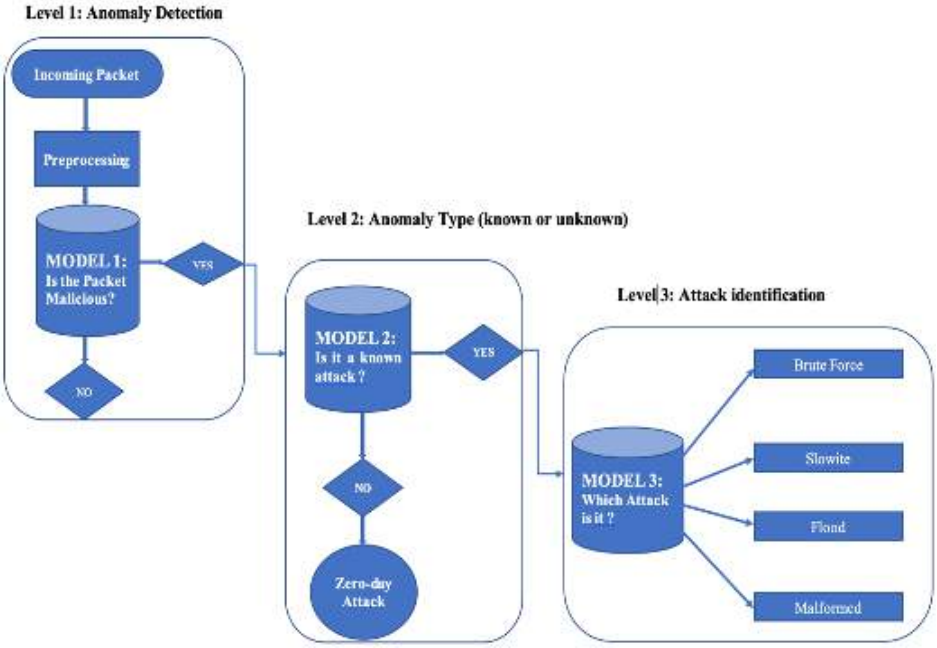


Figure 5.2 – Description du Framework proposé

des auto-encodeurs similaires, tandis qu'au troisième niveau, un perceptron multicouche a été mis en œuvre pour servir de classificateur multi-classes dans l'identification des attaques.

MODEL 1 et MODEL 2

Dans les deux premiers niveaux ($Niveau1et Niveau2$) de notre système, nous avons mis en œuvre deux auto-encodeurs que nous avons baptisés respectivement $MODEL\ 1$ et $MODEL\ 2$. Nous avons opté pour l'utilisation d'auto-encodeurs en raison de leur remarquable efficacité dans la détection des anomalies et des schémas non conventionnels présents dans les données. Ces auto-encodeurs sont spécialement entrainés pour reconstruire les données d'entrée avec une perte minimale, leur permettant ainsi d'identifier avec précision les variations significatives dans le trafic réseau. Dans le chapitre précédent, chapitre 4, nous avons largement exploré les concepts sous-jacents des auto-encodeurs, ainsi que leurs applications dans la détection d'anomalies. Par conséquent, nous éviterons de répéter ces détails dans cette section.

L'utilisation de deux auto-encodeurs distincts dans les deux premiers niveaux de notre approche offre une approche multicouche puissante pour la détection d'anomalies, ce qui se traduit par une amélioration notable de la sensibilité et de la spécificité du système. Cette approche multicouche nous permet de capturer des nuances subtiles dans les données,

ce qui renforce la capacité du système à identifier les comportements anormaux avec une grande précision.

Différents paramètres sont testés afin de construire un modèle optimal. Le choix du nombre de couches ainsi que du nombre de neurones par couche a été fait empiriquement, en explorant diverses combinaisons pour trouver la configuration la plus performante. Les auto-encodeurs implémentés sont configurés comme suit :

— Une couche d'entrée et une couche de sortie de 33 neurones, correspondant à la taille du vecteur caractéristique.
— Un encodeur composé de trois couches cachées avec des dimensions respectives de 33, 32 et 24 neurones.
— Un décodeur composé également de trois couches cachées, avec des dimensions de 24, 32 et 33 neurones.
— Un code (Latent representation) de 16 neurones est introduit pour réduire la dimensionnalité de la représentation latente, facilitant ainsi l'apprentissage des caractéristiques les plus importantes.

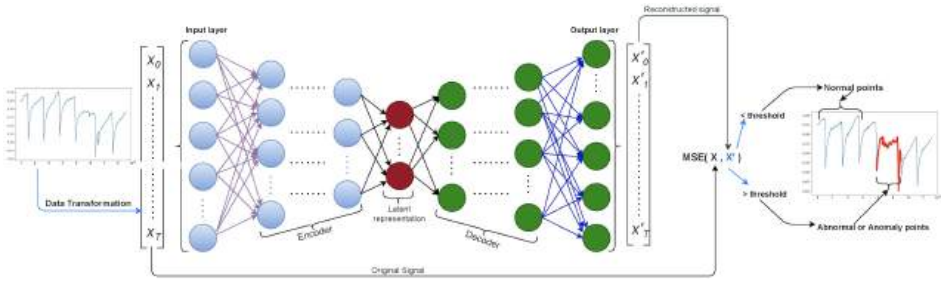


Figure 5.3 – Architecture de l'auto-encodeur implémenté comme MODEL 1 et MODEL 2. Source image, Nicholaus et al. 2021

Dans la conception du modèle, nous avons opté pour l'utilisation de l'unité linéaire rectifiée $(ReLU)$ comme fonction d'activation pour les couches cachées, offrant ainsi une non-linéarité essentielle à la représentation des données. Pour la couche de sortie, nous avons choisi la fonction sigmoïde, idéale pour la classification binaire et la génération de probabilités. L'algorithme d'optimisation $Adam$ a été sélectionné pour sa capacité à converger rapidement vers des solutions de qualité, grâce à son adaptation des taux d'apprentissage pour chaque paramètre du modèle. En ce qui concerne l'évaluation de la performance du modèle, nous avons utilisé la fonction de perte d'erreur quadratique moyenne (MSE), qui mesure la différence entre les valeurs prédites et les valeurs réelles, offrant ainsi une estimation précise de l'erreur à chaque étape de l'apprentissage.

MODEL 3

Dans le troisième niveau, nous utilisons un perceptron multi-couches (MLP) tel que décrit dans (Widiasari, Nugroho et Widyawan 2017). Un MLP est un type de réseau de neurones artificiels à propagation avant composé de plusieurs couches, généralement trois ou plus, interconnectées par des neurones. La structure du réseau implique une série de couches interconnectées, formant un graphe dirigé de la couche d'entrée à la couche de sortie. Chaque neurone, à l'exception de la couche d'entrée, applique une fonction d'activation non linéaire à son entrée. Le nombre de couches cachées dépend de la complexité de la tâche à accomplir. Les neurones dans une couche sont entièrement connectés aux neurones de la couche suivante via des connexions pondérées, qui sont mises à jour lors du processus d'apprentissage. Les MLP utilisent la rétropropagation comme méthode d'apprentissage supervisé, ajustant les poids pour minimiser les erreurs.

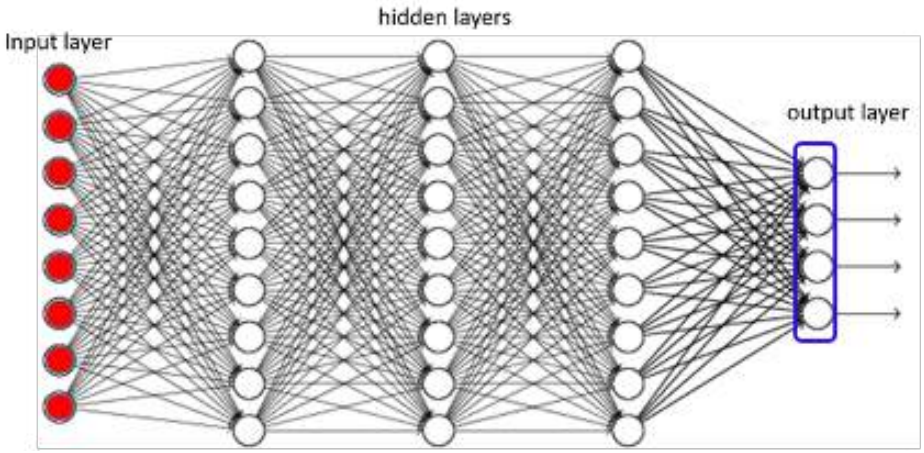


Figure 5.4 – Exemple d'architecture d'un MLP avec Trois couches cachées

Comme illustré dans la Figure 5.4, notre architecture MLP implémentée se compose de cinq couches. Le MLP construit comprend une couche d'entrée avec 33 neurones et une couche de sortie avec 4 neurones, représentant le nombre de classes à prédire. Il intègre quatre couches cachées, avec respectivement 100, 150, 100 et 33 neurones. Ce modèle sert de classificateur multi-classes, formé sur un sous-ensemble de MQTTSet contenant des données de quatre catégories d'attaques distinctes : bruteforce, flood, malformed et slowite. Pour les fonctions d'activation, ReLU et Softmax sont utilisés respectivement dans les couches cachées et la couche de sortie. Pendant la phase d'entraînement, l'optimisation Adam est utilisée, avec la cross-entropie catégorique comme fonction de perte.

5.4 Résultats

Dans cette section, la performance du système proposé est évaluée. Comme spécifié précédemment, le système est composé de trois niveaux et à chaque niveau est implémenté un modèle d'apprentissage profond. Par conséquent, les résultats sont également divisés en trois sous-sections, à savoir Niveau 1, Niveau 2 et Niveau 3. Nous allons examiner les performances du système à chaque niveau et analyser les résultats obtenus pour évaluer son efficacité globale.

5.4.1 Résultats du niveau 1

Ce niveau représente l'entrée principale dans le système, où s'amorce la première étape cruciale de la détection. Tel que précisé dans la section précédente, c'est à ce stade que s'opère l'utilisation d'un auto-encodeur désigné sous le nom de $MODEL$.

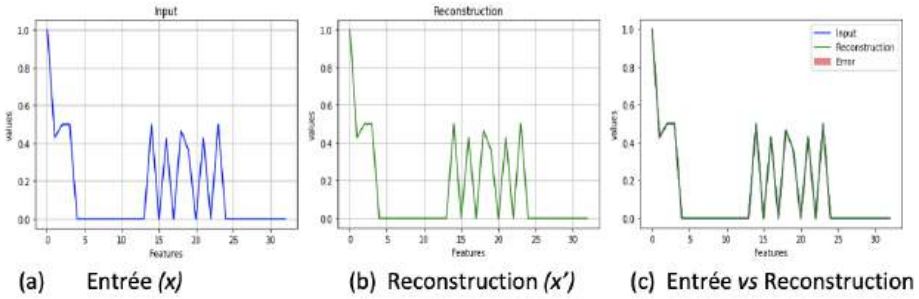


Figure 5.5 – MODEL 1 : Exemple d'entrée normale

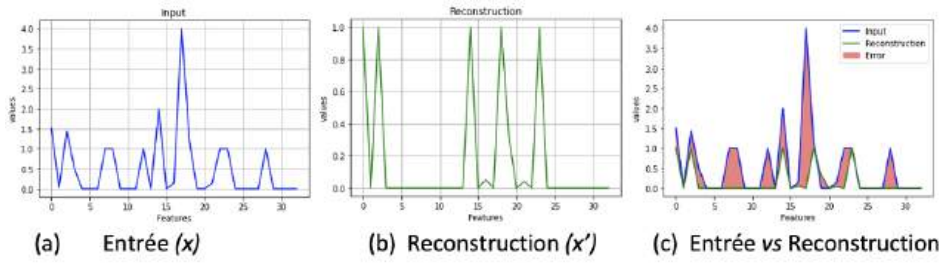


Figure 5.6 – MODEL 1 : Exemple d'entrée malicieuse

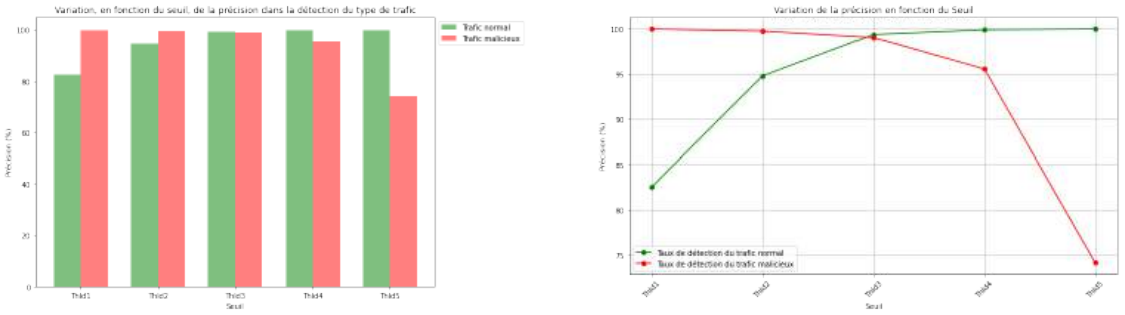
Les performances de ce modèle sont présentées dans la Figure 5.5 et la Figure 5.6, illustrant respectivement sa capacité à identifier le trafic normal et à détecter les activités malveillantes. Dans cette section, nous analyserons de manière approfondie les résultats engendrés par le $MODEL$, afin d'évaluer son efficacité dans la détection des anomalies et des schémas de trafic illicite.

La Figure 5.5 et la Figure 5.6 exposent la manière dont le premier niveau ($MODEL\ 1$) reconstruit respectivement le trafic normal et le trafic malveillant. Nous notons avec intérêt que sur la Figure 5.5, l'entrée et la sortie se superposent parfaitement. Cette concordance démontre une reconstruction impeccable de l'entrée correspondant au trafic normal. En revanche, pour ce qui est de l'entrée malveillante, l'écart de reconstruction est significatif, révélant ainsi une nette distinction entre l'entrée et la sortie reconstruite (Figure 5.6).

Ainsi, afin de procéder à des prédictions, nous définissons un seuil d'erreur au-delà duquel une entrée est catégorisée comme normale ou malveillante. Si l'erreur de reconstruction se situe en dessous du seuil, l'entrée est classée comme normale ; autrement, elle est considérée comme malveillante.

Tableau 5.2 – Variation de la précision globale (Accuracy) en fonction du seuil

Nom du seuil	Valeur du seuil	Trafic normal	Trafic malicieux	Trafic Global
Thld1	moyenne$(train_loss) + 0.1 \times$ std$(train_loss)$	82.56%	99.99%	83.69%
Thld2	moyenne$(train_loss) + 0.25 \times$ std$(train_loss)$	94.79%	99.75%	95.11%
Thld3	moyenne$(train_loss) +$ std$(train_loss)$	99.37%	99.03%	99.35%
Thld4	moyenne$(train_loss) + 5 \times$ std$(train_loss)$	99.90%	95.55%	99.62%
Thld5	moyenne$(train_loss) + 15 \times$ std$(train_loss)$	99.99%	74.17%	98.31%

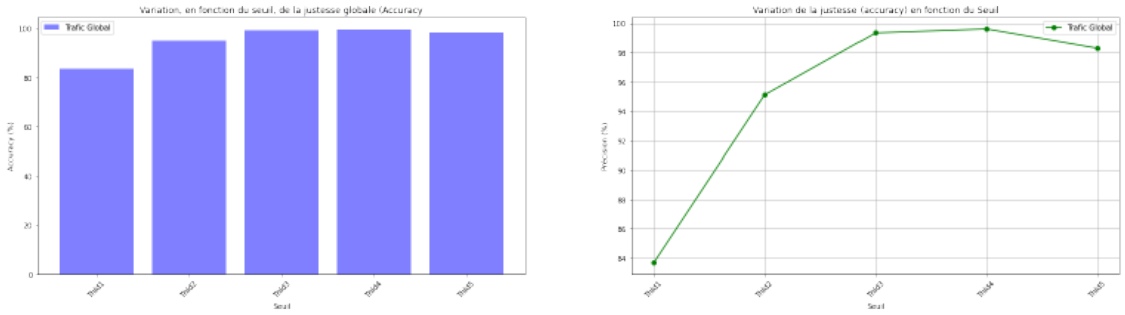


(a) Histogramme des précisions en fonction du seuil

(b) Courbes de variations des précisions en fonction du seuil

Figure 5.7 – MODEL 1 : Taux de détection du trafic normal et du trafic malicieux, en fonction du seuil

Le Tableau5.2 offre une analyse détaillée des performances du modèle d'auto-encodeur appelé $MODEL\ 1$, évaluées à travers l'utilisation de cinq seuils distincts, désignés de $Th1$ à $Th5$. Ces performances sont évaluées en termes de taux de détection du trafic normal, du trafic malicieux, et de l'exactitude globale (accuracy), toutes exprimées en pourcentage. En ce qui concerne le taux de détection du trafic normal, nous observons une amélioration significative à mesure que le seuil augmente. Initialement, à $Thld1$, le taux de détection est de 82.56%, s'élevant progressivement jusqu'à 99.99% à $Thld5$. Cette tendance indique une efficacité croissante du modèle dans la détection du trafic normal à des seuils plus élevés. En revanche, le taux de détection du trafic malicieux montre une dynamique opposée. Alors qu'à $Thld1$, le taux de détection est de 99.99%, il diminue progressivement pour at-



(a) Histogramme de la justesse (Accuracy) en fonction du seuil

(b) Courbes de variation de la justesse (Accuracy) en fonction du seuil

Figure 5.8 – MODEL 1 : Taux de détection globale du type de trafic (Accuracy), en fonction du seuil

teindre 74.17% à Thld5. Cette diminution suggère que le modèle devient moins précis dans la détection du trafic malicieux à des seuils plus élevés. L'exactitude globale (Accuracy) du modèle suit une tendance similaire à celle du taux de détection du trafic normal, augmentant de $Thld1$ à $Thld5$ et atteignant une valeur maximale de 98.31% à $Thld5$. Cette mesure englobe à la fois la capacité du modèle à détecter le trafic normal et malicieux, ainsi que sa capacité à éviter les faux positifs et faux négatifs.

Les performances du modèle varient considérablement selon le seuil choisi. Bien que des seuils plus élevés améliorent généralement la détection du trafic normal, ils peuvent également compromettre la détection du trafic malicieux, entraînant une diminution de l'accuracy. Par conséquent, le choix du seuil doit être soigneusement équilibré en fonction des priorités de détection du trafic normal et malicieux, ainsi que de la tolérance aux faux positifs et faux négatifs.

La Figure 5.7 se décompose en deux graphiques, la Figure 5.7a et la Figure 5.7b, chacun offrant un aperçu précieux des performances de $MODEL\ 1$. La Figure 5.7a présente un histogramme illustrant le taux de détection du trafic normal et malicieux. Les barres vertes représentent le taux de détection du trafic normal, tandis que les barres rouges représentent le taux de détection du trafic malicieux. On observe une tendance similaire à celle Tableau5.2 : le taux de détection du trafic normal augmente à mesure que le seuil augmente, tandis que le taux de détection du trafic malicieux diminue. La Figure 5.7b présente les courbes de variation du taux de détection du trafic normal et malicieux. La courbe en vert représente le taux de détection du trafic normal, tandis que la courbe en rouge représente le taux de détection du trafic malicieux. Cette visualisation met en évidence la relation inverse entre la détection du trafic normal et malicieux en fonction du seuil. Ces résultats permettent une compréhension visuelle plus approfondie de l'impact du choix du seuil sur les performances du modèle. Elles soulignent l'importance de sélectionner un seuil approprié en fonction des exigences spécifiques de détection.

La figure 5.8 présente deux graphiques, la Figure 5.8a et la Figure 5.8b, offrant un aperçu de la justesse globale (accuracy) du modèle en fonction des différents seuils. La Figure 5.8a est un histogramme montrant la justesse globale pour chaque seuil. Les barres bleues représentent la justesse globale (accuracy) pour chaque seuil. La Figure 5.8b est la courbe de variation de l'accuracy en fonction du seuil. Ces visualisations confirment la tendance observée dans le Tableau 5.2 et fournissent une compréhension visuelle de l'impact du choix du seuil sur l'accuracy globale du modèle. Elles mettent en évidence l'importance de sélectionner un seuil approprié pour atteindre un équilibre optimal entre la détection du trafic normal et malicieux, tout en maximisant l'accuracy globale du modèle. Deux seuils, $Th3$ et $Th4$, se démarquent parmi les autres, offrant les meilleures performances globales.

En définitive, l'analyse approfondie des performances de $(MODEL\ 1)$ à travers le tableau, les figures et les observations souligne l'importance cruciale du choix du seuil dans la détection efficace du trafic normal et malicieux. Alors que des seuils plus élevés améliorent généralement la détection du trafic normal, ils peuvent compromettre la détection du trafic malicieux, entraînant une baisse de l'accuracy globale du modèle. Cette dynamique met en évidence le besoin de trouver un équilibre délicat lors du choix du seuil, en tenant compte des priorités de détection spécifiques et de la tolérance aux faux positifs et faux négatifs. Les deux seuils les plus performants, $Th3$ et $Th4$, se distinguent par leur capacité à fournir les meilleures performances globales, soulignant ainsi leur importance dans le processus de détection. En somme, cette analyse fournit des indications précieuses pour guider la sélection du seuil optimal et maximiser l'efficacité du modèle d'auto-encodeur dans la détection du trafic.

5.4.2 Résultats du niveau 2

Comme explicité dans la section méthodologie, ce palier de classification s'avère crucial dans la détermination de la nature des intrusions en cours, que ce soit une attaque déjà répertoriée ou un schéma d'attaque inédit.

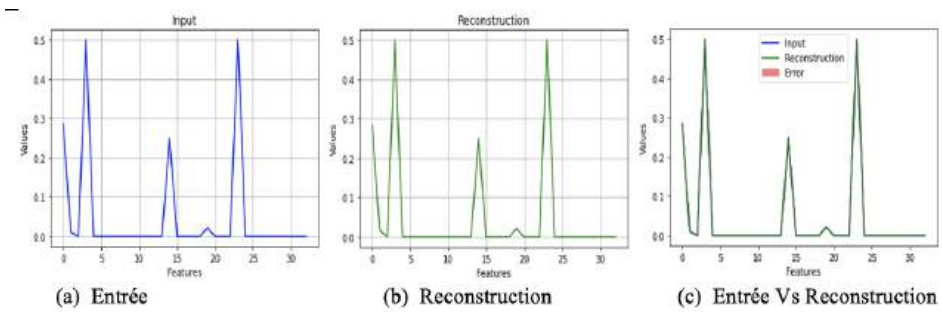
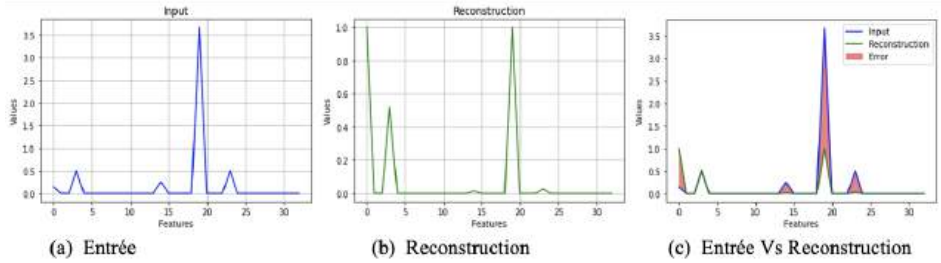


Figure 5.9 – MODEL 2 : Exemple d'attaque connue



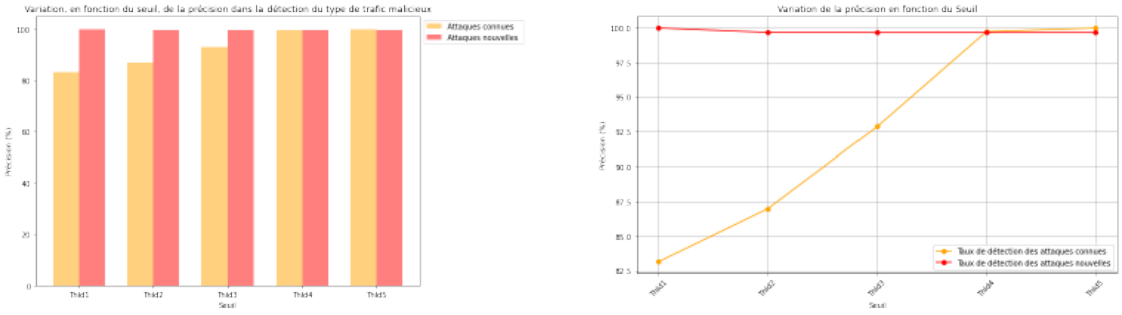
(a) Entrée (b) Reconstruction (c) Entrée Vs Reconstruction

Figure 5.10 – MODEL 2 : Exemple d'attaque nouvelle

Les représentations graphiques fournies dans la Figure 5.9 et la Figure 5.10 illustrent nettement la disparité entre la reconstruction des données correspondant aux attaques connues et celles relatives aux attaques inconnues. Les résultats obtenus révèlent que le modèle parvient à reconstruire de manière précise les entrées associées aux attaques déjà documentées. Cependant, une nette dégradation de la qualité de reconstruction est observée lorsque des les entrées correspondent à de nouveaux types d'attaques, se manifestant par un taux d'erreur significatif. Cette disparité entre les deux scénarios permet d'envisager la définition d'un seuil discriminant, similaire à celui établi au niveau 1, afin d'assurer une prédiction fiable et précise.

Tableau 5.3 – Performances de $MODEL\ 2$ en fonction du seuil

Seuil	Valeur du seuil	Attaques Connues	Attaques nouvelles	Trafic Malicieux Global
Thld1	moyenne$(train_loss) + 0.075 \times$ std$(train_loss)$	83.16%	100%	99.13%
Thld2	moyenne$(train_loss) + 0.125 \times$ std$(train_loss)$	86.98%	99.68%	99.03%
Thld3	moyenne$(train_loss) +$ std$(train_loss)$	92.88%	99.67%	99.32%
Thld4	moyenne$(train_loss) + 7 \times$ std$(train_loss)$	99.74%	99.67%	99.67%
Thld5	moyenne$(train_loss) + 10 \times$ std$(train_loss)$	99.99%	99.67%	99.69%

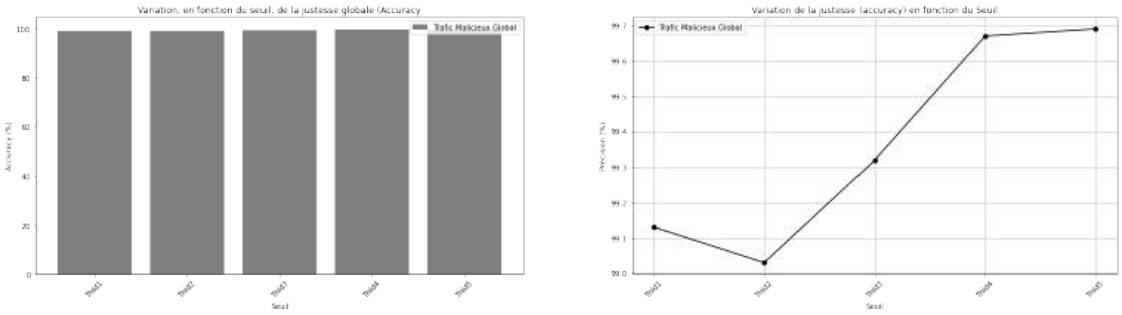


(a) Histogramme des précisions en fonction du seuil (b) Variations des précisions en fonction du seuil

Figure 5.11 – MODEL 2 : Taux de détection des attaque connues et des attaques nouvelles, en fonction du seuil

Pour évaluer les performances du modèle d'auto-encodeur $MODEL\ 2$ au Niveau 2, une série de tests ont été effectués en utilisant cinq seuils différents, tout comme au Niveau 1.

Le Tableau 5.3 révèle plusieurs aspects cruciaux de la capacité du modèle à détecter



(a) Histogramme de la justesse (Accuracy) en fonction du seuil

(b) Variation de la justesse (Accuracy) en fonction du seuil

Figure 5.12 – MODEL 2 : Taux de détection globale des deux types d'attaques (Accuracy), en fonction du seuil

les attaques. Tout d'abord, en examinant le taux de détection des attaques connues, nous constatons une amélioration significative à mesure que le seuil augmente. Cette tendance suggère une meilleure adaptation du modèle aux attaques déjà répertoriées lorsqu'il fonctionne à des seuils plus élevés. En revanche, le taux de détection des attaques nouvelles reste relativement stable pour tous les seuils testés, ce qui indique une certaine robustesse du modèle face à des attaques inédites.

Un aspect important à considérer est la justesse globale (accuracy) du modèle, qui combine sa capacité à détecter à la fois les attaques connues et nouvelles, tout en minimisant les faux positifs et faux négatifs. Comme prévu, l'accuracy augmente avec l'augmentation du seuil, atteignant un maximum de 99.69% à Thld5. Cette observation souligne l'importance du réglage du seuil pour maximiser la performance globale du modèle.

la Figure 5.11 complète l'analyse du Tableau 5.3 en offrant une représentation visuelle claire des performances du modèle en fonction du seuil. La Figure 5.11a présente un histogramme comparant le taux de détection des attaques connues et nouvelles pour chaque seuil testé. Les barres orange représentent le taux de détection des attaques connues, tandis que les barres rouges représentent celui des attaques nouvelles. Cette visualisation confirme les observations du Tableau 5.3, mettant en évidence l'augmentation du taux de détection des attaques connues avec l'augmentation du seuil. La Figure 5.11b offre une perspective supplémentaire en illustrant la variation du taux de détection des attaques connues et nouvelles en fonction du seuil. Les courbes tracées confirment la tendance observée dans le Tableau 5.3 et renforcent ainsi la cohérence des résultats.

La Figure 5.12 complète l'ensemble d'informations en fournissant des détails sur la justesse globale de la détection du trafic malicieux en fonction du seuil. La Figure 5.12a présente un histogramme montrant la justesse globale pour chaque seuil, mettant en évidence une augmentation de cette mesure avec l'augmentation du seuil. La Figure 5.12b offre une

représentation graphique de la variation de la justesse globale en fonction du seuil. Cette courbe confirme la tendance observée dans le tableau et les autres figures, soulignant ainsi l'importance du seuil dans l'optimisation des performances du modèle.

L'ensemble des résultats présentés dans le tableau et les figures fournissent une compréhension approfondie des performances de $MODEL\ 2$, mettant en lumière l'importance du seuil dans la détection efficace des attaques connues et nouvelles.

5.4.3 Résultats du niveau 3

Le modèle intitulé $MODEL\ 3$, mis en œuvre au Niveau 3 de notre Framework de détection d'intrusion, se révèle être un classificateur multi-classe efficace pour identifier les attaques en cours. Basé sur un réseau de neurones artificiels de type MLP, ce modèle offre une couche de détection avancée et spécifique dans notre système de sécurité. Pour évaluer les performances de ce modèle, nous nous sommes appuyés sur la matrice de confusion et le rapport de classification qui fournissent des métriques permettant d'avoir une vue d'ensemble détaillée de la capacité du modèle à classifier les différentes attaques.

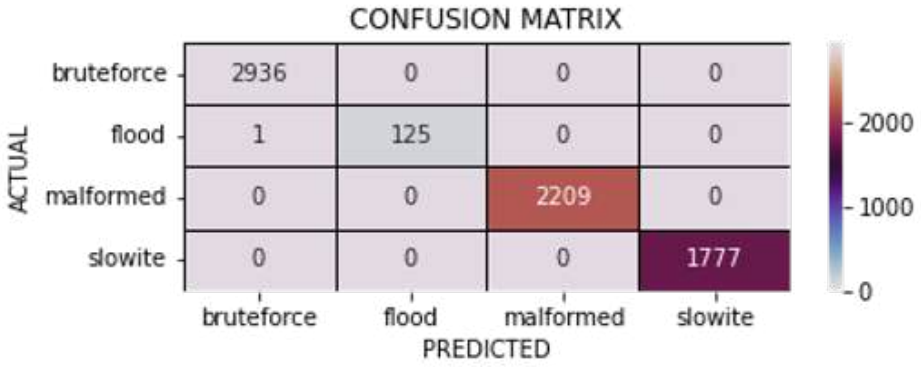


Figure 5.13 – MODEL 3 : Matrice de Confusion

	precision	recall	f1-score	support
Bruteforce	1.00	1.00	1.00	2936
Flood	1.00	0.99	1.00	126
Malformed	1.00	1.00	1.00	2209
Slowite	1.00	1.00	1.00	1777
accuracy			1.00	7048
macro avg	1.00	1.00	1.00	7048
weighted avg	1.00	1.00	1.00	7048

Figure 5.14 – MODEL 3 : Rapport de classification

Sur la Figure 5.13, nous avons la matrice de confusion qui fournit une vue détaillée des prédictions du modèle par rapport aux véritables étiquettes de classe. Dans notre cas, la matrice de confusion révèle que MODEL 3 a réussi à classifier avec une précision remarquable chaque type d'attaque présente dans notre ensemble de données. Les nombres sur la diagonale principale de la matrice indiquent le nombre d'observations correctement classées pour chaque classe, tandis que les zéros en dehors de la diagonale suggèrent une absence de fausses prédictions entre les différentes classes.

Le rapport de classification donné par la Figure 5.14 complète cette évaluation en fournissant des mesures de précision, de rappel et de score F1 pour chaque classe, ainsi que des mesures agrégées telles que l'accuracy, la moyenne des scores F1 et d'autres statistiques macro et pondérées. Dans notre cas, le rapport de classification démontre des scores parfaits de précision, de rappel et de score F1 pour chaque classe, ce qui témoigne de la capacité exceptionnelle de $MODEL\ 3$ à discriminer efficacement entre les différentes attaques. De plus, l'accuracy globale de 100% indique que le modèle a réussi à classifier correctement toutes les instances de notre ensemble de données, confirmant ainsi sa fiabilité et son efficacité.

Les performances de $MODEL\ 3$ sont remarquables, avec une capacité exceptionnelle à identifier précisément chaque type d'attaque dans un réseau IoT. Sa robustesse et sa précision en font un élément essentiel de notre système de détection d'intrusion, offrant une protection avancée contre les menaces potentielles et contribuant ainsi à renforcer la sécurité globale du réseau.

5.5 Conclusion

Face à la nécessité croissante de sécuriser les réseaux IoT contre les menaces d'intrusion, ce chapitre propose une approche novatrice reposant sur un système de détection d'intrusion à trois niveaux basé sur le deep learning. En se fondant sur les principes de l'apprentissage non supervisé et supervisé, ce système combine l'efficacité des auto-encodeurs pour la détection d'anomalies avec la capacité des réseaux de neurones artificiels à classifier les attaques spécifiques.

L'évaluation approfondie des performances de chaque niveau révèle une efficacité remarquable dans la détection des attaques connues et émergentes, tout en minimisant les faux positifs et faux négatifs. Les modèles d'auto-encodeurs implantés aux niveaux 1 et 2 ont démontré une capacité précise à identifier les anomalies dans le trafic réseau, distinguant avec succès les schémas de comportement normal des activités malveillantes. De plus, le modèle de classification multi-classes au niveau 3 a exhibé une précision remar-

quable dans l'identification des différentes catégories d'attaques, offrant ainsi une solution complète pour renforcer la sécurité des réseaux IoT. En utilisant un ensemble de données diversifié et représentatif des scénarios d'attaques IoT, ce chapitre met en évidence la robustesse et la généralisabilité du système proposé. Les résultats obtenus suggèrent que cette approche à trois niveaux basée sur le deep learning peut être une stratégie prometteuse pour sécuriser efficacement les réseaux IoT contre un large éventail de menaces d'intrusion, y compris les attaques zero-day.

En somme, ce chapitre offre une contribution significative à la recherche sur la sécurité des systèmes IoT en proposant une approche innovante et efficace pour la détection d'intrusion. Les perspectives futures pourraient inclure l'exploration de techniques d'optimisation supplémentaires pour améliorer les performances du système, ainsi que son déploiement et son évaluation dans des environnements réels pour valider sa robustesse et sa viabilité opérationnelle.

Conclusion Générale et Perspectives

L'Internet des Objets (IoT) a redéfini les paradigmes technologiques et industriels, offrant des solutions innovantes et ouvrant de nouvelles perspectives pour l'automatisation et l'efficacité opérationnelle. Cette thèse a exploré en profondeur les multiples facettes de l'IoT, ses applications et les défis de sécurité qui lui sont associés. À travers cinq chapitres, une analyse détaillée a été réalisée pour mettre en lumière l'importance de l'IoT, l'application de l'intelligence artificielle en cybersécurité, la classification des cyber-attaques, l'utilisation des autoencodeurs pour la détection d'anomalies, et enfin, la proposition d'un système de détection d'intrusion à trois niveaux basé sur le deep learning.

Le premier chapitre a fourni un panorama complet des applications de l'IoT et de son architecture complexe. Il a révélé comment l'IoT impacte des secteurs clés tels que la santé, l'industrie, les villes intelligentes, l'agriculture, la gestion de l'énergie et le transport, en optimisant les processus et en améliorant la prise de décision en temps réel. Cependant, ce potentiel immense s'accompagne de défis majeurs en matière de sécurité, soulignant la nécessité de mesures de protection rigoureuses et de stratégies de sécurité robustes pour garantir la fiabilité et la durabilité de ces technologies. L'application de l'intelligence artificielle (IA) en cybersécurité, abordée dans le deuxième chapitre, a démontré son rôle transformationnel dans la sécurisation des systèmes IoT. Les technologies de machine learning et de deep learning ont prouvé leur capacité à détecter rapidement et précisément les anomalies et à classifier de manière proactive les menaces potentielles. Malgré les défis liés à leur mise en œuvre, tels que la nécessité de modèles légers et la protection des données sensibles, les avancées dans l'apprentissage non supervisé et les réseaux neuronaux offrent des solutions adaptatives et efficaces pour renforcer la sécurité des systèmes IoT. Le troisième chapitre a mis en avant la classification des cyber-attaques dans les réseaux IoT en utilisant des algorithmes de machine learning et de deep learning. L'évaluation des modèles sur un dataset IoT Botnet a démontré l'efficacité de ces approches pour détecter et classer les attaques. Les résultats ont souligné la précision et l'efficacité des modèles de classification, fournissant des insights précieux pour le choix des algorithmes appropriés pour différents types de cyber-attaques. L'utilisation des autoencodeurs pour la détection d'anomalies, explorée dans le quatrième chapitre, a montré que ces outils sont puissants pour identifier des schémas anormaux dans les données, indicateurs potentiels d'activités malveillantes. Les variations des seuils de décision permettent d'ajuster la sensibilité de la détection en fonction des besoins spécifiques du système de sécurité, offrant une flexibilité précieuse pour optimiser la détection des anomalies.Enfin, le cinquième chapitre a proposé un système de

détection d'intrusion à trois niveaux basé sur le deep learning. Cette approche novatrice combine l'efficacité des autoencodeurs pour la détection d'anomalies avec la capacité des réseaux de neurones artificiels à classifier les attaques spécifiques. L'évaluation des performances du système a révélé une efficacité remarquable dans la détection des attaques connues et émergentes, tout en minimisant les faux positifs et faux négatifs. Cette contribution significative à la recherche sur la sécurité des systèmes IoT offre des perspectives prometteuses pour le déploiement de solutions robustes et efficaces.

Les résultats obtenus dans cette thèse ont des implications pratiques et théoriques majeures. Ils démontrent le potentiel immense de l'IA et des techniques de machine learning et de deep learning pour la sécurité des réseaux IoT. Ces technologies permettent de renforcer la sécurité de manière proactive et adaptative, en analysant les flux de données en temps réel et en détectant précocement les menaces. Pour l'avenir, ces modèles seront aussi testés sur des jeux de données différents pour mesurer leur robustesse. Il est essentiel de continuer à innover et à améliorer les modèles d'apprentissage pour s'adapter aux nouvelles menaces et aux évolutions technologiques. Le déploiement pratique et l'évaluation des solutions proposées dans des environnements réels sont cruciaux pour valider leur robustesse et leur viabilité opérationnelle. Des tests sur le terrain permettront de mesurer l'efficacité des approches combinées et d'identifier les améliorations possibles. L'IoT continue de croître à un rythme rapide, et les défis de sécurité nécessitent une vigilance continue et une innovation constante. Les chercheurs et les praticiens doivent collaborer pour développer des solutions de sécurité avancées qui peuvent s'adapter aux évolutions technologiques et aux nouvelles menaces. L'intégration des technologies SDN (Software-Defined Networking) et leur combinaison avec les méthodes d'apprentissage automatique et profond offre une gestion plus flexible et dynamique des incidents de sécurité, facilitant l'application rapide et adaptative des politiques de sécurité.

Liste des Publications

Cette thèse a été réalisée dans le cadre du projet IoT-SDC [2](IoT Smart Data classifier). Le projet est composé de deux thèmes. Le premier est celui qui est développé dans cette thèse. Le deuxième est intitulé : *Nez électronique et intelligence artificielle pour la caractérisation sensorielle des tomates locales.* Il est aussi développé dans une autre thèse. Ce projet a été financé par le DSTN (Digital Science and Technology Network) [3] qui est une communauté d'experts scientifiques en sciences des données, intelligence artificielle, calcul haute performance, cybersécurité, internet des objets, qui souhaite relever les défis posés par la transformation numérique pour un développement durable en Afrique.

Conférences

— **A. Diallo**, L. Affognon, C. Diallo and E. C. Ezin, *"A Three-Level Deep Learning Intrusion Detection System for IoT Network,"* 2023 International Conference on Electrical, Communication and Computer Engineering (ICECCE), Dubai, United Arab Emirates, 2023, pp. 1-6, doi : 10.1109/ICECCE61019.2023.10442798.

— **A. Diallo**, L. Affognon, C. Diallo, and E.C. Ezin, *"Towards the implementation of a dynamic IDS for IoT : Anomaly detection in MQTT traffic."* on 13th Conference on Research in Computer Science and its Applications, CNRIA 2023. May 25-27 2023.

— **A. Diallo**, L. Affognon, C. Diallo and E. C. Ezin, *"Deep Learning Based Binary and Multi-class Classification Comparison for Anomaly Detection,"* 2022 International Conference on Engineering and Emerging Technologies (ICEET), Kuala Lumpur, Malaysia, 2022, pp. 1-6, doi : 10.1109/ICEET56468.2022.10007171.

— **A. Diallo** and C. Diallo, *"Human Activity Recognition in Smart Home using Deep Learning Models,"* 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2021, pp. 1511-1515, doi : 10.1109/CSCI54926.2021.00294.

2. https://ace-partner.org/dstn/projet/iot-sdc
3. https://ace-partner.org/dstn/

Posters

— **A. Diallo**, L. Affognon, C. Diallo, and E.C. Ezin, *"Autoencoders for Intrusion Detection in IoT Networks"*, présenté aux Journées Scientifiques de l'UFR des Sciences Appliquées et de Technologie, Université Gaston Berger, Saint-Louis, Sénégal, dans les locaux du CEA-MITIC, les 11 et 12 mai 2023.

Publications dans le cadre du projet IoT-SDC

— L. Affognon, **A. Diallo** Diallo, Chérif Ezin, Eugène, *"Design of an Experimental Electronic Nose for Data Collection for Food Quality,"*IEEE EUROCON 2023 - 20th International Conference on Smart Technologies, Torino, Italy, 2023, pp. 406-411, doi : 10.1109/EUROCON56442.2023.10199067.

— Affognon, Lionel **Diallo, Abdoulaye** Diallo, Chérif Ezin, Eugène. (2023). *A Survey on Statistical and Machine Learning Algorithms Used in Electronic Noses for Food Quality Assessment.* SN Computer Science. 4. 10.1007/s42979-023-02052-0.

— L. Affognon, **A. Diallo** Diallo, Chérif Ezin, Eugène, *"Supervised Learning Models for Tomato Quality Prediction Using Electronic Nose Data,"* 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa, 2023, pp. 1-6, doi : 10.1109/ICECET58911.2023.10389485.

— Affognon, Lionel Ezin, Eugène **Diallo, Abdoulaye** Diallo, Chérif. (2023). *Electronic Nose Architecture for Tomato Data Collection.* 138-155. 10.1007/978-3-031-47457-6_9.

Bibliographie

Abir, S. M. Abu Adnan et al. (2021). "IoT-Enabled Smart Energy Grid: Applications and Challenges". In : *IEEE Access* 9, p. 50961-50981. doi : 10.1109/ACCESS.2021.3067331.

Abomhara, Mohamed, Geir Køien et Mohammed Alghamdi (fév. 2021). *Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks.*

Abu Al-Haija, Qasem (2022). "Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks". In : *Frontiers in Big Data* 4. issn : 2624-909X. doi : 10.3389/fdata.2021.782902. url : https://www.frontiersin.org/articles/10.3389/fdata.2021.782902.

Aggarwal, Alankrita, Mamta Mittal et Gopi Battineni (2021). "Generative adversarial network: An overview of theory and applications". In : *International Journal of Information Management Data Insights* 1.1, p. 100004. issn : 2667-0968. doi : https://doi.org/10.1016/j.jjimei.2020.100004. url : https://www.sciencedirect.com/science/article/pii/S2667096820300045.

Ahmad, Tanveer et Dongdong Zhang (2021). "Using the internet of things in smart energy systems and networks". In : *Sustainable Cities and Society* 68, p. 102783.

Ahmad, Zeeshan et al. (2021). "Anomaly Detection Using Deep Neural Network for IoT Architecture". In : *Applied Sciences* 11.15. issn : 2076-3417. doi : 10.3390/app11157050. url : https://www.mdpi.com/2076-3417/11/15/7050.

Ahmed, Eshtiak et al. (2020). "Internet of Things (IoT): Vulnerabilities, Security Concerns and Things to Consider". In : *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, p. 1-6. doi : 10.1109/ICCCNT49239.2020.9225283.

Akpakwu, Godfrey Anuga et al. (2018). "A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges". In : *IEEE Access* 6, p. 3619-3647. doi : 10.1109/ACCESS.2017.2779844.

Aksoy, Ahmet et Mehmet Hadi Gunes (2019). “Automated IoT Device Identification using Network Traffic”. In : *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, p. 1-7. doi : 10.1109/ICC.2019.8761559.

Alom, Md. Zahangir et al. (mars 2018). “The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches”. In.

Alzubaidi, Laith et al. (2021). “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”. In : *Journal of big Data* 8, p. 1-74.

Anand, Pooja et al. (2020). “IoVT: Internet of Vulnerable Things? Threat Architecture, Attack Surfaces, and Vulnerabilities in Internet of Things and Its Applications towards Smart Grids”. In : *Energies* 13.18, p. 4813.

Andrea, Ioannis, Chrysostomos Chrysostomou et George Hadjichristofi (2015). “Internet of Things: Security vulnerabilities and challenges”. In : *2015 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, p. 180-187.

Anthi, Eirini et al. (2019). “A Supervised Intrusion Detection System for Smart Home IoT Devices”. In : *IEEE Internet of Things Journal* 6.5, p. 9042-9053. doi : 10.1109/JIOT.2019.2926365.

Apostol, Ioana et al. (2021a). “IoT Botnet Anomaly Detection Using Unsupervised Deep Learning”. In : *Electronics* 10.16. issn : 2079-9292. doi : 10.3390/electronics10161876. url : https://www.mdpi.com/2079-9292/10/16/1876.

— (2021b). “IoT Botnet Anomaly Detection Using Unsupervised Deep Learning”. In : *Electronics* 10.16. issn : 2079-9292. doi : 10.3390/electronics10161876. url : https://www.mdpi.com/2079-9292/10/16/1876.

Asghari, Parvaneh, Amir Masoud Rahmani et Hamid Haj Seyyed Javadi (2019). “Internet of Things applications: A systematic review”. In : *Computer Networks* 148, p. 241-261.

Ashiku, Lirim et C.H. Dagli (juin 2021). “Network Intrusion Detection System using Deep Learning”. In : *Procedia Computer Science* 185, p. 239-247. doi : 10.1016/j.procs.2021.05.025.

Ashraf, Javed et al. (juill. 2020). “A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions”. In : *Electronics* 9. doi : 10.3390/electronics9071177.

B, Delipetrev, Tsinaraki C et Kostic U (2020). “Historical Evolution of Artificial Intelligence”. In : KJ-NA-30221-EN-N (online). ISSN : 1831-9424 (online). DOI : 10.2760/801580(online).

Bai, Lei et al. (2018). “Automatic device classification from network traffic streams of internet of things”. In : *2018 IEEE 43rd conference on local computer networks (LCN)*. IEEE, p. 1-9.

Baker, Stephanie B., Wei Xiang et Ian Atkinson (2017). “Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities”. In : *IEEE Access* 5, p. 26521-26544. DOI : 10.1109/ACCESS.2017.2775180.

Baker, Stephen (2011). *Final Jeopardy: The story of Watson, the computer that will transform our world*. HMH.

Banu, Masoodhu et Sujatha C (nov. 2017). “IoT Architecture a Comparative Study”. In : DOI : 10.12732/ijpam.v117i8.10.

Bassene, Avewe et Bamba Gueye (2021). “A group-based IoT devices classification through network traffic analysis based on machine learning approach”. In : *Towards new e-Infrastructure and e-Services for Developing Countries: 12th EAI International Conference, AFRICOMM 2020, Ebène City, Mauritius, December 2-4, 2020, Proceedings 12*. Springer, p. 185-202.

Bauer, Martin et al. (2013). “IoT Reference Model”. In : *Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model*. Sous la dir. d’Alessandro Bassi et al. Berlin, Heidelberg : Springer Berlin Heidelberg, p. 113-162. ISBN : 978-3-642-40403-0. DOI : 10.1007/978-3-642-40403-0_7. URL : https://doi.org/10.1007/978-3-642-40403-0_7.

Bengio, Yoshua, Nicholas Léonard et Aaron Courville (2013). “Estimating or propagating gradients through stochastic neurons for conditional computation”. In : *arXiv preprint arXiv:1308.3432*.

Berahmand, Kamal et al. (2024). “Autoencoders and their applications in machine learning: a survey”. In : *Artificial Intelligence Review* 57.2, p. 28.

Borkar, Suresh R. (2020). “7 - Long-term evolution for machines (LTE-M)”. In : *LPWAN Technologies for IoT and M2M Applications*. Sous la dir. de Bharat S. Chaudhari et Marco Zennaro. Academic Press, p. 145-166. ISBN : 978-0-12-818880-4. DOI : https:

//doi.org/10.1016/B978-0-12-818880-4.00007-7. URL : https://www.sciencedirect.com/science/article/pii/B9780128188804000077.

Campbell, Murray, A Joseph Hoane Jr et Feng-hsiung Hsu (2002). "Deep blue". In : *Artificial intelligence* 134.1-2, p. 57-83.

Carleo, Giuseppe et al. (déc. 2019). "Machine learning and the physical sciences". In : *Reviews of Modern Physics* 91. DOI : 10.1103/RevModPhys.91.045002.

Cekerevac, Zoran et al. (2017). "Internet of Things and the Man-in-the-Middle Attacks – Security and Economic Risks". In : *MEST Journal* 5.2, p. 15-25. DOI : 10.12709/mest.05.05.02.03.

Chaabouni, Nadia et al. (2019). "Network Intrusion Detection for IoT Security Based on Learning Techniques". In : *IEEE Communications Surveys Tutorials* 21.3, p. 2671-2701. DOI : 10.1109/COMST.2019.2896380.

Charbuty, Bahzad et Adnan Abdulazeez (2021). "Classification based on decision tree algorithm for machine learning". In : *Journal of Applied Science and Technology Trends* 2.01, p. 20-28.

Chen, Jim X (2016). "The evolution of computing: AlphaGo". In : *Computing in Science & Engineering* 18.4, p. 4-7.

Chen, Mingzhe et al. (2019). "Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial". In : *IEEE Communications Surveys Tutorials* 21.4, p. 3039-3071. DOI : 10.1109/COMST.2019.2926625.

Chen, Shuangshuang et Wei Guo (2023). "Auto-Encoders in Deep Learning—A Review with New Perspectives". In : *Mathematics* 11.8. ISSN : 2227-7390. DOI : 10.3390/math11081777. URL : https://www.mdpi.com/2227-7390/11/8/1777.

Chung, Junyoung et al. (déc. 2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In.

Ciklabakkal, Ege et al. (2019). "ARTEMIS: An Intrusion Detection System for MQTT Attacks in Internet of Things". In : *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, p. 369-3692. DOI : 10.1109/SRDS47363.2019.00053.

Cultice, Tyler, Dan Ionel et Himanshu Thapliyal (2020). “Smart Home Sensor Anomaly Detection Using Convolutional Autoencoder Neural Network”. In : *2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, p. 67-70. doi : 10.1109/iSES50453.2020.00026.

Czarnowski, Ireneusz et Piotr Jędrzejowicz (2021). “Supervised Classification Problems–Taxonomy of Dimensions and Notation for Problems Identification”. In : *IEEE Access* 9, p. 151386-151400. doi : 10.1109/ACCESS.2021.3125622.

Dawoud, Ahmed et al. (2020a). “Internet of Things Intrusion Detection: A Deep Learning Approach”. In : *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, p. 1516-1522. doi : 10.1109/SSCI47803.2020.9308293.

Dawoud, Ahmed et al. (déc. 2020b). “Internet of Things Intrusion Detection: A Deep Learning Approach”. In : p. 1516-1522. doi : 10.1109/SSCI47803.2020.9308293.

Dawoud, Ahmed et al. (2020c). “Internet of Things Intrusion Detection: A Deep Learning Approach”. In : *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, p. 1516-1522. doi : 10.1109/SSCI47803.2020.9308293.

De Resende, Adriano AM et al. (2022). “Traffic Classification of Home Network Devices using Supervised Learning.” In : *ICAART (3)*, p. 114-120.

Diallo, A. et al. (2023a). “Towards the implementation of a dynamic IDS for IoT: Anomaly detection in MQTT traffic”. In : *Research in Computer Science and Its Applications: 13th Conference on Research in Computer Science and its Applications, CNRIA 2023.*

Diallo, Abdoulaye et al. (2022). “Deep Learning Based Binary and Multi-class Classification Comparison for Anomaly Detection”. In : *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, p. 1-6. doi : 10.1109/ICEET56468.2022.10007171.

Diallo, Abdoulaye et al. (2023b). “A Three-Level Deep Learning Intrusion Detection System for IoT Network”. In : *2023 International Conference on Electrical, Communication and Computer Engineering (ICECCE)*, p. 1-6. doi : 10.1109/ICECCE61019.2023.10442798.

Digital Matter (2022). *Connectivity LTE-M vs NB-IoT: Coverage, Providers, and Roaming*. Accessed: 2022-06-01. url : https://support.digitalmatter.com/en_

US/networks-and-sim-card-options/connectivity-lte-m-vs-nb-iot-coverage-providers-and-roaming.

Ding, Jie et al. (2020). "IoT Connectivity Technologies and Applications: A Survey". In : *IEEE Access* 8, p. 67646-67673. doi : 10.1109/ACCESS.2020.2985932.

Drăgulinescu, Ana Maria Claudia et al. (2020). "LoRa-based medical IoT system architecture and testbed". In : *Wireless Personal Communications*, p. 1-23.

El Hakim, Ahmed (2018). "Internet of things (IoT) system architecture and technologies". In : *White Paper* 10, p. 1-5.

Elsayed, Nelly et al. (2021). "Intrusion Detection System in Smart Home Network Using Bidirectional LSTM and Convolutional Neural Networks Hybrid Model". In : *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, p. 55-58. doi : 10.1109/MWSCAS47672.2021.9531683.

G. Uganya, Radhika Baskar et N. Vijayaraj (2021). "A Survey on Internet of Things: Applications, Recent Issues, Attacks, and Security Mechanisms". In : *Journal of Circuits, Systems and Computers*. doi : 10.1142/S0218126621300063.

Gamundani, Attlee M. (2015). "An impact review on Internet of Things attacks". In : *2015 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*. IEEE, p. 110-115.

Ge, Mengmeng et al. (2019). "Deep Learning-Based Intrusion Detection for IoT Networks". In : *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, p. 256-25609. doi : 10.1109/PRDC47002.2019.00056.

GeeksforGeeks (2022). *5-Layer Architecture of Internet of Things*. Accessed: 2022-06-01. url : https://www.geeksforgeeks.org/5-layer-architecture-of-internet-of-things/.

Gondchawar, Nikesh, RS Kawitkar et al. (2016). "IoT based smart agriculture". In : *International Journal of advanced research in Computer and Communication Engineering* 5.6, p. 838-842.

Goodfellow, Ian, Yoshua Bengio et Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Goodfellow, Ian J. et al. (2014). *Generative Adversarial Networks*. arXiv : 1406.2661 [stat.ML].

Graves, Alex (août 2013). “Generating Sequences With Recurrent Neural Networks”. In.

GridSearchCV in scikit-learn: A Comprehensive Guide (s. d.). Consulté le 15 juin 2023. URL : https://dev.to/anurag629/gridsearchcv-in-scikit-learn-a-comprehensive-guide-2a72.

Guo, Yifan et al. (oct. 2020). “Unsupervised Anomaly Detection in IoT Systems for Smart Cities”. In : *IEEE Transactions on Network Science and Engineering* 7, p. 2231-2242. DOI : 10.1109/TNSE.2020.3027543.

Gurung, Sandeep, Mirnal Kanti Ghose et Aroj Subedi (2019). “Deep learning approach on network intrusion detection system using NSL-KDD dataset”. In : *International Journal of Computer Network and Information Security* 11.3, p. 8-14.

Habibi Lashkari, Arash et al. (jan. 2017). “Characterization of Tor Traffic using Time based Features”. In : p. 253-262. DOI : 10.5220/0006105602530262.

Hamid, Harth Ghassan et Zainab T Alisa (2021). “Survey on IoT application layer protocols”. In : *Indonesian Journal of Electrical Engineering and Computer Science* 21.3, p. 1663-1672.

Hastie, Trevor, Robert Tibshirani et Jerome Friedman (fév. 2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*. ISBN : 0387848576.

Herrero, Rolando (2022). *Fundamentals of IoT communication technologies*. Springer.

Hindy, Hanan et al. (2020). *MQTT-IoT-IDS2020: MQTT Internet of Things Intrusion Detection Dataset*. DOI : 10.21227/bhxy-ep04. URL : https://dx.doi.org/10.21227/bhxy-ep04.

Hindy, Hanan et al. (jan. 2021). “Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset)”. In : p. 73-84. ISBN : 978-3-030-64757-5. DOI : 10.1007/978-3-030-64758-2_6.

Hinton, G E et R R Salakhutdinov (juill. 2006). “Reducing the dimensionality of data with neural networks”. In : *Science* 313.5786, p. 504-507. DOI : 10.1126/science.1127647. URL : http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed&uid=16873662&cmd=showdetailview&indexed=google.

Hochreiter, Sepp et Jürgen Schmidhuber (1997). “Long short-term memory”. In : *Neural computation* 9.8, p. 1735-1780.

Hölldobler, Steffen, Sibylle Möhle et Anna Tigunova (2017). “Lessons Learned from AlphaGo.” In : *YSIP*, p. 92-101.

Huc, Aleks et Denis Trcek (avr. 2021). “Anomaly Detection in IoT Networks: From Architectures to Machine Learning Transparency”. In : *IEEE Access* PP, p. 1-1. doi : 10.1109/ACCESS.2021.3073785.

Hwang, Ren-Hung et al. (fév. 2020). “An Unsupervised Deep Learning Model for Early Network Traffic Anomaly Detection”. In : *IEEE Access* 8, p. 1-1. doi : 10.1109/ACCESS.2020.2973023.

Hyperparameter Tuning with GridSearchCV (s. d.). Consulté le 15 juin 2023. url : https://www.mygreatlearning.com/blog/gridsearchcv.

Ibrahim, Heba Ezzat, Sherif M. Badr et Mohamed A. Shaheen (2012). *Adaptive Layered Approach using Machine Learning Techniques with Gain Ratio for Intrusion Detection Systems*. arXiv : 1210.7650 [cs.CR].

Infosec, SISA (2023). *The OWASP IoT Top 10 Vulnerabilities and How to Mitigate Them*. Accessed: 13 juillet 2023, Dernière MAJ: 17 Mai 2023. url : https://www.sisainfosec.com/blogs/the-owasp-iot-top-10-vulnerabilities-and-how-to-mitigate-them/.

Iqbal, Mehzabien, Abu Abdullah et Farzana Shabnam (jan. 2020). “An Application Based Comparative Study of LPWAN Technologies for IoT Environment”. In : p. 1857-1860. doi : 10.1109/TENSYMP50017.2020.9230597.

Iqbal, Talha et Hazrat Ali (2018). “Generative adversarial network for medical images (MI-GAN)”. In : *Journal of medical systems* 42.11, p. 231.

Jabraeil Jamali, Mohammad Ali et al. (2020). “IoT Architecture”. In : *Towards the Internet of Things: Architectures, Security, and Applications*. Cham : Springer International Publishing, p. 9-31. isbn : 978-3-030-18468-1. doi : 10.1007/978-3-030-18468-1_2. url : https://doi.org/10.1007/978-3-030-18468-1_2.

Jha, Anupama, Meenu Dave et Supriya Madan (jan. 2019). “Comparison of Binary Class and Multi-Class Classifier Using Different Data Mining Classification Techniques”. In : *SSRN Electronic Journal*. doi : 10.2139/ssrn.3464211.

Jmila, Houda et al. (2022). “A Survey of Smart Home IoT Device Classification Using Machine Learning-Based Network Traffic Analysis”. In : *IEEE Access* 10, p. 97117-97141. doi : 10.1109/ACCESS.2022.3205023.

Jurcut, Anca et al. (juin 2020). “Security Considerations for Internet of Things: A Survey”. In : *SN Computer Science* 1. doi : 10.1007/s42979-020-00201-3.

Kabir, Md et Xiao Luo (août 2020). “Unsupervised Learning for Network Flow Based Anomaly Detection in the Era of Deep Learning”. In : p. 165-168. doi : 10.1109/BigDataService49289.2020.00032.

Karagiannis, Vasileios et al. (2015). “A survey on application layer protocols for the internet of things”. In : *Transaction on IoT and Cloud computing* 3.1, p. 11-17.

Kassim, Mohamed Rawidean Mohd (2020). “Iot applications in smart agriculture: Issues and challenges”. In : *2020 IEEE conference on open systems (ICOS)*. IEEE, p. 19-24.

Khan, Akram et al. (2018). “Design of an IoT smart home system”. In : *2018 15th Learning and Technology Conference (L&T)*. IEEE, p. 1-5.

Khan, Muhammad Almas et al. (oct. 2021). “A Deep Learning-Based Intrusion Detection System for MQTT Enabled IoT”. In : *Sensors* 21. doi : 10.3390/s21217016.

Khanna, Abhishek et Sanmeet Kaur (2020). “Internet of things (IoT), applications and challenges: a comprehensive review”. In : *Wireless Personal Communications* 114, p. 1687-1762.

Kingma, Diederik P et Max Welling (2022). *Auto-Encoding Variational Bayes*. arXiv : 1312.6114 [stat.ML].

Kizza, Joseph (jan. 2009). *A Guide to Computer Network Security*. isbn : 978-1-84800-916-5. doi : 10.1007/978-1-84800-917-2.

Koroniotis, Nickolaos et al. (nov. 2018). *Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset*.

Kulkarni, Vrushali Y et Pradeep K Sinha (2013). “Random forest classifiers: a survey and future research directions”. In : *Int. J. Adv. Comput* 36.1, p. 1144-1153.

Lampropoulos, Georgios, Kerstin Siakas et Theofylaktos Anastasiadis (2019). “Internet of things in the context of industry 4.0: An overview”. In : *International Journal of Entrepreneurial Knowledge*, p. 4-19.

LeCun, Yann, Yoshua Bengio et Geoffrey Hinton (mai 2015). “Deep learning”. In : *nature* 521.7553, p. 436-444. doi : 10.1038/nature14539.

Lee, Minhyeok et Junhee Seok (2019). “Controllable generative adversarial network”. In : *Ieee Access* 7, p. 28158-28169.

Legrand, Adrien et al. (déc. 2018a). “Study of Autoencoder Neural Networks for Anomaly Detection in Connected Buildings”. In : p. 1-5. doi : 10.1109/GCIoT.2018.8620158.

— (2018b). “Study of Autoencoder Neural Networks for Anomaly Detection in Connected Buildings”. In : *2018 IEEE Global Conference on Internet of Things (GCIoT)*, p. 1-5. doi : 10.1109/GCIoT.2018.8620158.

Lemeš, Samir (oct. 2023). “The Role of Software Engineering in Industry 4.0”. In : doi : 10.5644/PI2023.209.05.

Lewandowska, Natalia (jan. 2024). *Intrusion Detection Systems: Categories, attack detection and response.* doi : 10.20944/preprints202402.0008.v1.

Li, Pengzhi, Yan Pei et Jianqiang Li (2023). “A comprehensive survey on design and application of autoencoder in deep learning”. In : *Applied Soft Computing* 138, p. 110176. issn : 1568-4946. doi : https://doi.org/10.1016/j.asoc.2023.110176. url : https://www.sciencedirect.com/science/article/pii/S1568494623001941.

Lighthill, James (1973). “Artificial Intelligence: A General Survey”. In : *Artificial Intelligence: A paper symposium*. Science Research Council. UK.

Lin, Peng, Kejiang Ye et Cheng-Zhong Xu (juin 2019). “Dynamic Network Anomaly Detection System by Using Deep Learning Techniques”. In : p. 161-176. isbn : 978-3-030-23501-7. doi : 10.1007/978-3-030-23502-4_12.

Lindemann, Benjamin et al. (2021). “A survey on long short-term memory networks for time series prediction”. In : *Procedia CIRP* 99. 14th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 15-17 July 2020, p. 650-655. issn : 2212-8271. doi : https://doi.org/10.1016/j.procir.2021.03.088. url : https://www.sciencedirect.com/science/article/pii/S2212827121003796.

Liu, Yanli, Yourong Wang et Jian Zhang (2012). “New machine learning algorithm: Random forest”. In : *Information Computing and Applications: Third International Conference, ICICA 2012, Chengde, China, September 14-16, 2012. Proceedings 3*. Springer, p. 246-252.

Lopez-Martin, Manuel et al. (2017). “Network traffic classifier with convolutional and recurrent neural networks for Internet of Things”. In : *IEEE access* 5, p. 18042-18050.

Mahesh, Batta (jan. 2019). “Machine Learning Algorithms -A Review”. In : doi : 10.21275/ART20203995.

Maimon, Oded Z et Lior Rokach (2014). *Data mining with decision trees: theory and applications*. T. 81. World scientific.

Malgwi, Yusuf, Goni Ibrahim et Mahmud Bamanga Ahmad (jan. 2022). “Artificial Neural Network Model for Intrusion Detection System”. In : *Mediterranean Journal of Basic and Applied Sciences* 06, p. 20-26. doi : 10.46382/MJBAS.2022.6103.

Marian-Vladut, Toma (nov. 2021). “IoT potential in Industry 4.0 and Education 4.0 context”. In : *Journal of Applied Computer Science Mathematics* 15, p. 37-43. doi : 10.4316/JACSM.202102005.

Mayzaud, Anthéa, Rémi Badonnel et Isabelle Chrisment (2016). “A Taxonomy of Attacks in RPL-based Internet of Things”. In : *International journal of network security* 18.3, p. 459-473.

McCarthy, John et al. (2006). “A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955”. In : *AI Magazine* 27.4, p. 12. doi : 10.1609/aimag.v27i4.1904. url : https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1904.

Meira, Jorge et al. (2020). “Performance evaluation of unsupervised techniques in cyber-attack anomaly detection”. In : *Journal of Ambient Intelligence and Humanized Computing* 11.11, p. 4477-4489.

Mishra, Nivedita et Sharnil Pandya (2021). “Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review”. In : *IEEE Access* 9, p. 59353-59373.

Mitchell, Tom M (1997). “Artificial neural networks”. In : *Machine learning* 45.81, p. 127.

Murphy, Kevin (jan. 2012). *Machine Learning: A Probabilistic Perspective*. T. 58.

Nahida Islam Fahiba Farhin, Ishrat Sultana M. Shamim Kaiser Md. Sazzadur Rahman Mufti Mahmud A. S. M. Sanwar Hosen Gi Hwan Cho (2021). "Towards Machine Learning Based Intrusion Detection in IoT Networks". In : *Computers, Materials & Continua* 69.2, p. 1801-1821. ISSN : 1546-2226. DOI : 10.32604/cmc.2021.018466. URL : http://www.techscience.com/cmc/v69n2/43889.

Naseer, Sheraz et al. (2018). "Enhanced Network Anomaly Detection Based on Deep Neural Networks". In : *IEEE Access* 6, p. 48231-48246. DOI : 10.1109/ACCESS.2018.2863036.

Navada, Arundhati et al. (2011). "Overview of use of decision tree algorithms in machine learning". In : *2011 IEEE Control and System Graduate Research Colloquium*, p. 37-42. DOI : 10.1109/ICSGRC.2011.5991826.

Nawir, Mukrimah et al. (2016). "Internet of Things (IoT): Taxonomy of Security Attacks". In : *2016 3rd International Conference on Electronic Design (ICED)*. IEEE, p. 321-326.

Naxoo (2022). *L'Internet des Objets (Internet of Things, IoT)*. Accessed: 2022-05-31. URL : https://www.naxoo.ch/linternet-des-objets-internet-of-things-iot/.

Neal, Radford M. (2006). "Pattern Recognition and Machine Learning". In : *Technometrics* 49, p. 366 -366. URL : https://api.semanticscholar.org/CorpusID:31993898.

Neshenko, Nataliia et al. (2019). "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations". In : *IEEE Communications Surveys Tutorials* 21.3, p. 2702-2733. DOI : 10.1109/COMST.2019.2910750.

Nicholaus, Isack Thomas et al. (2021). "Anomaly Detection of Water Level Using Deep Autoencoder". In : *Sensors* 21.19. ISSN : 1424-8220. DOI : 10.3390/s21196679. URL : https://www.mdpi.com/1424-8220/21/19/6679.

Nikolov, Neven (2020). "Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems". In : *2020 XXIX International Scientific Conference Electronics (ET)*, p. 1-4. DOI : 10.1109/ET50336.2020.9238208.

Nõmm, Sven et Hayretdin Bahsi (déc. 2018). "Unsupervised Anomaly Based Botnet Detection in IoT Networks". In : p. 1048-1053. DOI : 10.1109/ICMLA.2018.00171.

Okano, Marcelo T (2017). “IOT and industry 4.0: the industrial new revolution”. In : *International Conference on Management and Information Systems*. T. 25, p. 26.

Oladimeji, Damilola et al. (2023). “Smart transportation: an overview of technologies and applications”. In : *Sensors* 23.8, p. 3880.

Optimizing Performance: SelectKBest for Efficient Feature Selection in Machine Learning (s. d.). Consulté le 20 juin 2023. URL : https://medium.com/@Kavya2099/optimizing-performance-selectkbest-for-efficient-feature-selection-in-machine-learning-3b635905ed48.

Otoum Ph.D P.Eng, Yazan, Dandan Liu et Amiya Nayak (mars 2022). “DL-IDS: a deep learning–based intrusion detection framework for securing IoT”. In : *Transactions on Emerging Telecommunications Technologies* 33. DOI : 10.1002/ett.3803.

Painuli, Deepak et al. (2021). “20 - Forecast and prediction of COVID-19 using machine learning”. In : *Data Science for COVID-19*. Sous la dir. d’Utku Kose et al. Academic Press, p. 381-397. ISBN : 978-0-12-824536-1. DOI : https://doi.org/10.1016/B978-0-12-824536-1.00027-7. URL : https://www.sciencedirect.com/science/article/pii/B9780128245361000277.

Payatu (2018). *IoT Security – Part 3 (101 – IoT Top Ten Vulnerabilities)*. Dernière MAJ: 19 janvier 2018, Visité: 18/05/22. URL : https://payatu.com/masterclass/iot-security-part-3-101-iot-top-ten-vulnerabilities.

Pisner, Derek A et David M Schnyer (2020). “Support vector machine”. In : *Machine learning*. Elsevier, p. 101-121.

Priya Dharshini K.and Gopalakrishnan, D.and Shankar C. K. et R. Ramya (2022). “A Survey on IoT Applications in Smart Cities”. In : *Immersive Technology in Smart Cities: Augmented and Virtual Reality in IoT*. Sous la dir. de Sagaya Aurelia et Sara Paiva. Cham : Springer International Publishing, p. 179-204. ISBN : 978-3-030-66607-1. DOI : 10.1007/978-3-030-66607-1_9. URL : https://doi.org/10.1007/978-3-030-66607-1_9.

Ranzato, Marc’Aurelio et al. (jan. 2006). “Efficient Learning of Sparse Representations with an Energy-Based Model”. In.

Regulagadda, Ravi et al. (2023). “Accurate Diagnosis of Cardiac Disease with and Without Using GRIDSEARCHCV”. In : *2023 IEEE 5th International Conference on Cybernetics,*

Cognition and Machine Learning Applications (ICCCMLA), p. 26-29. DOI : 10.1109/ICCCMLA58983.2023.10346700.

Rigatti, Steven J (2017). "Random forest". In : *Journal of Insurance Medicine* 47.1, p. 31-39.

Rizvi, Syed et al. (2020). "Identifying the Attack Surface for IoT Network". In : *Internet of Things*, p. 100162. DOI : 10.1016/j.iot.2020.100162.

Rosenblatt, F. (1957). *The perceptron - A perceiving and recognizing automaton*. Rapp. tech. 85-460-1. Ithaca, New York : Cornell Aeronautical Laboratory.

Saarika, PS, K Sandhya et T Sudha (2017). "Smart transportation system using IoT". In : *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*. IEEE, p. 1104-1107.

Saba, Tanzila et al. (avr. 2022). "Anomaly-based intrusion detection system for IoT networks through deep learning model". In : *Computers Electrical Engineering* 99, p. 107810. DOI : 10.1016/j.compeleceng.2022.107810.

Samrin, Rafath et D Vasumathi (2017). "Review on anomaly based network intrusion detection system". In : *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, p. 141-147. DOI : 10.1109/ICEECCOT.2017.8284655.

Samuel, A. L. (1959). "Some Studies in Machine Learning Using the Game of Checkers". In : *IBM Journal of Research and Development* 3.3, p. 210-229. DOI : 10.1147/rd.33.0210.

Sarawi, Shadi et al. (juill. 2017). "Internet of Things (IoT) Communication Protocols : Review". In : DOI : 10.1109/ICITECH.2017.8079928.

Saritha, Saritha et V Sarasvathi (2017). "A study on application layer protocols used in IoT". In : *2017 International Conference on Circuits, Controls, and Communications (CCUBE)*, p. 155-159. DOI : 10.1109/CCUBE.2017.8394143.

Serbanati, Alexandru, Carlo Maria Medaglia et Ugo Biader Ceipidor (2011). "Building Blocks of the Internet of Things: State of the Art and Beyond". In : URL : https://api.semanticscholar.org/CorpusID:3992830.

Sewak, Mohit, Sanjay Sahay et Hemant Rathore (jan. 2020). "An Overview of Deep Learning Architecture of Deep Neural Networks and Autoencoders". In : *Journal of Compu-*

tational and Theoretical Nanoscience 17, p. 182-188. DOI : 10.1166/jctn.2020.8648.

SHAHID, Mustafizur R. et al. (2018). "IoT Devices Recognition Through Network Traffic Analysis". In : *2018 IEEE International Conference on Big Data (Big Data)*, p. 5187-5192. DOI : 10.1109/BigData.2018.8622243.

SHARMA, Neelam et Saurabh MUKHERJEE (déc. 2012). "A Novel Multi-Classifier Layered Approach to Improve Minority Attack Detection in IDS". In : *Procedia Technology* 6, p. 913-921. DOI : 10.1016/j.protcy.2012.10.111.

SHONE, Nathan et al. (fév. 2018). "A Deep Learning Approach to Network Intrusion Detection". In : *IEEE Transactions on Emerging Topics in Computational Intelligence* 2, p. 41-50. DOI : 10.1109/TETCI.2017.2772792.

SHORTLIFFE, Edward (oct. 1976). "Computer-based medical consultations: MYCIN". In : *Artificial Intelligence - AI* 388. DOI : 10.1097/00004669-197610000-00011.

SINGH, Gurpreet et Manoj SACHAN (2014). "Multi-layer perceptron (MLP) neural network technique for offline handwritten Gurmukhi character recognition". In : *2014 IEEE International Conference on Computational Intelligence and Computing Research*, p. 1-5. DOI : 10.1109/ICCIC.2014.7238334.

SOMVANSHI, Madan et al. (2016). "A review of machine learning techniques using decision tree and support vector machine". In : *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, p. 1-7. DOI : 10.1109/ICCUBEA.2016.7860040.

SONAR, Krushang et Hardik UPADHYAY (2014). "A Survey: DDOS Attack on Internet of Things". In : *International Journal of Engineering Research and Development* 10.11, p. 58-63.

STILLER, Burkhard, Eryk SCHILLER et Corinna SCHMITT (jan. 2021). "An Overview of Network Communication Technologies for IoT". In.

STILLER, Burkhard et al. (2020). "An overview of network communication technologies for IoT". In : *Handbook of Internet-of-Things* 12.

STOJKOSKA, Biljana L Risteska et Kire V TRIVODALIEV (2017). "A review of Internet of Things for smart home: Challenges and solutions". In : *Journal of cleaner production* 140, p. 1454-1464.

Suthaharan, Shan (2016). “Support Vector Machine”. In : *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*. Boston, MA : Springer US, p. 207-235. isbn : 978-1-4899-7641-3. doi : 10.1007/978-1-4899-7641-3_9. url : https://doi.org/10.1007/978-1-4899-7641-3_9.

Sutton, Richard S. et Andrew G. Barto (2018). *Reinforcement Learning: An Introduction*. 2nd. Adaptive Computation and Machine Learning Series. Cambridge, MA : The MIT Press. url : https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf.

Tien, Chin Wei et al. (juill. 2021). “Using Autoencoders for Anomaly Detection and Transfer Learning in IoT”. In : *Computers* 10, p. 88. doi : 10.3390/computers10070088.

Toshniwal, Akanksha, Kavi Mahesh et Ranga Reddy Jayashree (oct. 2020). “Overview of Anomaly Detection techniques in Machine Learning”. In : p. 808-815. doi : 10.1109/I-SMAC49090.2020.9243329.

TURING, A. M. (oct. 1950). “I.—COMPUTING MACHINERY AND INTELLIGENCE”. In : *Mind* LIX.236, p. 433-460. issn : 0026-4423. doi : 10.1093/mind/LIX.236.433. eprint : https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf. url : https://doi.org/10.1093/mind/LIX.236.433.

Ullah, Imtiaz (déc. 2020). “A Technique for Generating a Botnet Dataset for Anomalous Activity Detection in IoT Networks”. In : doi : 10.1109/SMC42975.2020.9283220.

Ullah, Imtiaz et Qusay H. Mahmoud (2021a). “Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks”. In : *IEEE Access* 9, p. 103906-103926. doi : 10.1109/ACCESS.2021.3094024.

— (2021b). “Network Traffic Flow Based Machine Learning Technique for IoT Device Identification”. In : *2021 IEEE International Systems Conference (SysCon)*, p. 1-8. doi : 10.1109/SysCon48628.2021.9447099.

Usama, Muhammad et al. (sept. 2017). “Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges”. In : *IEEE Access* PP. doi : 10.1109/ACCESS.2019.2916648.

VACCARI, Ivan et al. (2020). "MQTTset, a New Dataset for Machine Learning Techniques on MQTT". In : *Sensors* 20.22. ISSN : 1424-8220. DOI : 10.3390/s20226578. URL : https://www.mdpi.com/1424-8220/20/22/6578.

VINCENT, Pascal et al. (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". In : *J. Mach. Learn. Res.* 11, p. 3371-3408. URL : https://api.semanticscholar.org/CorpusID:17804904.

WIDIASARI, Indrastanti, Lukito NUGROHO et Widyawan WIDYAWAN (nov. 2017). "Deep learning multilayer perceptron (MLP) for flood prediction model using wireless sensor network based hydrology time series data mining". In : p. 1-5. DOI : 10.1109/INNOCIT.2017.8319150.

WINOGRAD, Terry (1972). "Understanding natural language". In : URL : https://api.semanticscholar.org/CorpusID:56798209.

Y. LeCun, Qu'est-ce que l'intelligence artificielle (s. d.). Consulté le 15 Septembre 2021. URL : https://www.college-de-france.fr/media/yann-lecun/UPL4485925235409209505_Intelligence_Artificielle_____Y._LeCun.pdf.

YANG, Feng-Jen (déc. 2019). "An Extended Idea about Decision Trees". In : p. 349-354. DOI : 10.1109/CSCI49370.2019.00068.

YASSEIN, Muneer Bani, Mohammed Q SHATNAWI et al. (2016). "Application layer protocols for the Internet of Things: A survey". In : *2016 International Conference on Engineering & MIS (ICEMIS)*. IEEE, p. 1-4.

YI, Xin, Ekta WALIA et Paul BABYN (2019). "Generative adversarial network in medical imaging: A review". In : *Medical image analysis* 58, p. 101552.

YOON, John (juin 2019). "Using a Deep-Learning Approach for Smart IoT Network Packet Analysis". In : p. 291-299. DOI : 10.1109/EuroSPW.2019.00039.

ZHAO, Junbo, Michael MATHIEU et Yann LECUN (2016). "Energy-based generative adversarial network". In : *arXiv preprint arXiv:1609.03126*.

ZHOU, Yichun, Xuyang SONG et Mengyuan ZHOU (2021). "Supply Chain Fraud Prediction Based On XGBoost Method". In : *2021 IEEE 2nd International Conference on Big Data,*

Artificial Intelligence and Internet of Things Engineering (ICBAIE), p. 539-542. DOI : 10.1109/ICBAIE52039.2021.9389949.

Zou, Jinming, Yi Han et Sung-Sau So (2009). "Overview of Artificial Neural Networks". In : *Artificial Neural Networks: Methods and Applications*. Sous la dir. de David J. Livingstone. Totowa, NJ : Humana Press, p. 14-22. ISBN : 978-1-60327-101-1. DOI : 10.1007/978-1-60327-101-1_2. URL : https://doi.org/10.1007/978-1-60327-101-1_2.

Šimec, Alen (2017). "Cyber-attacks and Internet of Things as a threat to critical infrastructure". In : *University of applied sciences Zagreb, Vrbik 8, Croatia*.