RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABOU BEKR BELKAID TLEMCEN
FACULTÉ DE TECHNOLOGIE
DÉPARTEMENT DE TELECOMMUNICATIONS

THÈSE
Présentée pour l'obtention du diplôme de
DOCTORAT en Télécommunications
Spécialité : Technologie de l'information et de la communication

Par
M^{me} **FERHI Wafaa**

Détection des anomalies dans l'Internet des Objets

Thèse soutenue publiquement en 2026 à Tlemcen devant le jury composé de :

Pr. Kadri Benamar	Université de Tlemcen	Président
Dr. Merzoug Mohammed	Université de Tlemcen	Examinateur
Dr. Zerrouki Hadj	Université de Sidi Bel Abbes	Examinateur
Pr. Hadjila Mourad	Université de Tlemcen	Directeur de thèse
Dr. Moussaoui Djilali	Université de Tlemcen	Co-Directeur de thèse
Dr. M'hamedi Mohammed	ESSA - Tlemcen	Invité

Année universitaire 2025-2026

Remerciements

Louange à Allah qui ma donné patience et courage pour mener à bien ce travail de thèse malgré les difficultés rencontrées. Je tiens à remercier en premier lieu Mr Hadjila Mourad et Mr Moussaoui Djilali, mes directeurs de thèse pour leurs précieux conseils.

Je remercie également les membres du jury qui ont accepté l'évaluation de notre travail, et je cite en l'occurrence Pr Kadri Benamar, Dr Merzoug Mohammed, Dr Zerrouki Hadj, et Dr M'Hamedi Mohammed.

Je remercie aussi ma famille pour son soutien, son écoute et ses encouragements tout au long de cette thèse.

Je dédie ce modeste travail
A mes très chers parents, qu'Allah tout puissant les protège
A mon mari et mes soeurs.

Résumé

L'essor de l'Internet des Objets (IoT) et de son pendant industriel (IIoT) a profondément transformé les systèmes cyber-physiques modernes. Cette connectivité ubiquitaire, moteur d'efficacité et d'automatisation, a cependant accru la complexité, l'hétérogénéité et la vulnérabilité des réseaux connectés. Dans ce contexte, la détection d'anomalies joue un rôle essentiel pour assurer la fiabilité, la résilience et la sécurité des infrastructures IoT. Cette thèse s'inscrit dans cette problématique en proposant des approches d'apprentissage profond capables d'identifier des comportements anormaux dans des flux de données massifs, dynamiques et hétérogènes. Après avoir établi un cadre conceptuel unifié définissant les notions d'anomalies, de défaillance et d'intrusion, les travaux ont porté sur la conception et l'évaluation de deux modèles complémentaires. Le premier modèle, fondé sur une architecture hybride CNNDNN, a été appliqué au jeu de données Edge-IIoTset pour la détection d'attaques dans des environnements IIoT. Les résultats expérimentaux montrent une amélioration significative des performances de classification sur plusieurs configurations de classes, confirmant la robustesse du modèle proposé. Le second modèle repose sur un Réseau de Neurones à Graphes Hétérogènes (DHGNN), appliqué au jeu de données CIC-Darknet2020, pour la classification du trafic chiffré. Cette approche tire parti des relations entre entités du réseau pour identifier les comportements suspects sans analyse du contenu. Les résultats obtenus démontrent la pertinence de combiner ingénierie des données, apprentissage profond et modélisation relationnelle pour renforcer la sécurité et la surveillance intelligente des systèmes IoT. Des perspectives sont enfin ouvertes vers l'apprentissage fédéré, l'adaptation en ligne et l'explicabilité des modèles.

Mots-clés : Internet des Objets, Détection intelligente,anomalies, Apprentissage profond, Metriques, Sécurité, Dataset.

Abstract

The rapid expansion of the Internet of Things (IoT) and its industrial counterpart (IIoT) has profoundly reshaped modern cyber-physical systems. While ubiquitous connectivity enables automation and efficiency, it also increases the complexity, heterogeneity, and vulnerability of connected infrastructures. In this context, anomaly detection has become essential to ensuring the reliability, resilience, and security of IoT environments. This thesis addresses these challenges by proposing deep learning-based approaches capable of identifying abnormal behaviors in large-scale, dynamic, and heterogeneous data streams. After defining a unified conceptual framework that distinguishes anomalies, failures, and intrusions, two complementary models were designed and evaluated. The first model, based on a hybrid CNNDNN architecture, was applied to the Edge-IIoTset dataset for detecting cyberattacks in IIoT environments. Experimental results show significant improvements in detection accuracy across multiple class configurations, demonstrating the robustness and adaptability of the proposed architecture. The second model employs a Dual Heterogeneous Graph Neural Network (DHGNN) on the CIC-Darknet2020 dataset for encrypted traffic classification. By leveraging relationships between network entities, the model successfully detects suspicious behaviors without relying on content inspection. The results highlight the relevance of combining data engineering, deep learning, and relational modeling to enhance intelligent monitoring and security in IoT systems. Future directions include federated learning, online adaptation, and explainable AI for trustworthy anomaly detection.

Keywords: Internet of Things, Intelligent Detection, Anomalies, Deep Learning, Metrics, Security, Dataset.

مـــلـــخـــص

التوسع السريع لإنترنت الأشياء (IoT) ونظيره الصناعي (IIoT) قد أعاد تشكيل الأنظمة السيبرانية الفيزيائية الحديثة بشكل عميق. بينما تتيح الاتصال الشامل الأتمتة والكفاءة، فإنها تزيد أيضًا من تعقيد وتنوع وهشاشة البنى التحتية المتصلة. في هذا السياق، أصبح **اكتشاف الشذوذ (السلوكات الغير معتادة)** ضروريًا لضمان موثوقية ومرونة وأمان بيئات إنترنت الأشياء. تتناول هذه الأطروحة هذه التحديات من خلال اقتراح نهج قائم على **التعلم العميق** قادر على تحديد السلوكيات الشاذة في تدفقات البيانات الكبيرة، الديناميكية، والمتنوعة. بعد تحديد إطار مفاهيمي موحد يميز بين السلوكات الغير العادية والأعطال والتدخلات، تم تصميم وتقييم نموذجين تكميليين. النموذج الأول، القائم على **بنية هجينة** من **CNN-DNN**، تم تطبيقه على مجموعة بيانات **Edge-IIoTset** لاكتشاف الهجمات السيبرانية في بيئات IIoT. تُظهر النتائج التجريبية تحسنًا كبيرًا في دقة الكشف عبر تكوينات الفئات المتعددة، مما يُظهر متانة ومرونة البنية المقترحة. النموذج الثاني يستخدم **شبكة عصبية رسومية ثنائية غير متجانسة (DHGNN)** على مجموعة بيانات **CIC-Darknet2020** لتصنيف حركة المرور المشفرة. من خلال الاستفادة من العلاقات بين كيانات الشبكة، ينجح النموذج في **اكتشاف السلوكيات المشبوهة دون الاعتماد على فحص المحتوى**. تُبرز النتائج أهمية دمج هندسة البيانات، والتعلم العميق، والنمذجة **الترابطيّة**

لتعزيز المراقبة الذكية والأمان في أنظمة إنترنت الأشياء. تشمل الاتجاهات المستقبلية التعلم الفيدرالي، والتكيف عبر الإنترنت، والذكاء الاصطناعي القابل للتفسير للكشف الموثوق عن الشذوذ.

الكلمات المفتاحية: إنترنت الأشياء، الكشف الذكي، الشذوذ، التعلم العميق، المقاييس، الأمان، مجموعة البيانات.

Table des matières

Remerciements **i**

Résumé **iii**

Abstract **iii**

Table des matières **vi**

Table des figures **x**

Liste des tableaux **xii**

Introduction générale **1**
0.1 Contexte 1
0.2 Problématique 2
0.3 Contribution 3
0.4 Organisation de la thèse 4

1 Internet des Objets : Fondements, Architectures et Enjeux de Sécurité **6**
1.1 Introduction 7
1.2 Présentation de l'IoT 8
1.3 Évolution de l'Internet des Objets (IoT) 9
1.3.1 Période précurseur : fondations conceptuelles et technologiques (1920-1970) 10
1.3.1.1 Contributions théoriques précoces (années 1920-1960) 10
1.3.1.2 Premiers réseaux et systèmes embarqués (années 1960-1970) 11
1.3.2 Naissance de la communication machine-à-machine (M2M) et émergence du paradigme IoT (1980-1999) 11
1.3.2.1 Premier appareil IoT (1982) 11
1.3.2.2 RFID et réseaux de capteurs sans fil (années 1990) . 11
1.3.2.3 Formalisation du terme "Internet des Objets" (1999) 12
1.3.3 Normalisation et premiers cadres IoT (2000-2008) 12
1.3.3.1 IPv6 et connectivité omniprésente (années 2000) . . 12
1.3.3.2 Premiers déploiements industriels et urbains (2005-2008) 12

1.3.4 Adoption massive et Internet industriel des objets (2010-2018) 12
1.3.4.1 Intégration du Cloud et des Big Data (2010-2014) . . 12
1.3.4.2 Efforts de normalisation et enjeux de sécurité (2015-2018) 13
1.3.5 Convergence technologique de l'IA, de la 5G et du Edge Computing dans l'IoT moderne (2020-présent) 13
1.3.5.1 IoT alimenté par l'intelligence artificielle 13
1.3.5.2 5G et informatique de bord 13
1.4 Domaines d'application clés de l'Internet des Objets 14
1.4.1 Applications domestiques et résidentielles 15
1.4.1.1 Écosystème de la maison intelligente (Smart Home) . 15
1.4.1.2 Santé connectée (Santé numérique / Digital Health) . 16
1.4.2 Secteurs industriels et logistiques (Internet Industriel des Objets IIoT) 16
1.4.3 Infrastructures urbaines et villes intelligentes (Smart Cities) . 17
1.5 Architecture de l'Internet des Objets (IoT) 18
1.5.1 Architecture stratifiée à trois couches 18
1.5.1.1 Couche de perception (Physical Layer) 18
1.5.1.2 Couche réseau (Network Layer) 20
1.5.1.3 Couche application (Service Layer) 20
1.6 Paradigmes de calcul distribué dans l'IoT et l'IIoT 20
1.6.1 Cloud Computing : le noyau analytique centralisé 20
1.6.2 Edge Computing : intelligence et autonomie de proximité . . . 21
1.6.3 Fog Computing : la couche intermédiaire hiérarchique 21
1.7 Paysage des menaces IoT/IIoT : analyse multi-couches 21
1.7.1 Menaces au niveau des dispositifs (Perception) 22
1.7.2 Menaces au niveau réseau (Communication) 22
1.7.3 Menaces au niveau application et services 22
1.7.4 Acteurs et capacités 22
1.7.5 Secteurs critiques et profils de risques 22
1.7.6 Cadres et bonnes pratiques 23
1.7.7 Dynamiques émergentes : 5G, IA offensive, chaîne d'approvisionnement 23
1.8 Conclusion 24

2 Anomalies et Détection dans l'Internet des Objets 25
2.1 Introduction 26
2.2 Lien entre l'IoT et les anomalies 27
2.2.1 Définition et conceptualisation des anomalies dans l'IoT 27
2.2.2 Les catégories d'anomalies dans les systèmes IoT 27
2.2.3 Le rôle de la détection d'anomalies dans les environnements IoT 28
2.3 Hiérarchisation conceptuelle des anomalies systémiques : défaillances, intrusions et anomalies comportementales 29
2.3.1 Défaillance : une violation de la fiabilité 29
2.3.2 Intrusion : une violation de la sécurité 29
2.3.3 Anomalie comportementale : une déviation observable 29
2.4 Détection d'anomalies sous contraintes : défis de sécurité dans les environnements IoT 30
2.4.1 Hétérogénéité architecturale et topologique 30

2.4.2 Contraintes de ressources 30
2.4.3 Complexités centrées sur les données 31
2.5 De la Détection Statique vers l'Intelligence Adaptative 31
2.5.1 Limites structurelles des approches traditionnels 31
2.5.1.1 Définition 2.1 Détection par seuil fixe 31
2.5.1.2 Proposition 2.1 Insuffisance du modèle statique . . . 32
2.5.2 Fondements théoriques de l'apprentissage adaptatif 32
2.5.2.1 Définition 2.2 Concept drift 32
2.5.2.2 Formulation de la mise à jour des paramètres 33
2.5.3 Principaux algorithmes d'apprentissage pour l'adaptation intelligente 33
2.5.3.1 Apprentissage automatique classique (Machine Learning) 33
2.5.3.2 Réseaux de neurones profonds (Deep Neural Networks, DNN) 34
2.5.3.3 Réseaux de neurones sur graphes (Graph Neural Networks, GNN) 34
2.6 Renforcement de la cybersécurité dans l'Internet des Véhicules (IoV) : détection d'anomalies et d'intrusions par Deep Learning 34
2.6.1 Contexte et motivation 34
2.6.2 Synthèse de travaux récents sur la sécurité de l'IoV 35
2.6.3 Paysage des attaques dans l'Internet des Véhicules 36
2.6.4 Jeu de données CICIoV2024 37
2.6.5 Prétraitement des données 37
2.6.5.0.1 Label Encoding 38
2.6.5.0.2 Feature Selection 38
2.6.5.0.3 Feature Scaling 38
2.6.5.0.4 One-Hot Encoding 38
2.6.5.0.5 Train-Test Split 39
2.6.6 Architecture du modèle DNN 39
2.6.7 Configurations de classification et résultats expérimentaux . . 39
2.6.8 Analyse comparative avec des modèles classiques 41
2.7 Conclusion 42

3 Détection d'anomalies pour l'IIoT : analyse du dataset Edge-IIoTset avec des distributions de classes variées 43
3.1 Introduction 43
3.2 Travaux connexes 46
3.3 Sécurité de l'IIoT : menaces, vulnérabilités et défis 48
3.4 Cadre conceptuel de l'étude 49
3.4.1 Réseaux de neurones profonds 49
3.4.2 Réseaux de neurones convolutionnels 51
3.4.3 Mesures d'évaluation 52
3.5 Modèle proposé 53
3.5.1 Jeu de données 53
3.5.2 Approches expérimentales 54
3.5.2.1 Classification binaire 55
3.5.2.2 Classification multiclasse 55
3.6 Résultats et discussion 59

3.7 Conclusion 63

4 Classification du trafic et des applications du darknet à l'aide d'un réseau neuronal à graphes hétérogènes 68
4.1 Introduction 68
4.2 Travaux connexes et revue de la littérature 70
4.3 Réseau de neurones à graphes hétérogènes 73
4.3.1 Construction de données graphiques 73
4.3.2 Passage de messages hétérogènes 73
4.3.3 Architecture du modèle 74
4.4 Methodologie 75
4.4.1 Description du dataset 75
4.4.2 Exploration et prétraitement des données 75
4.4.3 Conception du modèle 76
4.5 Résultats expérimentaux 78
4.5.1 Classification du trafic (cas à 4 classes) 80
4.5.2 Classification d'applications (cas à 8 classes) 83
4.6 Conclusion 87

Conclusion générale 89

Liste des publications 91

Bibliographie 93

Table des figures

1.1 Ecosystème IoT : convergence des mondes réel et numérique. 9
1.2 Chronologie de l'évolution de l'Internet des Objets de 1920 à 2025 . . 10
1.3 Domaines d'application de l'Internet des Objets. 14
1.4 Architecture d'une maison intelligente 15
2.1 Taxonomie des anomalies dans l'IoT 28
2.2 Prétraitement des données CICIoV2024 38
2.3 Matices de confusion pour la détection d'attaques IoV 40
2.4 Matrices de confusion (6 classes) pour la détection d'attaques IoV . . 41
3.1 Exemples d'attaques de sécurité IIoT à travers différentes couches . . 49
3.2 Défis liés à l'IIoT 50
3.3 Exemple de perceptron multicouche 51
3.4 Exemple de modèle utilisant la rétropropagation 51
3.5 Distribution en barres des 9 classes 57
3.6 Distribution en barres des 6 classes 57
3.7 Distribution en barres pour 10 classes (incluant la classe DA) 58
3.8 Méthodologie proposée dans cette étude 59
3.9 Exactitude pour la distribution à 15 classes 61
3.10 Exactitude pour la distribution à 6 classes 61
3.11 Exactitude pour la distribution à 10 classes 62
3.12 Exactitude pour la distribution à 9 classes 62
3.13 Fonction de perte pour la distribution à 6 classes 63
3.14 Fonction de perte pour la distribution à 9 classes 64
3.15 Fonction de perte pour la distribution à 10 classes 66
3.16 Fonction de perte pour la distribution à 15 classes 66
4.1 Processus de conception du modèle. 77
4.2 Proportion de chaque classe dans le dataset CIC-Darknet2020 Cas de trafic 80
4.3 Perte du modèle en fonction d'une époque durant le processus d'entraînement - Cas de trafic 81
4.4 Proportion de chaque classe mal classée - Cas de trafic 82
4.5 Matrice de confusion pour la classification multi-classes des flux - Cas de trafic 83
4.6 Perte du modèle en fonction des époques durant le processus dapprentissage Cas dapplication. 84
4.7 Proportion de chaque classe mal classée Cas dapplication 85

4.8 Matrice de confusion pour la classification multi-classes des flux - Cas d'applicatitions. 86

Liste des tableaux

1.1 Résultats empiriques issus de l'application de l'IIoT et des technologies connexes 17
1.2 Comparatif des principales normes et standards d'architecture IoT . . 19
1.3 Typologie synthétique des acteurs de la menace IoT 23
1.4 Exemples de profils sectoriels (exposition × impact) 23
1.5 Vulnérabilités typiques par couche et exemples 24

2.1 Surfaces d'attaque, vulnérabilités et impacts potentiels dans les systèmes automobiles 36
2.2 Distribution des classes dans le jeu de données CICIoV2024. 37
2.3 Architecture du modèle DNN pour la classification des attaques IoV. 39
2.4 Comparaison des performances globales entre le DNN proposé et des modèles de machine learning classiques. 42

3.1 Nombre d'instances pour `Attack_label` 54
3.2 Nombre d'instances par type de trafic dans Edge-IIoTset (`Attack_type`) 54
3.3 Métriques de classification binaire. 59
3.4 Métriques d'évaluation pour la distribution à 15 classes. 63
3.5 Métriques d'évaluation pour la distribution à 9 classes. 64
3.6 Métriques d'évaluation pour la distribution à 10 classes. 65
3.7 Métriques d'évaluation pour la distribution à 6 classes. 65
3.8 Résumé des résultats. 65
3.9 Comparaison des résultats avec les travaux antérieurs utilisant le jeu de données *Edge-IIoTset*. 66

4.1 DHGNN Model Node Features 76
4.2 Hyperparamètres du modèle DHGNN et plage d'exploration 79
4.3 Évaluation de la performance des différentes méthodes de classification de trafic. 82
4.4 Comparaison des résultats de rappel des méthodologies de classification d'applications 87
4.5 Analyse comparative de la classification d'applications 87

Introduction générale

Sommaire

0.1 Contexte 1
0.2 Problématique 2
0.3 Contribution 3
0.4 Organisation de la thèse 4

0.1 Contexte

La numérisation massive des systèmes et la prolifération des objets connectés ont profondément transformé les environnements cyber-physiques. De la maison à l'usine, des réseaux urbains aux infrastructures critiques, l'Internet des Objets (IoT) et sa déclinaison industrielle (IIoT) constituent désormais une trame invisible qui capte, transporte et alimente la décision par les données. Cette connectivité ubiquitaire, moteur d'efficacité et d'automatisation, s'accompagne toutefois d'une expansion considérable de la surface d'attaque, d'une hétérogénéité technologique marquée et d'une variabilité continue des comportements observés. Dans ce contexte, la détection d'anomalies et la classification du trafic ne relèvent plus d'un simple filtrage statistique ; elles exigent des approches intelligentes, adaptatives, capables d'intégrer le contexte, d'embrasser la diversité des flux et d'exploiter les dépendances structurelles entre dispositifs et communications.

L'Internet des Objets (IoT) représente aujourd'hui l'une des pierres angulaires de la transformation numérique. En connectant une multitude d'objets intelligents capteurs, actionneurs, dispositifs mobiles et systèmes industriels, il permet la collecte, l'échange et le traitement en temps réel d'un volume colossal de données. Cette interconnexion ubiquitaire ouvre la voie à de nombreuses applications : villes intelligentes, santé connectée, industrie 4.0, réseaux d'énergie, transport autonome, et bien d'autres.

Cependant, cette expansion rapide s'accompagne d'une complexification sans précédent des environnements réseau. Les systèmes IoT sont distribués, dynamiques et fortement hétérogènes tant au niveau matériel que logiciel. Ils reposent sur des protocoles variés, des capacités de calcul limitées et des infrastructures souvent dépourvues de mécanismes de sécurité robustes. Cette situation rend l'IoT particulièrement vulnérable à des menaces de plus en plus sophistiquées telles que les attaques par déni

de service (DoS/DDoS), les intrusions, la compromission de nœuds ou encore l'exfiltration de données.

Dans le domaine industriel, cette problématique devient critique. L'Industrial Internet of Things (IIoT) relie des systèmes de contrôle et des chaînes de production entières, où une intrusion peut provoquer non seulement des pertes économiques considérables mais aussi des risques physiques pour les opérateurs. La cybersécurité de l'IoT/IIoT s'impose donc comme un enjeu majeur de fiabilité et de résilience.

Face à la complexité croissante du trafic réseau et à la diversité des menaces, les approches traditionnelles de détection basées sur des signatures ou des règles statiques se révèlent insuffisantes. Dans ce contexte, l'intelligence artificielle, et en particulier les méthodes d'apprentissage automatique et d'apprentissage profond, offrent de nouvelles perspectives. Elles permettent d'extraire des motifs cachés, de détecter des comportements anormaux et de s'adapter à l'évolution du trafic.

L'objectif général de cette thèse s'inscrit dans cette dynamique : concevoir et évaluer des méthodes de détection d'anomalies efficaces, adaptatives et robustes pour les environnements IoT et IIoT.

0.2 Problématique

L'IoT repose sur une architecture distribuée et hétérogène, intégrant des dispositifs aux capacités de calcul limitées, fonctionnant dans des environnements ouverts et interconnectés. Ces caractéristiques, bien que favorisant l'agilité et la scalabilité des systèmes, engendrent de nouveaux défis en matière de surveillance, d'analyse et de sécurité.

La première difficulté réside dans la diversité et la dynamique des données : les flux générés par les objets connectés sont massifs, variés, bruités et non stationnaires. Les comportements dits "normaux" peuvent évoluer au fil du temps, rendant obsolètes les seuils et modèles statiques.

La seconde difficulté concerne les contraintes de ressources. Les dispositifs IoT opèrent souvent avec des capacités restreintes en calcul, mémoire et énergie, ce qui limite l'implémentation de solutions complexes sur le plan local.

Une troisième difficulté majeure touche la qualité et la représentation des données. Les jeux de données utilisés pour la détection d'anomalies présentent souvent une forte hétérogénéité au niveau des features (caractéristiques) : valeurs manquantes, redondantes ou non normalisées. Ces données brutes nécessitent un traitement préalable rigoureux incluant le nettoyage, le codage, le filtrage, la vectorisation et parfois la réduction de dimensionnalité afin d'assurer leur compatibilité avec les modèles d'apprentissage. La pertinence du modèle dépend alors directement de la qualité de cette étape de prétraitement, qui conditionne la stabilité et la fiabilité des prédictions.

S'ajoute à cela le déséquilibre des classes, caractéristique des environnements réels où les attaques et anomalies sont beaucoup plus rares que le trafic légitime. Ce déséquilibre complique la formation de modèles supervisés performants et favorise les biais de détection.

Enfin, la topologie en constante évolution des réseaux IoT avec des dispositifs qui rejoignent ou quittent le réseau dynamiquement rend la modélisation du comportement global encore plus ardue.

Ces contraintes soulèvent plusieurs questions de recherche :

- Comment représenter efficacement la structure et le comportement des réseaux IoT pour améliorer la détection d'anomalies ?
- Quelles architectures d'apprentissage permettent d'intégrer les dépendances spatio-temporelles tout en restant légères et adaptatives ?
- Comment concilier précision, rapidité et résilience face aux dérives conceptuelles et à l'évolution du trafic ?
- Comment traiter et encoder efficacement les features issues de données hétérogènes pour garantir la robustesse du modèle ?
- Comment évaluer et généraliser ces modèles dans des contextes réels, hétérogènes et soumis à des contraintes de ressources ?

Ces interrogations constituent le socle de la démarche scientifique adoptée dans cette thèse.

0.3 Contribution

Les travaux réalisés dans cette thèse s'inscrivent dans le cadre de la détection d'anomalies dans les systèmes IoT et IIoT à l'aide de modèles d'apprentissage profond et de représentations intelligentes des flux réseau. Ils couvrent l'ensemble du cycle de traitement, depuis la compréhension des environnements IoT jusqu'à la conception et l'évaluation de modèles d'apprentissage avancés. Les contributions majeures peuvent être synthétisées comme suit :

Élaboration d'un cadre conceptuel unifié : Une analyse approfondie du paysage IoT/IIoT a été menée afin de formaliser la notion d'anomalie, d'établir une taxonomie rigoureuse (défaillances, intrusions, anomalies comportementales) et de positionner les approches de détection existantes selon leurs limites et potentialités. Ce travail a permis de dégager les principales caractéristiques structurelles et dynamiques des environnements IoT et d'identifier les paramètres critiques influençant la détection d'anomalies.

Prétraitement et ingénierie des données IoT : Avant l'entraînement des modèles, un effort substantiel a été consacré à la préparation et au traitement des données. Les jeux de données utilisés notamment Edge-IIoTset et CIC-Darknet2020 ont fait l'objet d'un processus rigoureux de nettoyage, d'encodage, de filtrage et de vectorisation afin d'assurer la cohérence et la compatibilité des features avec les architectures profondes. Ce travail d'ingénierie des caractéristiques a permis d'optimiser la qualité de l'apprentissage, de réduire la redondance et d'améliorer la représentativité des données, renforçant ainsi la stabilité et la performance des modèles.

Proposition d'un modèle de détection d'anomalies basé sur l'apprentissage profond : Un modèle hybride CNNDNN a été conçu et appliqué au jeu de données Edge-IIoTset, englobant divers types d'attaques et de scénarios de trafic. Le modèle a été évalué selon plusieurs configurations de classes (2, 6, 9, 10 et 15 classes) afin d'étudier sa robustesse et sa capacité de généralisation. Les résultats ont montré une amélioration significative des performances de détection par rapport aux approches classiques, démontrant la pertinence des architectures profondes pour la surveillance intelligente de l'IIoT.

Exploration d'approches relationnelles pour la modélisation du trafic : Une extension de la méthodologie a été réalisée à travers l'utilisation d'un Réseau de Neurones à Graphes Hétérogènes (DHGNN), appliqué au dataset CIC-Darknet2020.

Ce modèle tire parti des relations structurelles entre hôtes et flux, permettant une classification performante du trafic chiffré et une meilleure compréhension des comportements anormaux dans les environnements complexes. Il met en évidence le potentiel des modèles à base de graphes pour la détection d'anomalies dans des contextes où le contenu est masqué ou partiellement observable.

Mise en place d'un cadre expérimental rigoureux et reproductible : L'ensemble des expériences a été conduit selon des protocoles systématiques d'échantillonnage, de normalisation, de validation croisée et d'évaluation (accuracy, F1-score, précision, rappel). Ce cadre expérimental garantit la fiabilité des résultats et facilite la comparaison avec les méthodes de l'état de l'art.

Ainsi, la thèse démontre la pertinence de combiner l'ingénierie des données, l'apprentissage profond et la modélisation relationnelle pour concevoir des systèmes de détection d'anomalies capables de s'adapter à la dynamique, à la diversité et à la complexité des réseaux IoT et IIoT.

0.4 Organisation de la thèse

Le manuscrit est structuré en quatre chapitres principaux, suivant une progression cohérente allant du contexte général jusqu'à la mise en œuvre expérimentale des modèles proposés :

Le chapitre 1 présente le contexte général de l'Internet des Objets (IoT) et de l'Internet Industriel des Objets (IIoT). Ce chapitre introduit les architectures, les domaines d'application et les principales vulnérabilités liées à ces environnements connectés. Il offre également une vue d'ensemble sur les enjeux de sécurité et les limites des approches classiques de détection d'anomalies, posant ainsi les bases conceptuelles de la recherche.

Le chapitre 2 développe le cadre théorique et conceptuel de la détection d'anomalies dans l'IoT. Il définit les différentes catégories d'anomalies, présente les principales approches existantes (statistiques, à base de règles, d'apprentissage automatique et profond) et discute leurs avantages et limites. Ce chapitre met également en évidence les défis spécifiques de l'IoT, tels que le déséquilibre des classes, la dérive des données, l'hétérogénéité des flux et la nécessité d'un prétraitement rigoureux des features pour garantir la qualité de l'apprentissage.

Le chapitre 3 décrit la méthodologie proposée pour la détection d'anomalies dans l'IIoT à l'aide d'un modèle hybride CNNDNN. Il présente les étapes de préparation et de traitement des données du jeu Edge-IIoTset, incluant le codage, la vectorisation et la normalisation des caractéristiques. Les différentes expérimentations menées selon plusieurs répartitions de classes (2, 6, 9, 10 et 15 classes) sont ensuite exposées, accompagnées d'une évaluation comparative détaillée des performances.

Le chapitre 4 étend la démarche vers la classification de trafic chiffré et la modélisation relationnelle à l'aide d'un Réseau de Neurones à Graphes Hétérogènes (DH-GNN), appliqué au jeu de données CIC-Darknet2020. Ce chapitre explore la capacité des modèles à base de graphes à exploiter les relations entre hôtes et flux pour détecter les comportements suspects, même lorsque le contenu du trafic est partiellement ou totalement chiffré. Les résultats obtenus sont comparés à l'état de l'art, confirmant la pertinence des représentations relationnelles pour la cybersécurité des environnements distribués.

Enfin, la conclusion générale synthétise l'ensemble des contributions de la re-

cherche. Elle met en évidence la complémentarité entre les différentes approches proposées depuis la préparation des données jusqu'à la conception de modèles intelligents et ouvre plusieurs perspectives de recherche, notamment en matière d'apprentissage fédéré, de détection en ligne, d'équilibrage de classes et d'explicabilité des modèles pour renforcer la confiance et la transparence des systèmes de détection d'anomalies dans l'IoT.

Chapitre 1

Internet des Objets : Fondements, Architectures et Enjeux de Sécurité

Sommaire

1.1 Introduction 7
1.2 Présentation de l'IoT 8
1.3 Évolution de l'Internet des Objets (IoT) 9
1.3.1 Période précurseur : fondations conceptuelles et technologiques (1920-1970) 10
1.3.1.1 Contributions théoriques précoces (années 1920-1960) 10
1.3.1.2 Premiers réseaux et systèmes embarqués (années 1960-1970) 11
1.3.2 Naissance de la communication machine-à-machine (M2M) et émergence du paradigme IoT (1980-1999) 11
1.3.2.1 Premier appareil IoT (1982) 11
1.3.2.2 RFID et réseaux de capteurs sans fil (années 1990) 11
1.3.2.3 Formalisation du terme "Internet des Objets" (1999) 12
1.3.3 Normalisation et premiers cadres IoT (2000-2008) 12
1.3.3.1 IPv6 et connectivité omniprésente (années 2000) . 12
1.3.3.2 Premiers déploiements industriels et urbains (2005-2008) 12
1.3.4 Adoption massive et Internet industriel des objets (2010-2018) 12
1.3.4.1 Intégration du Cloud et des Big Data (2010-2014) 12
1.3.4.2 Efforts de normalisation et enjeux de sécurité (2015-2018) 13
1.3.5 Convergence technologique de l'IA, de la 5G et du Edge Computing dans l'IoT moderne (2020-présent) 13
1.3.5.1 IoT alimenté par l'intelligence artificielle 13
1.3.5.2 5G et informatique de bord 13
1.4 Domaines d'application clés de l'Internet des Objets . . 14
1.4.1 Applications domestiques et résidentielles 15
1.4.1.1 Écosystème de la maison intelligente (Smart Home) 15
1.4.1.2 Santé connectée (Santé numérique / Digital Health) 16
1.4.2 Secteurs industriels et logistiques (Internet Industriel des Objets IIoT) 16

1.4.3 Infrastructures urbaines et villes intelligentes (Smart Cities) 17
1.5 Architecture de l'Internet des Objets (IoT) 18
1.5.1 Architecture stratifiée à trois couches 18
1.5.1.1 Couche de perception (Physical Layer) 18
1.5.1.2 Couche réseau (Network Layer) 20
1.5.1.3 Couche application (Service Layer) 20
1.6 Paradigmes de calcul distribué dans l'IoT et l'IIoT . . . 20
1.6.1 Cloud Computing : le noyau analytique centralisé 20
1.6.2 Edge Computing : intelligence et autonomie de proximité . 21
1.6.3 Fog Computing : la couche intermédiaire hiérarchique . . . 21
1.7 Paysage des menaces IoT/IIoT : analyse multi-couches 21
1.7.1 Menaces au niveau des dispositifs (Perception) 22
1.7.2 Menaces au niveau réseau (Communication) 22
1.7.3 Menaces au niveau application et services 22
1.7.4 Acteurs et capacités 22
1.7.5 Secteurs critiques et profils de risques 22
1.7.6 Cadres et bonnes pratiques 23
1.7.7 Dynamiques émergentes : 5G, IA offensive, chaîne d'approvisionnement 23
1.8 Conclusion 24

1.1 Introduction

L'Internet des Objets (IoT *Internet of Things*) s'impose aujourd'hui comme l'un des piliers de la transformation numérique mondiale. En reliant des milliards d'objets physiques capteurs, actionneurs, véhicules, dispositifs médicaux ou infrastructures industrielles à des réseaux intelligents, l'IoT façonne un écosystème global où le monde réel et le monde numérique interagissent en permanence. Cette connectivité généralisée ouvre la voie à d'innombrables applications : optimisation des processus industriels, villes intelligentes, santé connectée, gestion énergétique, ou encore agriculture de précision. L'extension de ce paradigme à l'industrie, connue sous le nom d'*Industrial Internet of Things* (IIoT), marque une nouvelle étape vers l'automatisation et l'intelligence distribuée à grande échelle.

Cependant, cette évolution rapide s'accompagne de défis majeurs. L'IoT se caractérise par une hétérogénéité extrême des dispositifs, des protocoles et des environnements de déploiement. Les contraintes de calcul, de mémoire et d'énergie propres aux objets connectés rendent difficile l'intégration de mécanismes de sécurité avancés, tandis que la croissance exponentielle des données génère de nouveaux risques en matière de confidentialité, de fiabilité et d'intégrité. Ces vulnérabilités offrent aux acteurs malveillants de vastes surfaces d'attaque, allant des intrusions logicielles à la compromission de réseaux entiers. Dès lors, la sécurité des systèmes IoT n'est plus une option, mais un impératif technologique et stratégique.

Ce premier chapitre a pour objectif de situer le cadre conceptuel et technologique de l'IoT, d'en retracer l'évolution historique et d'en examiner les principales architectures, applications et problématiques de sécurité. Dans un premier temps, il présente la genèse et les fondements du paradigme IoT, depuis ses origines théoriques jusqu'à sa convergence avec les technologies émergentes telles que l'intelligence artificielle,

le *cloud* et le *edge computing*. Ensuite, il expose les principaux domaines d'application, illustrant la diversité des cas d'usage et des enjeux sociotechniques associés. Une attention particulière est accordée à l'architecture en couches de l'IoT et aux paradigmes de calcul distribués (Cloud, Fog et Edge), qui définissent les modalités de traitement et de circulation de l'information dans ces environnements complexes.

Enfin, le chapitre analyse les enjeux de cybersécurité propres aux écosystèmes IoT et IIoT : vulnérabilités structurelles, contraintes de ressources, hétérogénéité des dispositifs et insuffisance des modèles de protection classiques. Cette réflexion met en évidence la nécessité de concevoir des approches de sécurité adaptatives, capables d'évoluer avec les données et les contextes d'exécution. Ces considérations introduisent naturellement le chapitre suivant, consacré à la détection d'anomalies et à l'émergence de techniques d'apprentissage intelligent pour la protection proactive des systèmes connectés.

1.2 Présentation de l'IoT

L'Internet des Objets (IoT), terme introduit par Kevin Ashton en 1999 [1], désigne un paradigme technologique dans lequel des objets physiques munis de capteurs, d'actionneurs et de modules de communication sont interconnectés par des réseaux afin de collecter, échanger et exploiter des données, souvent sans intervention humaine directe [2]. Ces objets, également qualifiés de "connectés" ou "intelligents", peuvent interagir entre eux et avec des systèmes informatiques externes dans le cadre de communications machine-à-machine (M2M), facilitant ainsi l'automatisation et l'optimisation de processus dans des environnements variés [3]. L'émergence de l'IoT s'inscrit dans une convergence de progrès technologiques : miniaturisation des composants électroniques, démocratisation des communications sans fil, baisse du coût des capteurs et augmentation des capacités de calcul embarquées. L'intégration de l'IoT s'est accélérée au cours de la dernière décennie, soutenue par la disponibilité de réseaux à haut débit (4G, 5G et bientôt 6G) et par l'essor de l'architecture distribuée telle que le Edge Computing et le Fog Computing, qui rapprochent le traitement des données de leur source [4]. L'Internet des Objets (IoT) constitue une avancée technologique majeure ayant transformé en profondeur divers secteurs, allant de l'industrie manufacturière aux soins de santé. En intégrant capteurs, logiciels et technologies de communication, l'IoT crée des environnements intelligents dans des domaines variés tels que l'habitat, l'industrie, la santé et les transports, avec une adoption croissante à l'échelle mondiale. Dans le domaine domestique, l'IoT favorise l'automatisation des habitations connectées, illustrée par le projet ADEPT d'IBM et Samsung, où des appareils ménagers interconnectés régulent leur consommation énergétique et se coordonnent de manière autonome et sécurisée. En santé, les dispositifs IoT assurent la surveillance à distance des patients, transmettant en temps réel des données médicales aux professionnels, réduisant ainsi les visites hospitalières [5]. Les villes intelligentes exploitent également l'IoT pour optimiser la gestion énergétique grâce à des compteurs intelligents et des systèmes de distribution automatisés, favorisant une utilisation plus durable des ressources. Dans le secteur industriel, l'IoT, et plus particulièrement l'Internet des Objets Industriel (IIoT), renforce l'efficacité opérationnelle, soutient l'automatisation à grande échelle et facilite la prise de décision fondée sur les données [5].

Ces dernières années, l'Internet des Objets est devenu un levier majeur de transfor-

mation numérique, soutenu par les investissements de grands acteurs tels que Google, Samsung et Apple [6]. Selon Cisco, le potentiel économique global de l'IoT pourrait atteindre 14 000 milliards USD, avec déjà 25 milliards d'objets connectés avant 2020 et une projection de 50 milliards de connexions permanentes à moyen terme [6]. Les estimations récentes indiquent qu'il y a 29,4 milliards d'appareils connectés en 2023 [7] et jusqu'à 38 milliards d'ici 2026, notamment grâce à l'essor des secteurs manufacturier et de la santé, qui concentreraient une part importante des 6 200 milliards USD estimés pour le marché global à cet horizon [8]. Parallèlement, la production de capteurs devrait fortement croître dans des domaines stratégiques comme l'énergie et les mines, l'automobile ou la santé [8]. Cette expansion s'accompagne d'une augmentation massive du volume de données générées et échangées, ce qui renforce la nécessité de solutions efficaces de gestion, d'interopérabilité et de sécurité. L'IoT, bien qu'apportant des bénéfices considérables, expose en effet à des vulnérabilités importantes, notamment en raison de l'hétérogénéité des dispositifs, de la diversité des protocoles utilisés et des contraintes de ressources qui limitent l'intégration de mécanismes de protection robustes [5].

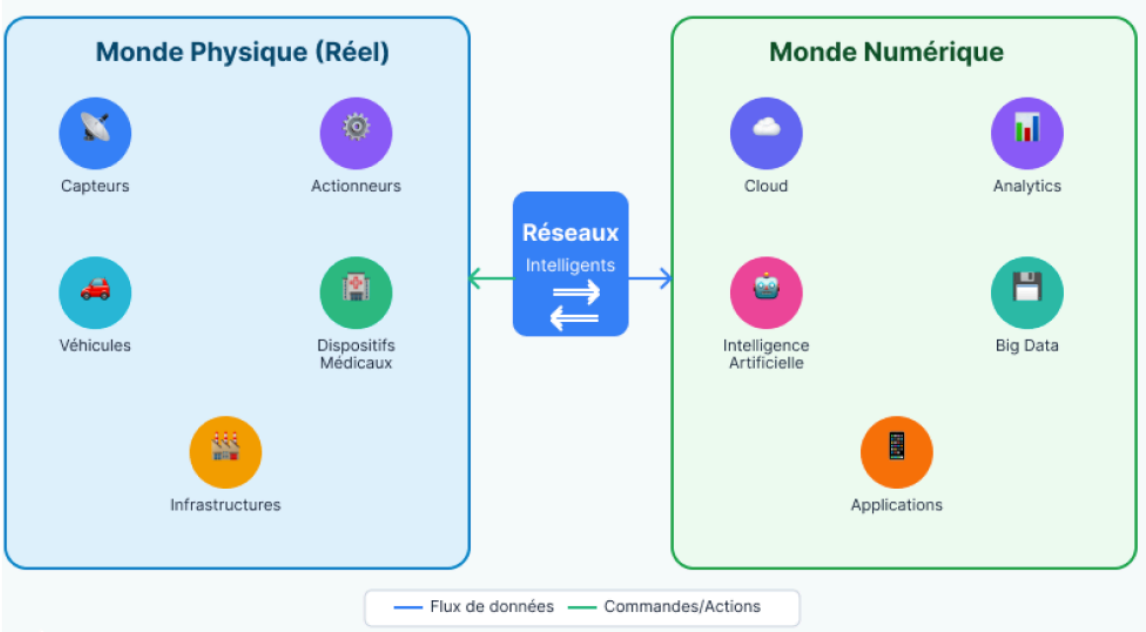

Figure 1.1 *– Ecosystème IoT : convergence des mondes réel et numérique.*

Ainsi, l'IoT s'impose aujourd'hui comme un pilier de la transformation numérique mondiale, En reliant des milliards d'objets physiques capteurs, actionneurs, véhicules, dispositifs médicaux ou infrastructures industrielles à des réseaux intelligents, l'IoT façonne un écosystème global où le monde réel et le monde numérique interagissent en permanence (voir figure 1.1).

1.3 Évolution de l'Internet des Objets (IoT)

L'évolution de l'Internet des Objets peut être retracée à travers cinq grandes périodes historiques, chacune caractérisée par des avancées technologiques majeures et des changements paradigmatiques (figure 1.2).

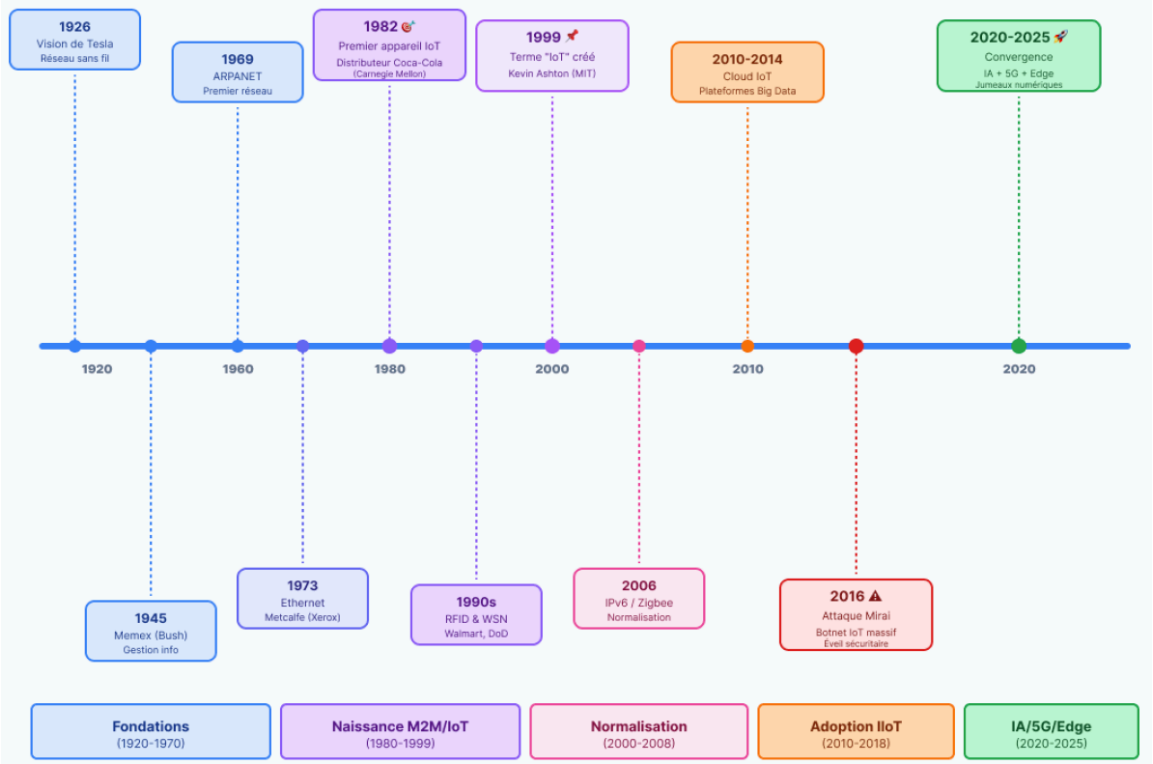

Figure 1.2 *– Chronologie de l'évolution de l'Internet des Objets de 1920 à 2025*

1.3.1 Période précurseur : fondations conceptuelles et technologiques (1920-1970)

1.3.1.1 Contributions théoriques précoces (années 1920-1960)

Les origines conceptuelles de l'IoT peuvent être attribuées à des chercheurs visionnaires qui ont prévisionné la communication en réseau entre les dispositifs. L'interview de Nikola Tesla en 1926 dans le magazine *Collier's* proposait un système sans fil qui connecterait tous les appareils à l'échelle mondiale, affirmant que ce réseau transformerait la Terre en un "énorme cerveau" [9]. Cette vision prémonitoire a établi le concept central de l'IoT, à savoir la connectivité intégrale, des années avant que les moyens techniques n'existent pour le mettre en œuvre.

L'essai de Vannevar Bush en 1945, *As We May Think*, a introduit le dispositif Memex, un ordinateur analogique théorique capable de stocker et de récupérer des informations par le biais de liens associatifs [10]. Bien que principalement axé sur la gestion de l'information, le travail de Bush anticipait des concepts clés de l'IoT, y compris l'association de données et la collecte à partir de systèmes en réseau. Ses idées ont influencé les innovations ultérieures en matière d'hypertexte et de technologie informatique en infrastructure réseau, qui se révéleraient essentiels pour les architectures de l'IoT.

Les mémos de J.C.R. Licklider en 1962 sur un "Réseau Informatique Intergalactique" à l'ARPA ont articulé la première vision globale des systèmes informatiques interconnectés à l'échelle mondiale [11]. Son travail a établi des principes fondamentaux pour la mise en réseau distribuée qui permettraient plus tard les systèmes IoT, y compris les concepts d'accès à distance aux données et de communication machine-à-machine qui sous-tendent les implémentations modernes de l'IoT.

1.3.1.2 Premiers réseaux et systèmes embarqués (années 1960-1970)

L'exploitation technologique des concepts de l'IoT a commencé avec le développement des technologies de réseau fondamentales. ARPANET, lancé en 1969, a démontré la faisabilité de la mise en réseau à commutation de paquets, créant le premier prototype opérationnel de ce qui deviendrait Internet [12]. Cette étude a prouvé que des dispositifs géographiquement dispersés pouvaient échanger des données de manière fiable, établissant ainsi la base technique pour les réseaux IoT ultérieurs.

L'invention de l'Ethernet par Robert Metcalfe en 1973 chez Xerox PARC a fourni un protocole standardisé pour la mise en réseau locale qui deviendrait crucial pour la connectivité des dispositifs IoT [13]. Ce protocole utilise un mécanisme d'accès distribué au médium permettant aux dispositifs de partager un même canal de communication. Le développement du CSMA/CD (Carrier Sense Multiple Access with Collision Detection) dans les normes Ethernet a résolu des défis clés concernant la gestion des collisions sur le réseau [14], un aspect essentiel pour garantir la communication fiable entre plusieurs dispositifs IoT sur un réseau partagé.

1.3.2 Naissance de la communication machine-à-machine (M2M) et émergence du paradigme IoT (1980-1999)

1.3.2.1 Premier appareil IoT (1982)

La première mise en œuvre reconnue de l'IoT a été développée à l'Université Carnegie Mellon en 1982, où des étudiants diplômés en informatique ont modifié un distributeur de Coca-Cola pour le connecter à ARPANET [15]. Ce projet innovant a équipé la machine de micro-interrupteurs pour surveiller les niveaux d'inventaire et d'un capteur thermique pour suivre la température des boissons, avec des rapports de statut accessibles sur le réseau. Bien que primitif selon les normes modernes, cette mise en œuvre a démontré trois principes fondamentaux de l'IoT : la détection embarquée, la connectivité réseau et la surveillance à distance de l'état des objets physiques.

1.3.2.2 RFID et réseaux de capteurs sans fil (années 1990)

La commercialisation de la technologie RFID dans les années 1990 a marqué une avancée significative vers des mises en œuvre pratiques de l'IoT. Les grands détaillants comme Walmart et les agences gouvernementales, y compris le Département de la Défense des États-Unis, ont adopté la RFID pour le suivi de la chaîne d'approvisionnement, prouvant la valeur commerciale des objets physiques en réseau [16]. Ces mises en œuvre ont démontré comment des identifiants numériques uniques pouvaient être attachés à des objets physiques, permettant un suivi automatisé et une gestion des stocks des fonctionnalités essentielles de l'IoT.

Les développements parallèles dans les réseaux de capteurs sans fil (WSNs) ont élargi les applications potentielles de l'IoT. Le projet Smart Dust a exploré des réseaux de nœuds de capteurs à l'échelle millimétrique capables de surveiller les conditions environnementales [17]. Ces innovations ont prouvé la faisabilité du déploiement de réseaux de capteurs distribués, en abordant des défis techniques clés dans la gestion de l'énergie et la communication sans fil qui informeraient plus tard les conceptions de systèmes IoT.

1.3.2.3 Formalisation du terme "Internet des Objets" (1999)

La conception moderne de l'IoT a été formellement établie en 1999 lorsque Kevin Ashton, travaillant au Auto-ID Center du MIT, a inventé le terme "Internet des Objets" [18]. Le travail d'Ashton se concentrait sur l'utilisation de la RFID pour créer un système mondial de suivi des biens physiques, mais sa définition s'est élargie pour englober la vision plus large d'un réseau connectant des objets physiques grâce à la technologie embarquée. Cette conceptualisation a marqué la transition des applications industrielles spécialisées au paradigme plus large de l'IoT que nous reconnaissons aujourd'hui.

1.3.3 Normalisation et premiers cadres IoT (2000-2008)

1.3.3.1 IPv6 et connectivité omniprésente (années 2000)

La normalisation de l'IPv6 en 2006 a résolu la limitation critique de l'espace d'adressage de l'IPv4, permettant l'identification unique de milliards de dispositifs IoT [19]. Cette expansion était essentielle pour soutenir la croissance anticipée des déploiements IoT, fournissant suffisamment d'adresses pour les dizaines de milliards de dispositifs connectés projetés dans les décennies à venir. Le développement de Zigbee en 2003, basé sur la norme IEEE 802.15.4, a fourni un protocole crucial pour la communication sans fil à faible puissance et à courte portée. IEEE 802.15.4 définit la couche physique et la couche contrôle d'accès au média pour les réseaux personnels sans fil à faible vitesse (LR-WPAN). Zigbee construit au-dessus de cette norme un protocole de réseau maillé fiable adapté aux applications embarquées. Cette configuration a permis une communication efficace entre les dispositifs IoT tout en optimisant la consommation d'énergie, répondant ainsi à deux défis clefs dans les implémentations IoT, ce qui a favorisé l'adoption de Zigbee dans l'automatisation domestique et la surveillance industrielle [20, 21].

1.3.3.2 Premiers déploiements industriels et urbains (2005-2008)

Durant cette période, l'IoT s'est particulièrement développé dans le domaine industriel et urbain. Les technologies RFID et les réseaux sans fil ont été largement adoptés pour la traçabilité des biens et la gestion automatisée des stocks [16]. Le concept de "Villes Intelligentes" (Smart Cities) a été adopté par diverses entreprises technologiques dès 2005. Les travaux d'IBM sur les "Villes plus Intelligentes" (Smarter Cities) ont débuté fin 2008 dans le cadre de leur initiative "Smarter Planet". Des villes comme Singapour montrent une forte implication dans ce domaine de compétition mondiale [22].

1.3.4 Adoption massive et Internet industriel des objets (2010-2018)

1.3.4.1 Intégration du Cloud et des Big Data (2010-2014)

L'émergence de plateformes IoT basées sur le cloud a fourni une infrastructure essentielle pour gérer des déploiements IoT à grande échelle [23]. Ces plateformes offrent des solutions évolutives pour la gestion des dispositifs, le stockage des données et l'analyse en temps réel, supprimant ainsi le besoin pour les entreprises de

développer des infrastructures IoT propriétaires. L'intégration de l'analyse de données massives avec les flux IoT a permis de nouvelles applications dans la maintenance prédictive et l'optimisation opérationnelle à travers les industries.

Parallèlement, les objets connectés se sont rapidement imposés comme un champ majeur d'application de l'IoT dans le secteur industriel. La connexion des équipements à des plateformes cloud permet la surveillance en temps réel des conditions et la mise en œuvre de la maintenance prédictive, contribuant à réduire les temps d'arrêt coûteux, notamment dans les industries à forte intensité de capital [24].

1.3.4.2 Efforts de normalisation et enjeux de sécurité (2015-2018)

La croissance rapide de l'IoT a nécessité le développement de protocoles de communication standardisés. MQTT (Message Queuing Telemetry Transport) et CoAP (Constrained Application Protocol) ont émergé comme des protocoles légers optimisés pour les dispositifs IoT avec une puissance de traitement limitée [25]. Ces normes ont répondu aux exigences uniques des communications IoT, y compris la connectivité intermittente et les contraintes de bande passante, tout en garantissant l'interopérabilité entre divers appareils et plateformes.

Les préoccupations en matière de sécurité sont devenues primordiales avec l'attaque du botnet Mirai en 2016, qui a compromis des milliers d'appareils IoT non sécurisés pour lancer des attaques DDoS massives [26]. Cet incident a révélé des vulnérabilités critiques dans les pratiques de sécurité de l'IoT et a stimulé des efforts pour développer des cadres de sécurité plus robustes pour les dispositifs IoT, y compris des changements de mot de passe obligatoires et des mécanismes de mise à jour de firmware sécurisés.

1.3.5 Convergence technologique de l'IA, de la 5G et du Edge Computing dans l'IoT moderne (2020-présent)

1.3.5.1 IoT alimenté par l'intelligence artificielle

L'intégration de l'intelligence artificielle avec l'IoT a permis des applications plus sophistiquées. Les algorithmes d'apprentissage automatique appliqués aux données des capteurs IoT facilitent la maintenance prédictive dans les environnements industriels, en analysant les schémas de vibration des équipements et les signatures thermiques pour anticiper les défaillances avant qu'elles ne se produisent [27]. La technologie des jumeaux numériques crée des répliques virtuelles de systèmes physiques qui peuvent simuler les performances dans diverses conditions, permettant ainsi l'optimisation sans perturber les opérations réelles [28].

1.3.5.2 5G et informatique de bord

Le déploiement des réseaux 5G répond aux exigences critiques de l'IoT en matière de communication à faible latence et de haute densité de dispositifs [29]. Ces capacités permettent des applications IoT sensibles au temps, telles que la coordination des véhicules autonomes et la robotique industrielle. Simultanément, les architectures de calcul de périphérie traitent les données plus près de leur source, réduisant ainsi la latence et les exigences de bande passante tout en améliorant la confidentialité grâce au traitement localisé des données [30]. Les approches d'apprentissage fédéré étendent

ces avantages en permettant l'entraînement collaboratif de modèles sur des dispositifs IoT distribués sans collecte de données centralisée.

En résumé, l'Internet des Objets est passé d'un concept visionnaire à une infrastructure globale reposant sur la connectivité ubiquitaire, l'intelligence distribuée et l'intégration de l'IA. Chaque étape de cette évolution des réseaux pionniers aux architectures cognitives actuelles a contribué à façonner les fondations techniques et sécuritaires des systèmes connectés contemporains.

1.4 Domaines d'application clés de l'Internet des Objets

Le développement de l'Internet des objets a entraîné des transformations profondes dans de multiples secteurs. Cette section explore les principaux domaines d'application, en se concentrant sur leurs fonctionnalités de base, leurs impacts plus larges et les défis qu'ils posent intrinsèquement. La figure 1.3 présente une vue d'ensemble des principaux domaines d'application, illustrant la diversité et l'étendue de l'écosystème IoT.

Figure 1.3 *– Domaines d'application de l'Internet des Objets.*

1.4.1 Applications domestiques et résidentielles

1.4.1.1 Écosystème de la maison intelligente (Smart Home)

L'écosystème de la maison intelligente constitue l'un des segments les plus dynamiques de l'Internet des objets, avec des projections estimant à plus de 50 milliards de dispositifs connectés d'ici 2025 [31]. Il intègre divers systèmes domestiques tels que le contrôle environnemental, la sécurité, la gestion de l'énergie et le divertissement dans un réseau intelligemment automatisé qui améliore le confort, l'efficacité et la commodité. Les applications principales incluent :

- **Systèmes d'automatisation domestique** : contrôle intelligent de l'éclairage, du chauffage, de la ventilation et de la climatisation (HVAC) et des appareils électroménagers ;
- **Sécurité et surveillance** : serrures intelligentes, caméras, détecteurs de mouvement et systèmes d'alarme ;
- **Gestion de l'énergie** : compteurs intelligents, thermostats et solutions de surveillance ;
- **Systèmes de divertissement** : téléviseurs connectés, haut-parleurs et plateformes de streaming ;
- **Assistants vocaux** : intégration avec Amazon Alexa, Google Assistant et Apple HomeKit.

La figure 1.4 illustre l'architecture complète d'une maison intelligente, montrant l'intégration des différents systèmes domestiques dans un réseau intelligemment automatisé.

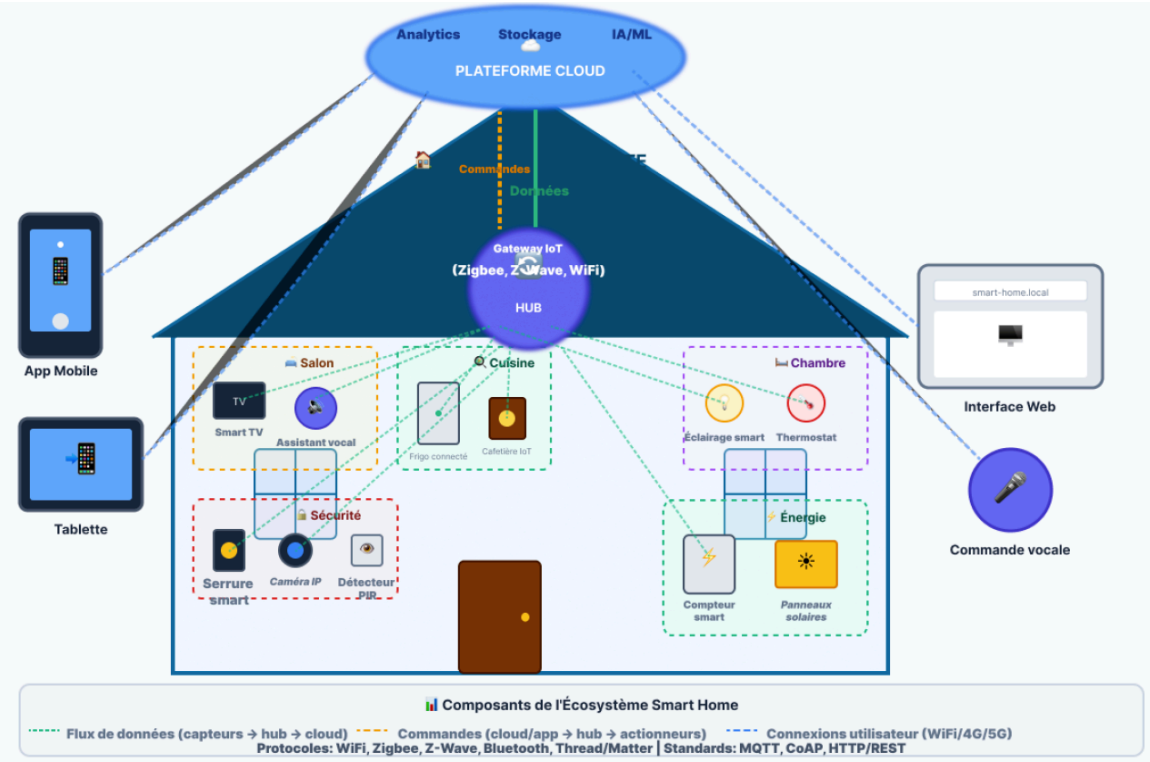

Figure 1.4 *– Architecture d'une maison intelligente*

1.4.1.2 Santé connectée (Santé numérique / Digital Health)

L'IoT dans le domaine de la santé communément appelé l'Internet des objets médicaux (IoMT, *Internet of Medical Things*) transforme la prestation de soins de santé en passant d'un soin épisodique et réactif à une gestion de la santé continue et proactive. En intégrant des dispositifs portables, des systèmes de surveillance à distance, des plateformes de télémédecine et des équipements médicaux intelligents, l'IoMT permet la collecte de données en temps réel, l'analyse prédictive et la prise de décisions fondée sur des preuves, améliorant ainsi les résultats des patients et réduisant les réadmissions hospitalières [32].

L'IoT dans le domaine de la santé englobe un large éventail de technologies, y compris :

- **Surveillance à distance des patients (RPM)** : suivi continu des signes vitaux pour la gestion des maladies chroniques et l'intervention précoce ;
- **Dispositifs de santé portables** : montres intelligentes, traqueurs de fitness et capteurs de qualité médicale pour la collecte de données en temps réel ;
- **Plateformes de télémédecine** : consultations virtuelles, diagnostics à distance et applications de santé mobile ;
- **Équipement médical intelligent** : dispositifs d'imagerie connectés, pompes à perfusion et lits d'hôpital améliorant l'efficacité clinique ;
- **Gestion des médicaments** : distributeurs de pilules intelligents et outils de suivi de l'adhérence ;
- **Vie assistée ambiante (AAL)** : systèmes favorisant l'autonomie des personnes âgées et handicapées, tels que la détection de chutes et les solutions d'éclairage adaptatif.

1.4.2 Secteurs industriels et logistiques (Internet Industriel des Objets IIoT)

L'Internet des Objets Industriels (IIoT) est un pilier angulaire de la Quatrième Révolution Industrielle, intégrant des systèmes cyber-physiques, l'analyse de données massives et le cloud computing pour permettre des usines intelligentes et une logistique intelligente. En convergeant la technologie opérationnelle (OT) et la technologie de l'information (IT), l'IIoT stimule l'automatisation, la flexibilité et la prise de décision basée sur les données dans les secteurs de la fabrication, de l'énergie et de la logistique.

La mise en œuvre pratique de l'IIoT se reflète dans plusieurs domaines clés des opérations industrielles, y compris [33–35] :

- **Maintenance prédictive** : prévision des défaillances d'équipement basée sur des capteurs, réduction des temps d'arrêt et optimisation des calendriers de maintenance ;
- **Optimisation de la chaîne d'approvisionnement** : suivi des actifs en temps réel, automatisation des inventaires et planification des itinéraires logistiques ;
- **Contrôle de la qualité** : inspection automatisée, détection des défauts et surveillance de la conformité ;
- **Gestion de l'énergie** : surveillance et optimisation de la consommation d'énergie industrielle et de l'empreinte carbone ;

- **Sécurité et conformité** : surveillance de la sécurité des travailleurs, détection des dangers et suivi de la conformité environnementale.

Les récentes avancées, telles que l'informatique de périphérie, les jumeaux numériques et la connectivité 5G, renforcent encore la réactivité, l'évolutivité et la résilience de ces systèmes. Par ailleurs, l'association de l'IIoT avec l'intelligence artificielle (IA) apparaît comme un catalyseur majeur, permettant d'exploiter pleinement le potentiel des données industrielles.

Tableau 1.1 *– Résultats empiriques issus de l'application de l'IIoT et des technologies connexes*

Source	**Apport ou bénéfice mesuré**
Deloitte (2023) [36]	Les entreprises digitalement matures se révèlent plus résilientes et compétitives.
Capgemini Research Institute (2021) [37]	Gains moyens de 15% en efficacité opérationnelle et plus de 25% d'amélioration globale des systèmes ; citation de la NTU (Singapour) avec 31% d'économies d'énergie et 9,6 kt CO_2 évitées grâce aux jumeaux numériques.
Maserati [37]	Réduction des coûts et délais de développement de 30%.
Rolls-Royce [37]	Prolongation de 50% des intervalles de maintenance.
Schneider Electric [38]	Diminution de 10% des coûts énergétiques.
McKinsey (2021) [38]	Réduction des défauts produits de 1 à 4% selon les secteurs industriels.

Les résultats du tableau 1.1 confirment que, bien que les bénéfices puissent varier selon les contextes, l'IIoT, combiné à l'IA et aux technologies associées, constitue un levier de performance mesurable et significatif.

1.4.3 Infrastructures urbaines et villes intelligentes (Smart Cities)

L'idée de villes intelligentes ("smart cities") dépasse largement la simple digitalisation des infrastructures urbaines. Elle représente une évolution dans la gestion et l'organisation du cadre urbain, où des technologies de pointe notamment l'Internet des objets (IoT), l'intelligence artificielle (IA) et l'analyse de données sont mises en œuvre pour construire des écosystèmes urbains durables, performants et axés sur les citoyens [39, 40].

Les villes intelligentes visent à fusionner le développement économique et la responsabilité écologique en intégrant l'intelligence dans des infrastructures essentielles comme l'énergie, la mobilité et les services publics. Elles encouragent également la résilience, l'inclusion et le bien-être de la société.

1.5 Architecture de l'Internet des Objets (IoT)

L'architecture de l'Internet des objets (IoT) ne repose pas sur un modèle universel figé, mais sur une diversité de cadres conceptuels et technologiques élaborés par différents organismes de normalisation. Cette pluralité s'explique par la nature hétérogène et en constante évolution de l'IoT, qui englobe des domaines variés des réseaux industriels aux applications médicales , chacun imposant ses propres contraintes de performance, de sécurité et d'interopérabilité.

Ainsi, plutôt que d'imposer une architecture unique, les organisations internationales (ISO/IEC, IEEE, ITU-T, IIC, oneM2M, AIOTI, etc.) ont développé des *architectures de référence*, des *frameworks conceptuels* et des *modèles fonctionnels* destinés à guider la conception, le déploiement et la gouvernance des systèmes IoT. Ces modèles visent à établir un vocabulaire commun, à assurer l'interopérabilité entre écosystèmes hétérogènes et à promouvoir la fiabilité et la sécurité dans les infrastructures connectées.

Les architectures de référence diffèrent par leur **portée** (générique ou sectorielle), leur **niveau d'abstraction** (conceptuel, fonctionnel ou technique) et leur **objectif principal** (interopérabilité, performance, confiance ou convergence OT/IT). Le **tableau 1.2** ci-dessous présente un comparatif synthétique des principaux standards et cadres normatifs proposés pour l'IoT, illustrant la complémentarité de leurs approches.

Cette comparaison met en évidence que les architectures IoT actuelles ne sont pas concurrentes, mais plutôt *complémentaires et interopérables*. Elles reflètent une vision modulaire et évolutive du système IoT, où chaque norme contribue à un niveau particulier d'abstraction depuis les modèles conceptuels jusqu'aux implémentations industrielles. Dans la section suivante, nous examinerons plus en détail les modèles d'architecture fonctionnelle les plus couramment utilisés, notamment les architectures à trois et cinq couches.

1.5.1 Architecture stratifiée à trois couches

L'architecture typique des systèmes IoT s'articule autour d'une structuration hiérarchique tripartite, conceptualisée selon un modèle stratifié (*layered architecture*). Ce modèle vise à séparer les préoccupations fonctionnelles de manière systématique, tout en assurant modularité, évolutivité et interopérabilité [47–49]. Dans sa configuration classique, il distingue trois couches principales :

1.5.1.1 Couche de perception (Physical Layer)

La couche de perception constitue l'interface fondamentale entre le monde physique et l'infrastructure numérique. Elle intègre des dispositifs de transduction capables de convertir les grandeurs physiques (température, pression, mouvement, signaux biométriques, etc.) en données numériques exploitables. Cette couche englobe ainsi un large éventail de capteurs environnementaux, biométriques, industriels ou contextuels ainsi que des actionneurs permettant d'exercer une rétroaction directe sur l'environnement physique [50, 51].

Tableau 1.2 *– Comparatif des principales normes et standards d'architecture IoT*

Standard	**Norme & Référence**	**Type d'Architecture**	**Portée & Domaine**	**Modèle Architectural Principal**	**Composants / Vues Clés et Finalité**
ISO/IEC 30141 :2018	ISO/IEC 30141 :2018 [41]	Architecture de Référence (Générique)	Systèmes IoT génériques ; approche descendante pour architectures spécifiques	Modèle conceptuel (CM) + Modèle de référence (RM) + Vues multiples	4 vues : fonctionnelle, déploiement système, réseau, usage + 6 domaines (PED, SCD, ASD, OMD, UD, RAID). **Finalité** : vocabulaire commun et modularité.
IIC IIRA v1.10	Industrial Internet Reference Architecture v1.10 [42]	Architecture de Référence (Spécifique au domaine industriel)	Internet Industriel des Objets (IIoT)	4 points de vue alignés sur ISO/IEC/IEEE 42010	Vues : business, usage, fonctionnelle, implémentation. **Différenciateur** : convergence OT/IT et patterns d'implémentation spécialisés.
IEEE P2413	IEEE P2413 [43]	Framework Architectural (Unificateur)	Multi-domaines (santé, bâtiments, énergie, transport, fabrication)	Framework extensible avec modèle de référence intégré	Modèle de référence + blueprint pour abstraction des données + "quadruple" de confiance. **Finalité** : interopérabilité inter-domaines.
ITU-T Y.2060	ITU-T Y.2060 (06/2012) [44]	Modèle de Référence (Fondamental)	Vue d'ensemble de l'infrastructure IoT mondiale	Modèle à 4 couches (Application, Service, Réseau, Dispositif)	Capacités transversales : sécurité, gestion, identification. **Différenciateur** : approche télécoms et vision business globale.
oneM2M TS-0001	oneM2M TS-0001-V4.23.0 (2024) [45]	Architecture Fonctionnelle (Couche de Services Commune)	Systèmes M2M/IoT indépendants du réseau sous-jacent	Modèle à 3 couches : Application, CSE, NSE	Entités fonctionnelles : AE, CSE, NSE + points de référence (Mca, Mcc, Mcn). **Finalité** : interopérabilité RESTful et sécurité de bout en bout.
AIOTI HLA R5.0	AIOTI High Level Architecture Release 5.0 (2020) [46]	Architecture de Haut Niveau (HLA)	Systèmes IoT généraux, applicables aux grands pilotes (LSP)	Modèle à 3 couches : Application, IoT, Réseau	Entités : App Entity, IoT Entity, Networks + 5 interfaces (AppIoTNetwork). **Différenciateur** : approche centrée sur la "chose", intelligence distribuée, compatibilité cloud/edge.

1.5.1.2 Couche réseau (Network Layer)

La couche réseau assure la connectivité et l'acheminement des données issues de la perception vers les infrastructures de traitement et de stockage. Elle mobilise un ensemble hétérogène de technologies de communication, aussi bien filaires que sans fil, sélectionnées et optimisées en fonction des contraintes de portée, de débit, de consommation énergétique et de fiabilité. Cette couche représente un maillon critique, puisqu'elle conditionne à la fois la performance, la résilience et la sécurité des flux de données au sein de l'écosystème IoT [48, 49, 52].

1.5.1.3 Couche application (Service Layer)

La couche applicative regroupe la logique métier et les services de haut niveau. Elle a pour rôle de transformer les données brutes en informations à forte valeur ajoutée, destinées à la prise de décision et à l'interaction avec les utilisateurs finaux. Cette couche intègre des fonctionnalités avancées telles que l'analyse prédictive, l'apprentissage automatique, la gestion des services numériques, ainsi que l'interopérabilité avec les systèmes d'information existants [48, 50, 51]. Elle constitue ainsi l'interface directe entre les infrastructures IoT et les besoins opérationnels ou sociétaux auxquels elles répondent.

1.6 Paradigmes de calcul distribué dans l'IoT et l'IIoT

1.6.1 Cloud Computing : le noyau analytique centralisé

Le *cloud computing* constitue le pilier centralisé et fondamental du modèle architectural de l'Internet des objets (IoT). Il désigne la mise à disposition à la demande de ressources informatiques évolutives serveurs, stockage, bases de données, réseaux et outils analytiques via Internet à partir de centres de données distants à grande échelle [53]. Dans le contexte de l'IoT, le cloud agit comme le réservoir ultime et le moteur analytique pour les immenses flux de données générés par les dispositifs distribués [54]. Sa valeur principale réside dans sa capacité à exécuter des tâches computationnelles intensives qui ne sont pas critiques en temps réel, mais qui exigent une vision globale et une profondeur historique importante [55]. Ces tâches incluent l'analyse de tendances à long terme, l'entraînement de modèles complexes d'apprentissage automatique, ainsi que l'agrégation de données à grande échelle provenant d'ensembles étendus de dispositifs connectés. Le cloud offre également des services essentiels de niveau entreprise tels que la gestion des utilisateurs, la visualisation des données et l'intégration avec d'autres systèmes métiers (ERP, CRM) [56]. Cependant, ce modèle présente des limites intrinsèques : une latence élevée liée à la distance avec les centres de données, une forte consommation de bande passante, et des vulnérabilités potentielles dues à la centralisation [57]. Ainsi, bien qu'indispensable pour la consolidation et l'analyse globale, le cloud reste inadapté aux applications exigeant une prise de décision autonome et en temps réel à la périphérie du réseau.

1.6.2 Edge Computing : intelligence et autonomie de proximité

L'*edge computing* (ou informatique en périphérie) représente une évolution majeure, visant à décentraliser le traitement des données en le rapprochant physiquement et logiquement des sources de génération [58]. Ce paradigme pallie les limites du modèle cloud en effectuant des traitements analytiques directement sur les dispositifs IoT ou à proximité immédiate. L'edge computing opère selon différents niveaux de proximité. Au niveau *extrême edge*, l'intelligence est intégrée dans les capteurs, actionneurs et contrôleurs, capables d'exécuter des tâches telles que le filtrage de données, la détection d'anomalies ou le contrôle local avec une latence de quelques millisecondes [59]. À un niveau supérieur, les passerelles *edge* agrègent les données de plusieurs nœuds locaux pour effectuer des analyses plus complexes, des conversions de protocoles ou des compressions avant de transmettre au cloud uniquement les informations pertinentes [60].

Les avantages de l'*edge computing* sont multiples :

- une réduction drastique de la latence, garantissant la réactivité en temps réel ;
- une diminution significative de la bande passante utilisée grâce au prétraitement local ;
- une résilience accrue des opérations, permettant la continuité de service même en cas de coupure réseau [61].

Ce paradigme est aujourd'hui incontournable dans les systèmes industriels, les véhicules autonomes et les environnements à criticité temporelle [62].

1.6.3 Fog Computing : la couche intermédiaire hiérarchique

Le *fog computing* introduit une couche intermédiaire hiérarchique située entre les dispositifs de périphérie et le cloud. Il s'agit d'une extension distribuée du paradigme du cloud vers les réseaux locaux, visant à rapprocher la puissance de calcul des lieux de production des données [63]. Le fog ne remplace pas l'*edge computing* mais agit comme une couche de coordination et d'orchestration pour plusieurs nœuds périphériques [64]. Les nœuds de brouillard (*fog nodes*), souvent situés dans des infrastructures locales (usines, micro-centres urbains), possèdent une capacité de calcul et de stockage supérieure à celle des dispositifs edge. Ils assurent des services tels que la corrélation de données issues de multiples sources, le traitement d'événements complexes, ou encore la gestion locale des ressources (par exemple, l'équilibrage énergétique dans une microgrille) [65]. En servant de point d'agrégation et de traitement intermédiaire, la couche fog réduit la charge sur le cloud, améliore la latence et permet de déployer des applications d'intelligence locale distribuée [66]. Elle s'avère essentielle pour des scénarios nécessitant à la fois contextualisation, réactivité et scalabilité.

1.7 Paysage des menaces IoT/IIoT : analyse multi-couches

Cette section dresse une cartographie structurée des menaces touchant l'IoT/IIoT en s'alignant sur l'architecture en couches et sur les paradigmes de calcul présentés précédemment (Cloud/Edge/Fog). Elle s'appuie sur la littérature académique et les cadres

de référence reconnus afin de relier vulnérabilités techniques, acteurs malveillants, secteurs critiques et orientations de défense [67–71].

1.7.1 Menaces au niveau des dispositifs (Perception)

Au niveau *perception/dispositifs*, les attaques exploitent les contraintes matérielles et logicielles : identifiants par défaut, mécanismes de mise à jour insuffisants, expositions physiques, faiblesses cryptographiques embarquées. Les compromissions de firmware, les attaques par force brute d'identifiants et les vecteurs *side-channel* sont typiques. La littérature met en évidence la persistance de ces failles dans des environnements hétérogènes et distribués [72, 73]. L'épisode *Mirai* illustre l'exploitation massive d'identifiants par défaut pour l'enrôlement de dispositifs dans un botnet [74].

1.7.2 Menaces au niveau réseau (Communication)

La couche réseau est ciblée via l'exploitation de protocoles légers (p. ex. MQTT, CoAP) lorsqu'ils sont mal configurés ou insuffisamment protégés, via des interceptions *man-in-the-middle*, du brouillage sélectif ou des attaques de déni de service distribuées. Les attaques contre les protocoles et piles de communication (Wi-Fi, LTE/5G) par exemple la réinstallation de clés (KRACK) soulignent la centralité de la sécurité des liaisons et de la segmentation réseau [75–77].

1.7.3 Menaces au niveau application et services

Dans la couche application/plateforme (cloud, *mobile*, *web*), les vulnérabilités classiques (API peu sûres, authentification/autorisation faibles, validation d'entrée insuffisante) se combinent à des problématiques d'intégration et d'orchestration (edge/fog/cloud). Les conséquences incluent injections (SQL/XSS), détournements de sessions et manipulations de données [67, 78]. La mise en corrélation de flux multisources, si elle n'est pas protégée, peut amplifier la portée d'une compromission locale.

1.7.4 Acteurs et capacités

Le paysage des menaces mêle acteurs étatiques (APT), organisations cybercriminelles, hacktivistes et opportunistes. Les premiers se distinguent par la persistance, l'outillage avancé et l'appétence pour les infrastructures critiques ; les seconds industrialisent les campagnes (extorsion, botnets, fraude) ; les hacktivistes exploitent l'exposition des services publics ; les opportunistes capitalisent sur les défauts de configuration [79–81].

1.7.5 Secteurs critiques et profils de risques

Les profils de risques varient selon les secteurs : l'IoMT (santé) présente un risque d'impact *safety-critical* ; l'IIoT (manufacture/énergie) confronte la convergence IT/OT ; les villes intelligentes subissent l'exposition d'une surface publique étendue ; l'énergie et les réseaux intelligents cumulent dépendances systémiques [81–84].

***Tableau 1.3** – Typologie synthétique des acteurs de la menace IoT*

Catégorie	**Motivations**	**Cibles/Caractéristiques**
Étatiques/APT	Renseignement, sabotage	Infrastructures critiques, opérations persistantes, outillage sur mesure [79, 81]
Cybercriminalité	Gain financier, rançongiciels	Santé, PME/ETI, plateformes cloud ; opérations à l'échelle [80]
Hacktivisme	Message politique	Services publics, *smart cities* ; perturbation/visibilité
Opportunistes	Expérimentation, ressources	IoT domestique/consommateur ; scans/automatisation

***Tableau 1.4** – Exemples de profils sectoriels (exposition × impact)*

Secteur	**Préoccupations clés**	**Scénarios d'impact**
Santé (IoMT)	Protocoles hérités, données patients	Sécurité des patients, continuité des soins [82]
Industrie (IIoT)	Convergence IT/OT, systèmes legacy	Arrêts de production, fuite PI [81]
Villes intelligentes	Échelle, accessibilité publique	Rupture de services, vie privée [83]
Énergie/Utilities	Dépendances critiques	Instabilité réseau, effets en cascade [84]

1.7.6 Cadres et bonnes pratiques

Les cadres de référence structurent l'analyse et l'amélioration de la posture de sécurité : l'OWASP IoT Top 10 fournit une grille de lecture des faiblesses récurrentes côté écosystème/applications ; les modèles d'architecture (p. ex. SGAM pour *smart grid*) mettent en évidence les dépendances inter-couches et les exigences d'interopérabilité et d'authentification/attestation [85, 86]. Leur usage outille la priorisation des contrôles et la vérification de conformité.

1.7.7 Dynamiques émergentes : 5G, IA offensive, chaîne d'approvisionnement

Trois dynamiques renforcent l'exposition : (i) la 5G accroît la densité de connexion et déporte des fonctions réseau, élargissant la surface d'attaque ; (ii) l'IA offensive facilite l'automatisation de la découverte/exploitation de vulnérabilités ; (iii) les risques *supply chain* (micrologiciels, composants, services tiers) contournent les défenses périmétriques [69, 71]. Ces tendances plaident pour des approches *zero-trust*, l'attestation de bout en bout et la vérification continue.

Le tableau 1.5 récapitule les vulnérabilités typiques par couche pour guider la priorisation des contrôles.

***Tableau 1.5** – Vulnérabilités typiques par couche et exemples*

Couche	**Vulnérabilités récurrentes**	**Exemples documentés**
Dispositifs	Identifiants par défaut, mises à jour non sûres, faiblesses crypto, exposition physique	Botnet *Mirai* (identifiants) [74] ; analyses [72, 73]
Réseau	Protocoles non chiffrés/mal configurés, segmentation insuffisante, attaques MITM/-DoS	KRACK (WPA2) [76] ; synthèses [75, 77]
Application/Services	API non sécurisées, authN/authZ faibles, validation d'entrée insuffisante	Revue et recommandations [67, 78]

1.8 Conclusion

L'évolution de l'Internet des objets (IoT) a conduit à la formation d'un écosystème numérique vaste, interconnecté et distribué, reposant sur une architecture stratifiée et des paradigmes de calcul complémentaires tels que le *cloud*, le *fog* et l'*edge computing*. Ces infrastructures, conçues pour assurer connectivité, traitement et réactivité, favorisent l'intelligence distribuée mais introduisent également une complexité systémique et de nouvelles surfaces d'exposition aux menaces.

L'analyse des vulnérabilités et des risques, conduite selon les différentes couches de l'architecture IoT, met en évidence la multiplicité et l'interdépendance des points d'attaque depuis les dispositifs physiques jusqu'aux plateformes applicatives. Les menaces contemporaines se caractérisent par leur nature évolutive, leur automatisation croissante et la convergence des domaines IT, OT et IoT, rendant les approches de sécurité statiques ou périmétriques insuffisantes.

Dans ce contexte, la sécurité de l'IoT ne peut plus être envisagée comme un ensemble de mécanismes ponctuels, mais comme un système dynamique intégrant l'adaptation continue, la surveillance proactive et l'apprentissage automatique. C'est dans cette perspective que s'inscrit le chapitre suivant, consacré à la **détection d'anomalies dans les systèmes IoT**. Ce dernier abordera les fondements conceptuels, les modèles d'analyse et les approches d'apprentissage adaptatif permettant d'identifier, d'interpréter et d'anticiper les comportements anormaux au sein de réseaux connectés complexes.

Chapitre 2

Anomalies et Détection dans l'Internet des Objets

Sommaire

2.1 Introduction 26
2.2 Lien entre l'IoT et les anomalies 27
2.2.1 Définition et conceptualisation des anomalies dans l'IoT . . 27
2.2.2 Les catégories d'anomalies dans les systèmes IoT 27
2.2.3 Le rôle de la détection d'anomalies dans les environnements IoT 28
2.3 Hiérarchisation conceptuelle des anomalies systémiques : défaillances, intrusions et anomalies comportementales . 29
2.3.1 Défaillance : une violation de la fiabilité 29
2.3.2 Intrusion : une violation de la sécurité 29
2.3.3 Anomalie comportementale : une déviation observable . . . 29
2.4 Détection d'anomalies sous contraintes : défis de sécurité dans les environnements IoT 30
2.4.1 Hétérogénéité architecturale et topologique 30
2.4.2 Contraintes de ressources 30
2.4.3 Complexités centrées sur les données 31
2.5 De la Détection Statique vers l'Intelligence Adaptative 31
2.5.1 Limites structurelles des approches traditionnels 31
2.5.1.1 Définition 2.1 Détection par seuil fixe 31
2.5.1.2 Proposition 2.1 Insuffisance du modèle statique . 32
2.5.2 Fondements théoriques de l'apprentissage adaptatif 32
2.5.2.1 Définition 2.2 Concept drift 32
2.5.2.2 Formulation de la mise à jour des paramètres . . . 33
2.5.3 Principaux algorithmes d'apprentissage pour l'adaptation intelligente 33
2.5.3.1 Apprentissage automatique classique (Machine Learning) 33
2.5.3.2 Réseaux de neurones profonds (Deep Neural Networks, DNN) 34
2.5.3.3 Réseaux de neurones sur graphes (Graph Neural Networks, GNN) 34
2.6 Renforcement de la cybersécurité dans l'Internet des Véhicules (IoV) : détection d'anomalies et d'intrusions par Deep Learning 34

2.6.1 Contexte et motivation 34
2.6.2 Synthèse de travaux récents sur la sécurité de l'IoV 35
2.6.3 Paysage des attaques dans l'Internet des Véhicules 36
2.6.4 Jeu de données CICIoV2024 37
2.6.5 Prétraitement des données 37
2.6.5.0.1 Label Encoding 38
2.6.5.0.2 Feature Selection 38
2.6.5.0.3 Feature Scaling 38
2.6.5.0.4 One-Hot Encoding 38
2.6.5.0.5 Train-Test Split 39
2.6.6 Architecture du modèle DNN 39
2.6.7 Configurations de classification et résultats expérimentaux . 39
2.6.8 Analyse comparative avec des modèles classiques 41
2.7 Conclusion 42

2.1 Introduction

L'essor de l'Internet des Objets (IoT) a profondément transformé les systèmes cyber-physiques en multipliant les dispositifs connectés, hétérogènes et distribués. Cette expansion s'accompagne toutefois d'une augmentation significative de la surface d'attaque, d'une complexification des dépendances entre composants et d'une variabilité permanente des données collectées. Dans cet environnement mouvant, la *détection d'anomalies* constitue un levier central pour préserver la fiabilité opérationnelle, garantir l'intégrité des données et renforcer la sécurité des infrastructures. Elle permet d'identifier, précocement et de manière contextualisée, des dérives de comportement susceptibles de révéler des défaillances, des intrusions ou de simples changements d'usage.

Ce chapitre propose un cadre conceptuel et méthodologique unifié pour appréhender les anomalies dans l'IoT. Nous commençons par définir et situer la notion d'anomalie au regard des spécificités des environnements connectés, puis nous établissons une taxonomie distinguant clairement trois catégories de phénomènes systémiques : *défaillances*, *intrusions* et *anomalies comportementales*. Cette hiérarchisation terminologique, articulée aux contraintes propres à l'IoT (hétérogénéité des architectures, ressources limitées, multimodalité et dérive des données), fournit la base nécessaire à l'analyse critique des approches de détection.

Sur cette base, nous mettons en évidence les limites des techniques traditionnelles (règles, seuils statiques, signatures) face à la non-stationnarité et à la diversité contextuelle des flux IoT. Nous motivons alors le passage vers des approches *adaptatives* et *intelligentes* capables d'apprendre des représentations robustes, d'intégrer l'information contextuelle et d'évoluer en continu. Trois familles de solutions sont au cœur de cette transition :

- l'apprentissage automatique (*machine learning*) pour des modèles légers et interprétables,
- l'apprentissage profond (DNN/CNN) pour la capture de structures non linéaires et de motifs hiérarchiques, et
- les réseaux de neurones sur graphes (GNN) pour exploiter la topologie des communications et les dépendances entre objets.

Enfin, le chapitre présente les principes théoriques de l'apprentissage adaptatif (formulations, mise à jour incrémentale des paramètres, prise en compte de la dérive), avant d'introduire les algorithmes et architectures qui seront mobilisés et comparés dans la suite du manuscrit. L'objectif est double : fournir au lecteur un socle conceptuel rigoureux pour raisonner sur les anomalies en contexte IoT, et justifier, sur des bases solides, le choix d'approches d'apprentissage modernes adaptées aux contraintes réelles des systèmes connectés.

2.2 Lien entre l'IoT et les anomalies

2.2.1 Définition et conceptualisation des anomalies dans l'IoT

Une anomalie, ou valeur aberrante, est généralement définie comme tout point de données, événement ou comportement s'écartant significativement des normes attendues ou établies [87]. Dans les environnements de l'Internet des objets (IoT), les anomalies ne se limitent pas à des valeurs statistiquement extrêmes ; elles reflètent souvent des déviations fonctionnelles, opérationnelles ou sécuritaires. Yasarathna, Liyanage et Le-Khac (2025) [88] définissent la détection d'anomalies comme l'identification de comportements inattendus ou suspects dans les données réseau. Cook, Msrl et Fan (2020) [89] la décrivent comme les conséquences mesurables d'un changement inattendu dans l'état d'un système au-delà de sa norme locale ou globale. Cette perspective souligne la nature contextuelle des anomalies dans l'IoT : un comportement perçu comme anormal dans un dispositif peut être parfaitement normal dans un autre, selon son contexte opérationnel. Cette complexité découle de l'hétérogénéité intrinsèque de l'IoT, de son architecture distribuée et des capacités de calcul limitées de ses composants. La frontière entre comportement normal et anormal devient alors dynamique et difficile à définir [90].

2.2.2 Les catégories d'anomalies dans les systèmes IoT

Selon la littérature [89, 90], les anomalies dans l'IoT peuvent être regroupées en trois catégories principales :

- **Anomalies ponctuelles** : se produisent lorsqu'un seul point de données s'écarte considérablement du reste de l'ensemble. Par exemple, une augmentation soudaine de la température ou de l'utilisation de la bande passante peut indiquer un dysfonctionnement du capteur ou une intrusion réseau.
- **Anomalies contextuelles** : dépendent du contexte temporel ou environnemental. Une valeur peut être normale dans certaines conditions mais anormale dans d'autres un phénomène courant dans les séries temporelles générées par les dispositifs IoT.
- **Anomalies collectives** : apparaissent lorsqu'une séquence de points de données, bien qu'individuellement normaux, représente collectivement un comportement anormal. Ces anomalies révèlent souvent des irrégularités corrélées ou des attaques coordonnées.

Cette typologie montre la nécessité de mécanismes de détection capables d'interpréter les corrélations, motifs et dynamiques évolutives entre appareils et dans le temps. La figure 2.1 représente une classification hiérarchique des trois catégories principales

d'anomalies : ponctuelles (points isolés), contextuelles (dépendantes du contexte), et collectives (séquences anormales). Cette taxonomie illustre la complexité de la détection dans les environnements IoT hétérogènes.

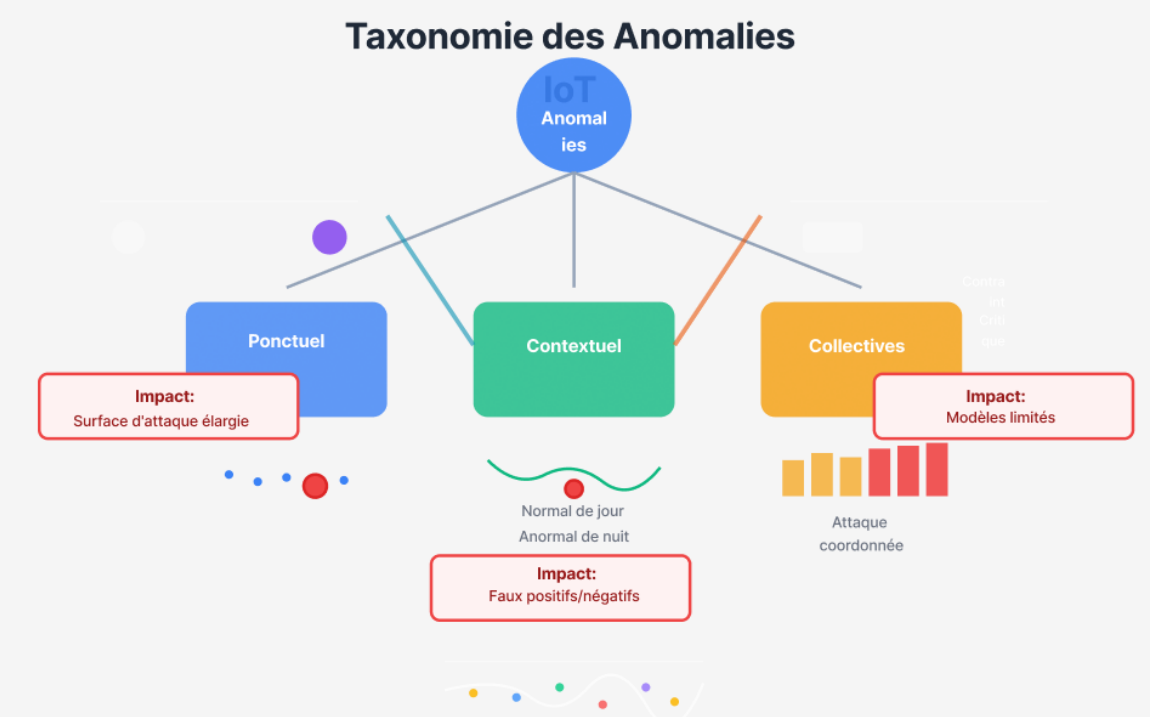

Figure 2.1 *– Taxonomie des anomalies dans l'IoT*

2.2.3 Le rôle de la détection d'anomalies dans les environnements IoT

La détection d'anomalies n'est pas simplement un outil d'identification des irrégularités ; elle constitue une fonction stratégique essentielle pour le fonctionnement, l'optimisation et la sécurité des systèmes IoT [91]. Ses principaux apports incluent :

- **Sécurité renforcée** : l'identification d'écarts dans le trafic ou le comportement des appareils permet de détecter des intrusions et de prévenir des menaces émergentes [92].
- **Maintenance prédictive** : la détection précoce d'écarts dans les performances des machines favorise la maintenance proactive, réduisant les temps d'arrêt et les coûts [93].
- **Assurance qualité des données** : la détection des dysfonctionnements de capteurs et incohérences de données garantit l'intégrité des flux collectés [94].
- **Optimisation des ressources** : l'identification des usages inefficaces ou excessifs améliore la gestion énergétique [95].
- **Gestion énergétique intelligente** : dans les bâtiments et villes intelligents, la détection d'anomalies soutient l'efficacité énergétique [96].
- **Surveillance de la santé et de l'environnement** : la détection de lectures anormales issues de capteurs physiologiques ou environnementaux peut déclencher des interventions rapides [97, 98].

- **Réduction des coûts opérationnels** : en logistique, la détection d'anomalies en temps réel optimise les chaînes d'approvisionnement [99].

En somme, les anomalies dans les systèmes IoT représentent des écarts révélateurs de défaillances, d'inefficacités ou de violations de sécurité. Leur détection précise est donc essentielle au maintien de l'intégrité, de la fiabilité et de la résilience des infrastructures connectées. La nature dynamique et multidimensionnelle des données IoT fait de la détection d'anomalies un défi technique majeur, mais aussi un pilier clé pour des écosystèmes plus sûrs et adaptatifs.

2.3 Hiérarchisation conceptuelle des anomalies systémiques : défaillances, intrusions et anomalies comportementales

Un lexique précis est essentiel pour les disciplines de la fiabilité et de la cybersécurité. Clarifier les distinctions entre défaillance, intrusion et anomalie comportementale établit une base conceptuelle solide pour des diagnostics robustes et des mécanismes de réponse efficaces.

2.3.1 Défaillance : une violation de la fiabilité

Une défaillance est un événement où un système ou un composant cesse de fournir un service correct pour en délivrer un incorrect, échouant ainsi à remplir sa fonction prévue [100]. Elle représente une violation fondamentale de la fiabilité du système. Le domaine de la défaillance relève principalement de la fiabilité et de la disponibilité des systèmes, sans intention malveillante. Les causes sont souvent non intentionnelles : pannes matérielles (disque, mémoire), erreurs logicielles (pointeur nul, conditions de course), ou épuisement des ressources. La question clé du diagnostic devient alors : le système fonctionne-t-il conformément à ses exigences fonctionnelles ?

2.3.2 Intrusion : une violation de la sécurité

Une intrusion désigne un acte malveillant réussi contournant les contrôles de sécurité d'un système pour compromettre sa confidentialité, son intégrité ou sa disponibilité [101, 102]. Elle s'inscrit dans le champ de la cybersécurité et implique des causes intentionnelles : exploitation de vulnérabilités, accès non autorisé via des identifiants compromis, ou attaques par ransomware. La question centrale devient : un adversaire a-t-il franchi le périmètre de sécurité du système ?

2.3.3 Anomalie comportementale : une déviation observable

Une anomalie comportementale est une déviation mesurable par rapport à un comportement de référence établi pour un système, un réseau ou un utilisateur [103]. Elle constitue une observation, non un diagnostic : elle peut signaler une défaillance, une intrusion [104] ou simplement un changement bénin dans l'activité du système. La question fondamentale devient : le comportement observé s'écarte-t-il du modèle normatif attendu ?

2.4 Détection d'anomalies sous contraintes : défis de sécurité dans les environnements IoT

La prolifération de l'Internet des objets (IoT) a entraîné une transformation profonde des systèmes en réseau, marquée par l'expansion massive de dispositifs intelligents interconnectés et distribués. Cette convergence a permis de nombreuses applications de l'automatisation industrielle aux villes intelligentes , mais a également introduit un niveau inédit de complexité et de vulnérabilité. Comme le soulignent Hosseinzadeh et Sinopoli (2023) [105], la tension fondamentale réside dans la sécurisation de systèmes conçus avant tout pour la fonctionnalité et l'interopérabilité, plutôt que pour la cybersécurité. Ainsi, les paradigmes classiques de détection conçus dans des environnements homogènes et riches en ressources se révèlent inadaptés aux contraintes IoT [106]. Les défis de détection peuvent être décrits selon trois dimensions interdépendantes :

- **(1) l'hétérogénéité architecturale**,
- **(2) les contraintes de ressources**,
- **(3) la complexité des données**.

2.4.1 Hétérogénéité architecturale et topologique

L'Internet des objets n'est pas une infrastructure uniforme, mais un ensemble dynamique de sous-systèmes hétérogènes différant par leurs architectures matérielles, protocoles et environnements d'exploitation. Ce manque d'uniformité complique l'établissement de politiques de sécurité cohérentes et crée une surface d'attaque élargie, comportant de multiples points de vulnérabilité [107]. Les dispositifs IoT varient non seulement par leur puissance de calcul allant des passerelles aux capteurs contraints mais aussi par leurs versions logicielles, leurs rôles réseau et leurs paramètres de sécurité. Les bases de référence comportementales deviennent vite obsolètes dans un contexte dynamique, nécessitant des systèmes d'apprentissage continu. De plus, la diversité organisationnelle et les domaines de confiance multiples exigent des cadres de détection distribués ou fédérés capables d'opérer dans des environnements non fiables.

2.4.2 Contraintes de ressources

La majorité des dispositifs IoT opèrent sous de fortes contraintes de calcul, de mémoire et d'énergie, limitant l'usage de mécanismes complexes. Les limitations computationnelles restreignent l'usage de modèles profonds sur les appareils à faibles ressources, notamment pour l'inférence en temps réel [106]. Les contraintes de mémoire et d'énergie freinent l'enregistrement de données historiques ou la transmission continue pour analyse distante. Comme le notent Anthi et ses collègues, ces contraintes obligent à privilégier des modèles légers ou heuristiques, souvent au détriment de la précision. Les approches hybrides combinant inférence locale et analyse cloud offrent un compromis prometteur, mais posent de nouveaux défis de latence et de confidentialité [104].

2.4.3 Complexités centrées sur les données

Les environnements IoT génèrent des flux de données hétérogènes, multimodaux et rapides, souvent bruités, incomplets et dépendants du contexte [103]. Trois défis majeurs se posent :

- **Multimodalité** : les données proviennent de capteurs, réseaux ou journaux système aux caractéristiques distinctes. Leur corrélation requiert des techniques avancées de fusion souvent trop coûteuses pour les nœuds à faibles ressources.
- **Volatilité et dérive conceptuelle** : les comportements "normaux" évoluent avec les conditions d'usage, rendant obsolètes les modèles entraînés sur données historiques. Sans adaptation, cela conduit à des taux élevés de fausses alarmes [107].
- **Rareté des données étiquetées** : la collecte d'ensembles annotés reste rare dans l'IoT pour des raisons de confidentialité et de coût, d'où l'intérêt croissant pour les approches non ou semi-supervisées [108].

Enfin, les mécanismes de préservation de la vie privée, tels que l'apprentissage fédéré, bien qu'indispensables, peuvent masquer des informations contextuelles clés pour la détection fine des comportements anormaux. Ce compromis entre confidentialité et précision reste l'un des dilemmes majeurs de la recherche actuelle.

En résumé, la détection d'anomalies dans les environnements IoT constitue un champ spécifique de la cybersécurité, distinct des approches traditionnelles. La diversité architecturale, la rareté des ressources et la complexité des données exigent des systèmes légers, adaptatifs et contextuels capables d'opérer sous contrainte et incertitude. Comme le rappellent Hosseinzadeh et Sinopoli (2023) [105], les futures solutions devront être aussi dynamiques et diversifiées que les écosystèmes qu'elles visent à protéger.

2.5 De la Détection Statique vers l'Intelligence Adaptative

2.5.1 Limites structurelles des approches traditionnels

Les systèmes de détection d'anomalies traditionnels s'appuient sur des règles prédéfinies et des seuils statiques, supposant implicitement que le comportement nominal du système reste stationnaire et parfaitement caractérisable *a priori*. Cette hypothèse, rarement vérifiée dans les environnements IoT réels, constitue une limitation fondamentale [108, 109].

2.5.1.1 Définition 2.1 Détection par seuil fixe

Soit $x \in \mathbb{R}^d$ un vecteur de caractéristiques observées à l'instant t, où d représente la dimensionnalité de l'espace des caractéristiques (par exemple $d = 5$ pour {trafic réseau, température CPU, mémoire, latence, taux d'erreurs}). Soit $f : \mathbb{R}^d \to \mathbb{R}$ une fonction de score d'anomalie (par exemple une moyenne, une variance ou une distance euclidienne). La règle de détection s'exprime formellement comme suit :

$$\text{Détection}(x) = \begin{cases} 1, & \text{si } f(x) > \tau, \\ 0, & \text{sinon.} \end{cases} \tag{2.1}$$

où $\tau \in \mathbb{R}$ est un seuil prédéfini, généralement fixé empiriquement ou par expertise métier.

2.5.1.2 Proposition 2.1 Insuffisance du modèle statique

Ce paradigme présente une faiblesse structurelle : le seuil τ demeure constant même lorsque la distribution sous-jacente des données $P(X)$ évolue dans le temps. Or, dans un réseau IoT hétérogène, plusieurs facteurs induisent une non-stationnarité intrinsèque [109, 110] :

- déploiement dynamique de nouveaux capteurs → changement de $P(X)$;
- variations circadiennes et saisonnières → périodicité temporelle ;
- dégradation matérielle progressive → dérive graduelle ;
- modifications contextuelles (climat, charge réseau, conditions physiques) → variabilité multimodale.

Ainsi, si $P_t(X) \neq P_{t+\Delta}(X)$, alors :

1. **Faux positifs :** si la distribution se déplace vers des valeurs plus élevées, des observations normales peuvent dépasser τ ;
2. **Faux négatifs :** si le bruit augmente, des anomalies réelles peuvent passer sous τ.

Cette inadéquation est mesurable par la divergence de KullbackLeibler [111] :

$$D_{KL}(P_t \parallel P_{t+\Delta}) = \int P_t(x) \log \frac{P_t(x)}{P_{t+\Delta}(x)}\, dx. \tag{2.2}$$

Lorsque $D_{KL} > \varepsilon$, le modèle devient obsolète. De telles dérives sont fréquentes dans les systèmes connectés où la dynamique des flux rend impossible l'utilisation de seuils fixes [108, 110].

2.5.2 Fondements théoriques de l'apprentissage adaptatif

Un modèle adaptatif se distingue par sa capacité à ajuster ses paramètres de manière incrémentale pour refléter les changements dans la distribution des données. Il repose sur le principe de la descente de gradient stochastique, fondement de la plupart des modèles modernes d'apprentissage profond [112, 113].

2.5.2.1 Définition 2.2 Concept drift

Le *concept drift* caractérise l'évolution temporelle de la distribution conditionnelle des données :

$$P_t(Y \mid X, C) \neq P_{t+\Delta}(Y \mid X, C), \tag{2.3}$$

où :

- $X \in \mathbb{R}^d$: vecteur de caractéristiques observées ;
- $Y \in \{0, 1\}$: étiquette binaire (0 = normal, 1 = anomalie) ;
- $C \in \Omega$: contexte opérationnel (type de capteur, localisation, heure, environnement).

Cette non-stationnarité peut être abrupte, graduelle, incrémentale ou récurrente, selon la nature du phénomène [110].

Dans ce cadre, l'*apprentissage adaptatif* ne désigne pas un algorithme particulier, mais un principe général consistant à permettre à un modèle d'ajuster ses paramètres en fonction des évolutions des données et du contexte. Il s'agit d'un paradigme transversal qui englobe à la fois les approches de *Machine Learning* (ML) et celles de *Deep Learning* (DL), incluant notamment les réseaux de neurones profonds (DNN) et les réseaux de neurones sur graphes (GNN). Ces modèles constituent le socle des systèmes de détection intelligents

2.5.2.2 Formulation de la mise à jour des paramètres

La mise à jour adaptative est exprimée par la règle d'optimisation suivante [112, 113] :

$$\theta_{t+1} = \theta_t - \eta \frac{\partial L(X_t, Y_t; \theta_t)}{\partial \theta_t}, \tag{2.4}$$

où :

- θ_t représente le vecteur des paramètres (poids, biais) à l'instant t ;
- $L(X_t, Y_t; \theta_t)$ est la fonction de perte (par ex. cross-entropy) ;
- $\eta > 0$ est le taux d'apprentissage régulant la vitesse d'adaptation.

Cette relation traduit le principe d'ajustement incrémental : à chaque itération, le modèle se déplace dans la direction opposée au gradient de la perte afin de minimiser l'erreur de prédiction. Elle constitue le cœur de l'apprentissage adaptatif et permet aux modèles de conserver leur efficacité face aux dérives temporelles et contextuelles dans les environnements IoT [112, 113].

2.5.3 Principaux algorithmes d'apprentissage pour l'adaptation intelligente

Les stratégies d'apprentissage adaptatif peuvent être implémentées à travers plusieurs familles d'algorithmes selon la nature des données, la disponibilité d'étiquettes et la complexité du problème. Les plus utilisées dans les systèmes IoT relèvent de trois grandes catégories : apprentissage automatique classique (ML), apprentissage profond (DNN/CNN) et apprentissage sur graphes (GNN).

2.5.3.1 Apprentissage automatique classique (Machine Learning)

Les méthodes de Machine Learning constituent la première génération d'outils de détection [108, 109]. Elles apprennent une frontière de décision à partir d'un ensemble de données étiquetées ou non. Les modèles les plus courants incluent :

- **Support Vector Machines (SVM)** : construction d'un hyperplan séparateur maximal entre classes ;
- **Random Forest (RF)** : ensemble d'arbres de décision améliorant la robustesse ;
- **K-Nearest Neighbors (KNN)** : classification selon la majorité des voisins les plus proches ;
- **Isolation Forest** : approche non supervisée efficace pour les anomalies rares.

Ces algorithmes sont rapides et adaptés aux dispositifs IoT contraints, mais leur capacité d'adaptation reste limitée face à la dérive conceptuelle.

2.5.3.2 Réseaux de neurones profonds (Deep Neural Networks, DNN)

Les DNN représentent une évolution majeure du ML en apprenant automatiquement des représentations hiérarchiques des données [114]. Chaque couche extrait des caractéristiques de plus haut niveau (structure spatiale, dépendances temporelles). Les architectures les plus utilisées pour la détection d'anomalies IoT incluent :

- réseaux entièrement connectés ;
- réseaux convolutionnels (CNN) ;
- autoencodeurs pour la détection non supervisée.

Ces modèles capturent des corrélations non linéaires complexes et s'adaptent aux variations contextuelles, ce qui les rend particulièrement pertinents dans des environnements IoT dynamiques.

2.5.3.3 Réseaux de neurones sur graphes (Graph Neural Networks, GNN)

Les environnements IoT étant naturellement structurés en graphes, chaque capteur ou dispositif peut être modélisé comme un nœud, et les communications comme des arêtes. Les GNN exploitent cette structure pour propager et agréger l'information entre nœuds connectés [115]. Leur mise à jour repose sur le mécanisme de *message passing* :

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot \mathrm{AGG}\{h_u^{(l)} : u \in \mathcal{N}(v)\}\right), \tag{2.5}$$

où $h_v^{(l)}$ représente l'état du nœud v à la couche l, et $\mathcal{N}(v)$ l'ensemble de ses voisins.

2.6 Renforcement de la cybersécurité dans l'Internet des Véhicules (IoV) : détection d'anomalies et d'intrusions par Deep Learning

2.6.1 Contexte et motivation

L'Internet des Véhicules (IoV) s'impose comme un prolongement naturel de l'Internet des Objets (IoT) appliqué aux systèmes de transport intelligents. Il repose sur l'interconnexion de véhicules, d'infrastructures routières et de services cloud afin d'améliorer la sécurité routière, la gestion du trafic et l'expérience de conduite [116]. Au coeur de cette architecture se trouve le bus *Controller Area Network* (CAN), qui permet la communication temps réel entre les unités de contrôle électroniques (ECU) et les différents capteurs et actionneurs du véhicule [117].

Cette intégration massive de communications numériques ouvre cependant la voie à de nouvelles vulnérabilités. La surface d'attaque des véhicules modernes s'étend désormais bien au-delà des interfaces physiques traditionnelles, incluant l'accès via les ports de diagnostic, les interfaces sans fil, les liaisons V2X (Vehicle-to-Everything) et les services connectés [118, 119]. Des attaques telles que l'injection de messages sur le

bus CAN, les dénis de service (DoS) ou les attaques de *spoofing* peuvent compromettre des fonctions critiques et menacer directement la sécurité des passagers.

Les systèmes de détection d'intrusion (IDS) classiques, souvent fondés sur des signatures ou des règles statiques, peinent à suivre l'évolution et la complexité des menaces dans un environnement aussi dynamique que l'IoV [120]. Dans ce contexte, les approches d'apprentissage profond apparaissent comme une voie prometteuse : elles permettent de modéliser des comportements normaux complexes, d'identifier des anomalies subtiles et de s'adapter à de nouvelles formes d'attaques [121].

Le travail présenté dans cette section s'inscrit dans cette dynamique et propose un cadre de détection d'intrusions basé sur un réseau de neurones profond (Deep Neural Network, DNN), entraîné et évalué sur un jeu de données récent et réaliste dédié à l'IoV : le *CICIoV2024*.

2.6.2 Synthèse de travaux récents sur la sécurité de l'IoV

Les recherches récentes sur la cybersécurité de l'IoV couvrent plusieurs axes complémentaires : détection d'anomalies, IDS spécifiques au bus CAN, sécurité des architectures Fog/Edge, et cadres de forensic réseau.

Dans [122], les auteurs s'intéressent aux réseaux IoV assistés par le Fog (*Fog-assisted IoV*). Ils identifient plusieurs zones encore peu explorées et proposent un mécanisme de détection d'anomalies basé sur un autoencodeur convolutionnel. Le modèle est évalué sur la base NSL-KDD à l'aide de la F1-score, avec des performances supérieures aux approches antérieures, ce qui illustre la pertinence des architectures profondes pour la détection temps réel.

Le travail [123] introduit *CANShield*, un cadre de détection d'intrusion au niveau signal pour le bus CAN. L'architecture se décompose en trois modules : un prétraitement pour transformer les flux de trames CAN en séries temporelles exploitables, un analyseur basé sur plusieurs autoencodeurs profonds, puis un module de décision fondé sur un ensemble de modèles. Ce type d'approche montre que le passage du niveau trame au niveau signal enrichit les informations disponibles pour la détection.

Dans un contexte plus large de systèmes industriels, [124] compare un empilement de modèles de deep learning à onze algorithmes de machine learning classiques (*XGBoost*, *Random Forest*, etc.) appliqués à des systèmes SCADA. Contrairement à l'intuition, *XGBoost* se révèle le plus performant, illustrant le fait que les réseaux profonds ne dominent pas systématiquement, et que le choix du modèle doit être guidé par la nature des données.

Sur le plan de la sécurité globale de l'IoV, [125] propose une revue détaillée de l'architecture de l'IoV, des vulnérabilités des réseaux intra- et inter-véhiculaires, ainsi que des différentes familles d'IDS proposées pour ce domaine. Les critères de conception et d'évaluation des IDS sont discutés, mettant en évidence l'importance de prendre en compte l'hétérogénéité des protocoles et les contraintes temps réel.

Par ailleurs, [126] examine l'apport de l'intelligence artificielle à l'investigation et à la *network forensics* sur différents terrains (IoT, Cloud, *smart grid*, forensique véhicule). L'étude met en avant l'intérêt des approches hybrides combinant règles expertes, machine learning et deep learning pour reconstruire les scénarios d'attaque.

Des travaux plus spécifiquement orientés IoV, comme [127–129], proposent des cadres d'IDS basés sur des modèles d'arbres (*Random Forest*, *XGBoost*, *LightGBM*) ou des architectures hybrides, et soulignent l'importance de la sélection de caractéristiques pour réduire la complexité sans dégrader la détection.

Enfin, plusieurs contributions analysent la surface d'attaque globale des systèmes IoV, en insistant sur le rôle central du bus CAN et des ECU [130–132]. Elles convergent vers la nécessité de combiner durcissement des protocoles, segmentation, IDS embarqués et analyse comportementale.

Le cadre proposé dans cette section prolonge ces travaux en combinant : (i) un jeu de données récent spécifiquement conçu pour l'IoV (*CICIoV2024*), (ii) un prétraitement rigoureux des flux CAN, et (iii) un DNN optimisé pour la classification multi-classes des attaques de type DoS et *spoofing*.

2.6.3 Paysage des attaques dans l'Internet des Véhicules

L'IoV repose sur la coopération de multiples composants : ECU, capteurs, bus internes, liaisons V2X, infrastructures et services distants. Chacun de ces éléments constitue une surface d'attaque potentielle. En s'appuyant sur la littérature existante [130–132], on peut structurer ces surfaces d'attaque selon le tableau 2.1 qui sont inspirées de [131, 132].

Tableau 2.1 *– Surfaces d'attaque, vulnérabilités et impacts potentiels dans les systèmes automobiles*

Surface d'attaque	**Type de vulnérabilité**	**Exemples d'attaques et impacts**
ECU (invasif)	Dommages physiques, modification ou injection de code, abus du port de diagnostic	Dysfonctionnements critiques du véhicule, atteinte à la confidentialité, dommages matériels
ECU (non invasif)	*Side-channel*, attaques par force brute sur authentification ou identifiants	Vol de données, accès non autorisé aux fonctions logicielles
Capteurs	Interférences, *spoofing*, brouillage, attaques adversariales sur les signaux	Perception erronée de l'environnement, décisions inappropriées, risque d'accident
Liens intra-véhiculaires	Vulnérabilités de communication (filaire/sans fil), absence de chiffrement ou de segmentation	*Sniffing*, DoS, usurpation d'identité, rejouement, attaques *Man-in-the-Middle*
Liens inter-véhiculaires (V2X)	Vulnérabilités des protocoles V2V/V2I, gestion d'identité, authentification	Écoute, usurpation, DoS, perturbation de la signalisation coopérative
Infrastructure et services	Compromission d'équipements ou services tiers, clés ou identités compromises	Décisions erronées, interruptions de service, escalade d'attaque sur de larges zones

Cette typologie met en évidence la nature systémique des risques : une compromission localisée (par exemple d'un ECU ou d'un capteur) peut se propager via le bus CAN ou les liaisons V2X, et affecter des fonctions critiques (direction, freinage, assistance à la conduite). Les systèmes de détection d'intrusion doivent donc intégrer une

vision globale des flux, tout en restant compatibles avec les contraintes embarquées.

2.6.4 Jeu de données CICIoV2024

Le jeu de données *CICIoV2024* a été récemment publié par le *Canadian Institute for Cybersecurity* (CIC) de l'Université du Nouveau-Brunswick [133]. Il fournit des traces de communications CAN collectées en 2024 sur les ECU d'un véhicule Ford de 2019, dans un environnement contrôlé reproduisant différents scénarios d'attaque.

Le jeu de données contient des sessions distinctes correspondant à des situations normales (benign) et à plusieurs types d'attaques ciblant le bus CAN : attaques de déni de service (DoS) et attaques de *spoofing* portant sur diverses composantes (vitesse, régime moteur, position du volant, commande d'accélérateur, etc.). Les données sont disponibles en plusieurs formats (décimal, binaire, hexadécimal), ce qui offre une grande flexibilité pour la représentation des caractéristiques.

Le tableau 2.2 résume la distribution des classes telle qu'utilisée dans l'étude [133].

Tableau 2.2 *– Distribution des classes dans le jeu de données CICIoV2024.*

Catégorie	**Sous-catégorie**	**Nombre d'observations**
Benign	–	1 223 337
Attaques (DoS)	–	74 663
Attaques (Spoofing)	Steering Wheel	19 977
	RPM	54 900
	GAS	9 991
	Speed	24 951

Cette granularité permet de considérer plusieurs scénarios de classification : binaire (benign vs attaque), multi-classes agrégées (benign, DoS, spoofing) et multi-classes détaillées (benign, DoS, spoofing par composante).

2.6.5 Prétraitement des données

Avant l'entraînement du modèle de deep learning, un pipeline de prétraitement est appliqué (voir figure 2.2) afin de garantir la qualité et l'exploitabilité des données. Les opérations principales sont les suivantes :

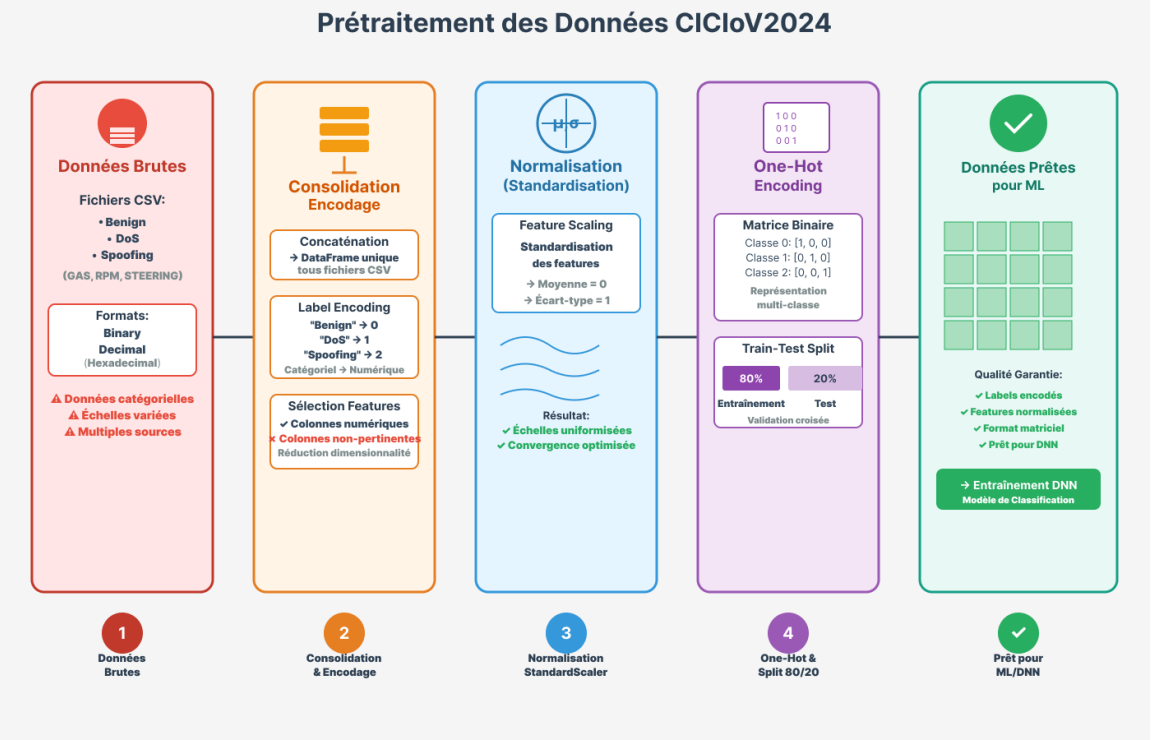

Figure 2.2 *– Prétraitement des données CICIoV2024*

2.6.5.0.1 Label Encoding Les étiquettes catégorielles décrivant les types de trafic (par exemple `Benign`, `DoS`, `Spoofing_Speed`, ...) sont d'abord converties en valeurs entières à l'aide du *Label Encoding*. Cette étape fournit une représentation numérique compacte des classes, nécessaire pour la manipulation interne et la construction ultérieure de cibles de classification [134].

2.6.5.0.2 Feature Selection Certaines colonnes ne contribuent pas directement à la tâche de détection (par exemple des identifiants non utilisés, des champs textuels ou des horodatages bruts). Une phase de *Feature Selection* est donc menée pour ne conserver que les caractéristiques pertinentes et purement numériques qui serviront d'entrée au DNN. Cette sélection permet de réduire la dimensionnalité, de limiter le bruit et de diminuer le risque de surapprentissage.

2.6.5.0.3 Feature Scaling Les caractéristiques retenues présentent souvent des échelles très différentes. Un *Feature Scaling* de type standardisation est appliqué : pour chaque caractéristique, on soustrait la moyenne et on divise par l'écart-type, de sorte que les données aient une moyenne nulle et une variance unitaire. La transformation adoptée est la suivante :

$$x_{\text{standardized}} = \frac{x - \mu}{\sigma} \tag{2.6}$$

où x désigne la valeur initiale, μ la moyenne et σ l'écart-type de la caractéristique. Cette étape facilite l'entraînement du DNN et évite qu'une caractéristique à grande amplitude ne domine les autres.

2.6.5.0.4 One-Hot Encoding Les étiquettes entières obtenues par *Label Encoding* sont transformées en vecteurs binaires via le *One-Hot Encoding*. Chaque classe

est associée à une dimension du vecteur, avec la valeur 1 pour la classe réelle et 0 pour les autres. Cette représentation est particulièrement adaptée à l'apprentissage supervisé en classification multi-classes, notamment lorsque la couche de sortie du réseau utilise une activation *softmax* [134].

2.6.5.0.5 Train-Test Split Enfin, le jeu de données est partitionné en deux sous-ensembles : un ensemble d'entraînement (80 %) et un ensemble de test (20 %). Cette séparation permet d'évaluer de manière fiable la capacité de généralisation du modèle sur des exemples jamais vus.

2.6.6 Architecture du modèle DNN

Le modèle proposé est un réseau de neurones profond (*Deep Neural Network*) de type séquentiel, composé de plusieurs couches entièrement connectées. L'architecture est conçue pour capturer des relations non linéaires complexes tout en restant suffisamment compacte pour être envisagée dans une intégration embarquée.

Le tableau 2.3 résume la structure du modèle.

Tableau 2.3 *– Architecture du modèle DNN pour la classification des attaques IoV.*

Couche (type)	**Taille de sortie**	**Commentaires**
Dense 1	128 unités	Couche d'entrée, activation ReLU, régularisation L2 ($\lambda = 0.001$)
Dense 2	64 unités	Activation ReLU, L2
Dense 3	32 unités	Activation ReLU, L2
Dense 4	32 unités	Activation ReLU, L2
Couche de sortie	6 unités	Activation *softmax* (6 classes : benign + 5 types d'attaque)

La fonction d'activation *Rectified Linear Unit* (ReLU), définie par $f(x) = \max(0, x)$, est utilisée dans les couches cachées pour introduire de la non-linéarité et faciliter la convergence. La régularisation L2 (*weight decay*) pénalise les poids de grande amplitude et contribue à améliorer la robustesse du modèle. La couche de sortie exploite une activation *softmax*, permettant d'interpréter la sortie comme une distribution de probabilité sur les classes.

Le modèle est entraîné avec une fonction de perte de type entropie croisée catégorielle, et optimisé à l'aide d'un optimiseur de type descente de gradient adaptative (par exemple Adam). Les métriques suivies durant l'entraînement et l'évaluation incluent l'accuracy, la précision, le rappel et la F1-score.

2.6.7 Configurations de classification et résultats expérimentaux

Trois configurations de classification sont étudiées afin d'évaluer la capacité d'adaptation du modèle :

- **Classification binaire** : séparation entre trafic *benign* et trafic malveillant (toutes attaques confondues) ;
- **Classification à 3 classes** : {Benign, Spoofing, DoS} ;
- **Classification à 6 classes** : {Benign, DoS, Spoofing_RPM, Spoofing_Speed, Spoofing_SteeringWheel, Spoofing_GAS}.

Chaque configuration est évaluée dans deux représentations de données (format décimal et format binaire) issues du jeu de données *CICIoV2024*. Pour chaque modèle, des matrices de confusion sont générées et analysées afin d'identifier les éventuelles confusions résiduelles entre classes.

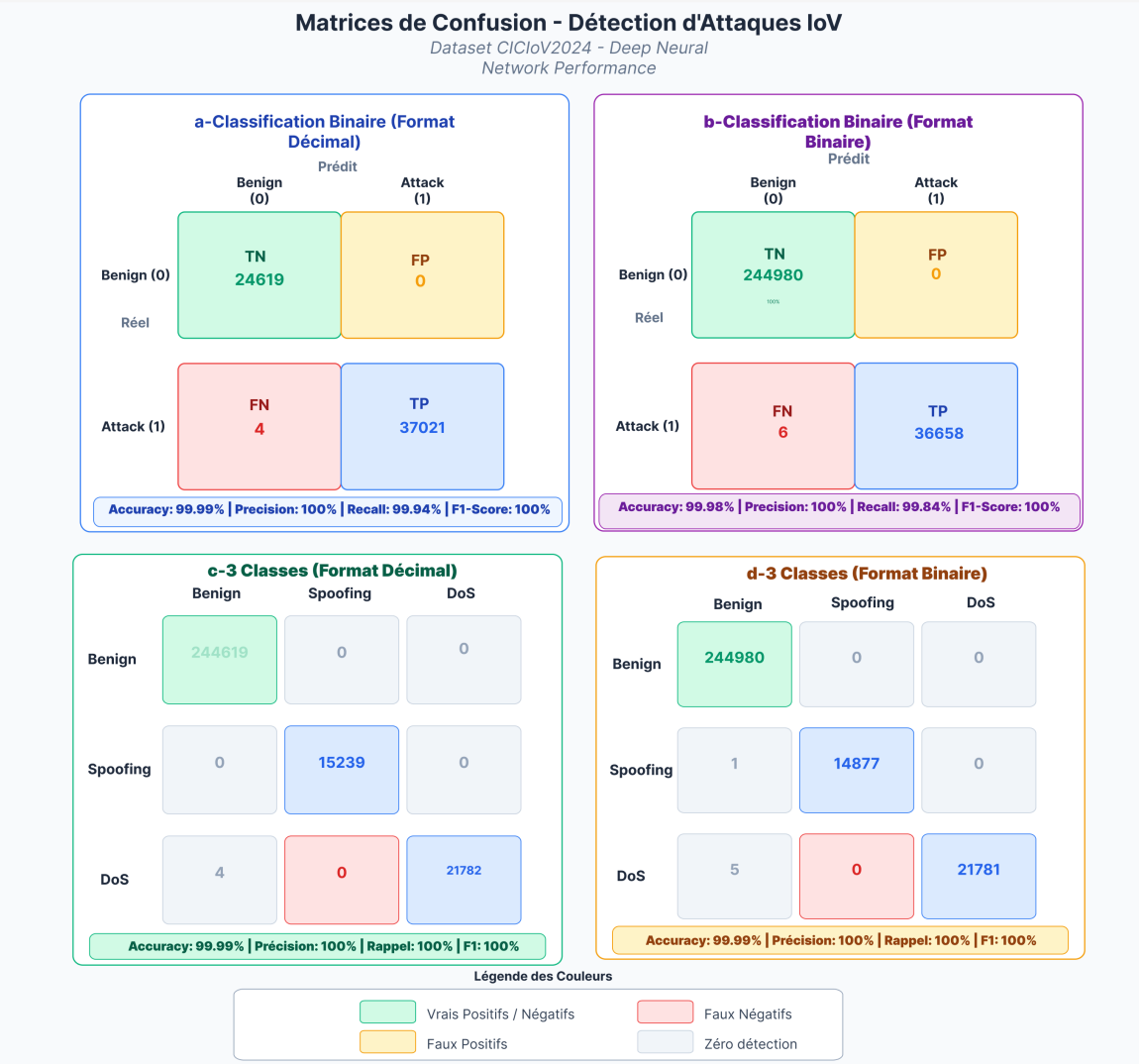

Figure 2.3 *– Matrices de confusion obtenues pour la détection d'attaques dans l'Internet of Vehicles (IoV), évaluées sur le jeu de données CICIoV2024 à laide dun réseau neuronal profond (DNN). Les quatre blocs correspondent à : (a) classification binaire en format décimal, (b) classification binaire en format binaire, (c) classification à trois classes en format décimal, (d) classification à trois classes en format binaire. Les métriques affichées (Accuracy, Précision, Rappel, F1-score)*

Comme illustré dans la figure 2.4, le passage à six classes permet dévaluer la capacité du modèle à distinguer finement plusieurs sous-catégories dattaques Spoofing

et DoS. Les performances obtenues avec une précision et un F1-score avoisinant les 100% confirment la capacité du DNN à capturer des schémas comportementaux complexes dans les environnements IoV.

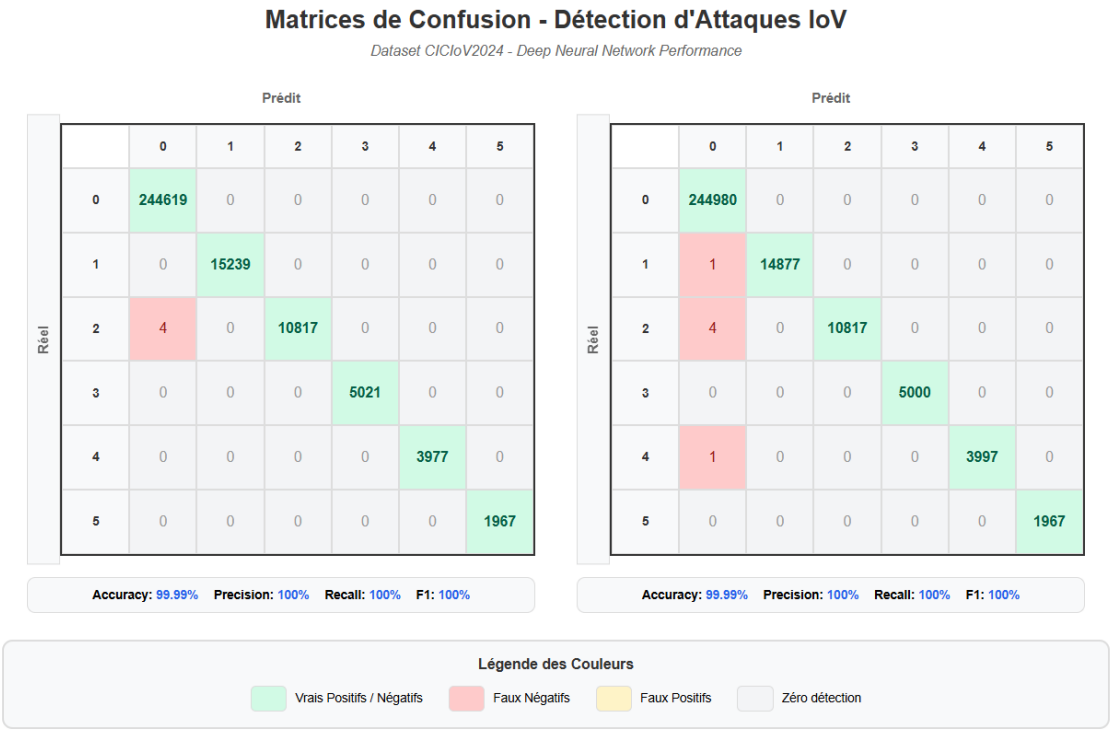

Figure 2.4 *– Matrices de confusion obtenues pour la classification en six classes.*

Les résultats montrent que, pour l'ensemble des scénarios, le modèle DNN atteint des performances très élevées, avec des accuracy, précisions, rappels et F1-score proches de 100% pour la majorité des classes. La modélisation multi-classes fine (6 classes) ne dégrade pas significativement les performances, ce qui indique que le réseau parvient à distinguer efficacement les motifs spécifiques inhérents à chaque type d'attaque.

2.6.8 Analyse comparative avec des modèles classiques

Afin de positionner le modèle proposé par rapport à des approches plus classiques, une étude comparative est menée avec plusieurs algorithmes de machine learning supervisé : *Logistic Regression*, *AdaBoost* et *Random Forest*. Les modèles sont entraînés et évalués dans des conditions similaires, en utilisant les mêmes jeux d'entraînement et de test.

Le tableau 2.4 synthétise les résultats globaux (accuracy, recall, precision, F1-score) pour les formats binaire et décimal. Ces résultats sont inspirés de [135].

Tableau 2.4 *– Comparaison des performances globales entre le DNN proposé et des modèles de machine learning classiques.*

Modèle	**Accuracy**	**Recall**	**Precision**	**F1**
Format binaire				
DNN (proposé)	0.99	1.00	1.00	1.00
Logistic Regression	0.95	0.68	0.74	0.63
AdaBoost	0.87	0.17	0.14	0.15
Random Forest	0.95	0.68	0.60	0.62
Format décimal				
DNN (proposé)	0.99	1.00	1.00	1.00
Logistic Regression	0.89	0.50	0.48	0.49
AdaBoost	0.92	0.66	0.48	0.51
Random Forest	0.96	0.76	0.76	0.76

On constate que le DNN surpasse nettement les autres modèles en termes d'accuracy globale et de F1-score, dans les deux représentations de données. *Random Forest* reste compétitif, en particulier sur le format décimal, mais ne parvient pas à atteindre le niveau de performance du réseau profond, notamment pour les classes d'attaque les moins représentées.

Ces résultats confirment l'intérêt de recourir au deep learning pour la détection d'intrusions dans l'IoV, à condition de disposer d'un jeu de données suffisamment riche et de procédures de prétraitement adaptées.

Les performances obtenues, très élevées sur les différents scénarios de classification, suggèrent que les DNN constituent une brique prometteuse pour des IDS embarqués capables de détecter des attaques de type DoS et *spoofing* sur le bus CAN. Les perspectives mentionnées dans l'article original incluent la mise en œuvre temps réel et l'intégration dans des architectures Edge/Fog, ce qui rejoint plus largement les enjeux de sécurité adaptative et de détection proactive d'anomalies abordés dans le reste de ce chapitre.

2.7 Conclusion

Dans la continuité de ces fondements, les approches d'apprentissage supervisé constituent une étape essentielle vers la détection intelligente. Elles permettent de formaliser le processus d'adaptation en définissant explicitement une fonction d'apprentissage à partir de données étiquetées, tout en intégrant les principes d'optimisation et de régularisation présentés précédemment.

Chapitre 3

Détection d'anomalies pour l'IIoT : analyse du dataset Edge-IIoTset avec des distributions de classes variées

Sommaire

3.1 Introduction 43
3.2 Travaux connexes 46
3.3 Sécurité de l'IIoT : menaces, vulnérabilités et défis . . . 48
3.4 Cadre conceptuel de l'étude 49
3.4.1 Réseaux de neurones profonds 49
3.4.2 Réseaux de neurones convolutionnels 51
3.4.3 Mesures d'évaluation 52
3.5 Modèle proposé 53
3.5.1 Jeu de données 53
3.5.2 Approches expérimentales 54
3.5.2.1 Classification binaire 55
3.5.2.2 Classification multiclasse 55
3.6 Résultats et discussion 59
3.7 Conclusion 63

3.1 Introduction

L'Internet des Objets (IdO) est décrit comme l'interconnexion de multiples appareils utilisant des identifiants uniques pour partager des données et autres informations pertinentes à travers un réseau sans l'intervention de personnes [136]. Grâce à des dispositifs de détection, tout objet physique peut être facilement actionné, réduisant ainsi le besoin de main-d'œuvre humaine [137]. Les applications de l'IdO ont été déployées dans pratiquement tous les domaines, allant de la santé et l'agriculture aux transports et à la fabrication. Ces applications ont révolutionné et transformé les industries en connectant les appareils, en collectant des données et en permettant des processus de décision intelligents [138]. De plus, à mesure que le monde industriel évolue vers des systèmes plus avancés et complexes, le besoin d'Internet des Objets Industriel (IdOI) a émergé. L'IdOI reprend les principes de l'IdO et les applique spécifiquement aux processus industriels, permettant la surveillance à distance, l'analyse intelligente et le contrôle des opérations industrielles. Il introduit un niveau supérieur

d'automatisation, d'évolutivité et d'efficacité, répondant aux défis et aux exigences uniques du secteur de la fabrication. Grâce à l'IdOI, les industries peuvent optimiser la production, améliorer l'utilisation des ressources et accroître la performance opérationnelle globale [139].

De plus, l'IdOI contribue à améliorer l'efficacité opérationnelle des industries grâce à la convergence entre les technologies de l'information (TI) et les technologies opérationnelles (TO). Les TI représentent l'infrastructure informatique traditionnelle que les entreprises utilisent pour gérer leurs données et leurs systèmes. Les TO désignent les technologies opérationnelles utilisées pour contrôler et surveiller les processus physiques dans les usines et autres environnements industriels. La convergence des TI et des TO permet d'atteindre de nouveaux niveaux d'automatisation, d'efficacité et de productivité dans le secteur manufacturier [140]. Par exemple, les dispositifs IdOI peuvent être utilisés pour surveiller l'état des équipements et prédire les défaillances potentielles [141]. Ces informations permettent de planifier une maintenance préventive, d'éviter des interruptions de production non planifiées coûteuses et d'améliorer la fiabilité globale des équipements, renforçant ainsi l'interconnectivité des systèmes [142].

Par ailleurs, l'intégration de l'IdOI et des Systèmes Cyber-Physiques (SCP) provoque un changement de paradigme supplémentaire dans le secteur manufacturier. Un SCP est un système qui intègre les mondes physique et virtuel, en favorisant la connectivité entre eux [143]. Ces systèmes représentent un domaine transformationnel où les mondes physique et cybernétique s'entrelacent de manière complexe, imprégnant le paysage opérationnel d'une intelligence et d'une efficacité accrues [144]. Ils utilisent des capteurs et des actionneurs pour collecter des données du monde physique et des logiciels pour analyser et agir sur ces données, promouvant une connectivité transparente [145]. L'intégration de l'IdOI et des SCP permet de nouveaux niveaux de collaboration entre les dispositifs intelligents, renforçant ainsi l'interconnectivité entre eux [146].

De plus, la croissance rapide de l'IdOI (IIoT) a entraîné une augmentation massive des appareils connectés, accroissant exponentiellement le volume de données générées [147]. Cette masse de données présente de nouveaux défis pour les entreprises en matière de sécurité des données et de détection d'anomalies. Compte tenu du volume important d'informations transmises sur les réseaux, garantir la confidentialité, l'intégrité et la disponibilité des données de l'IdOI est primordial pour protéger les infrastructures critiques et les informations sensibles [148].

Les systèmes IdOI interconnectés élargissent la surface d'attaque et les rendent vulnérables à des violations de sécurité potentielles. Les pirates et les attaquants malveillants peuvent exploiter des vulnérabilités dans les réseaux et les appareils pour obtenir un accès non autorisé, falsifier des données et perturber les opérations commerciales [149]. Par conséquent, des mesures de sécurité robustes doivent être mises en œuvre pour protéger l'écosystème de l'IdOI [150].

La détection d'anomalies dans les données est un autre aspect majeur qui doit être abordé dans le contexte de l'IdOI. Les anomalies, qui peuvent être des écarts par rapport aux modèles ou comportements normaux, peuvent indiquer des vulnérabilités ou des inefficacités potentielles en matière de sécurité [151]. En tirant parti de techniques d'analyse avancées, telles que l'intelligence artificielle (IA) et l'apprentissage profond, les organisations peuvent développer des modèles de détection d'anomalies efficaces capables d'identifier les anomalies, généralement en temps réel. Ces modèles peuvent détecter des anomalies dans le trafic de données, le comportement des appa-

reils ou les performances du système, permettant une réponse proactive aux menaces ou incidents potentiels [152].

Il est crucial de souligner certaines statistiques préoccupantes concernant les logiciels malveillants et les problèmes de cybersécurité. La propagation des logiciels malveillants d'un employé à un autre est passée de 61 % en 2020 à 75 % en 2022. Soixante-neuf pour cent des experts en cybersécurité affirment que l'équipe dédiée à la cybersécurité de leur entreprise est en sous-effectif. Avec 2,75 milliards d'attaques au cours des six premiers mois de 2022, le nombre de nouveaux types de logiciels malveillants détectés devrait être similaire à celui de l'année précédente. Les ransomwares et les logiciels malveillants pour l'IdO deviennent plus prévalents, avec 40 millions d'attaques par ransomware attendues par mois au premier semestre 2022 [153]. Bien que le nombre de nouvelles attaques par logiciels malveillants ait diminué en 2020 pour la première fois depuis 2015, selon le Rapport sur les Cybermenaces 2022 de SonicWall, les attaques par logiciels malveillants devraient atteindre 10,4 millions par an, un niveau équivalent à celui de 2018 [154]. De plus, selon le Rapport de Transparence de Google, 3,849 millions d'avertissements de navigation ont été affichés à des utilisateurs tentant d'accéder à des sites web potentiellement dangereux, et 1,6 million de personnes ont reçu des avertissements concernant des sites web potentiellement nuisibles dans leurs résultats de recherche [155].

Ces statistiques illustrent l'importance croissante de mesures de sécurité robustes dans le paysage de l'IdOI [156], telles que l'IA et l'apprentissage profond. La combinaison de ces technologies améliore la protection des données, la détection d'anomalies et les capacités d'atténuation des menaces, permettant aux entreprises de faire face à la marée montante des attaques par logiciels malveillants, des ransomwares et des tentatives de hameçonnage. Face à un paysage de la cybersécurité de plus en plus complexe, les entreprises peuvent se défendre de manière proactive contre les menaces émergentes, protéger les infrastructures critiques et garantir l'intégrité et la disponibilité des données de l'IdOI en déployant l'IA et l'apprentissage profond.

Les contributions principales de notre recherche sont présentées ci-dessous :

- Introduction d'un nouveau modèle combiné d'apprentissage profond, intégrant les Réseaux Neuronaux Convolutionnels (CNN) et les Réseaux Neuronaux Profonds (DNN), qui démontre des performances supérieures.
- Utilisation d'un jeu de données contemporain, connu sous le nom d'Edge-IIoTset, pour faciliter l'entraînement et l'évaluation du modèle MLCNNDNN proposé. Des distributions multiclasses variées sont introduites et analysées.
- L'efficacité du modèle proposé est évaluée dans des scénarios de classification binaire et multiclasse.
- L'évaluation de la performance de notre modèle intègre plusieurs métriques, incluant la justesse (accuracy) et la précision.

Le reste du chapitre est organisé comme suit : les travaux connexes seront présentés dans la section 2. La section 3 expose la sécurité de l'IdOI : menaces, vulnérabilités et défis. La section 4 fournit le cadre contextuel de l'étude. La section 5 décrit la solution proposée, basée sur l'apprentissage profond et les algorithmes de réseaux neuronaux convolutionnels. La section 6 présente les résultats et la discussion. Enfin, nous concluons notre travail et suggérons des pistes de recherche futures pour la détection d'anomalies dans l'environnement IdOI.

3.2 Travaux connexes

Dans des études récentes, les chercheurs ont proposé différentes approches pour traiter les vulnérabilités et les brèches de cybersécurité dans l'environnement de l'IIoT. Dans [157], les auteurs proposent une solution nommée NIDS-CNNLSTM (modèle de classification de détection d'intrusion basé sur l'apprentissage profond) pour relever les défis du taux de détection faible, de la précision de classification et du taux de fausse détection dans les modèles traditionnels de détection d'intrusion dans l'environnement de l'Internet Industriel des Objets (IIoT). NIDS-CNNLSTM combine un CNN (Réseau Neuronal Convolutif) à deux couches pour l'extraction de caractéristiques spatiales et un LSTM (Mémoire à Long Court Terme) unidirectionnel à deux couches pour l'extraction de caractéristiques temporelles. Le modèle est testé sur plusieurs jeux de données, incluant KDD CUP99, NSL_KDD et UNSW_NB15, et il démontre une convergence, une précision et des performances robustes dans des scénarios de classification binaire et multi-classes.

De même, dans [158], un autre groupe de chercheurs présente une approche novatrice basée sur des techniques d'apprentissage profond, utilisant les mêmes jeux de données. Leur cadre proposé se distingue en comparant et en opposant les méthodes non supervisées aux approches discriminatives d'apprentissage profond. Dans cette étude, un réseau antagoniste génératif (GAN) est utilisé pour détecter les cybermenaces spécifiquement dans les réseaux de l'Internet Industriel des Objets (IIoT) pilotés par l'IoT. Les méthodes proposées, incluant les CNN, RNN et DNN, démontrent une amélioration remarquable des performances d'environ 95% à 97% en termes de précision, de fiabilité et d'efficacité pour divers types d'attaques.

Dans l'étude menée par [159], l'efficacité de sept classifieurs d'apprentissage automatique, à savoir AdaBoost, Random Forest, Naïve Bayes, Arbre de Décision, MLP, KNN et QDA, a été testée via de multiples expériences de systèmes de détection d'intrusion (IDS). L'évaluation a utilisé le jeu de données CICIDS2017, qui comprend à la fois du trafic bénin et des cyberattaques courantes. Les résultats expérimentaux démontrent que le classifieur des K plus proches voisins (KNN) surpasse les autres méthodes conventionnelles d'apprentissage automatique en termes de précision, rappel, exactitude et score F1. Il est à noter que tous les classifieurs évalués, à l'exception de KNN, ont démontré un temps d'entraînement acceptable.

Dans un travail connexe par [160], utilisant le même jeu de données que [24], un modèle combinant des algorithmes d'apprentissage automatique et des techniques d'analyse en composantes principales (ACP) est proposé pour entraîner et prédire les attaques par déni de service distribué (DDoS). Le jeu de données CSE-CIC-IDS 2018 est également employé. Les modèles proposés pour la détection d'attaques DDoS ont démontré des performances supérieures et un temps d'entraînement amélioré.

Dans l'étude menée par [161], les chercheurs ont visé à traiter la détection des attaques par rejeu et par déni de service distribué (DDoS) sur une plateforme de ville intelligente en conditions réelles, en proposant un modèle hybride d'apprentissage profond. L'évaluation des performances de ce modèle a été réalisée en utilisant des jeux de données en temps réel de villes intelligentes. Les résultats expérimentaux ont démontré des taux de précision élevés : le jeu de données environnementales a atteint un taux de précision de 98,37 %, le jeu de données de la rivière intelligente a atteint 98,13 %, et le jeu de données du sol intelligent a atteint un taux de précision impressionnant de 99,51 %.

L'article [162] présente une approche novatrice dans laquelle les auteurs proposent

un modèle d'apprentissage profond pour créer des représentations équilibrées de jeux de données déséquilibrés. Ces nouvelles représentations sont ensuite utilisées dans un modèle de détection d'attaques par apprentissage profond ensembliste, spécifiquement conçu pour un environnement de Systèmes de Contrôle Industriels (ICS). Le modèle de détection d'attaques intègre des classifieurs de Réseau Neuronal Profond (DNN) et d'Arbre de Décision (DT) pour identifier les cyberattaques en s'appuyant sur les nouvelles représentations.

Une autre étude [163] présente une approche innovante d'apprentissage profond pour la détection d'intrusion, visant à relever ces défis. Les auteurs décrivent leur nouvelle méthode appelée autoencodeur profond non symétrique (NDAE), qui est spécifiquement conçue pour l'apprentissage de caractéristiques non supervisé.

Dans [164], les chercheurs présentent un modèle de détection d'intrusion basé sur les anomalies, novateur, qui utilise un modèle de réseau neuronal convolutif pour créer des modèles de classification binaire et multi-classes. L'efficacité du modèle proposé est évaluée à l'aide de plusieurs jeux de données de détection d'intrusion, incluant BoT-IoT, IoT Network Intrusion, MQTT-IoT-IDS2020 et IoT-23.

Les articles [[165], [166], [167], [168], [169]] font référence à de nombreuses études de synthèse intéressantes qui traitent des techniques d'IA et d'apprentissage profond pour les systèmes de détection d'intrusion (IDS). Ces synthèses examinent les jeux de données d'intrusion disponibles publiquement et utilisés dans les IDS récents afin de révéler les défis actuels et les orientations futures. La revue publiée dans [170] se concentre sur les avancées récentes concernant les jeux de données d'IDS, spécifiquement de CSE-CIC-IDS-2017 à CSE-CIC-IDS-2018. Cette mise à jour inclut l'ajout de nouvelles catégories d'attaques. L'analyse souligne l'importance de ces avancées pour diverses communautés de recherche intéressées par l'utilisation des nouveaux jeux de données d'IDS pour développer des systèmes de détection d'intrusion efficaces et performants, basés sur l'IA et l'exploration de données.

L'étude de revue présentée dans [171] explore et analyse les méthodes de détection et de prévention d'intrusion spécifiquement conçues pour atténuer les attaques DDoS. Cette recherche examine en détail la classification des IDS, explore diverses approches de détection d'anomalies, analyse différents modèles de systèmes de détection d'intrusion basés sur des jeux de données, et étudie une gamme de techniques d'apprentissage automatique et de réseaux neuronaux profonds utilisées pour le prétraitement des données et la détection de logiciels malveillants.

La synthèse mentionnée dans [172] offre un aperçu de l'extraction de structures de graphes et de l'entraînement de modèles de réseaux neuronaux à graphes (GNN) pour des tâches de classification en aval dans la détection d'intrusion basée sur le réseau et sur l'hôte. Les auteurs fournissent une revue complète et une catégorisation des approches et des jeux de données de l'état de l'art utilisés dans ce domaine. Ils soulignent le potentiel des GNN pour générer des représentations vectorielles efficaces permettant une détection robuste de différents types d'intrusions.

Similairement aux études de [171] et [161], la recherche menée dans [173] se concentre sur la détection et la classification des attaques DDoS. Dans leur étude, les auteurs utilisent un modèle hybride nommé AE-MLP, qui combine un Autoencodeur (AE) avec un Perceptron Multicouche (MLP). Ce modèle hybride est employé pour détecter et classifier avec précision différents types d'attaques DDoS.

Il est vrai que la technique utilisée dans les systèmes de détection d'intrusion (IDS) est d'une importance majeure. Cependant, il est également crucial de prendre en compte la distribution des données. En d'autres termes, la manière dont les données

sont distribuées peut avoir un impact significatif sur les performances et l'efficacité des systèmes de détection d'intrusion.

Dans ce contexte, les travaux de [174] ont proposé le système DeepBalance, qui exploite des techniques d'apprentissage profond et de suréchantillonnage flou pour la détection de vulnérabilités logicielles. L'objectif principal de leur travail est de résoudre le problème du déséquilibre des classes lors de la construction de modèles de classification pour identifier les fonctions vulnérables dans de nouveaux projets, en utilisant du code source existant qui contient des vulnérabilités connues. Pour résoudre ce problème, ils utilisent le suréchantillonnage flou, une méthode qui génère des échantillons synthétiques pour la classe sous-représentée du code vulnérable, rééquilibrant ainsi les données d'entraînement. En combinant l'apprentissage profond et le suréchantillonnage flou, le système DeepBalance vise à améliorer la précision et l'efficacité de la détection des vulnérabilités logicielles dans des scénarios réels.

Dans la même veine et toujours dans ce contexte, les chercheurs de [175] utilisent l'algorithme DSSTE (Difficult Set Sampling Technique). Le but de DSSTE est d'améliorer l'apprentissage des données de réseau déséquilibrées dans un modèle de classification, en augmentant le nombre d'échantillons minoritaires qui nécessitent un apprentissage. DSSTE vise à résoudre le problème du trafic réseau déséquilibré et à améliorer la précision de la classification pour la classe minoritaire.

Dans [176], les chercheurs analysent minutieusement et proposent des solutions pour les problèmes découlant du déséquilibre des jeux de données pendant les phases d'entraînement et d'inférence.

3.3 Sécurité de l'IIoT : menaces, vulnérabilités et défis

L'*Industrial Internet of Things* (IIoT) constitue une extension de l'*Internet of Things* (IoT) qui exploite des capteurs et des actionneurs pour connecter et automatiser les processus industriels. Il s'agit d'un élément clé de l'Industrie 4.0, née en Allemagne. L'IIoT joue un rôle crucial dans l'automatisation des processus internes et externes dans des secteurs tels que le transport, la fabrication et le marketing, en s'appuyant sur un large éventail de dispositifs interconnectés [177]. En plus des vulnérabilités héritées de l'IoT, l'IIoT intègre des composants dotés de capacités de détection, de traitement et de visualisation, ce qui augmente sa sensibilité à un large éventail d'attaques. La figure 3.1 illustre les différentes couches de l'architecture IIoT et met en évidence les attaques courantes se produisant à chaque couche. Cette représentation permet d'identifier les risques de sécurité potentiels et les vulnérabilités à travers l'ensemble du système IIoT.

Afin de réduire ces risques, les chercheurs et les développeurs travaillent sur un large éventail d'outils et de méthodologies initialement conçus pour l'IoT grand public. Ils tirent les enseignements des expériences et des erreurs observées dans le domaine de l'IoT grand public et s'appuient sur les nombreux résultats bien documentés en matière de sécurité de l'IoT. La figure 3.2 illustre quelques-uns des défis de sécurité généraux rencontrés dans les environnements IIoT.

Dans ce travail, nous nous concentrons sur le défi, en constante évolution, de la détection d'anomalies dans l'IIoT, qui constitue un aspect essentiel pour garantir la sécurité et l'intégrité opérationnelle de ces environnements. La détection d'anomalies consiste à identifier des motifs ou comportements anormaux qui s'écartent

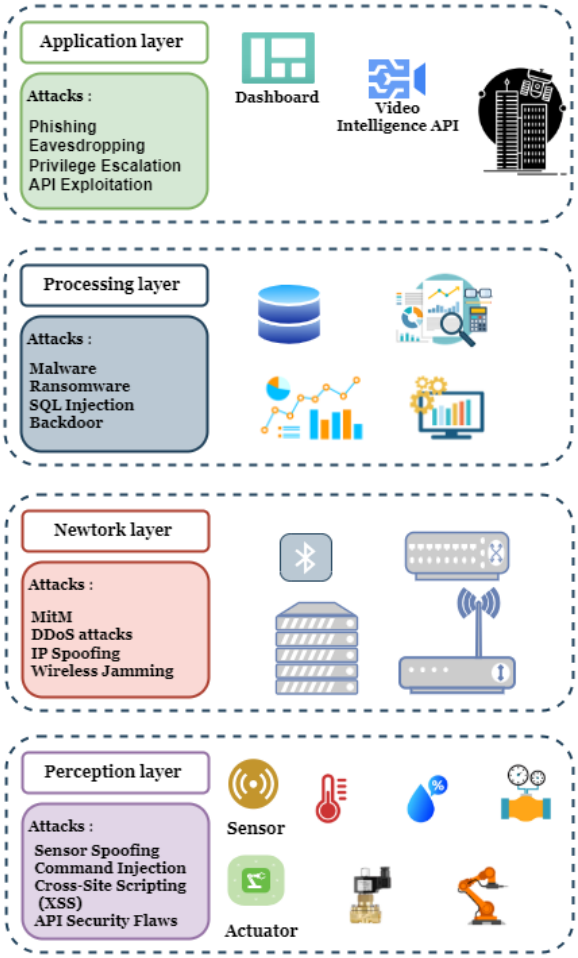

Figure 3.1 *– Exemples d'attaques de sécurité IIoT à travers différentes couches*

de manière significative des normes attendues dans les données générées par les dispositifs interconnectés [178]. En exploitant des techniques avancées, telles que les modèles d'apprentissage profond, la détection d'anomalies joue un rôle fondamental dans l'atténuation des risques et la protection des systèmes IIoT contre les menaces potentielles. En détectant et en traitant les anomalies en temps réel, la détection d'anomalies renforce la posture globale de sécurité des systèmes IIoT, permet une intervention rapide et minimise l'impact des événements anormaux [179].

3.4 Cadre conceptuel de l'étude

3.4.1 Réseaux de neurones profonds

L'apprentissage profond (*deep learning*) est un sous-domaine puissant de l'apprentissage automatique (*machine learning*) qui repose sur l'utilisation de réseaux de neurones artificiels pour apprendre à partir des données. Inspirés du fonctionnement du cerveau humain, les réseaux de neurones peuvent apprendre des motifs complexes dans les données, motifs que les approches classiques de machine learning auraient du mal, voire seraient incapables, de capturer [180]. Un réseau de neurones comporte généralement une couche d'entrée, une couche de sortie et une ou plusieurs couches

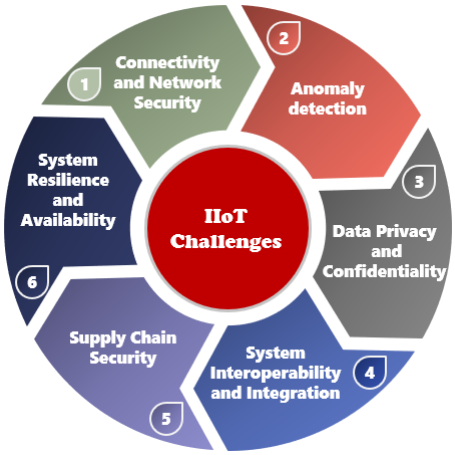

Figure 3.2 *– Défis liés à l'IIoT*

cachées. Lorsqu'il ne possède qu'une seule couche cachée, on parle de réseau de neurones peu profond (*shallow neural network*). Lorsqu'il comprend plusieurs couches cachées, il est qualifié de réseau de neurones profond (*deep neural network*, DNN), et le processus d'entraînement correspondant est appelé apprentissage profond [181].

Le perceptron constitue l'unité de base et le bloc fondamental des réseaux de neurones, y compris des DNN [182]. Il joue le rôle de neurone artificiel, capable de traiter plusieurs entrées en leur appliquant des poids spécifiques. Ces entrées pondérées sont ensuite sommées, puis un terme de biais est ajouté, et le résultat est passé à travers une fonction d'activation pour produire une sortie [183]. Mathématiquement, cette opération peut être exprimée comme :

$$z = w_1 x_1 + w_2 x_2 + ... + w_n x_n + b. \tag{3.1}$$

$$\begin{cases} f_1 &= w_{11} x_1 + w_{12} x_2 + b_1 \\ f_2 &= w_{21} x_1 + w_{22} x_2 + b_2 \\ f_3 &= w_{31} y_1 + w_{32} y_2 + b_3 \end{cases} \tag{3.2}$$

où $x_1, x_2, \ldots, x_n$ sont les variables d'entrée, $w_1, w_2, \ldots, w_n$ les poids correspondants, et b le terme de biais.

Initialement, les premiers DNN étaient construits sous la forme de perceptrons multicouches (*multilayer perceptrons*), dans lesquels plusieurs perceptrons étaient connectés en couches successives. Chaque perceptron recevait des entrées, réalisait des calculs en fonction de ses paramètres (poids et biais), puis produisait une sortie [184]. Dans le cas de trois perceptrons connectés, comme illustré à la figure 3.3, les deux premiers perceptrons reçoivent les entrées w_n et x_2, effectuent leurs calculs respectifs et génèrent les sorties y_1 et y_2. Ces dernières sont ensuite transmises au troisième perceptron, qui effectue à son tour des calculs pour produire la sortie finale y_3. Cette structure interconnectée de perceptrons permet au réseau d'apprendre et de réaliser des prédictions.

Cependant, avec l'évolution des DNN, leur entraînement repose désormais sur des méthodes plus sophistiquées, telles que la rétropropagation du gradient (*backpropagation*, voir figure 3.4). La rétropropagation permet un apprentissage plus efficace et

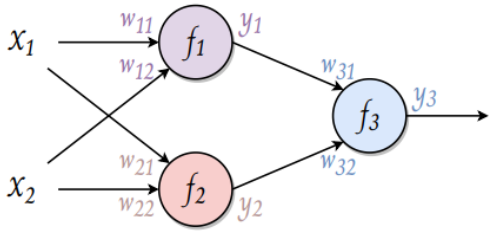

Figure 3.3 *– Exemple de perceptron multicouche*

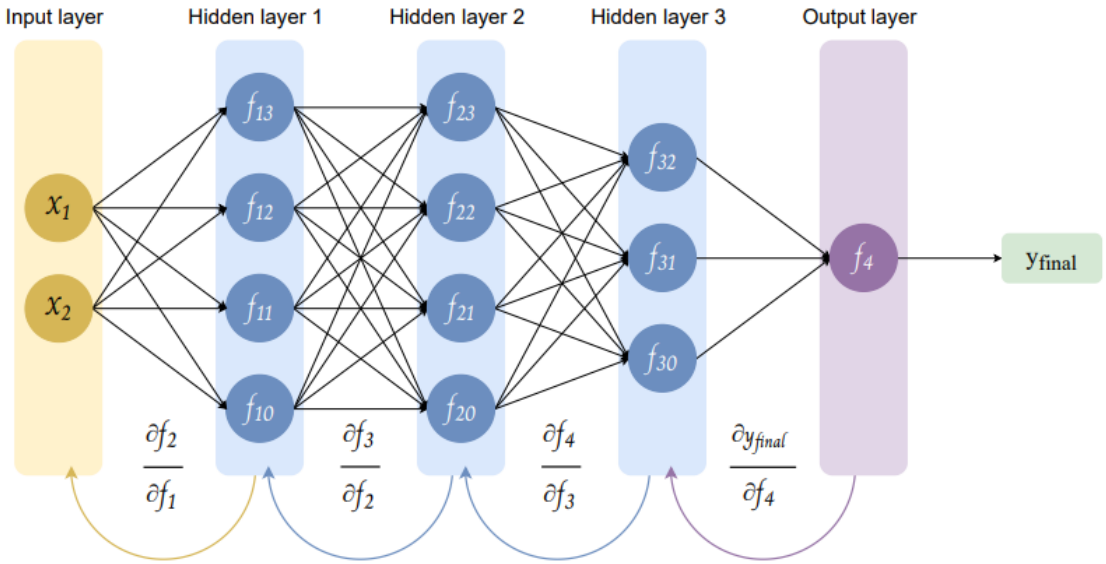

Figure 3.4 *– Exemple de modèle utilisant la rétropropagation*

plus performant en ajustant les poids et les biais des perceptrons du réseau à partir des erreurs calculées au cours de l'entraînement [112]. Cette technique a considérablement amélioré l'apprentissage et les performances des DNN modernes par rapport à la règle d'apprentissage initiale du perceptron.

3.4.2 Réseaux de neurones convolutionnels

Au fil de l'évolution des réseaux de neurones, des progrès importants ont été réalisés avec l'introduction des variantes de perceptrons multicouches et des réseaux de neurones convolutionnels (*Convolutional Neural Networks*, CNN). En 1989, le concept de modèles à perceptrons multicouches a marqué une étape clé dans le développement des réseaux de neurones. Toutefois, c'est Yann LeCun qui a véritablement révolutionné le domaine en proposant les premiers CNN [185].

Les CNN de LeCun s'inspirent de l'organisation et du fonctionnement du cortex visuel chez les animaux [186]. Ces réseaux sont conçus pour apprendre et traiter, de manière automatique et adaptative, des hiérarchies spatiales de caractéristiques. En intégrant des filtres mathématiques dans les premières couches du réseau, les CNN se sont révélés particulièrement efficaces pour les tâches de reconnaissance et de traitement d'images [187].

Les CNN constituent un cadre mathématique reposant généralement sur trois types de couches fondamentales : les couches convolutionnelles, les couches de pooling

et les couches entièrement connectées [188].

- **Couches convolutionnelles** : il s'agit des briques de base des CNN, qui réalisent l'opération de convolution sur les données d'entrée. Cette opération consiste à appliquer un ensemble de filtres apprenables sur l'entrée, afin d'extraire les caractéristiques pertinentes. Chaque filtre produit une carte de caractéristiques (*feature map*) en effectuant un produit scalaire entre ses poids et une petite région de l'entrée. De nombreux filtres sont utilisés afin de capturer des motifs variés dans les données.
- **Couches de pooling** : les couches de pooling sont utilisées pour réduire les dimensions spatiales des cartes de caractéristiques générées par les couches convolutionnelles. Elles réalisent un sous-échantillonnage (*downsampling*) des cartes, généralement au moyen d'opérations telles que le max pooling ou l'average pooling. Le pooling permet de diminuer la complexité computationnelle du réseau et introduit une certaine invariance aux translations, puisque la présence d'une caractéristique légèrement décalée dans l'espace conduit au même résultat après pooling.
- **Couches entièrement connectées** : ces couches produisent les prédictions finales à partir des caractéristiques extraites. Chaque neurone est connecté à tous les neurones de la couche précédente, formant un réseau entièrement connecté. Elles permettent de modéliser des relations complexes entre les caractéristiques. La sortie des couches entièrement connectées est généralement passée à une fonction d'activation pour générer la sortie finale du réseau.

En combinant ces couches de manière séquentielle et en ajustant leurs paramètres à l'aide de la rétropropagation et de la descente de gradient, les CNN peuvent apprendre des motifs complexes et réaliser des prédictions sur diverses tâches.

3.4.3 Mesures d'évaluation

L'évaluation des algorithmes utilisés pour la sécurisation de l'IIoT repose souvent sur différentes mesures de performance. Parmi celles-ci, on retrouve notamment l'exactitude (*accuracy*), la précision (*precision*), le rappel (*recall*), le score F1 (*F1-score*), le taux de vrais positifs (*true positive rate*), le taux de fausses alarmes (*false alarm rate*), le taux de faux positifs (*false positive rate*), la courbe ROC (*receiver operating characteristic*) et l'aire sous la courbe (AUC). Ces mesures sont couramment utilisées pour évaluer l'efficacité des modèles. Elles peuvent être calculées à partir d'une matrice de confusion, qui fournit une représentation structurée des résultats de classification. Dans cette matrice, les vrais positifs (TP) et les vrais négatifs (TN) correspondent respectivement au nombre d'échantillons d'attaque et d'échantillons normaux correctement classés par le modèle. À l'inverse, les faux positifs (FP) et les faux négatifs (FN) représentent les cas où des enregistrements normaux et d'attaque sont incorrectement classés.

- **Exactitude (Accuracy)** : elle représente la proportion globale de prédictions correctes et se calcule comme le rapport entre le nombre d'échantillons correctement classés (TP + TN) et le nombre total d'échantillons.
- **Précision (Precision)** : la précision mesure la proportion d'échantillons positifs correctement prédits (TP) parmi l'ensemble des échantillons prédits positifs (TP + FP). Elle renseigne sur la fiabilité des prédictions positives.

- **Rappel (Recall)** : également appelé sensibilité ou taux de vrais positifs (TPR), le rappel mesure la proportion d'échantillons positifs correctement prédits (TP) parmi l'ensemble des échantillons réellement positifs (TP + FN). Il indique la capacité du modèle à détecter les instances positives.
- **Score F1 (F1-score)** : le score F1 combine la précision et le rappel en une seule mesure équilibrée. Il correspond à la moyenne harmonique de la précision et du rappel et fournit une évaluation plus complète des performances du modèle, en particulier dans des contextes de classes déséquilibrées.

3.5 Modèle proposé

3.5.1 Jeu de données

Le choix du jeu de données est une étape critique pour les algorithmes de détection d'anomalies. Un jeu de données bien choisi peut conduire à de meilleures performances et à une meilleure capacité de généralisation du modèle de détection d'anomalies. Un bon jeu de données doit être représentatif des données sur lesquelles l'algorithme sera appliqué pour détecter des anomalies. Cela implique qu'il contienne une variété de points de données normaux et anormaux (attaques) et qu'il soit suffisamment volumineux pour couvrir les différents types d'anomalies que l'algorithme est censé détecter.

Cette étude utilise le jeu de données *Edge-IIoTset*, qui est un jeu de données de cybersécurité innovant et complet pour les applications IoT et IIoT. Il a été présenté dans un article de recherche intitulé *Edge-IIoTset : A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning* [189], rédigé par Mohamed Amine Ferrag, Othmane Friha, Djallel Hamouda, Leandros Maglaras et Helge Janicke. Le jeu de données a été construit à partir d'un banc d'essai IoT/IIoT spécialement conçu, intégrant un large éventail de dispositifs, de capteurs, de protocoles et de configurations Cloud/Edge. Il inclut des données provenant de plus de dix types de dispositifs IoT, parmi lesquels divers capteurs numériques de température et d'humidité, des capteurs ultrasoniques, des capteurs de niveau d'eau, des capteurs de pH, des capteurs d'humidité du sol, des capteurs de fréquence cardiaque et des capteurs de flamme.

Le jeu de données contient des informations sur 14 types d'attaques liées aux protocoles de connectivité IoT et IIoT, classées en cinq grandes catégories de menaces : attaques de type DoS/DDoS, collecte d'information (*information gathering*), attaques de type homme du milieu (*man-in-the-middle*), attaques par injection et attaques par logiciels malveillants (*malware*). Il inclut également des caractéristiques issues de plusieurs sources, telles que les alarmes, les ressources système, les journaux (*logs*) et le trafic réseau.

Le jeu de données *Edge-IIoTset* comporte 61 caractéristiques et deux variables cibles : `Attack_label` pour la classification binaire (tableau 3.1) et `Attack_type` pour la classification multiclasse. La variable `Attack_label` est spécifiquement destinée aux tâches de classification binaire, où l'objectif est de distinguer deux classes : "Attack" et "Normal". Elle attribue l'étiquette 1 aux instances représentant des attaques, et l'étiquette 0 aux instances correspondant à un trafic normal. La variable cible `Attack_type` est, quant à elle, utilisée pour la classification multiclasse, permettant de catégoriser les instances en différents types d'attaques ainsi qu'en trafic

normal.

Le tableau 3.2 présente un résumé du nombre d'instances correspondant aux différents types de trafic IoT observés dans le jeu de données *Edge-IIoTset*. Le jeu de données comprend à la fois du trafic normal et plusieurs types d'attaques liées aux protocoles de connectivité IoT et IIoT. Les types de trafic sont regroupés en deux classes : "Normal" et "Attack". La classe "Normal" représente un trafic légitime et bénin, tandis que la classe "Attack" regroupe les différentes catégories d'attaques. Le tableau 3.1 indique le nombre d'instances pour chaque type de trafic, offrant ainsi un aperçu global de la distribution des données.

Tableau 3.1 *– Nombre d'instances pour* `Attack_label`

Attack_label	**Nombre d'instances**
0	1,615,643
1	603,558

Tableau 3.2 *– Nombre d'instances par type de trafic dans Edge-IIoTset (*`Attack_type`*)*

Classe	Type de trafic	Instances
Normal	Normal	1,615,643
Attack	DDoS_UDP	121,568
Attack	DDoS_ICMP	116,436
Attack	SQL_injection	51,203
Attack	Password	50,153
Attack	Vulnerability_scanner	50,110
Attack	DDoS_TCP	50,062
Attack	DDoS_HTTP	49,911
Attack	Uploading	37,634
Attack	Backdoor	24,862
Attack	Port_Scanning	22,564
Attack	XSS	15,915
Attack	Ransomware	10,925
Attack	MITM	1,214
Attack	Fingerprinting	1,001
Total		2,496,644

3.5.2 Approches expérimentales

Le modèle proposé dans ce travail est un algorithme hybride composé d'un **réseau de neurones convolutionnel (CNN)** suivi d'un **réseau de neurones profond**

(DNN). Toutefois, avant de mettre en œuvre le modèle, une étape de **prétraitement** du jeu de données est réalisée afin d'améliorer ses performances.

Le **prétraitement** du jeu de données est une étape cruciale dans toute tâche de machine learning ou de deep learning. Il consiste à transformer et à préparer les données brutes de manière à les rendre adaptées à l'entraînement du modèle. Les étapes de prétraitement utilisées dans cette étude sont les suivantes :

- **Chargement du jeu de données** : le jeu de données est chargé à partir d'un fichier CSV.
- **Suppression des colonnes inutiles** : certaines colonnes ne sont pas nécessaires au modèle et sont supprimées à l'aide de la méthode `drop` du `DataFrame`.
- **Suppression des lignes contenant des valeurs manquantes** : les lignes comportant des valeurs manquantes sont supprimées à l'aide de la méthode `dropna`.
- **Mélange du DataFrame** : les lignes du `DataFrame` sont mélangées aléatoirement à l'aide de la fonction `shuffle` du module `sklearn.utils`, afin d'éviter tout biais lié à l'ordre initial des données.
- **Encodage des variables catégorielles** : certaines colonnes du jeu de données sont de nature catégorielle. Pour les convertir dans un format numérique compatible avec le modèle, un encodage *one-hot* est appliqué via la méthode `pd.get_dummies`.
- **Normalisation des caractéristiques** : les variables numériques sont standardisées pour avoir une moyenne nulle et un écart-type égal à 1 à l'aide du `StandardScaler` du module `sklearn.preprocessing`.
- **Encodage de la variable cible** : la variable cible `"Attack_type"`, représentant la classe d'attaque, est encodée par *label encoding*, ce qui associe un entier à chaque classe.
- **Séparation des données en ensembles d'entraînement et de test** : les données prétraitées sont séparées en ensembles d'entraînement et de test à l'aide de la fonction `train_test_split` du module `sklearn.model_selection`. L'ensemble d'entraînement sert à ajuster le modèle, tandis que l'ensemble de test est utilisé pour évaluer ses performances.

3.5.2.1 Classification binaire

Nous construisons un modèle **DNN** pour la classification binaire en utilisant le jeu de données décrit. Le modèle est constitué de couches denses (`Dense`) avec des fonctions d'activation **ReLU** et d'une couche de sortie avec une fonction d'activation **Sigmoid**. Nous compilons et entraînons le modèle à l'aide de l'optimiseur `Adam`, de la fonction de perte `binary_crossentropy` et de l'**exactitude** comme métrique d'évaluation. L'algorithme est présenté dans **l'Algorithme 1**.

3.5.2.2 Classification multiclasse

Dans un jeu de données bien structuré et prétraité efficacement, il est possible d'ajuster le nombre de classes de la variable cible en fonction des objectifs et du contexte d'utilisation du modèle. Dans *Edge-IIoTset*, la variable `"Attack_type"` est généralement utilisée pour effectuer une classification multiclasse. La manipulation du nombre de classes de la variable cible peut être bénéfique de plusieurs manières :

Algorithme 1 Modèle DNN pour la classification binaire

Entrées *x_train_scaled*, *y_train*, *x_test_scaled*, *y_test* Effectuer un encodage par labels sur *y_train* et *y_test* afin de convertir les étiquettes de classes en format numérique. **build_dnn_model** Créer un modèle `Sequential` Ajouter une couche `Dense` avec 256 neurones et une fonction d'activation ReLU, avec une dimension d'entrée égale au nombre de caractéristiques de *x_train_scaled*. Ajouter une deuxième couche `Dense` avec 164 neurones et une fonction d'activation ReLU. Ajouter une troisième couche `Dense` avec 82 neurones et une fonction d'activation ReLU. Ajouter une quatrième couche `Dense` avec 32 neurones et une fonction d'activation ReLU. Ajouter la couche de sortie `Dense` avec 1 neurone et une fonction d'activation Sigmoid (classification binaire). Appeler BUILD_DNN_MODEL() pour construire le modèle DNN de classification binaire. Compiler le modèle avec l'optimiseur Adam et la fonction de perte `binary_crossentropy`, en utilisant l'exactitude comme métrique d'évaluation. Entraîner le modèle pendant 25 époques avec une taille de lot (*batch size*) de 32. Valider le modèle en utilisant *x_test_scaled* et *y_test_encoded*.

- **Suppression des classes non essentielles** : certaines classes peuvent être supprimées du jeu de données si elles représentent des attaques rares ou peu pertinentes vis-à-vis de l'objectif fixé. Cela permet de simplifier le modèle et d'améliorer sa capacité à se concentrer sur les attaques les plus importantes. Dans ce travail, nous avons retenu uniquement les neuf classes d'attaque les plus fréquentes. Il s'agit des classes pour lesquelles le modèle a montré une forte exactitude et de bonnes performances de détection. Cette approche permet d'optimiser le modèle pour les types d'attaques les plus courants et les plus critiques, en améliorant ses performances globales et sa capacité de généralisation. La distribution des classes sélectionnées est illustrée à la figure 3.5, ce qui met en évidence leur importance dans le cadre du jeu de données et de nos objectifs de recherche.
- **Fusion de classes similaires** : dans certains cas, plusieurs classes peuvent être fusionnées en une seule si elles se rapportent au même type d'attaque ou à une catégorie similaire. Cela permet de réduire le nombre de classes tout en conservant la représentativité des données. Dans notre étude, nous avons mené une analyse approfondie du jeu de données et identifié 15 classes présentant des similarités en termes de caractéristiques et de comportements d'attaque. Pour accroître l'efficacité du modèle, nous avons fusionné ces 15 classes en 6 classes plus larges (voir figure 3.6), en regroupant les attaques qui partagent des motifs comparables. Cette consolidation simplifie le modèle tout en préservant les caractéristiques essentielles de chaque type d'attaque au sein d'un ensemble réduit de classes.
- **Agrégation de classes** : si certaines classes comportent peu de données ou présentent une exactitude de classification plus faible, elles peuvent être agrégées avec d'autres classes similaires afin d'augmenter leur représentativité et de renforcer la capacité prédictive du modèle. Dans notre étude, nous avons observé des classes présentant une exactitude plus faible ou un nombre d'instances limité, rendant difficile leur distinction individuelle. Toutefois, une caractéristique commune se dégageait : il s'agissait toutes d'attaques. En nous appuyant sur l'expérience acquise dans le cas des 9 classes, où certaines classes non essen-

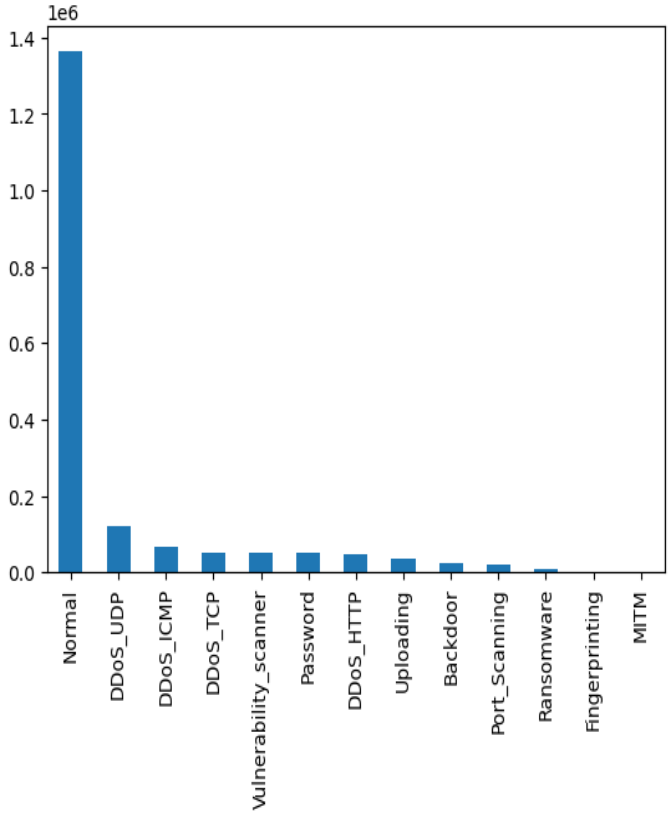

Figure 3.5 – *Distribution en barres des 9 classes*

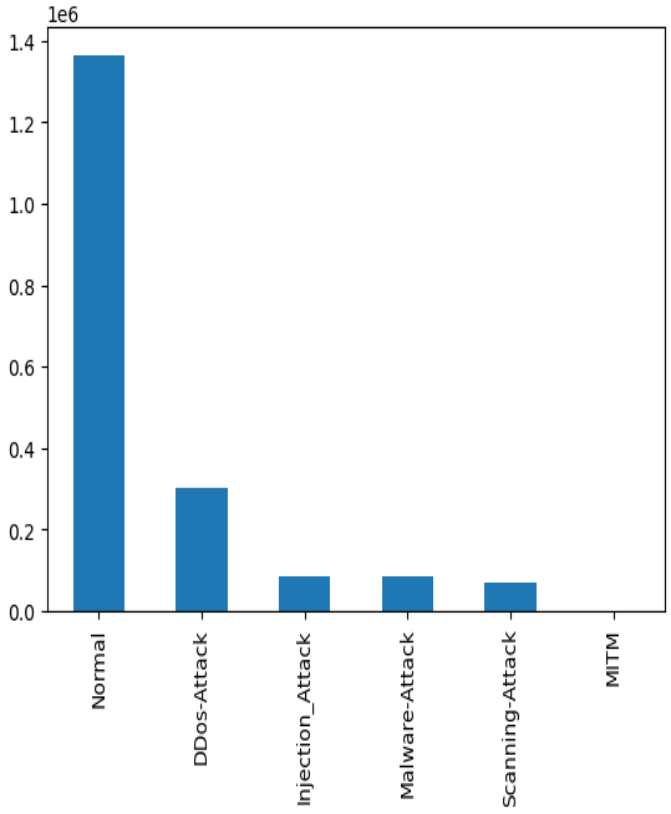

Figure 3.6 – *Distribution en barres des 6 classes*

tielles avaient été supprimées, nous avons adopté une stratégie différente : au lieu de les éliminer, nous les avons agrégées dans une nouvelle classe intitulée "DA" (*Different Attacks*), comme illustré à la figure 3.7. Cette nouvelle classe "DA" constitue une catégorie globale englobant toutes les attaques associées à une faible exactitude ou à un faible nombre d'instances. Cette agrégation permet de conserver l'information relative à ces attaques tout en atténuant les difficultés liées à leur classification individuelle.

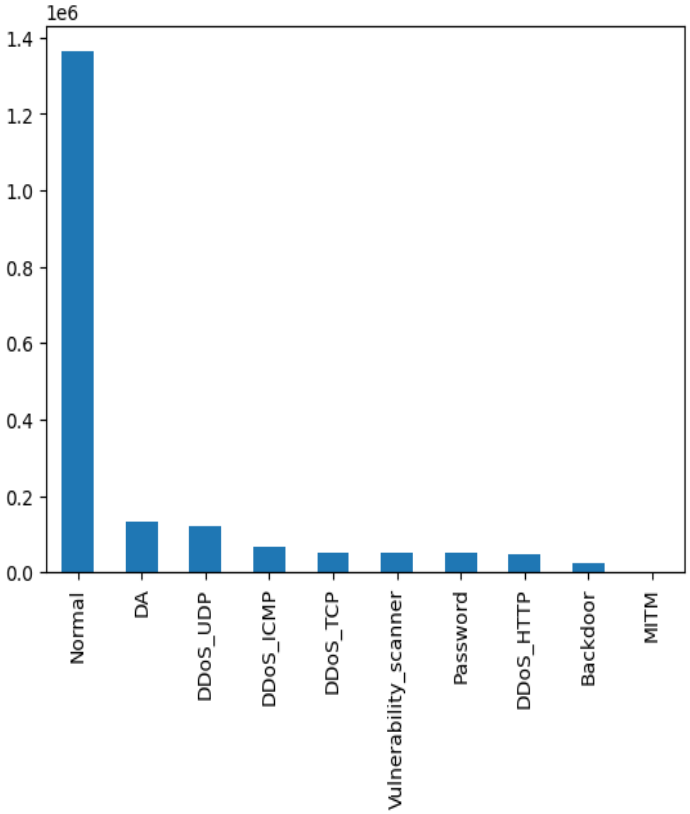

Figure 3.7 *– Distribution en barres pour 10 classes (incluant la classe DA)*

- **Duplication de classes** : dans certains cas, il est possible de dupliquer des classes pour représenter des sous-catégories d'attaques plus spécifiques. Cela permet au modèle de mieux distinguer des scénarios d'attaque différents et d'améliorer sa précision globale.

Une fois les données prétraitées et le nombre de classes déterminé, nous passons à la phase de conception du modèle et de définition de l'architecture (voir figure 3.8).

Pour concevoir notre modèle, nous nous appuyons sur les bonnes pratiques du deep learning. Notre architecture repose sur un réseau de neurones convolutionnel (CNN), en raison de sa capacité à extraire des caractéristiques pertinentes à partir des données d'entrée.

Une fois les caractéristiques extraites par le CNN, elles sont transmises à un réseau de neurones profond (DNN). Le DNN est composé de plusieurs couches entièrement connectées qui combinent les caractéristiques pour effectuer la classification finale des attaques. Ces couches apprennent des représentations plus abstraites des données, facilitant ainsi la prise de décision pour la classification. L'architecture du modèle MCCNNDNN (*multiclassification CNN DNN network*) est décrite dans **l'Algorithme 2**.

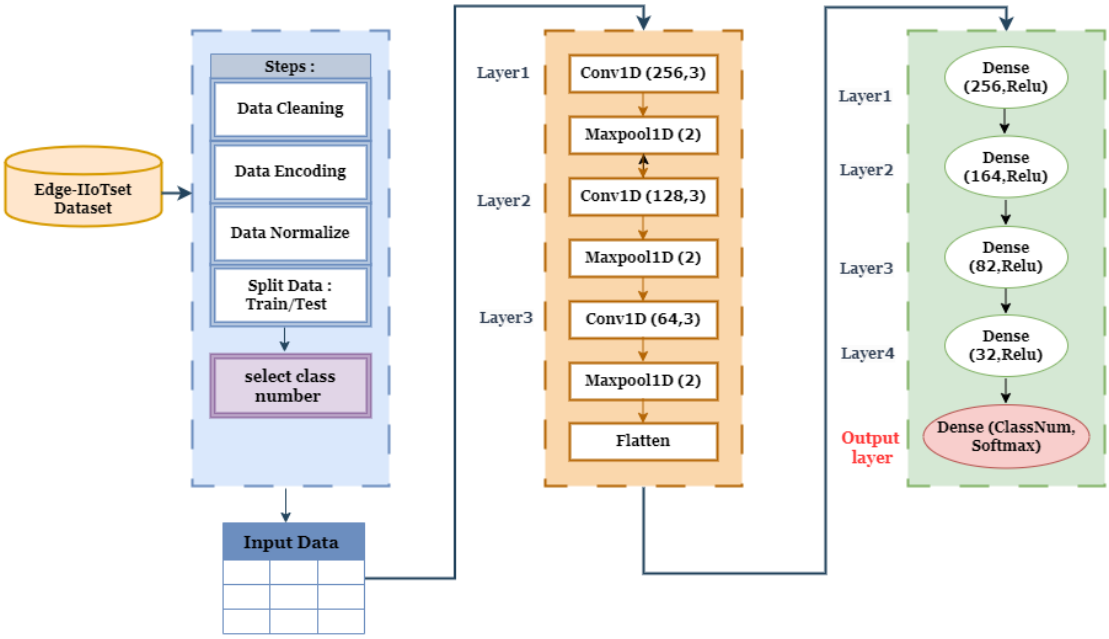

Figure 3.8 – *Méthodologie proposée dans cette étude*

3.6 Résultats et discussion

Cet article propose une méthode innovante et efficace pour les systèmes modernes de détection d'intrusion, qui jouent un rôle essentiel dans l'identification d'activités non autorisées au sein des réseaux informatiques. Malgré l'utilisation d'algorithmes à l'état de l'art pour catégoriser un large éventail de scénarios d'intrusion, leurs performances globales restent parfois sous-optimales. Cette limite pose des difficultés importantes pour la mise en œuvre pratique de ces architectures complexes de systèmes de détection d'intrusion (IDS) dans des environnements réseaux réels.

Le tableau 3.3 met en évidence les métriques de classification binaire pour deux classes distinctes, "Normal" et "Attack". Les métriques comprennent l'exactitude, la précision, le rappel et le score F1, chacune apportant un éclairage précieux sur les performances du modèle. Le tableau 3.3 illustre la capacité remarquable du modèle à distinguer les classes "Normal" et "Attack". Les scores parfaits obtenus pour l'exactitude, la précision, le rappel et le score F1 soulignent la capacité du modèle à classifier correctement les instances et à minimiser les erreurs de classification, ce qui le rend particulièrement adapté aux tâches de classification binaire.

Tableau 3.3 – *Métriques de classification binaire.*

Classe	Exactitude	Précision	Rappel	Score F1
Normal	1.00	1.00	1.00	1.00
Attack	1.00	1.00	1.00	1.00

Les figures 3.9, 3.10, 3.11 et 3.12 présentent les résultats d'exactitude obtenus sur les ensembles de test, de validation et d'entraînement en utilisant le jeu de données *Edge-IIoTset*, pour différents scénarios de distribution des classes. Il est particulièrement notable que la méthode proposée, le modèle Multi-Level Convolutional

Algorithme 2 Modèle MCCDDCNN

x_train, *y_train*, *x_tes*, *y_test* Initialiser le modèle avec `Sequential()` Ajouter une couche `Conv1D` avec `filters=256`, `kernel_size=3`, `activation='relu'`, et `input_shape=`($x_{\texttt{train}}$`.shape`$[1], 1$) Ajouter une couche `MaxPooling1D` avec `pool_size=2` Ajouter une deuxième couche `Conv1D` avec `filters=128`, `kernel_size=3`, `activation='relu'` Ajouter une deuxième couche `MaxPooling1D` avec `pool_size=2` Ajouter une troisième couche `Conv1D` avec `filters=64`, `kernel_size=3`, `activation='relu'` Ajouter une troisième couche `MaxPooling1D` avec `pool_size=2` Ajouter une couche `Flatten` Ajouter une couche `Dense` avec `units=256`, `activation='relu'`, et `kernel_regularizer=l2(0.00001)` Ajouter une couche `Dense` avec `units=164`, `activation='relu'`, `kernel_regularizer=l2(0.00001)` Ajouter une couche `Dense` avec `units=82`, `activation='relu'`, `kernel_regularizer=l2(0.00001)` Ajouter une couche `Dense` avec `units=32`, `activation='relu'`, `kernel_regularizer=l2(0.00001)` Ajouter la couche de sortie `Dense` avec `units=classnum` et `activation='softmax'` Définir l'optimiseur Adam avec un taux d'apprentissage de 0.001 Définir la fonction de perte comme `categorical_crossentropy` Compiler le modèle avec l'optimiseur et la fonction de perte Entraîner le modèle Ajuster le modèle sur x_{train} et y_{train} pendant 20 époques avec `batch_size=512` et `validation_split=0.2` Sauvegarder l'historique d'entraînement dans *history* Évaluer le modèle sur x_{test} et y_{test} et stocker les résultats dans *score*

Neural Network Deep Neural Network (MCCNNDNN), démontre systématiquement une exactitude élevée pour toutes les approches examinées. Plus précisément, parmi les différentes distributions de classes, l'approche à 9 classes se distingue avec un taux d'exactitude de 99,50 %. De même, l'approche à 6 classes présente un niveau d'exactitude robuste de 97,14 %. L'exactitude pour l'approche à 10 classes reste notable, atteignant 96,12 %, tandis que l'approche à 15 classes affiche une exactitude de 95,41 %. Les figures 3.13, 3.14, 3.15 et 3.16 montrent l'évolution de la fonction de perte. Ces résultats soulignent l'efficacité du modèle MCCNNDNN pour atteindre une haute exactitude dans des scénarios variés de distribution des classes, ce qui confirme son potentiel pour une détection robuste des intrusions dans le cadre complexe du jeu de données *Edge-IIoTset*.

Le tableau 3.4 présente les métriques d'évaluation, en termes de précision, de rappel et de score F1, pour le modèle MCCNNDNN appliqué à la distribution à 15 classes, incluant les classes "Normal", "Backdoor", "DDoS_HTTP", "DDoS_ICMP", "DDoS_TCP", "DDoS_UDP", "Fingerprinting", "MITM", "Password", "Port_Scanning", "Ransomware", "SQL_injection", "Uploading", "Vulnerability_scanner" et "XSS".

Le tableau 3.5 présente les performances du modèle MCCNNDNN pour la distribution à 9 classes. Les classes "Normal", "DDoS_UDP", "DDoS_ICMP" et "MITM" affichent une précision, un rappel et un score F1 parfaits, indiquant que le modèle se comporte de manière exceptionnelle pour ces classes. La classe "DDoS_TCP" présente également une précision très élevée (99 %), ce qui suggère la présence de quelques faux positifs, mais le rappel et le score F1 restent très élevés. Les classes "Vulnerability_scanner", "Password" et "Backdoor" montrent aussi de bonnes performances, bien que la classe "Password" présente un rappel plus faible, ce qui impacte son score F1. La classe "DDoS_HTTP" affiche une précision plus faible (87 %) mais un rappel

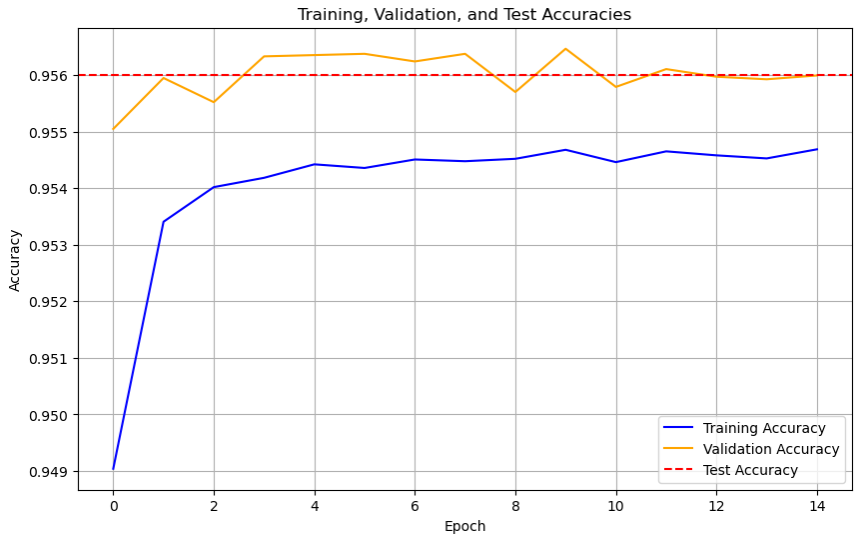

Figure 3.9 *– Exactitude pour la distribution à 15 classes*

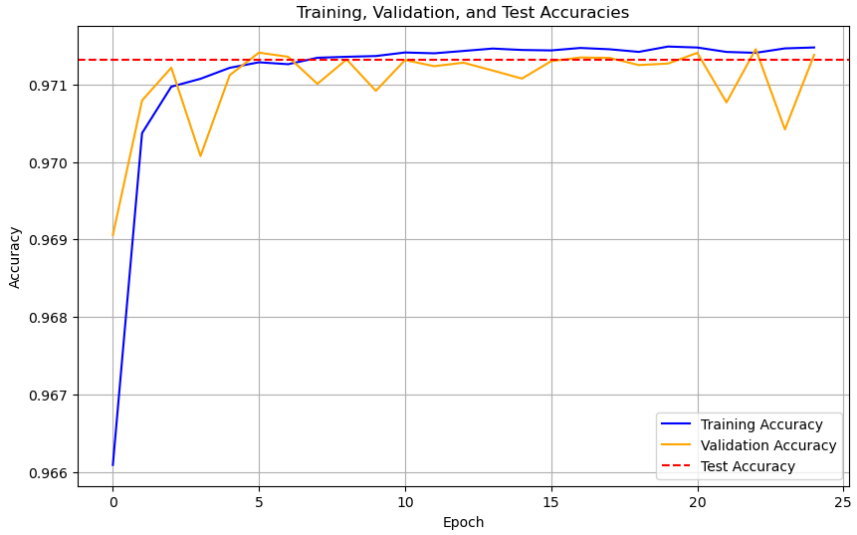

Figure 3.10 *– Exactitude pour la distribution à 6 classes*

élevé (98 %), conduisant à un score F1 satisfaisant (92 %).

Dans le tableau 3.6, le modèle démontre d'excellentes performances pour la distribution à 10 classes d'attaques. Pour les classes "DDoS_UDP" et "DDoS_ICMP", il atteint une précision et un rappel parfaits. Pour la classe "DDoS_TCP", la précision est de 82 % et le rappel de 100 %, ce qui donne un score F1 de 92 %. Pour la classe "DA", les performances sont raisonnables avec une précision de 71 % et un rappel de 85 %, conduisant à un score F1 de 77 %. La classe "Vulnerability_scanner" présente une précision de 91 % et un rappel de 81 %. Pour la classe "Password", la précision est parfaite (100 %) tandis que le rappel atteint 84 %. Enfin, pour la classe "Backdoor", le modèle obtient une précision de 99 % et un rappel de 95 %, soit un score F1 de 97 %. La classe "MITM" atteint une précision, un rappel et un score F1 parfaits (100 %).

Le tableau 3.7 présente les performances du modèle pour la distribution à 6 classes. Pour la classe "DDOS-ATTACK", le modèle atteint une précision de 68 % et un rappel de 99 %, ce qui conduit à un score F1 de 81 %. Pour les classes "INJECTION-ATTACK", "SCANNING-ATTACK" et "MITM", il obtient des performances élevées,

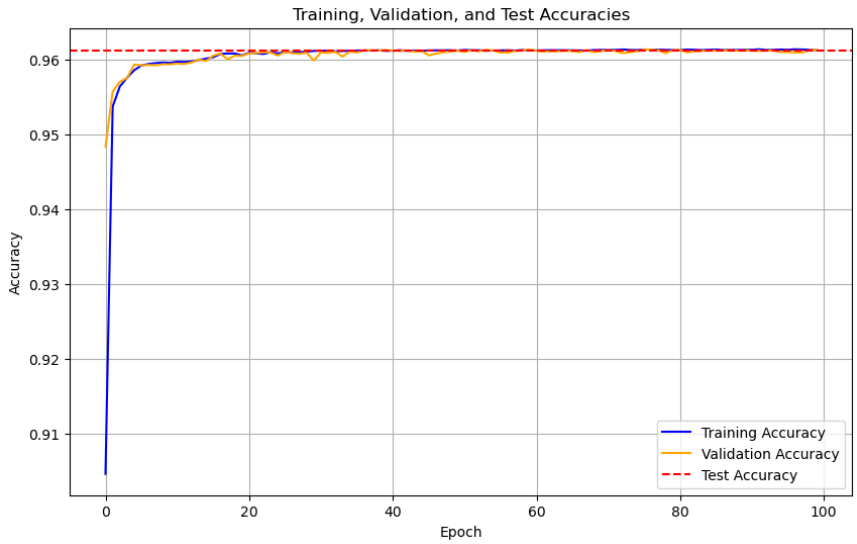

Figure 3.11 *– Exactitude pour la distribution à 10 classes*

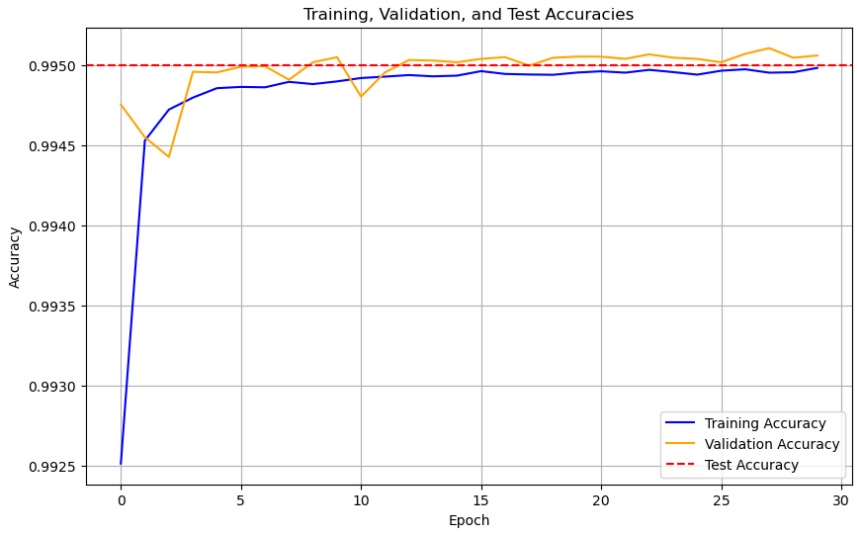

Figure 3.12 *– Exactitude pour la distribution à 9 classes*

avec des valeurs de précision, de rappel et de score F1 très bonnes, bien que "MITM" soit légèrement moins bien rappelée (0,73). Pour la classe "MALWARE-ATTACK", les performances sont plus modérées, avec une précision de 95 % mais un rappel de 51 %, soit un score F1 de 66 %.

Le tableau 3.8 résume les résultats obtenus en termes d'exactitude (*accuracy*) et de fonction de perte (*loss function*) pour les différentes distributions de classes.

Le tableau 3.9 présente une comparaison synthétique des performances de différents modèles de la littérature dans le domaine de la détection d'intrusion utilisant le jeu de données *Edge-IIoTset*. Notre modèle hybride MCCNNDNN se distingue par une exactitude supérieure, ce qui démontre sa robustesse et son potentiel pour renforcer les mécanismes de sécurité dans les environnements IIoT. Cette comparaison souligne les progrès réalisés dans les techniques de détection d'intrusion et contribue au développement de solutions efficaces pour la protection des systèmes IIoT.

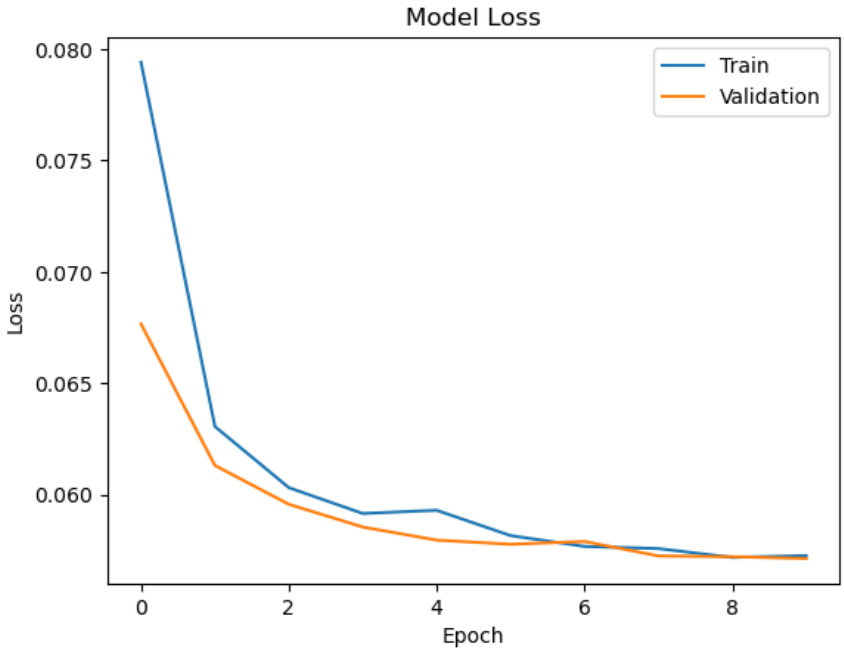

Figure 3.13 *– Fonction de perte pour la distribution à 6 classes*

Tableau 3.4 *– Métriques d'évaluation pour la distribution à 15 classes.*

Classe	Précision	Rappel	Score F1
Normal	1.00	1.00	1.00
Backdoor	0.94	0.97	0.96
DDoS_HTTP	0.74	0.96	0.84
DDoS_ICMP	1.00	1.00	1.00
DDoS_TCP	0.84	1.00	0.91
DDoS_UDP	1.00	1.00	1.00
Fingerprinting	0.35	0.46	0.40
MITM	1.00	1.00	1.00
Password	0.91	0.19	0.32
Port_Scanning	0.85	0.57	0.69
Ransomware	1.00	0.75	0.86
SQL_injection	0.46	0.91	0.61
Uploading	0.67	0.48	0.56
Vulnerability_scanner	1.00	0.83	0.90
XSS	0.62	0.35	0.44

3.7 Conclusion

Dans cette étude, notre objectif principal a été d'exploiter le jeu de données récent *Edge-IIoTset* afin de développer un modèle de détection d'intrusion à haute

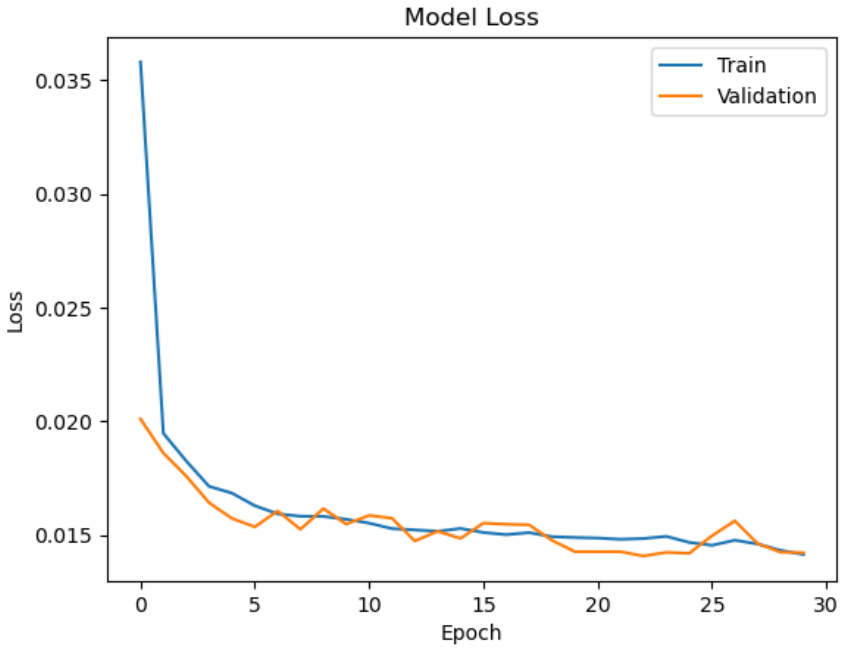

Figure 3.14 *– Fonction de perte pour la distribution à 9 classes*

Tableau 3.5 *– Métriques d'évaluation pour la distribution à 9 classes.*

Classe	**Précision**	**Rappel**	**Score F1**
Normal	1.00	1.00	1.00
DDoS_UDP	1.00	1.00	1.00
DDoS_ICMP	1.00	1.00	1.00
DDoS_TCP	0.99	1.00	1.00
Vulnerability_scanner	1.00	1.00	1.00
Password	0.98	0.85	0.91
DDoS_HTTP	0.87	0.98	0.92
Backdoor	1.00	0.98	0.99
MITM	1.00	1.00	1.00

Tableau 3.6 *– Métriques d'évaluation pour la distribution à 10 classes.*

Classe	**Précision**	**Rappel**	**Score F1**
Normal	1.00	1.00	1.00
DA	0.71	0.85	0.77
DDoS_UDP	1.00	1.00	1.00
DDoS_ICMP	1.00	0.99	1.00
DDoS_TCP	0.82	1.00	0.92
Vulnerability_scanner	0.91	0.81	0.92
Password	1.00	0.84	0.91
DDoS_HTTP	0.75	0.94	0.84
Backdoor	0.99	0.95	0.97
MITM	1.00	1.00	1.00

Tableau 3.7 *– Métriques d'évaluation pour la distribution à 6 classes.*

Classe	**Précision**	**Rappel**	**Score F1**
NORMAL	0.95	1.00	0.97
DDOS-ATTACK	0.68	0.99	0.81
INJECTION-ATTACK	1.00	1.00	1.00
MALWARE-ATTACK	0.95	0.51	0.66
SCANNING-ATTACK	1.00	1.00	1.00
MITM	0.98	0.73	0.84

Tableau 3.8 *– Résumé des résultats.*

Nombre de classes	**Exactitude**	**Fonction de perte**
2 classes	100 %	5.52×10^{-6}
6 classes	97.14 %	0.062
9 classes	99.50 %	0.014
10 classes	96.12 %	0.07
15 classes	95.41 %	0.08

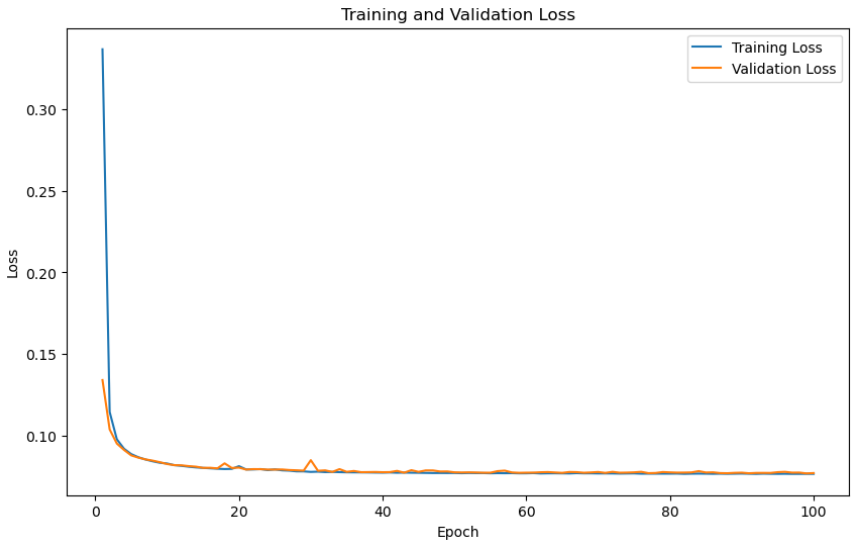

Figure 3.15 – *Fonction de perte pour la distribution à 10 classes*

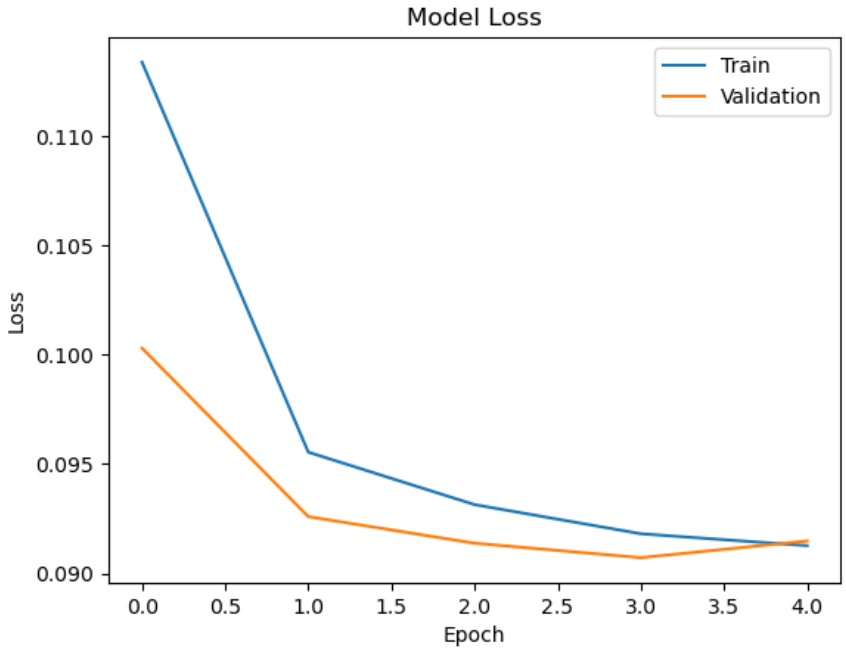

Figure 3.16 – *Fonction de perte pour la distribution à 15 classes*

Tableau 3.9 – *Comparaison des résultats avec les travaux antérieurs utilisant le jeu de données Edge-IIoTset.*

Auteurs	**Modèle**	**Exactitude**
M.A. Ferrag et al. [189]	DNN	96 %
Amina K. et al. [190]	CNN-LSTM	98.69 %
I. Tareq et al. [191]	Inception Time	94.94 %
Notre travail	**CNN-DNN**	**99.50 %**

performance pour les environnements IIoT. L'efficacité remarquable de notre modèle est largement attribuable au prétraitement rigoureux des données, étape cruciale qui constitue l'un des points forts de ce travail. L'utilisation de données soigneusement préparées et correctement encodées a contribué de manière significative à l'amélioration des performances du modèle.

Nous avons ensuite élargi notre analyse en considérant différentes distributions de classes, incluant des configurations à 2, 6, 9, 10 et 15 classes. Cette diversité de configurations nous a permis de mieux comprendre la distribution des attaques et leur impact dans le jeu de données *Edge-IIoTset*. Le choix de chaque configuration a été motivé par des considérations pratiques liées aux besoins du système et aux avantages attendus pour le modèle.

Notre approche hybride, combinant des réseaux de neurones convolutionnels (CNN) et des réseaux de neurones denses (DNN), a montré d'excellentes performances sur plusieurs métriques et pour différentes distributions de classes. En termes d'exactitude, les résultats sont particulièrement remarquables, atteignant 99,50 % pour la configuration à 9 classes. L'approche avec 6 classes a également montré une grande robustesse avec une exactitude de 97,14 %. L'exactitude pour la configuration à 10 classes demeure élevée, avec 96,12 %. Globalement, les modèles de détection d'intrusion proposés, et en particulier le modèle MCCNNDNN, ont démontré leur robustesse et leur capacité à traiter des scénarios d'attaque variés.

Dans nos travaux futurs, nous prévoyons d'explorer la combinaison avec d'autres techniques de détection, telles que les réseaux LSTM et les autoencodeurs. Nous envisageons également d'utiliser de nouveaux jeux de données et de déployer les modèles développés dans des environnements IIoT réels. Le modèle présenté ne constitue pas uniquement une contribution académique, mais également un point de départ vers un impact concret. La collaboration avec des partenaires industriels et des chercheurs représente une étape clé pour faire évoluer ce modèle d'une solution théorique performante vers un outil opérationnel pour la sécurité des systèmes IIoT.

Chapitre 4

Classification du trafic et des applications du darknet à l'aide d'un réseau neuronal à graphes hétérogènes

Sommaire

4.1 Introduction 68
4.2 Travaux connexes et revue de la littérature 70
4.3 Réseau de neurones à graphes hétérogènes 73
4.3.1 Construction de données graphiques 73
4.3.2 Passage de messages hétérogènes 73
4.3.3 Architecture du modèle 74
4.4 Methodologie 75
4.4.1 Description du dataset 75
4.4.2 Exploration et prétraitement des données 75
4.4.3 Conception du modèle 76
4.5 Résultats expérimentaux 78
4.5.1 Classification du trafic (cas à 4 classes) 80
4.5.2 Classification d'applications (cas à 8 classes) 83
4.6 Conclusion 87

4.1 Introduction

L'Internet, un réseau mondial de réseaux informatiques interconnectés facilitant l'échange d'informations, englobe divers appareils interconnectés tels que des ordinateurs, des serveurs et des objets connectés. Le World Wide Web (www), souvent simplement appelé le web, est un système de documents et de ressources interconnectés qui sont accessibles via Internet. Il a été développé à la fin du 20e siècle et est depuis devenu un élément fondamental de la vie moderne. Le web permet aux utilisateurs d'accéder à une vaste gamme d'informations, de communiquer avec d'autres personnes et de participer à diverses activités telles que le shopping, les loisirs et l'éducation via des navigateurs web. Les composants clés du World Wide Web incluent les pages web (documents écrits en HTML), les hyperliens (qui permettent la navigation entre les pages) et les serveurs web (qui hébergent et diffusent les pages web aux utilisateurs).

Le Deep Web, également appelé Web Invisible, représente une partie significative d'Internet qui n'est pas indexée par les moteurs de recherche, la rendant inaccessible via les navigateurs web standard. Il englobe des contenus confidentiels tels que des relevés financiers, des dossiers médicaux, des bases de données gouvernementales et d'autres données sensibles non destinées à être vues par le public [192]. En revanche, le Dark Web constitue une partie de l'Internet accessible uniquement via des logiciels spécialisés comme le navigateur Tor, et se caractérise par son anonymat. Tristement célèbre pour son association avec des activités illicites, il sert de marché pour des transactions illégales, incluant des drogues, des armes et des données volées, et facilite l'échange d'informations et de matériaux interdits [193].

L'expansion d'Internet a transformé la communication mondiale, permettant aux individus d'interagir par-delà les frontières avec une simple connexion. Le réseau Tor est une technologie qui offre aux utilisateurs confidentialité et anonymat en ligne. Il a été développé par une équipe d'informaticiens et de mathématiciens du Naval Research Laboratory (NRL) en réponse aux préoccupations concernant la vie privée et l'anonymat durant les premières phases du développement d'Internet [194]. Le réseau Tor utilise une technique appelée "routage en oignon" pour acheminer le trafic via plusieurs serveurs et le chiffre, rendant quasiment impossible pour quiconque de tracer les activités en ligne de l'utilisateur [195]. Le réseau Tor répartit les données sur plusieurs nœuds, réduisant ainsi le risque de surveillance ou d'interception des données [196]. Le Navigateur Tor, qui fait partie du réseau Tor, est un navigateur web spécialement conçu pour protéger la vie privée et l'anonymat des utilisateurs en ligne. Il résiste au profilage numérique (fingerprinting), ce qui permet aux utilisateurs de rester anonymes sur le web. Le Tor Project est une organisation à but non lucratif qui propose un navigateur web gratuit et privé pour protéger les utilisateurs contre la surveillance et le traçage en ligne.

Dans la communication anonyme, les données sont d'abord transmises à la couche de superposition de routage en oignon (Onion Routing Overlay Layer) avant d'atteindre le destinataire. Agissant comme un intermédiaire, le routeur oignon (Onion Router) établit une séquence de nœuds en chaîne, dont le nombre varie en fonction de la localisation du destinataire. Le nœud initial se connecte à l'expéditeur, tandis que le nœud terminal est lié au destinataire. Au lieu de divulguer les adresses IP, la chaîne de routage en oignon utilise une série de bits pour interagir avec le voisin le plus proche, renforçant la difficulté pour les attaquants cherchant à exposer les données privées des utilisateurs, car la chaîne ne peut identifier que son nœud routeur voisin immédiat [197].

Une autre technologie facilitant une communication internet sécurisée et privée est le réseau privé virtuel (VPN). En utilisant des protocoles de tunnellisation sur des réseaux existants, le VPN établit une connexion virtuelle point à point. Ce processus crée un tunnel sécurisé et chiffré entre l'appareil de l'utilisateur et le serveur VPN. Par conséquent, toutes les données transmises via ce tunnel sont cryptées, les protégeant contre tout accès non autorisé et préservant la vie privée et la sécurité en ligne de l'utilisateur [198]. La reconnaissance du trafic sur le dark web est essentielle pour contrer les activités cybercriminelles telles que le piratage, l'espionnage et les menaces persistantes avancées. Les techniques d'apprentissage profond conçues pour identifier et classer le trafic VPN et TOR sur le dark web jouent un rôle déterminant à cet égard. Ces méthodes permettent aux agences gouvernementales, aux organisations et aux experts en cybersécurité de détecter et de contrer efficacement la cybercriminalité sur le dark web.

Dans ce chapitre, nous proposons un Réseau de Neurones à Graphies Hétérogènes pour Darknet (DHGNN) pour relever le défi de la classification du trafic et des applications sur le darknet. L'utilisation d'un GNN dans ce domaine offre plusieurs avantages par rapport aux méthodes traditionnelles, tels qu'une précision accrue, une meilleure évolutivité et la capacité de traiter de vastes ensembles de données. Pour ce faire, nous avons utilisé le jeu de données CIC-Darknet2020, une base complète de données de trafic réseau accessibles au public, pour entraîner et évaluer notre modèle de réseau de neurones à graphies. Ce choix a été motivé par la robustesse du classifieur et ses performances supérieures par rapport aux classifieurs de trafic réseau antérieurs. Nous avons d'abord exploré les principales caractéristiques du jeu de données et analysé l'importance des caractéristiques afin de décider lesquelles retenir pour notre modèle. Ensuite, nous avons procédé à une étape de prétraitement pour obtenir un ensemble de données tabulaire qui sera converti en un jeu de données structuré en graphies avant d'être introduit dans notre classifieur DHGNN.

Le reste de ce chapitre est organisé comme suit : la Section II présente une synthèse de la littérature pertinente et des recherches antérieures dans le domaine. La Section III couvre les bases des Réseaux de Neurones à Graphies Hétérogènes, telles que la construction des données en graphies, la transmission de messages hétérogènes et l'architecture du modèle. La Section IV présente la méthodologie proposée, comprenant la description du jeu de données CIC-Darknet2020, l'exploration et le prétraitement des données, ainsi que la conception du modèle. La Section V fournit les résultats expérimentaux pour les deux scénarios de classification : le trafic et les applications. Ensuite, la Section VI est consacrée à la conclusion et aux suggestions pour les futures directions de recherche.

4.2 Travaux connexes et revue de la littérature

Dans des travaux récents, les chercheurs ont montré une préférence pour l'utilisation des réseaux de neurones graphiques (GNN) par rapport aux algorithmes d'apprentissage automatique traditionnels. Les GNN excellent dans l'identification des relations entre les caractéristiques des données représentées sous forme de graphes. Cette capacité est cruciale pour des tâches telles que l'analyse de réseaux sociaux, la modélisation d'interactions protéine-protéine et la détection d'anomalies, domaines où les méthodes traditionnelles peinent à intégrer directement ces relations.

Le travail présenté dans [199] se concentre sur une nouvelle approche de la sécurité des réseaux, spécifiquement sur la construction de réseaux d'activité et d'événements en utilisant des réseaux de neurones graphiques. Ce paradigme innovant est conçu pour contrer à la fois les attaques volumétriques et les menaces persistantes. Les auteurs ont utilisé deux jeux de données distincts : le jeu de données TOR-nonTOR et le jeu de données DDoS, obtenant des résultats de précision de 78

Dans leur étude innovante, Satriawan et al. [200] explorent le potentiel des réseaux de neurones graphiques (GNN) pour améliorer les capacités des systèmes de détection d'intrusion (IDS). Leur approche repose sur l'utilisation d'E-GraphSAGE, une variante de l'algorithme GraphSAGE conçue pour agréger efficacement et exploiter l'information du voisinage dans les données structurées en graphes. Cette méthodologie est particulièrement apte à capturer les modèles complexes sous-jacents souvent présents dans les données de trafic réseau, qui sont essentielles pour identifier et classer avec précision les activités malveillantes.

Dans une étude convaincante sur la détection et la classification de logiciels malveillants, les chercheurs de [201] se sont tournés vers les Réseaux de Neurones Graphiques (GNN) pour une approche plus nuancée. Utilisant le jeu de données CIC-AndMal2017, une collection complète d'échantillons de logiciels malveillants Android, l'étude examine la capacité des modèles GNN à détecter et à catégoriser efficacement différents types de malware. Grâce à une évaluation rigoureuse, l'étude évalue la performance des modèles sur plusieurs métriques clés, incluant la précision, le rappel, les courbes ROC (Receiver Operating Characteristic). Cette recherche renforce la viabilité des GNN en tant qu'outil puissant dans la lutte continue contre les logiciels malveillants.

Une étude intéressante [172] présente la mise en œuvre de l'apprentissage de représentations de graphes pour la détection d'intrusions basée sur le réseau et les hôtes. Elle fournit une vue d'ensemble complète des modèles de GNN, en détaillant les étapes pour construire un modèle, et inclut plusieurs exemples publiés dans ce domaine. Elle évalue également la robustesse de ces techniques face aux attaques adverses. Enfin, les auteurs passent en revue les forces et les limites de la détection d'intrusions basée sur les GNN et mettent en lumière les futures pistes de recherche.

Une autre étude dans [202] propose une investigation détaillée des attaques et défenses adverses sur les graphes, mettant en lumière la vulnérabilité des réseaux de neurones sur graphes (GNN) à ce type d'attaques, compte tenu de leurs performances exceptionnelles dans diverses tâches. L'étude examine en profondeur les stratégies, avantages et inconvénients des principaux algorithmes du domaine, en offrant une analyse comparative qui souligne la lutte constante entre l'amélioration de la robustesse des GNN et la correction des failles de sécurité via les défenses adverses.

Dans [203], les auteurs ont utilisé un réseau de neurones convolutif (CNN) pour classer le trafic Tor et VPN comme trafic darknet, et le trafic non-Tor et non-VPN comme trafic clearnet. Le modèle CNN développé a atteint une précision globale de 94 % pour classer le trafic réseau comme étant soit darknet soit bénin. De plus, le modèle a démontré une précision de 86 % dans la classification du type d'application utilisé pour générer le trafic. Le trafic d'application a été classé en catégories telles que la diffusion audio, la diffusion vidéo, VOIP, navigateur, chat, email, transfert de fichier ou pair-à-pair.

Sarkar et ses collègues [204] ont utilisé des réseaux de neurones profonds (DNN) pour classer le trafic Tor et non-Tor en s'appuyant sur le jeu de données UNB-CIC Tor et non-Tor (également désigné ISCXTor2016 [205]). Ils ont développé deux modèles distincts : DNN-A composé de trois couches, et DNN-B comprenant cinq couches. DNN-A a atteint une précision remarquable de 98,81 % pour distinguer les échantillons Tor des échantillons non-Tor, tandis que DNN-B a surpassé ce résultat avec une précision encore plus élevée de 99,89

Dans leur étude [206], l'accent a été mis exclusivement sur l'analyse des types de trafic à l'aide d'un jeu de données commun. Les chercheurs ont utilisé divers algorithmes de classification, incluant la forêt aléatoire (RF), les arbres de décision (DT), les k-plus proches voisins (KNN), le perceptron multicouche (MLP) et les arbres de décision avec boosting de gradient (GBDT). Ils ont catégorisé les données en deux groupes pour la classification binaire : bénin et darknet, tout en considérant également les quatre types de trafic de base pour une classification multi-classes (Tor, non-Tor, VPN ou non-VPN). Leurs résultats ont révélé que la forêt aléatoire (RF) obtenait des performances exceptionnelles en tant que classifieur de trafic, atteignant un score F1 impressionnant de 98,61 % pour la classification multi-classes.

Demertzis et al. [207] ont étendu les catégories d'applications à 11 sous-catégories et ont employé des Réseaux de Neurones Agnosticistes Pondérés (WANN) pour la classification. Contrairement aux Réseaux de Neurones Artificiels (RNA) standards, les WANN n'ajustent pas les poids des neurones mais modifient plutôt leur propre architecture de réseau de manière incrémentale. Les WANN évaluent différentes conceptions en fonction de leurs performances et de leur complexité, construisant de nouvelles couches du réseau à partir de l'architecture ayant obtenu le rang le plus élevé. Leur modèle WANN principal a atteint une précision de 92,68 % dans la classification de la couche application.

La recherche présentée dans [208] s'est concentrée sur la classification des types de trafic et d'applications en utilisant des Réseaux de Neurones Convolutifs (CNN) ainsi que deux autres méthodes d'apprentissage profond : les Mémoires à Long Court Terme (LSTM) et les Unités Récurrentes à Porte (GRU). Initialement, 20 caractéristiques ont été extraites via une Analyse en Composantes Principales (PCA), des Arbres de Décision (DT) et du Gradient Boosting Extrême (XGBoost), pour ensuite employer des architectures hybrides CNN-LSTM et CNN-GRU. Dans ce cadre, la couche CNN était responsable de l'extraction des caractéristiques à partir des données d'entrée, tandis que les LSTM et GRU étaient utilisées pour la prédiction de séquences basée sur ces caractéristiques. Il est à noter que la combinaison CNN-LSTM avec XGBoost pour la sélection de caractéristiques a démontré les scores F1 les plus élevés, atteignant une précision de 96 % pour la classification du type de trafic et de 89 % pour la classification du type d'application.

Marim et al. [209] ont analysé et catégorisé du trafic Darknet réel en utilisant le jeu de données CIC-Darknet2020. Ils ont employé l'extraction de caractéristiques et regroupé les sous-réseaux potentiels à l'aide d'une approche par n-grammes. De plus, ils ont évalué l'importance des caractéristiques principales identifiées via la méthode d'Élimination Récursive des Caractéristiques (RFE). Leurs résultats démontrent que des modèles simples tels que les Arbres de Décision et les Forêts Aléatoires atteignent une précision supérieure à 99 % dans la classification du trafic. Leur méthode offre une amélioration notable allant jusqu'à 13 % par rapport aux approches état de l'art existantes.

Alimoradi et al. [210] ont proposé un nouveau système d'aide à la décision appelé "Tor-VPN detector" pour classer le trafic darknet brut en quatre catégories : Tor, non-Tor, VPN et non-VPN. Le détecteur utilise une architecture de réseau neuronal profond comportant 79 neurones artificiels en entrée et 6 couches cachées pour découvrir les relations non linéaires complexes présentes dans les données brutes du trafic darknet. La performance de cette méthode proposée a été évaluée sur le jeu de données de référence DIDarknet. Leur modèle a surpassé les approches par réseaux neuronaux state-of-the-art pour la classification du trafic darknet, atteignant une précision de 96 %. Il est à noter que ces résultats impressionnants ont été obtenus sans utiliser aucune technique de prétraitement telle que l'extraction de caractéristiques ou les techniques de rééquilibrage des données, démontrant ainsi la puissance de leur modèle à traiter efficacement des données de trafic darknet brutes.

Zhu et al. [211] ont introduit le Graphe de Trafic Darknet (DTG), une représentation graphique capturant les interactions entre les clients locaux et les serveurs distants dans le trafic darknet. Sur cette base, ils ont intégré des Réseaux de Neurones sur Graphes (GNN) avec un mécanisme d'attention pour formuler les Réseaux de Neurones sur Graphes Darknet (DGNN). Ce modèle innovant exploite efficacement les caractéristiques du trafic bénin et du trafic darknet. Sur le jeu de données

CIC-Darknet2020, le DGNN a atteint des précisions remarquables de 98,52 % et 99,06 % respectivement pour la classification du trafic et des applications, surpassant ainsi les autres classifieurs.

4.3 Réseau de neurones à graphes hétérogènes

L'algorithme de Réseau de Neurones à Graphes Hétérogènes (HGNN) est conçu pour traiter et analyser des données graphes comprenant différents types de nœuds et d'arêtes [212]. Cette diversité des types de nœuds et d'arêtes caractérise précisément un graphe comme étant hétérogène [213]. L'algorithme HeteroGNN capture efficacement les relations complexes et riches présentes dans ce type de données [214], le rendant particulièrement utile pour des tâches telles que la classification de nœuds, la prédiction de liens et la classification de graphes dans divers domaines, incluant les réseaux sociaux, les graphes de connaissances et la détection de logiciels malveillants [172].

4.3.1 Construction de données graphiques

Cette phase est fondamentale et consiste à traduire les données brutes du trafic réseau en un format de graphe structuré qu'un GNN peut traiter. Considérons un graphe hétérogène G = (V, E), où V est l'ensemble des nœuds et E est l'ensemble des arêtes [215].

- L'ensemble des nœuds V est divisé en deux sous-ensembles, $V = V_h \cup V_f$, où V_h représente les nœuds hôtes et V_f les nœuds de flux.
- Les nœuds hôtes possèdent des caractéristiques $X_h \in \mathbf{R}^{(|V_h| \times d_h)}$, et les nœuds de flux possèdent une matrice de caractéristiques $X_f \in \mathbf{R}^{(|V_f| \times d_f)}$, où d_h et d_f représentent les dimensions des caractéristiques pour les nœuds hôtes et de flux, respectivement. Ces caractéristiques servent d'entrée au HGNN [214].
- Identifiez les relations ou connexions entre les nœuds qui seront représentées sous forme d'arêtes. Nous considérons que l'ensemble des arêtes E est divisé en deux types différents : flux et hôte $E = E_{hf} \cup E_{fh}$, où E_{hf} représente les arêtes allant des nœuds hôtes vers les nœuds de flux, et E_{fh} les arêtes allant des nœuds de flux vers les nœuds hôtes. Cette distinction est cruciale pour les graphes hétérogènes où différents types de nœuds et d'arêtes peuvent encoder des types distincts d'informations et de relations.

4.3.2 Passage de messages hétérogènes

Le cœur de l'algorithme HGNN réside dans son mécanisme hétérogène de passage de messages, qui met à jour les représentations vectorielles des nœuds en agrégeant les informations provenant des nœuds voisins de différents types [215,216]. Ce processus peut être formalisé comme suit pour un système à deux types de nœuds en utilisant la Convolution sur Graphes Hétérogènes (HConv).

Pour les arêtes E_{hf}, la règle de mise à jour des caractéristiques des nœuds de flux est :

$$H_f^{(l+1)} = \sigma(W_{hf}^{(l)} \cdot AGG(\left\{h_h^{(l)} | \forall (v_h, v_f) \in E_{hf}\right\})) \tag{4.1}$$

Pour les arêtes E_{fh}, la règle de mise à jour des caractéristiques des nœuds hôtes est :

$$H_h^{(l+1)} = \sigma(W_{fh}^{(l)} \cdot AGG(\left\{h_f^{(l)} | \forall (v_h, v_f) \in E_{fh}\right\})) \tag{4.2}$$

Ici, $H_f^{(l)}$ et $H_h^{(l)}$ désignent les représentations vectorielles des nœuds de flux et des nœuds hôtes à la couche l.

$W_{hf}^{(l)}$ et $W_{fh}^{(l)}$ sont les poids ajustables pour chaque type d'arête, σ représente une fonction d'activation non linéaire, et AGG désigne une fonction d'agrégation et son rôle dans la combinaison des informations provenant du voisinage d'un nœud :

- **Agrégation moyenne :**

$$AGG_{mean} = mean(H[v_i], (\forall (v_i, v_j) \in E) \tag{4.3}$$

- **Agrégation des sommes :**

$$AGG_{sum} = sum(H[v_i], \forall (v_i, v_j) \in E) \tag{4.4}$$

4.3.3 Architecture du modèle

Le modèle est composé de L couches HConv, où les représentations vectorielles des nœuds sont mises à jour de manière itérative [215, 217]. Initialement :

$$H_h^{(0)} = X_h \ \ and \ \ H_f^{(0)} = X_f \tag{4.5}$$

Après L couches, les représentations vectorielles des nœuds de flux, $H_f^{(L)}$, sont utilisées dans une couche de prédiction linéaire :

$$Y = W_{out} \cdot H_f^{(L)} + b_{out} \tag{4.6}$$

$Y \in \mathbb{R}^{|V_f| \times C}$ représente les logits pour C classes pour chaque nœud de flux, où W_{out} et b_{out} sont des paramètres ajustables.

La structure multicouche avec L couches HConv permet une extraction de caractéristiques et un apprentissage de représentations progressivement complexes. La transition des caractéristiques initiales des nœuds vers des représentations vectorielles finales capables de supporter des tâches de classification illustre la capacité du modèle à transformer des données brutes en informations actionnables.

Enfin, le modèle est optimisé par apprentissage supervisé, en minimisant la perte d'entropie croisée entre les scores prédits Y et les étiquettes vraies Y_{True} :

$$L = -\sum_i \sum_c Y_{True,ic} \cdot log(Y_{ic}) \tag{4.7}$$

Ici, lindice i parcourt les nuds de flux, tandis que c parcourt les classes. Les paramètres, l'indice i parcourt les nœuds de flux, et c parcourt les classes. Les paramètres $(W_{out}, W_{fh}^{(l)}, W_{hf}^{(l)}, b_{out})$ sont optimisés pour minimiser L à l'aide de techniques d'optimisation basées sur le gradient telles que Adam, Adamax, AdaGrad, RMSProp, ...

4.4 Methodologie

4.4.1 Description du dataset

Le jeu de données CIC-Darknet2020, publié par l'Institut Canadien pour la Cybersécurité en 2020, comprend des données de trafic réseau provenant de différents scénarios simulés, englobant à la fois des modèles normaux et malveillants. Son objectif est d'offrir aux chercheurs et aux praticiens une collection complète de données de trafic réseau pour contribuer au développement, à l'évaluation et à l'expérimentation des méthodes de cybersécurité.

Le jeu de données CIC-Darknet2020 est structuré en deux niveaux. Le premier niveau utilise une approche à double couche pour générer à la fois du trafic bénin et du trafic darknet. Le deuxième niveau englobe divers scénarios de trafic tels que la Navigation, le P2P, l'Audio-Stream, le Transfert, la Discussion, l'Email, la Vidéo-Stream et la VOIP au sein du trafic darknet.

Pour assurer sa représentativité, le jeu de données combine des jeux de données antérieurs comme ISCXTor2016 et ISCXVPN2016, en fusionnant leur trafic VPN et Tor dans les catégories darknet correspondantes [13].

4.4.2 Exploration et prétraitement des données

Il existe plusieurs caractéristiques intéressantes que nous pouvons utiliser pour notre modèle DHGNN :

- The **timestamp :** Nous pouvons traiter les données pour extraire des informations concernant le jour de la semaine et l'heure de la journée. De manière générale, le trafic réseau est saisonnier, et les connexions qui surviennent la nuit ou durant des jours inhabituels sont suspectes.
- Processing **IP addresses :** Traiter des adresses IP comme 192.168.200.4 peut être difficile en raison de leur nature non numérique et de leur adhésion à des règles complexes. Une approche pourrait consister à les catégoriser en quelques groupes basés sur la connaissance de notre configuration réseau locale. Alternativement, une solution largement utilisée et plus adaptable consiste à les convertir en représentation binaire, où '192' serait représenté par '11000000'.
- **Flow Duration :** Le nombre de paquets et le nombre d'octets sont des caractéristiques qui présentent généralement des distributions à queue lourde. Par conséquent, elles nécessiteront un traitement particulier si c'est effectivement le cas.

Le jeu de données CIC-Darknet2020 comprend divers types de données, incluant des valeurs numériques comme la durée des flux, des caractéristiques catégorielles telles que Label/Label.1 spécifiquement désignées pour la variable cible, ainsi que d'autres comme des horodatages ou des adresses IP. Pour la suite, nous déterminerons les représentations appropriées pour ces types de données, en nous appuyant sur des connaissances expertes.

- Nous allons encoder en one-hot les informations de la semaine en les extrayant de l'horodatage. Ensuite, nous renommerons les colonnes résultantes pour une meilleure lisibilité.

- Nous allons encoder les informations temporelles de la journée (heure, minute, seconde) en valeurs numériques continues, puis normaliser ces caractéristiques entre 0 et 1.
- Nous utilisons un encodage binaire pour traiter à la fois les adresses IP source et destination. Plutôt que d'utiliser les 32 bits pour encoder l'adresse IPv4 complète, nous ne conserverons que les 16 derniers bits, qui revêtent la plus grande importance dans ce contexte. Il est à noter que les 16 premiers bits représentent généralement soit 192.168 si l'hôte appartient au réseau interne, soit une autre valeur s'il est externe.
- Nous établissons une division entraînement/validation/test en utilisant des ratios de 80/10/10.
- Enfin, nous devons aborder la mise à l'échelle de trois caractéristiques : la durée des flux, le nombre de paquets et le nombre d'octets. Nous utilisons PowerTransformer() de scikit-learn pour modifier leurs distributions.

Le tableau 4.1 présente les caractéristiques des nœuds utilisées dans le modèle DH-GNN.

Tableau 4.1 *– DHGNN Model Node Features*

Feature Type	**Feature Description**
Host	Source IP (ipsrc_1 to ipsrc_16)
Host	Destination IP (ipdst_1 to ipdst_16)
Flow	Daytime
Flow	Day of the Week (Monday to Friday)
Flow	Flow Duration
Flow	Flow Packets/s
Flow	Flow Bytes/s
Flow	FIN Flag Count
Flow	SYN Flag Count
Flow	RST Flag Count
Flow	Protocol

4.4.3 Conception du modèle

La conception de notre modèle est illustrée par la figure 4.1.

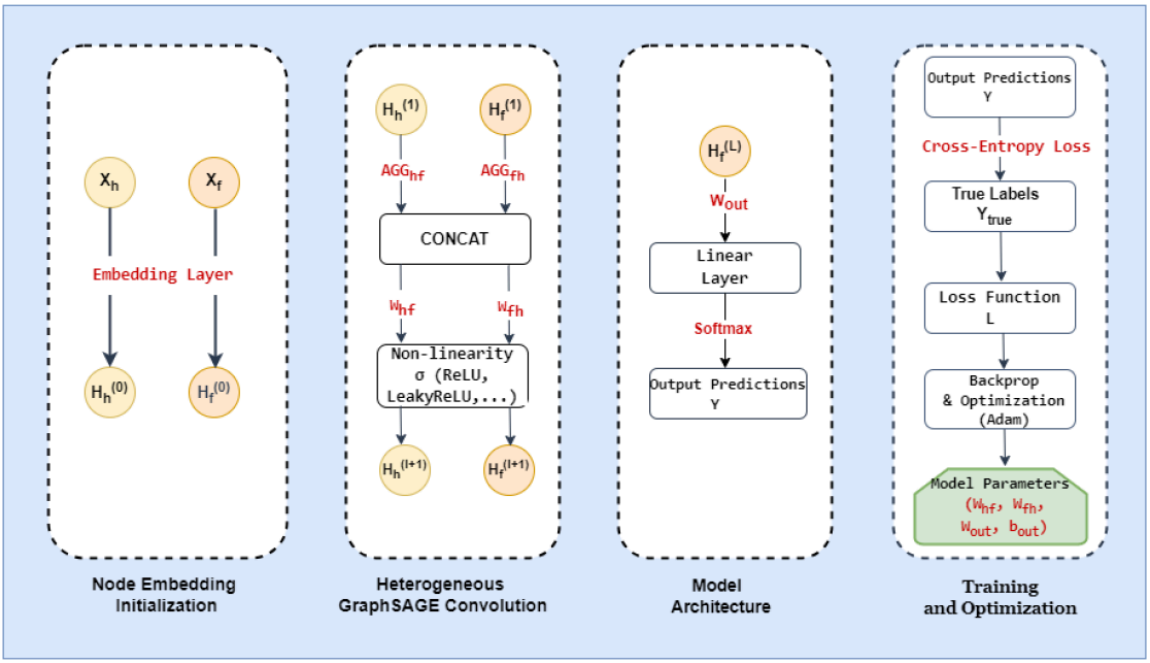

Figure 4.1 – *Processus de conception du modèle.*

Les caractéristiques d'entrée pour les nœuds hôtes X_h et les nœuds de flux X_f sont traitées par une couche d'embedding pour obtenir les représentations vectorielles initiales $H_h^{(0)}$ et $H_f^{(0)}$.

- Pour les nœuds hôtes, X_h inclut des caractéristiques telles que les bits des adresses IP source et destination.
- Pour les nœuds de flux, X_f inclut des caractéristiques telles que l'heure de la journée, le jour de la semaine, la durée du flux, le nombre d'octets et de paquets du flux, les indicateurs de drapeaux et le protocole.

Pour chaque couche HeteroGNN, les représentations vectorielles des nœuds de la couche précédente $H_h^{(l)}$ et $H_f^{(l)}$ sont traitées par des fonctions d'agrégation distinctes (AGG_{hf} et AGG_{fh}).

- AGG_{hf} agrège les informations des nœuds de flux voisins pour mettre à jour les représentations vectorielles des nœuds hôtes.
- AGG_{fh} agrège les informations des nœuds hôtes voisins pour mettre à jour les représentations vectorielles des nœuds de flux.
- Les fonctions d'agrégation utilisées sont des couches SAGEConv, qui calculent une somme pondérée des caractéristiques des voisins et appliquent une activation non linéaire.

Les représentations vectorielles agrégées provenant de AGG_{hf} et AGG_{fh} sont concaténées ensemble.

- Les représentations vectorielles concaténées sont ensuite projetées à l'aide de matrices de poids ajustables W_{hf} et W_{fh}, respectivement.

Une fonction d'activation non linéaire σ, telle que ReLU ou LeakyReLU, est appliquée aux représentations vectorielles projetées.

- Les représentations vectorielles résultantes deviennent les nouvelles représentations des nœuds hôtes et de flux ($H_h^{(l+1)}$ et $H_f^{(l+1)}$) pour la prochaine couche HeteroGNN.

- Après passage à travers toutes les couches HeteroGNN (sept couches), les représentations vectorielles finales des nœuds de flux ($H_f^{(L)}$) sont projetées à l'aide d'une matrice de poids W_{out}.

Les représentations vectorielles projetées sont traitées par une couche linéaire pour obtenir les logits.

- Une activation softmax est appliquée aux logits pour obtenir les prédictions finales (Y) sous forme de distributions de probabilité sur les classes.

Les prédictions de sortie (Y) sont comparées aux étiquettes vraies (Y_{true}) à l'aide d'une fonction de perte d'entropie croisée.

- La perte résultante (L) représente l'erreur entre les étiquettes prédites et les étiquettes vraies.

La perte (L) est rétropropagée à travers l'ensemble du modèle pour calculer les gradients des paramètres du modèle (W_{hf}, W_{fh}, W_{out}, et toutes les polarisations).

- Un algorithme d'optimisation, tel qu'Adam, est utilisé pour mettre à jour les paramètres du modèle sur la base des gradients calculés.
- Ce processus est répété pendant de multiples itérations (époques=100) pour optimiser le modèle et améliorer sa performance sur la tâche de classification.

4.5 Résultats expérimentaux

Dans le classifieur DHGNN, nous configurerons quatre ou huit couches de SAGEConv avec LeakyReLU pour chaque type de nœud (quatre couches pour la classification de trafic et huit couches pour la classification d'application). Ensuite, une couche linéaire produira un vecteur de dimension quatre ou huit, chaque dimension représentant une classe. De plus, nous entraînerons ce modèle de manière supervisée en utilisant la perte d'entropie croisée et l'optimiseur Adam.

Nous spécifions ensuite le GNN hétérogène, en incorporant trois paramètres : le nombre de dimensions cachées, le nombre de dimensions de sortie et le nombre de couches.

L'optimisation des hyperparamètres dans les GNN est cruciale, car elle influence directement la performance de classification du réseau. Nos expériences, détaillées dans le tableau 4.2, explorent minutieusement les paramètres clés en évaluant la précision dans des plages définies. Nous déterminons que la configuration optimale implique une profondeur de réseau de sept couches, entraînée sur 100 époques en utilisant l'optimiseur Adam et la fonction d'activation Leaky_ReLU.

Dans les sous-sections suivantes, nous présenterons les résultats pour les scénarios de classification de trafic et de classification d'applications.

Tableau 4.2 *– Hyperparamètres du modèle DHGNN et plage d'exploration*

Hyperparamètre	**Intervalle de valeurs**	**Valeur optimale**	**Description**
Dispositif (Device)	-	Dynamique	Déterminé automatiquement en fonction de la disponibilité de CUDA : *cuda* ou *cpu*.
Nombre de couches	[3, 4, 5, 6, 7]	7	Nombre total de couches dans le modèle, représentant la profondeur du réseau.
Dimension de la couche cachée	64	64	Dimension fixe des couches cachées du modèle.
Dimension de sortie	8 et 4	8 et 4	Dimension de la couche de sortie, fixée à 8 et 4 pour ce modèle.
Fonction d'activation	[Tanh, ReLU, LeakyReLU]	LeakyReLU	Fonctions d'activation évaluées, avec LeakyReLU retenue pour le modèle optimal.
Taux d'apprentissage (LR)	[0.001, 0.002, 0.003, 0.004, 0.005, 0.006]	0.004	Intervalle des taux d'apprentissage explorés, avec 0.004 sélectionné comme valeur optimale.
Optimiseur	[Adam, AdamW, RMSprop, Adamax]	Adam	Différents algorithmes doptimisation ont été évalués, Adam ayant été retenu pour le modèle optimal.
Fonction de perte	Entropie croisée	Entropie croisée	Fonction de perte utilisée pour l'entraînement, non modifiée durant les expérimentations.
Nombre total dépoques	[50, 100, 150, 200]	100	Nombre total dépoques testées, avec 100 retenu comme durée optimale dentraînement.

4.5.1 Classification du trafic (cas à 4 classes)

Notre attention s'est principalement portée sur la couche initiale du jeu de données CICDarknet2020, qui incluait des échantillons de données catégorisés comme trafic dark web (VPN et TOR) et trafic bénin (Non-VPN et Non-Tor).

Les quatre classes sont représentées par le diagramme circulaire illustré dans la figure 4.2 où le décompte de chaque valeur est le suivant : "Non-Tor" : 93 309, "NonVPN" : 23 861, "VPN" : 22 919, et "Tor" : 1 392 occurrences.

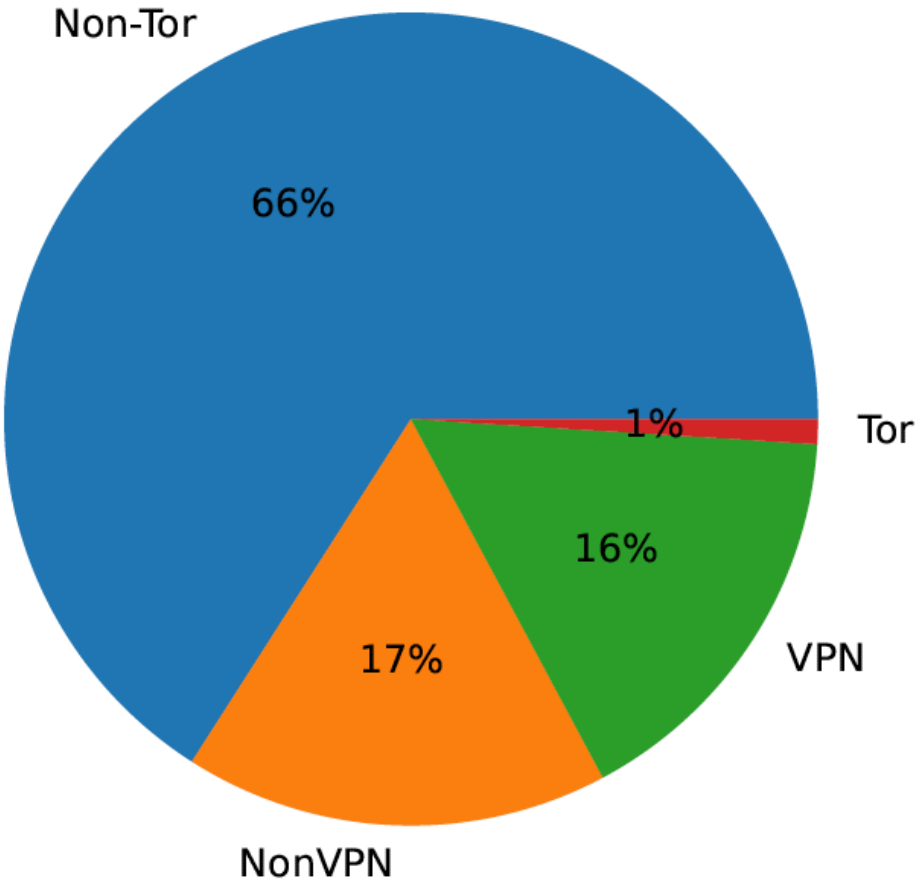

Figure 4.2 *– Proportion de chaque classe dans le dataset CIC-Darknet2020 Cas de trafic*

Nous évaluons la performance de l'approche proposée en utilisant les métriques d'évaluation suivantes : Précision, Rappel, F-mesure et Exactitude.

En classification de trafic, la précision de notre modèle DHGNN est de 99,80 %, indiquant un niveau très élevé de prédictions correctes.

Un graphique des erreurs d'entraînement et des erreurs de validation durant l'apprentissage du modèle est présenté par la figure 4.3.

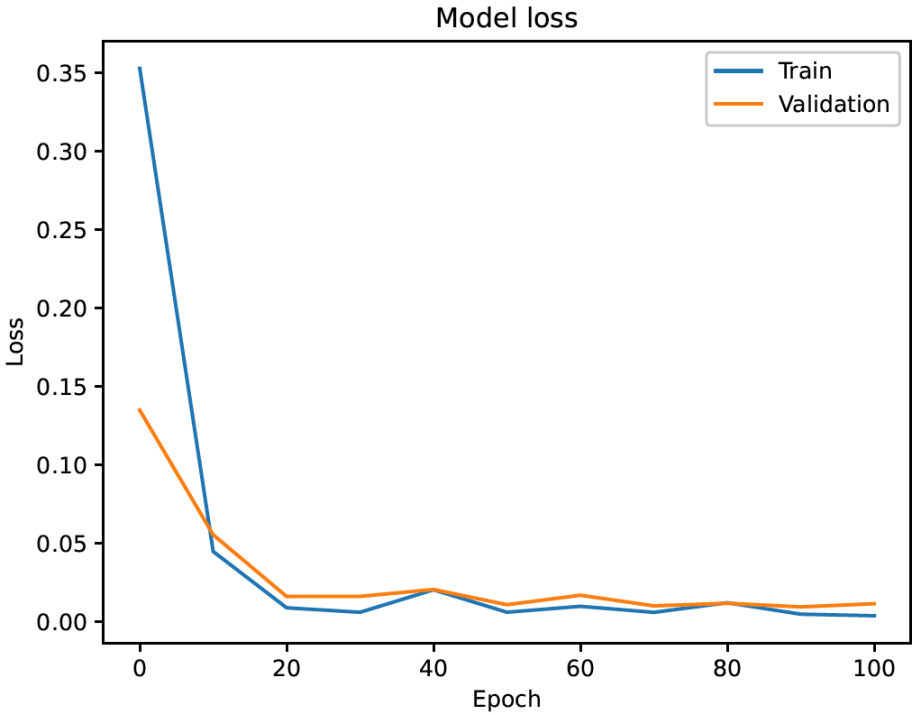

Figure 4.3 *– Perte du modèle en fonction d'une époque durant le processus d'entraînement - Cas de trafic*

Un graphique circulaire représentant la proportion d'échantillons mal classés est fourni par la figure 4.4. Si nous comparons ce diagramme aux proportions originales du jeu de données, nous constatons que le modèle performe mieux pour les classes majoritaires. Ceci n'est pas surprenant car les classes minoritaires sont plus difficiles à apprendre (moins d'échantillons), et ne pas les détecter est moins pénalisant (avec 93 309 flux Non-Tor contre 1 392 Tor). La détection des classes NonVPN et VPN pourrait être améliorée avec des techniques telles que le suréchantillonnage et l'introduction de pondérations de classes durant l'entraînement.

Des perspectives supplémentaires peuvent être obtenues en examinant la matrice de confusion (voir figure 4.5). Cette matrice de confusion affiche des résultats intéressants. Le tableau 4.3 compare la performance de divers modèles de réseaux de neurones sur graphes (GNN), incluant ResNet, GIN, GCN, GAT, GraphSAGE, Rev-GNN, DGNN et le modèle DHGNN proposé, sur une tâche de classification de trafic. Les métriques de performance rapportées sont l'exactitude, la précision, le F1-score et le rappel (TPR). Le classifieur DHGNN proposé atteint la plus haute exactitude de 0,9980, précision de 0,9898, F1-score de 0,9820 et rappel de 0,9322.

Tableau 4.3 – *Évaluation de la performance des différentes méthodes de classification de trafic.*

Model	Accuracy	Precision	F1	Recall (TPR)
ResNet [218]	0.9043	0.7625	0.7120	0.6991
GIN [219]	0.9358	0.8653	0.7569	0.6741
GCN [220]	0.9260	0.8464	0.7108	0.6158
GAT [221]	0.9235	0.8324	0.7021	0.6086
GraphSAGE [222]	0.9556	0.9142	0.8407	0.7791
RevGNN [223]	0.9131	0.8384	0.6348	0.5131
DGNN [211]	0.9852	0.9662	0.9488	0.8322
Our DHGNN	**0.9980**	**0.9898**	**0.9820**	**0.9748**

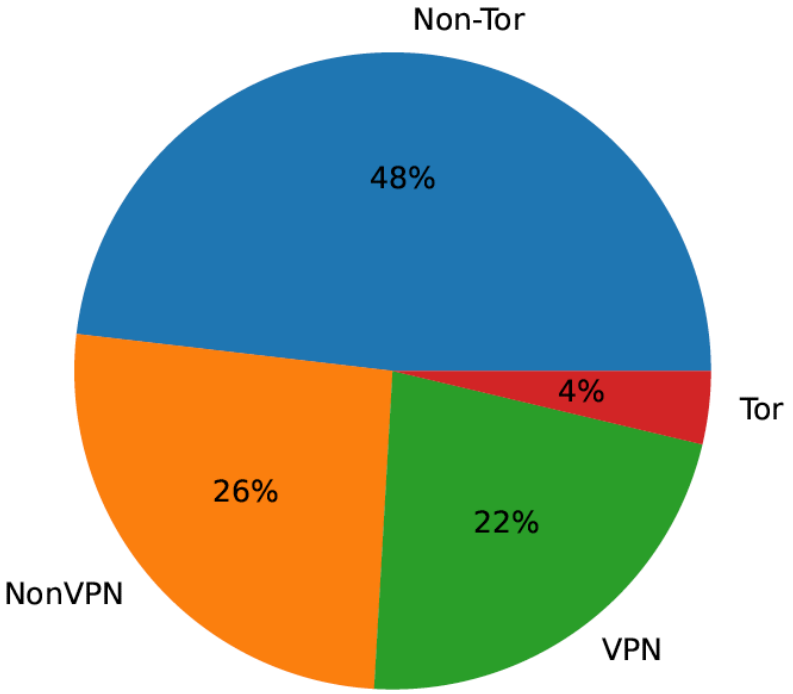

Figure 4.4 – *Proportion de chaque classe mal classée - Cas de trafic*

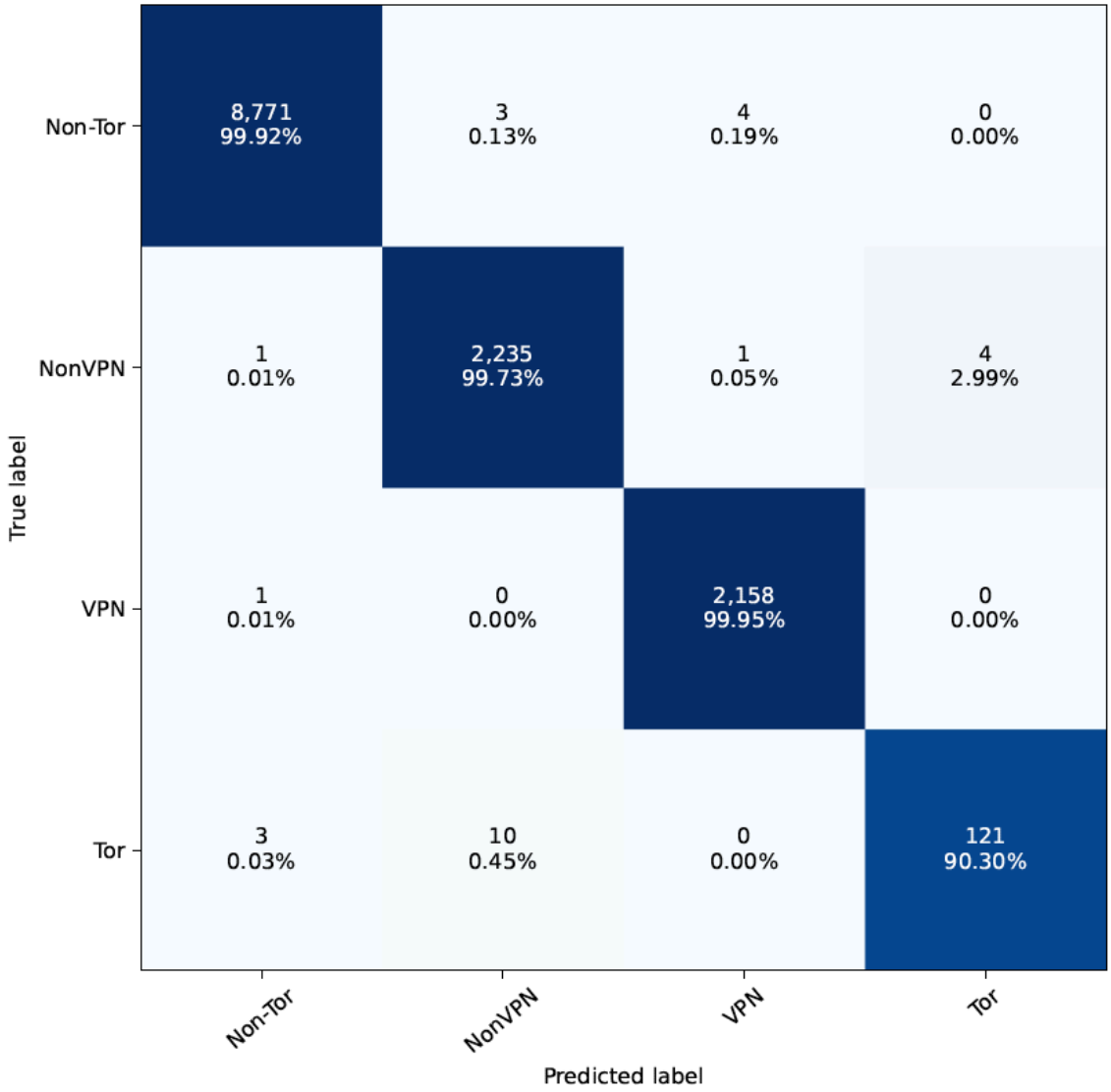

Figure 4.5 *– Matrice de confusion pour la classification multi-classes des flux - Cas de trafic*

4.5.2 Classification d'applications (cas à 8 classes)

Notre attention s'est maintenant portée sur la deuxième couche du jeu de données CICDarknet2020, qui incluait des échantillons de données étiquetés comme Audio-Stream, Chat, Email, P2P, VOIP, Video-Stream, Transfer et Browsing.

Nous utilisons les métriques d'évaluation Précision, Rappel, F-mesure et Exactitude pour évaluer la performance de l'approche proposée.

La figure 4.6 montre une courbe représentant la perte du modèle en fonction du nombre d'époques durant le processus d'entraînement. La courbe commence à une valeur de perte relativement élevée, indiquant que les prédictions initiales du modèle étaient assez imprécises ou éloignées des vraies valeurs au début du processus d'entraînement.

Dans les premières phases de l'entraînement, la courbe de perte présente une descente abrupte. Cette diminution rapide de la perte suggère que le modèle apprend rapidement et améliore significativement ses prédictions à chaque époque. Durant

cette phase, le modèle effectue des ajustements substantiels à ses paramètres internes pour minimiser l'erreur entre ses prédictions et les vraies valeurs.

Au fur et à mesure que l'entraînement progresse, le taux auquel la perte diminue commence à ralentir graduellement. Ceci indique que le modèle continue de s'améliorer, mais que les améliorations deviennent plus modestes.

En classification d'applications, la précision de notre modèle DHGNN est de 98,80 %, indiquant un niveau très élevé de prédictions correctes.

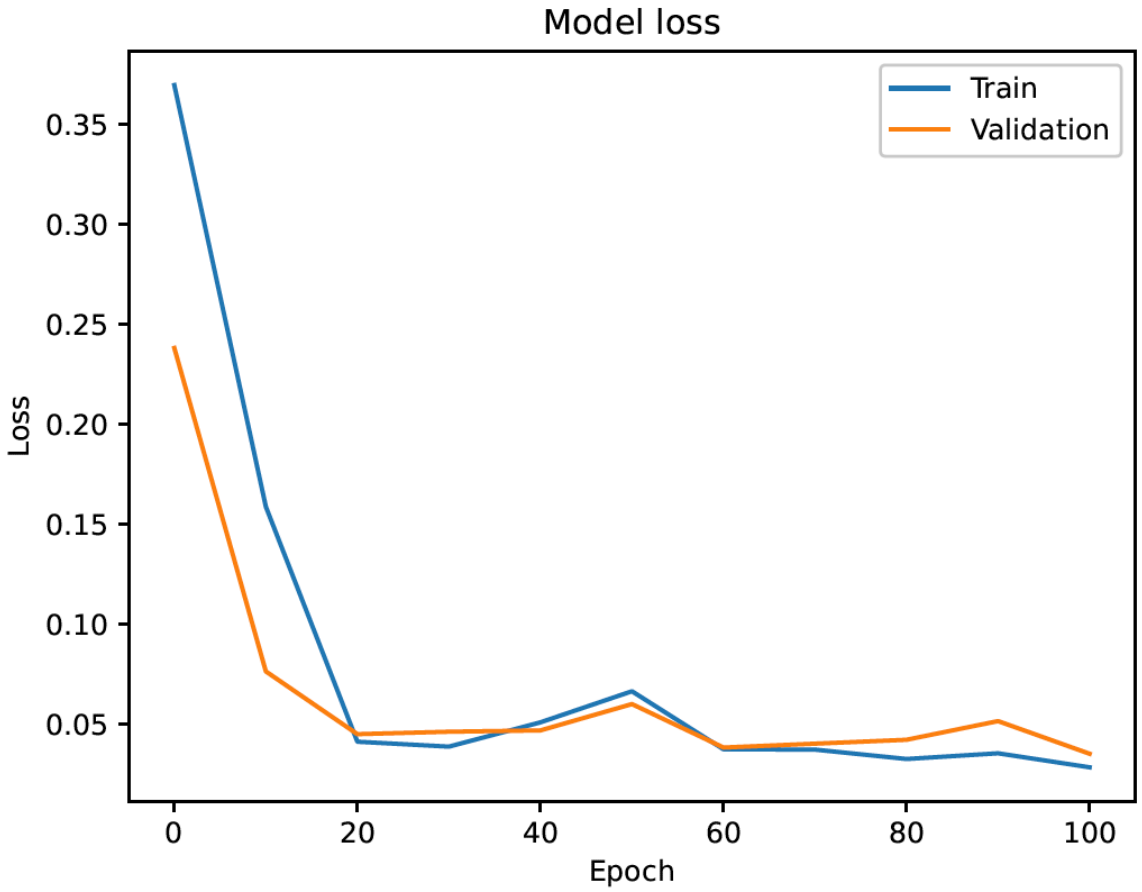

Figure 4.6 – *Perte du modèle en fonction des époques durant le processus dapprentissage Cas dapplication.*

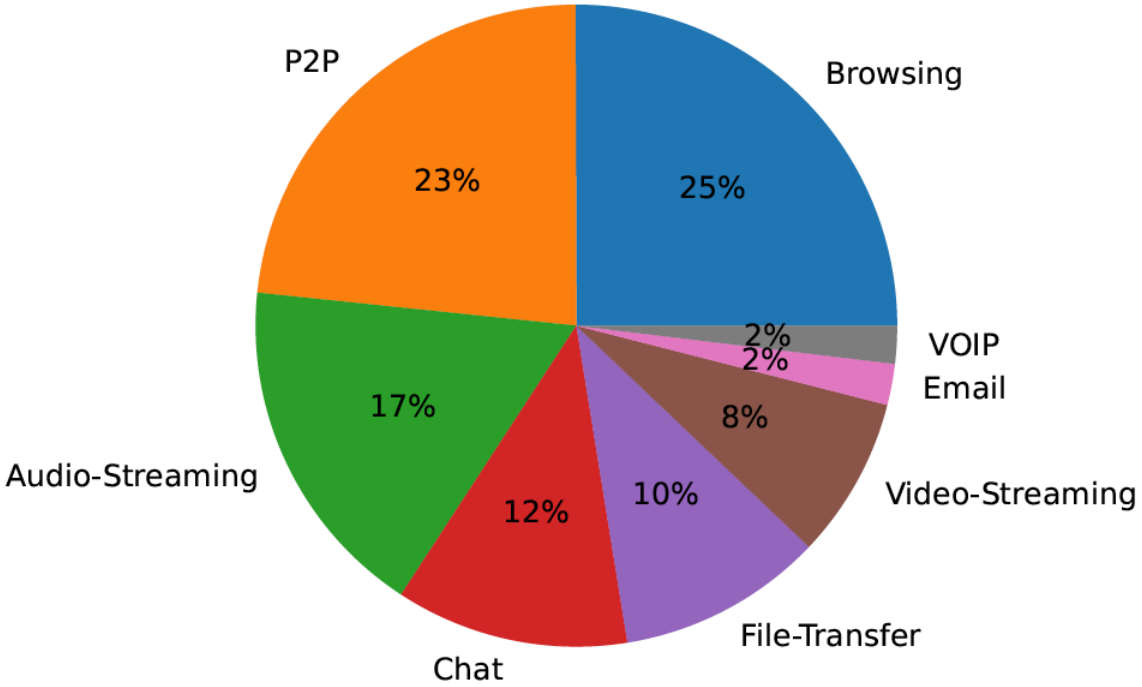

Figure 4.7 – *Proportion de chaque classe mal classée Cas dapplication*

Le diagramme circulaire fourni par la figure 4.7 montre la proportion de chaque classe mal classée dans le cas de la classification d'applications.

La figure 4.8 présente une matrice de confusion, qui est un tableau résumant la performance d'un modèle de classification en comparant les étiquettes prédites avec les étiquettes réelles pour un ensemble d'instances. La matrice de confusion comporte 8 lignes et 8 colonnes, représentant les 8 classes différentes de la tâche de classification. Les lignes correspondent aux étiquettes réelles, tandis que les colonnes correspondent aux étiquettes prédites.

À partir de cette matrice de confusion, nous pouvons observer que certaines classes présentent des taux d'erreur plus élevés que d'autres. Par exemple, la classe "VOIP" (dernière ligne) semble avoir un taux de mauvaise classification relativement élevé, avec des instances mal classées comme "Browsing", "Email" et "Audio-Streaming".

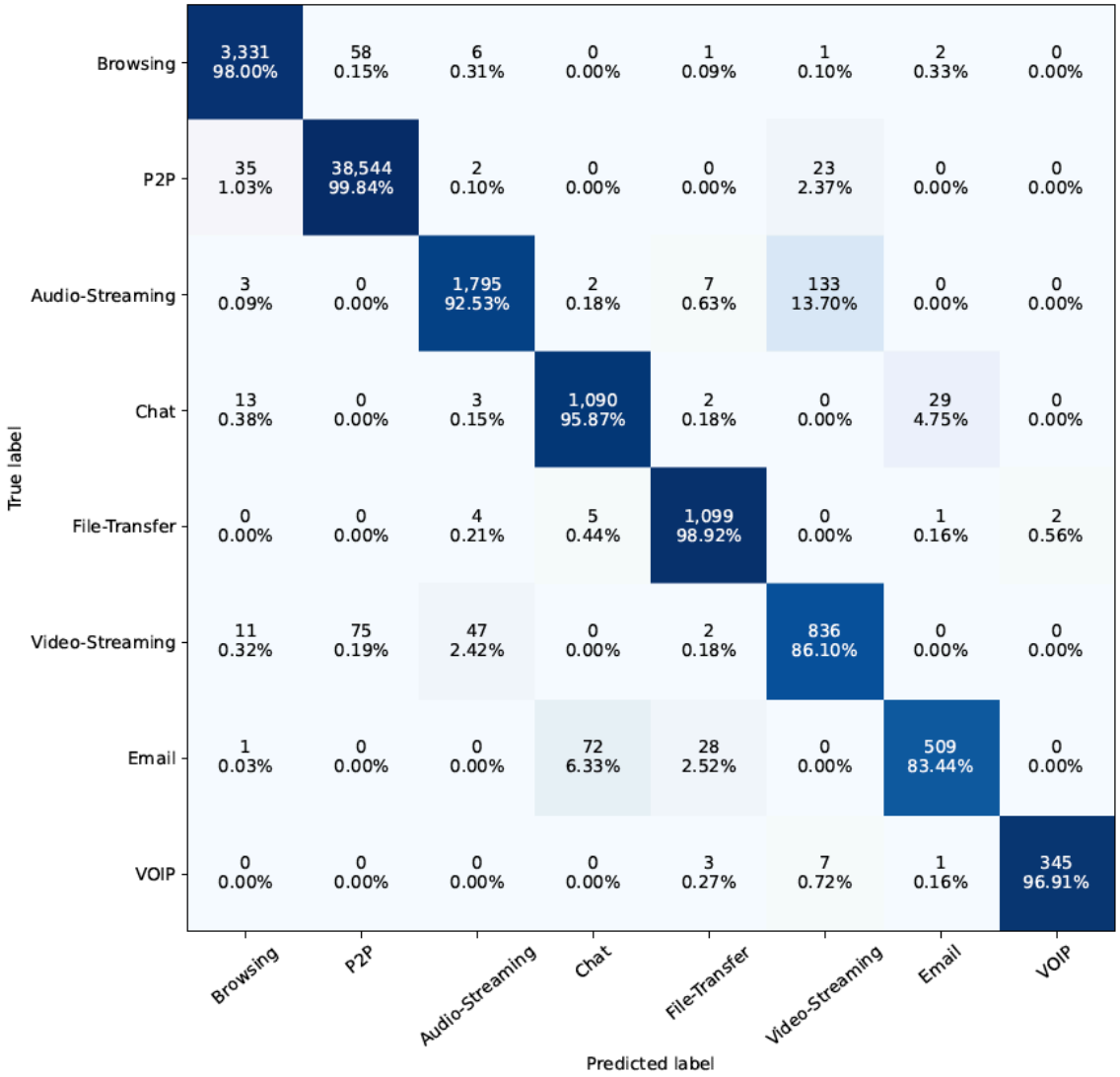

Figure 4.8 *– Matrice de confusion pour la classification multi-classes des flux - Cas d'applicatitions.*

Le tableau 4.4 présente les valeurs de rappel des différents modèles pour classifier diverses applications, telles que Audio-streaming, Browsing, Chat, Email, Transfer, P2P, VOIP et Video-streaming. Le classifieur DHGNN proposé atteint le rappel le plus élevé pour la plupart des applications, incluant Browsing (0,9800), Chat (0,9587), Transfer (0,9892), P2P (0,9984), VOIP (0,9691) et Video-stream (0,8610). DGNN [211] performe mieux pour la classification Audio-Streaming et Email avec un rappel de 0,9788 et 0,9682 respectivement.

Tableau 4.4 *– Comparaison des résultats de rappel des méthodologies de classification d'applications*

Model	Audio-Streaming	Browsing	Chat	Email	Transfer	P2P	VOIP	Video-Streaming
DarkDetect [208]	0.8100	0.8900	0.8100	0.8400	0.8300	0.9300	0.8700	0.8600
Didarknet [203]	0.9200	0.4700	0.8600	0.6700	0.7500	0.9500	0.6100	0.8800
ResNet [218]	0.8237	0.7808	0.8946	0.5553	0.6774	0.8898	0.7864	0.6104
GIN [219]	0.8928	0.3397	0.8341	0.1015	0.5059	0.7724	0.4706	0.3977
GCN [220]	0.8557	0.0072	0.7995	0.0020	0.3638	0.4968	0.5561	0.0240
GAT [221]	0.8061	0.0228	0.7838	0.1009	0.0731	0.7223	0.2545	0.0217
GraphSAGE [222]	0.9044	0.2423	0.8485	0.7804	0.5364	0.8756	0.8439	0.4438
RevGNN [223]	0.7472	0.0023	0.7723	0.0179	0.0181	0.0124	0.0089	0.0042
DGNN [211]	**0.9788**	0.7577	0.9458	**0.9682**	0.8804	0.9702	0.9548	0.8557
Our DHGNN	0.9253	**0.9800**	**0.9587**	0.8344	**0.9892**	**0.9984**	**0.9691**	**0.8610**

Le tableau 4.5 compare les performances de différentes méthodes, notamment le Random Forest (RF), les méthodes d'ensemble (RF+KNN+DT), ResGAT, DeepImage, CNN+LSTM, ainsi que les modèles proposés DHGNN et DGNN, appliqués à une tâche de classification d'applications. Les métriques présentées incluent la précision (accuracy), le score F1 et le modèle utilisé. Le modèle proposé DHGNN atteint le score F1 le plus élevé (0,9879) et une précision de 0,9880, surpassant ainsi les autres approches. Le modèle DGNN présente également de bonnes performances, avec une précision de 0,9906 et un score F1 de 0,9569.

Tableau 4.5 *– Analyse comparative de la classification d'applications*

Method	Accuracy	F1-score	Model
[224]	—	0.922	RF
[225]	0.9788	0.94	RF+KNN+DT
[226]	—	0.8807	ResGAT
[203]	0.86	0.86	DeepImage
[227]	0.9222	0.92	CNN+LSTM
[228]	0.8599	0.86	RF
[211]	**0.9906**	0.9569	DGNN
Our	0.9880	**0.9879**	DHGNN

4.6 Conclusion

Dans ce chapitre, nous avons proposé une approche de détection et de classification du trafic darknet basée sur un classificateur à réseau de neurones de graphes

hétérogènes (DHGNN). Ce classificateur a été évalué à laide du jeu de données CIC-Darknet2020, qui comprend quatre types de trafic (Tor, Non-Tor, VPN, Non-VPN) et huit catégories dapplications (Audio-Stream, Navigation, Chat, Courriel, P2P, Transfert, Vidéo-Stream, VOIP). Les résultats expérimentaux démontrent la supériorité des performances de classification du modèle DHGNN par rapport aux autres méthodes, mettant en évidence son potentiel pour renforcer les mesures de sécurité des réseaux en détectant efficacement le trafic provenant du darknet.

Les travaux futurs visent à approfondir cette approche en explorant dautres classificateurs basés sur les réseaux de neurones de graphes (GNN). Par ailleurs, nous prévoyons détudier des modèles hybrides combinant différentes architectures de GNN afin de tirer parti de leurs forces complémentaires, dans le but de concevoir un modèle plus robuste et plus adaptable. Nous reconnaissons également limportance de linterprétabilité des modèles en cybersécurité et avons lintention de développer des méthodes visant à améliorer lexplicabilité du modèle DHGNN, afin de rendre ses processus décisionnels plus transparents et compréhensibles. Cet accent mis sur linterprétabilité est essentiel pour renforcer la confiance, assurer la transparence opérationnelle et favoriser une compréhension approfondie des menaces détectées.

Conclusion générale

L'essor de l'Internet des Objets (IoT) et de sa déclinaison industrielle (IIoT) a profondément redéfini les frontières du monde numérique en connectant des milliards d'appareils, capteurs et systèmes interopérables. Cette connectivité ubiquitaire, si elle favorise la performance, l'efficacité et l'automatisation, a également amplifié la complexité et la vulnérabilité des infrastructures. La détection d'anomalies s'impose dans ce contexte comme un axe stratégique pour garantir la fiabilité, la résilience et la sécurité des systèmes connectés.

Cette thèse s'est inscrite dans cette dynamique, avec pour objectif principal de concevoir et d'évaluer des approches d'apprentissage profond et adaptatif pour la détection d'anomalies et la classification du trafic dans les environnements IoT et IIoT. Les travaux menés ont permis d'explorer l'ensemble de la chaîne de traitement, depuis la compréhension théorique des anomalies jusqu'à la mise en uvre de modèles intelligents, en passant par le prétraitement et l'ingénierie des données, étapes essentielles à la robustesse des modèles.

Sur le plan méthodologique, la recherche a d'abord posé les fondements conceptuels en clarifiant la notion d'anomalie au sein de l'IoT et en distinguant trois formes systémiques majeures : les défaillances, les intrusions et les anomalies comportementales. Cette formalisation a permis de structurer la problématique et de dégager les exigences fondamentales d'une détection efficace : adaptabilité, légèreté computationnelle, et contextualisation des signaux.

Dans une deuxième étape, un modèle hybride CNNDNN a été développé et testé sur le jeu de données Edge-IIoTset. L'architecture proposée a montré une capacité remarquable à généraliser sur plusieurs configurations de classes (2, 6, 9, 10 et 15 classes), atteignant des performances supérieures à celles des approches classiques, notamment en précision, rappel et F1-score. Ces résultats démontrent la pertinence des architectures profondes dans la capture des structures complexes et non linéaires des flux IoT.

Dans un troisième temps, les travaux ont été étendus au traitement de flux chiffrés via l'utilisation d'un Réseau de Neurones à Graphes Hétérogènes (DHGNN) appliqué au dataset CIC-Darknet2020. Cette approche a permis d'exploiter les relations entre entités du réseau (hôtes, paquets, sessions) pour détecter des comportements suspects sans accès au contenu du trafic. Les résultats obtenus confirment le potentiel des modèles relationnels pour la cybersurveillance des environnements distribués et illustrent une voie prometteuse vers la détection d'anomalies explicative et contextuelle.

Au-delà des performances obtenues, la thèse met en lumière plusieurs enseignements clés :

la qualité du prétraitement des données (codage, filtrage, vectorisation) conditionne directement la stabilité des modèles ;

la représentation des dépendances structurelles entre dispositifs renforce la capacité des modèles à comprendre les interactions complexes ;

enfin, la nécessité d'adaptation continue face à la dérive conceptuelle des données est un impératif pour toute solution en production dans l'IoT.

Cependant, plusieurs limites subsistent. Les modèles développés demeurent dépendants de la disponibilité de jeux de données annotés, souvent rares dans les environnements industriels réels. De plus, les contraintes computationnelles des dispositifs embarqués restreignent l'implémentation de modèles profonds complexes en périphérie (edge computing). Enfin, l'interprétabilité des modèles question centrale dans le domaine de la cybersécurité nécessite encore des efforts méthodologiques pour garantir la transparence et la confiance des opérateurs.

Ces constats ouvrent des perspectives de recherche prometteuses. Parmi elles :

le recours à l'apprentissage fédéré pour permettre la détection collaborative tout en préservant la confidentialité des données ;

l'intégration de mécanismes d'adaptation en ligne pour gérer la dérive conceptuelle en temps réel ;

l'exploration de modèles explicables et interprétables (XAI) pour renforcer la compréhension et la validation des décisions de détection ;

enfin, l'utilisation de modèles multimodaux combinant vision, graphes et signaux temporels pour une compréhension holistique du comportement des systèmes IoT.

En conclusion, cette thèse a contribué à la consolidation d'un cadre d'analyse et de modélisation intelligent pour la détection d'anomalies dans les environnements IoT et IIoT. Elle a démontré la pertinence de combiner apprentissage profond, ingénierie des données et modélisation relationnelle pour répondre aux défis croissants de sécurité et de fiabilité des infrastructures connectées. Les avancées obtenues constituent une base solide pour le développement futur de systèmes de détection plus autonomes, adaptatifs et explicables, capables d'évoluer au rythme des technologies et des menaces.

Liste des publications

- International peer-reviewed journals

[1] Wafaa Ferhi, Djilali Moussaoui, Mourad Hadjila, Al Baraa Bouidaine, Anomaly detection for IIoT : analyzing Edge-IIoTset dataset with varied class distributions, Scientific and Technical Journal of Information Technologies, Mechanics and Optics, Vol. 25, No.5, p. 876-887, September-October 2025. doi : 10.17586/2226-1494-2025-25-5-876-887

[2] Etchiali, A., Ferhi, W., Mhamedi, M., Hadjila, M., Merzoug, M., Hachemi, M.H. (2025). Darknet traffic and application classification using Heterogeneous Graph Neural Network. Ingénierie des Systèmes dInformation, Vol. 30, No. 10, pp. 2571-2581. https ://doi.org/10.18280/isi.301005

[3] M. Hadjila, M. Merzoug, W. Ferhi, D. Moussaoui, A.B. Bouidaine, M.H. Hachemi, "Obfuscated malware detection using deep neural network with ANOVA feature selection on CIC-MalMem-2022 dataset", Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2024, vol. 24, no 5, p. 849-857, doi : 10.17586/2226-1494-2024-24-5-849-857

[4] A.Bekkouche, M.Merzoug, M.Hadjila, and W.Ferhi. "Towards Early Breast Cancer Detection : A Deep Learning Approach", Engineering, Technology & Applied Science Research Vol. 14, No. 5, 2024, 17517-17523

[5] A.B. Bouidaine, D. Moussaoui, M. Hadjila, W. Ferhi, M. H. Hachemi. Deep Learning-Based Anomaly and Intrusion Detection Using the CSE-CIC-IDS2018 Dataset. Engineering, Technology & Applied Science Research. Vol. 15, No. 4, p. 24782-24787, August 2025, https ://doi.org/10.48084/etasr.11173

- International conferences with peer review

[6] FERHI, Wafaa, HADJILA, Mourad, DJILLALI, Djilali Moussaoui, and Al Baraa Bouidaine. Machine Learning-based Classification of Diabetes Disease : A Case Study with Orange Data Mining. In : 2023 International Conference on Electrical Engineering and Advanced Technology (ICEEAT). IEEE, 2023. p. 1-6.

[7] W. Ferhi, M. Hadjila, D. Moussaoui and A. B. Bouidaine, "Anomaly Detection in IoT : State-of-the-Art Techniques and Implementation Insights," 2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC), Setif, Algeria, 2024, pp. 1-7, doi : 10.1109/ICEEAC61226.2024.10576293.

[8] FERHI, Wafaa, HADJILA, Mourad, MOUSSAOUI, Djillali, et al. Enhancing Cybersecurity in the Internet of Vehicles (IoV) : A Deep Learning Approach for Anomaly and Intrusion Detection. In : GLOBECOM 2024-2024 IEEE Global Communications Conference. IEEE, 2024. p. 517-522.

[9] Al Baraa Bouidaine, Wafaa Ferhi, Djilali Moussaoui, Mourad Hadjila, 'Heart Disease Evaluation using Deep Learning Techniques', International Congress on

Health Sciences and Medical Technologies (ICHSMT 23), (Online) Tlemcen Algeria, 26-28 December 2023.

[10] Albaraa BOUIDAINE, Wafaa FERHI, Mourad HADJILA, and Djillali MOUSSAOUI. Deep Learning Classifier for DDoS Attacks Detection Across CSE-CIC-IDS2018 Dataset. In the First International Conference on Artificial Intelligence and Sustainable Development (ICAISD25), held on April 12th-13th, 2025, Ahmed Zabana University-Relizane, Algeria.

- National conferences

[11] FERHI, Wafaa, HADJILA, Mourad, DJILLALI, Djilali Moussaoui, and Al Baraa Bouidaine. "IoT Anomaly Detection Strategies : A Roadmap for Effective Research and Implementation", Algerian Doctoral Conference on Computer Science ADCCS2024.

Bibliographie

[1] K. Ashton, "That internet of things thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[2] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of things (iot) : A literature review," *Journal of Computer and Communications*, vol. 3, no. 5, pp. 164–173, 2015.

[3] E. Borgia, "The internet of things vision : Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, 2014.

[4] G. Misra, V. Kumar, A. Agarwal, and K. Agarwal, "Internet of things (iot) a technological analysis and survey on vision, concepts, challenges, innovation directions, technologies, and applications," *American Journal of Electrical and Electronic Engineering*, vol. 4, no. 1, pp. 23–32, 2016.

[5] V. A. Effendy, "Data security in internet of things systems based on distributed blockchain," *Journal of the American Institute*, vol. 2, no. 5, pp. 696–704, 2025.

[6] A. A. Badawi, "Internet of things (iot) : Origins, embedded technologies, smart applications, and its growth in the last decade," 2024.

[7] M. M. Pohan and B. Soewito, "Injection attack detection on internet of things device with machine learning method," Jurnal Riset Sistem Informasi dan Teknik Informatika (JURASIK), pp. 204–212, 2023. [Online]. Available : https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

[8] M. Chui, M. Collins, and M. Patel, "The internet of things : Catching up to an accelerating opportunity," 2021.

[9] N. Tesla, "Interview by j. b. kennedy," Colliers Magazine, 1926, uRL : https ://www.organism.earth/library/document/colliers-interview-nikola-tesla.

[10] V. Bush, "As we may think," *The Atlantic Monthly*, 1945, uRL : https ://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/.

[11] J. C. R. Licklider, "Memos on the intergalactic computer network," ARPA Memoranda, 1962, uRL : https ://historyofcomputercommunications.info/.

[12] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "The past and future history of the internet," *Communications of the ACM*, vol. 40, no. 2, pp. 102–108, 1997.

[13] R. M. Metcalfe and D. R. Boggs, "Ethernet : Distributed packet switching for local computer networks," *Communications of the ACM*, vol. 19, no. 7, pp. 395–404, 1976.

[14] *IEEE Standard for Ethernet (IEEE Std 802.3-2018)*, IEEE Standards Association Std., 2018. [Online]. Available : https://standards.ieee.org/standard/802_3-2018.html

[15] "En toute discrétion : l'iot en action," *Bulletin Magazine*, 2023. [Online]. Available : https://www.bulletin.ch/fr/news-detail/en-toute-discretion-liot-en-action.html

[16] R. Want, "An introduction to rfid technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.

[17] J. M. Kahn, R. H. Katz, and K. S. Pister, "Next century challenges : Mobile networking for smart dust," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 271–278.

[18] K. Ashton, "That internet of things thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[19] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, Std. RFC 8200, 2017.

[20] *IEEE Std 802.15.4-2003 : Wireless MAC and PHY Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Standards Association Std., 2003. [Online]. Available : https://standards.ieee.org/standard/802_15_4-2003.html

[21] Zigbee Alliance, "Zigbee specification," 2005. [Online]. Available : https://zigbeealliance.org/

[22] C. Harrison and I. A. Donnelly, "A theory of smart cities," in *Proceedings of the 55th Annual Meeting of the ISSS*, Hull, UK, 2011.

[23] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things : A survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[24] M. E. Porter and J. E. Heppelmann, "How smart, connected products are transforming competition," *Harvard Business Review*, vol. 92, no. 11, pp. 64–88, 2014.

[25] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things : A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[26] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran *et al.*, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1093–1110.

[27] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics : A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[28] M. Grieves and J. Vickers, "Digital twin : Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*. Cham : Springer, 2016, pp. 85–113.

[29] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing : Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[30] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning : Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[31] T. Magara and Y. Zhou, "Internet of things (iot) of smart homes : Privacy and security," *Journal of Electrical and Computer Engineering*, vol. 2024, no. 1, p. 7716956, 2024.

[32] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven iot ehealth : Promises and challenges of iot in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659–676, 2018.

[33] J. Lee, H. Davari, J. Singh, and V. Pandhare, "Industrial artificial intelligence for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 18, pp. 20–23, 2018.

[34] L. D. Xu, E. L. Xu, and L. Li, "Industry 4.0 : State of the art and future trends," *International Journal of Production Research*, vol. 56, no. 8, pp. 2941–2962, 2018.

[35] M. Ben-Daya, E. Hassini, and Z. Bahroun, "Internet of things and supply chain management : A literature review," *International Journal of Production Research*, vol. 57, no. 15–16, pp. 4719–4742, 2019.

[36] Deloitte, "2023 manufacturing industry outlook," 2023. [Online]. Available : https://www2.deloitte.com/us/en/insights/industry/manufacturing/manufacturing-industry-outlook.html

[37] C. R. Institute, "Smart factories at scale : Seizing the trillion-dollar opportunity," 2021.

[38] M. Chui, M. Collins, and M. Patel, "The internet of things : Catching up to an accelerating opportunity," 2021.

[39] V. Albino, U. Berardi, and R. M. Dangelico, "Smart cities : Definitions, dimensions, performance, and initiatives," *Journal of Urban Technology*, vol. 22, no. 1, pp. 3–21, 2015.

[40] R. Kitchin, *The Data Revolution : Big Data, Open Data, Data Infrastructures and Their Consequences*. SAGE Publications, 2014.

[41] *ISO/IEC 30141 :2018 – Internet of Things (IoT) – Reference Architecture*, International Organization for Standardization Std., 2018. [Online]. Available : https://www.iso.org/standard/65695.html

[42] Industry IoT Consortium, "Industrial internet reference architecture (iira) version 1.10," 11 2022, retrieved 2025-11-09. [Online]. Available : https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/11/IIRA-v1.10.pdf

[43] O. Logvinov, B. Kraemer, C. Adams *et al.*, "Standard for an architectural framework for the internet of things (iot) – ieee p2413," IEEE P2413 Working Group, Technical Report, 2016.

[44] *ITU-T Y.2060 (06/2012) : Overview of the Internet of Things*, International Telecommunication Union Std., 06 2012. [Online]. Available : https://www.itu.int/rec/T-REC-Y.2060-201206-I

[45] *oneM2M TS-0001-V4.23.0 : Functional Architecture*, oneM2M Std., 09 2024. [Online]. Available : https://www.onem2m.org/technical/published-specifications/release-4

[46] AIOTI, "High level architecture (hla) release 5.0," 2020. [Online]. Available : https://aioti.eu/wp-content/uploads/2020/12/AIOTI_HLA_R5_201221_Published.pdf

[47] L. Atzori, A. Iera, and G. Morabito, "The internet of things : A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[48] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things : A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[49] N. Kumar, P. K. Mallick, S. Misra *et al.*, "A review on architecture, technologies, protocols and challenges in internet of things," in *Smart Trends in Computing and Communications*. Springer, 2021, pp. 1–13.

[50] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot) : A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[51] Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira, "Artificial intelligence-based anomaly detection of energy consumption in buildings : A review, challenges and perspectives," *Applied Energy*, vol. 287, p. 116601, 2021.

[52] A. Khraisat, A. Gouglidis, A. Ahmed, and J. Slay, "A critical review of intrusion detection systems in the internet of things : Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 18, 2021.

[53] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[54] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot) : A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[55] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things : A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[56] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things : A survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[57] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing : A taxonomy, survey and future directions," in *Internet of Everything*. Springer, 2018, pp. 103–130.

[58] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing : Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[59] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[60] M. Chiang and T. Zhang, "Fog and iot : An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[61] C. Mouradian, D. Naboulsi, S. Yangui, R. Glitho, M. Morrow, and P. Polakos, "A comprehensive survey on fog computing : State-of-the-art and research

challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[62] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing : Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[63] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the MCC Workshop on Mobile Cloud Computing*, 2012, pp. 13–16.

[64] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, "Key ingredients in an iot recipe : Fog computing, cloud computing, and more fog computing," in *IEEE CAMAD*, 2014, pp. 325–329.

[65] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing : Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[66] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing : A taxonomy, survey and future directions," in *Internet of Everything*. Springer, 2018, pp. 103–130.

[67] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things : The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[68] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security : A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[69] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security : an exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[70] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. S. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[71] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, "A survey on security in internet of things with a focus on the impact of emerging technologies," *Internet of Things*, vol. 19, p. 100564, 2022.

[72] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[73] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world : Present and future challenges," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2483–2495, 2018.

[74] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran *et al.*, "Understanding the mirai botnet," in *26th USENIX Security Symposium*. USENIX Association, 2017, pp. 1093–1110. [Online]. Available : https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[75] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of things : Security vulnerabilities and challenges," in *2015 IEEE Symposium on Computers and Communication (ISCC)*, 2015, pp. 180–187.

[76] M. Vanhoef and F. Piessens, "Key reinstallation attacks : Forcing nonce reuse in wpa2," in *Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)*, 2017, pp. 1313–1328.

[77] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of iot new features on security and privacy : New threats, existing solutions, and challenges yet to be solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606–1616, 2019.

[78] M. A. Khan and K. Salah, "Iot security : Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[79] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on internet of things : Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017.

[80] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[81] A. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial internet of things," in *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1–6.

[82] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things : Perspectives and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014.

[83] A. Whitmore, A. Agarwal, and L. D. Xu, "The internet of thingsa survey of topics and trends," *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[84] Y. Huang, M. Esmalifalak, H. Nguyen, R. Zheng, Z. Han, H. Li, and L. Song, "Bad data injection in smart grid : Attack and defense mechanisms," *IEEE Communications Magazine*, vol. 51, no. 1, pp. 27–33, 2013.

[85] OWASP, "Owasp iot top 10 – the internet of things security guidance," https://owasp.org/www-project-internet-of-things/, 2023, consulté pour synthèse des faiblesses récurrentes IoT.

[86] T. Mavroeidakos and V. Chaldeakis, "On the security of iot device identity and attestation mechanisms," *Journal of Cybersecurity*, vol. 6, no. 1, p. tyaa008, 2020.

[87] TechTarget, "Anomaly detection," n.d., accessed 2025-11-03. [Online]. Available : https://www.techtarget.com/searchdatamanagement/definition/anomaly-detection

[88] T. L. Yasarathna, M. Liyanage, and N.-A. Le-Khac, "Deep learning based autonomous anomaly detection for security in sdn-iot networks," *IEEE Open Journal of the Communications Society*, 2025.

[89] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for iot time-series data : A survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, 2020. [Online]. Available : https://doi.org/10.1109/JIOT.2020.2974828

[90] L. Erhan, "Smart anomaly detection in sensor systems : A multi-perspective review," *Information Fusion*, vol. 67, pp. 64–79, 2021. [Online]. Available : https://doi.org/10.1016/j.inffus.2020.09.012

[91] W. Ferhi, M. Hadjila, D. Moussaoui, and A. Bouidaine, "Anomaly detection in iot : State-of-the-art techniques and implementation insights," in *Proceedings of the 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC)*. IEEE, 2024, pp. 1–7. [Online]. Available : https://doi.org/10.1109/ICEEAC60857.2024.10627532

[92] Wazuh, "Enhancing it security with anomaly detection in wazuh," October 12 2023, accessed 2025-11-03. [Online]. Available : https://wazuh.com

[93] P. Kamat and R. Sugandhi, "Anomaly detection for predictive maintenance in industry 4.0a survey," in *E3S Web of Conferences*, vol. 170, 2020, p. 02007. [Online]. Available : https://doi.org/10.1051/e3sconf/202017002007

[94] J. Mao, H. Wang, and J. Spencer, B. F., "Toward data anomaly detection for automated structural health monitoring," *Structural Health Monitoring*, vol. 20, no. 4, pp. 1609–1626, 2021. [Online]. Available : https://doi.org/10.1177/1475921720942310

[95] Y. Himeur, K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira, "Artificial intelligence-based anomaly detection of energy consumption in buildings," *Applied Energy*, vol. 287, p. 116601, 2021. [Online]. Available : https://doi.org/10.1016/j.apenergy.2021.116601

[96] Buildings IoT, "Leverage iot for energy efficiency in building management," December 28 2023, accessed 2025-11-03. [Online]. Available : https://buildingsiot.com

[97] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal, "Iot healthcare analytics : The importance of anomaly detection," in *Proceedings of the IEEE Conference on Advanced Information Networking and Applications*, 2016, pp. 994–997. [Online]. Available : https://doi.org/10.1109/AINA.2016.136

[98] J. Liu, P. Wang, D. Jiang, J. Nan, and W. Zhu, "An integrated data-driven framework for surface water quality anomaly detection and early warning," *Journal of Cleaner Production*, vol. 251, p. 119145, 2020. [Online]. Available : https://doi.org/10.1016/j.jclepro.2019.119145

[99] J. Gillespie, T. P. da Costa, X. Cama-Moncunill, T. Cadden, J. Condell, T. Cowderoy, and R. Ramanathan, "Real-time anomaly detection in cold chain transportation using iot technology," *Sustainability*, vol. 15, no. 3, p. 2255, 2023. [Online]. Available : https://doi.org/10.3390/su15032255

[100] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004. [Online]. Available : https://doi.org/10.1109/TDSC.2004.2

[101] National Institute of Standards and Technology, "Glossary of key information security terms (nist ir 7298 rev. 2)," U.S. Department of Commerce, Tech. Rep., 2013. [Online]. Available : https://doi.org/10.6028/NIST.IR.7298r2

[102] ——, "Security and privacy controls for information systems and organizations (nist sp 800-53 rev. 5)," U.S. Department of Commerce, Tech. Rep., 2020. [Online]. Available : https://doi.org/10.6028/NIST.SP.800-53r5

[103] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection : A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009. [Online]. Available : https://doi.org/10.1145/1541880.1541882

[104] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "The security of ai systems : A practical primer and framework for intrusion detection," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)*. ACM, 2023. [Online]. Available : https://doi.org/10.1145/3576915.3623220

[105] M. Hosseinzadeh and B. Sinopoli, "Security and privacy in the internet of things : Challenges and opportunities," *Informatica*, vol. 47, no. 1, 2023. [Online]. Available : https://doi.org/10.31449/inf.v47i1.4171

[106] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019. [Online]. Available : https://doi.org/10.1109/JIOT.2019.2928369

[107] A. R. Bhat and S. Venkataramanan, "Leveraging artificial intelligence for enhanced threat detection, response, and anomaly identification in resource-constrained iot networks," *Journal of Cybersecurity and Privacy*, vol. 4, no. 2, pp. 345–362, 2024. [Online]. Available : https://doi.org/10.3390/jcp4020018

[108] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016. [Online]. Available : https://doi.org/10.1016/j.jnca.2015.11.016

[109] M. Khraisat, Y. Gouglidis, A. Ahmed, and J. Slay, "A critical review of intrusion detection systems in the internet of things : Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 18, 2021. [Online]. Available : https://doi.org/10.1186/s42400-021-00077-7

[110] J. Gama, I. Žliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014. [Online]. Available : https://doi.org/10.1145/2523813

[111] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951. [Online]. Available : https://doi.org/10.1214/aoms/1177729694

[112] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[113] D. P. Kingma and J. Ba, "Adam : A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available : https://arxiv.org/abs/1412.6980

[114] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available : https://doi.org/10.1038/nature14539

[115] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, M. Sun, and J. Tang, "Graph neural networks : A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020. [Online]. Available : https://doi.org/10.1016/j.aiopen.2021.01.001

[116] M. M. A. Muslam, "Enhancing security in vehicle-to-vehicle communication : A comprehensive review of protocols and techniques," *Vehicles*, vol. 6, no. 1, pp. 450–467, 2024.

[117] S. F. Lokman, A. T. Othman, and M. H. Abu-Bakar, "Intrusion detection system for automotive controller area network (can) bus system : a review," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–17, 2019.

[118] Canis Labs, "Canis labs can security," https://canislabs.com/cansecurity/, 2024, accessed 28 April 2024.

[119] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems : techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[120] P. Sharma, M. Patel, and A. Prasad, "A systematic literature review on internet of vehicles security," arXiv preprint arXiv :2212.08754, 2022.

[121] L. E. Alatabani, E. S. Ali, and R. A. Saeed, "Deep learning approaches for iov applications and services," in *Intelligent Technologies for Internet of Vehicles*. Cham : Springer International Publishing, 2021, pp. 253–291.

[122] S. Yaqoob, A. Hussain, F. Subhan, G. Pappalardo, and M. Awais, "Deep learning based anomaly detection for fog-assisted iovs network," *IEEE Access*, vol. 11, pp. 19 024–19 038, 2023.

[123] M. H. Shahriar, Y. Xiao, P. Moriano, W. Lou, and Y. T. Hou, "Canshield : Deep learning-based intrusion detection framework for controller area networks at the signal-level," *IEEE Internet of Things Journal*, 2023.

[124] W. Wang, F. Harrou, B. Bouyeddou, S. M. Senouci, and Y. Sun, "Cyber-attacks detection in industrial systems using artificial intelligence-driven methods," *International Journal of Critical Infrastructure Protection*, vol. 38, p. 100542, 2022.

[125] H. Taslimasa, S. Dadkhah, E. C. P. Neto, P. Xiong, S. Ray, and A. A. Ghorbani, "Security issues in internet of vehicles (iov) : A comprehensive survey," *Internet of Things*, p. 100809, 2023.

[126] S. Rizvi, M. Scanlon, J. McGibney, and J. Sheppard, "Application of artificial intelligence to network forensics : Survey, challenges and future directions," *IEEE Access*, vol. 10, pp. 110 362–110 384, 2022.

[127] M. S. Korium, M. Saber, A. Beattie, A. Narayanan, S. Sahoo, and P. H. Nardelli, "Intrusion detection system for cyberattacks in the internet of vehicles environment," *Ad Hoc Networks*, vol. 153, p. 103330, 2024.

[128] J. Prakash, L. Murali, N. Manikandan, N. Nagaprasad, and K. Ramaswamy, "A vehicular network based intelligent transport system for smart cities using machine learning algorithms," *Scientific Reports*, vol. 14, no. 1, p. 468, 2024.

[129] B. Lampe and W. Meng, "Intrusion detection in the automotive domain : A comprehensive review," *IEEE Communications Surveys & Tutorials*, 2023.

[130] L. M. Ang, K. P. Seng, G. K. Ijemaru, and A. M. Zungeru, "Deployment of iov for smart cities : Applications, architecture, and challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2018.

[131] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of CAN bus security challenges," *Sensors*, vol. 20, no. 8, p. 2364, 2020.

[132] L. Pan, X. Zheng, H. X. Chen, T. Luan, H. Bootwala, and L. Batten, "Cyber security attacks to modern vehicular systems," *Journal of Information Security and Applications*, vol. 36, pp. 90–100, 2017.

[133] Canadian Institute for Cybersecurity, "Cic iov dataset 2024," https://www.unb.ca/cic/datasets/iov-dataset-2024.html, 2024, accessed 29 April 2024.

[134] Tutorialspoint, "One hot encoding and label encoding explained," https://www.tutorialspoint.com/one-hot-encoding-and-label-encoding-explained, 2024, accessed 29 April 2024.

[135] E. C. P. Neto, H. Taslimasa, S. Dadkhah, S. Iqbal, P. Xiong, T. Rahman, and A. A. Ghorbani, "CICIoV2024 : Advancing realistic ids approaches against dos and spoofing attack in iov can bus," *Journal of Internet of Things*, 2024, submitted.

[136] H. Jaidka, N. Sharma, and R. Singh, "Evolution of iot to iiot : Applications & challenges," in *Proceedings of the international conference on innovative computing & communications (ICICC)*, 2020.

[137] L. Farhan, R. Kharel, O. Kaiwartya, M. Quiroz-Castellanos, A. Alissa, and M. Abdulsalam, "A concise review on internet of things (iot)-problems, challenges and opportunities," in *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*. IEEE, 2018, pp. 1–6.

[138] T. Chalishazar, "Peerbits.exploring the applications of iot in different industries," 2023, accessed on June 24, 2023. [Online]. Available : https://www.peerbits.com/blog/iot-applications-in-different-industries.html

[139] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things : Architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[140] V. R. Kebande, "Industrial internet of things (iiot) forensics : The forgotten concept in the race towards industry 4.0," *Forensic Science International : Reports*, vol. 5, p. 100257, 2022.

[141] P. Rosso, "Restartsme, industrial iot and its application in critical environments," 2023, accessed on February 28, 2023. [Online]. Available : https://restartsmes.eu/2023/02/28/industrial-iot-and-its-application-in-critical-environments

[142] IBM, "Ibm, accelerate the journey to ai," 2023, accessed on June 24, 2023. [Online]. Available : https://www.ibm.com/downloads/cas/A4N9NVYL

[143] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," *Computers in Industry*, vol. 100, pp. 212–223, 2018.

[144] N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Cyber–physical systems forensics : Today and tomorrow," *Journal of Sensor and Actuator Networks*, vol. 9, no. 3, p. 37, 2020.

[145] M. Javaid, A. Haleem, R. P. Singh, R. Suman, and E. S. Gonzalez, "Understanding the adoption of industry 4.0 technologies in improving environmental sustainability," *Sustainable Operations and Computers*, vol. 3, pp. 203–217, 2022.

[146] A. A. Mirani, G. Velasco-Hernandez, A. Awasthi, and J. Walsh, "Key challenges and emerging technologies in industrial iot architectures : A review," *Sensors*, vol. 22, no. 15, p. 5836, 2022.

[147] M. Younan, E. H. Houssein, M. Elhoseny, and A. A. Ali, "Challenges and recommended technologies for the industrial internet of things : A comprehensive review," *Measurement*, vol. 151, p. 107198, 2020.

[148] T. Gebremichael, L. P. Ledwaba, M. H. Eldefrawy, G. P. Hancke, N. Pereira, M. Gidlund, and J. Akerberg, "Security and privacy in the industrial internet of things : Current standards and future challenges," *IEEE Access*, vol. 8, pp. 152 351–152 366, 2020.

[149] T. Wilmington, "Qomodo, securing the industrial internet of things : How embedded iot security software can help," 2023, accessed on June 24, 2023. [Online]. Available : https://www.qomodo.io/blog/securing-the-industrial-internet-of-things-how-embedded-iot-security-software-can-help

[150] V. Priya, I. S. Thaseen, T. R. Gadekallu, M. K. Aboudaif, and E. A. Nasr, "Robust attack detection approach for iiot using ensemble classifier," *arXiv preprint arXiv :2102.01515*, 2021.

[151] G. S. Madhuri and M. U. Rani, "Anomaly detection techniques," in *2018 IADS International Conference on Computing, Communications & Data Engineering (CCODE)*, 2018.

[152] M. Munir, M. A. Chattha, A. Dengel, and S. Ahmed, "A comparative analysis of traditional and deep learning-based anomaly detection methods for streaming data," in *2019 18th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2019, pp. 561–566.

[153] S. Cook, "Comparitech, malware statistics in 2023 : Frequency, impact, cost & more," 2023, accessed on June 21, 2023. [Online]. Available : https://www.comparitech.com/antivirus/malware-statistics-facts/

[154] SonicWall, "Sonicwall,sonicwall cyber threat report," 2023, accessed on June 24, 2023. [Online]. Available : https://www.sonicwall.com/fr-fr/sonicwall-cyber-threat-report-thank-you/?TR_name=asdas

[155] K. Indah, "Increditools., 33 key malware statistics in 2023," 2023, accessed on June 25, 2023. [Online]. Available : https://increditools.com/malware-statistics/

[156] E. Chachak, "Cyberdb, the future of ai in cybersecurity : Are we ready ?" 2023, accessed on June 25, 2023. [Online]. Available : https://www.cyberdb.co/the-future-of-ai-in-cybersecurity-are-we-ready/

[157] J. Du, K. Yang, Y. Hu, and L. Jiang, "Nids-cnnlstm : Network intrusion detection classification model based on deep learning," *IEEE Access*, vol. 11, pp. 24 808–24 821, 2023.

[158] I. A. Kandhro, S. M. Alanazi, F. Ali, A. Kehar, K. Fatima, M. Uddin, and S. Karuppayah, "Detection of real-time malicious intrusions and attacks in iot empowered cybersecurity infrastructures," *IEEE Access*, vol. 11, pp. 9136–9148, 2023.

[159] M. Alrowaily, F. Alenezi, and Z. Lu, "Effectiveness of machine learning based intrusion detection systems," in *Security, Privacy, and Anonymity in Computation, Communication, and Storage : 12th International Conference, SpaCCS*

2019, Atlanta, GA, USA, July 14–17, 2019, Proceedings 12. Springer, 2019, pp. 277–288.

[160] N. T. Cam and N. G. Trung, "An intelligent approach to improving the performance of threat detection in iot," *IEEE Access*, 2023.

[161] A. A. Elsaeidy, A. Jamalipour, and K. S. Munasinghe, "A hybrid deep learning approach for replay and ddos attack detection in a smart city," *IEEE Access*, vol. 9, pp. 154 864–154 875, 2021.

[162] A. Al-Abassi, H. Karimipour, A. Dehghantanha, and R. M. Parizi, "An ensemble deep learning-based cyber-attack detection in industrial control system," *IEEE Access*, vol. 8, pp. 83 965–83 973, 2020.

[163] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[164] I. Ullah and Q. H. Mahmoud, "Design and development of a deep learning-based model for anomaly detection in iot networks," *IEEE Access*, vol. 9, pp. 103 906–103 926, 2021.

[165] D. Gümüşbaş, T. Yıldırım, A. Genovese, and F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," *IEEE Systems Journal*, vol. 15, no. 2, pp. 1717–1731, 2020.

[166] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection : Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.

[167] G. Singh and N. Khare, "A survey of intrusion detection from the perspective of intrusion datasets and machine learning techniques," *International Journal of Computers and Applications*, vol. 44, no. 7, pp. 659–669, 2022.

[168] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection : A survey and an objective comparison," *Journal of Network and Computer Applications*, vol. 169, p. 102767, 2020.

[169] E. Ashraf, N. F. Areed, H. Salem, E. H. Abdelhady, and A. Farouk, "Iot based intrusion detection systems from the perspective of machine and deep learning : A survey and comparative study." *Delta University Scientific Journal*, vol. 5, no. 2, 2022.

[170] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Computer Science*, vol. 167, pp. 636–645, 2020.

[171] N. Mishra and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions : A systematic review," *IEEE Access*, vol. 9, pp. 59 353–59 377, 2021.

[172] T. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "Graph neural networks for intrusion detection : A survey," *IEEE Access*, 2023.

[173] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "Ae-mlp : A hybrid deep learning approach for ddos detection and classification," *IEEE Access*, vol. 9, pp. 146 810–146 821, 2021.

[174] S. Liu, G. Lin, Q.-L. Han, S. Wen, J. Zhang, and Y. Xiang, "Deepbalance : Deep-learning and fuzzy oversampling for vulnerability detection," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 7, pp. 1329–1343, 2019.

[175] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," *Ieee Access*, vol. 9, pp. 7550–7563, 2020.

[176] A. Ito, K. Saito, R. Ueno, and N. Homma, "Imbalanced data problems in deep learning-based side-channel attacks : Analysis and solution," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3790–3802, 2021.

[177] S. K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, and M. Guizani, "The duo of artificial intelligence and big data for industry 4.0 : Applications, techniques, challenges, and future research directions," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12 861–12 885, 2021.

[178] D. K. Sharma, T. Dhankhar, G. Agrawal, S. K. Singh, D. Gupta, J. Nebhen, and I. Razzak, "Anomaly detection framework to prevent ddos attack in fog empowered iot networks," *Ad Hoc Networks*, vol. 121, p. 102603, 2021.

[179] L. Zhou and H. Guo, "Anomaly detection methods for iiot networks," in *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. IEEE, 2018, pp. 214–219.

[180] P. Goyal, S. Pandey, and K. Jain, "Deep learning for natural language processing," *New York : Apress*, 2018.

[181] M. Moocarme, M. Abdolahnejad, and R. Bhagwat, *The Deep Learning with Keras Workshop : Learn how to define and train neural network models with just a few lines of code*. Packt Publishing Ltd, 2020.

[182] R. A. Chinnathambi, S. J. Plathottam, T. Hossen, A. S. Nair, and P. Ranganathan, "Deep neural networks (dnn) for day-ahead electricity price markets," in *2018 IEEE electrical power and energy conference (EPEC)*. IEEE, 2018, pp. 1–6.

[183] M. Abdel-Basset, N. Moustafa, H. Hawash, and W. Ding, *Deep Learning Techniques for IoT Security and Privacy*. Springer, 2022, vol. 997.

[184] F. Rosenblatt, "The perceptron : a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[185] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[186] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology*, vol. 195, no. 1, pp. 215–243, 1968.

[187] K. Fukushima, "A neural network model for selective attention in visual pattern recognition," *Biological Cybernetics*, vol. 55, no. 1, pp. 5–15, 1986.

[188] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks : an overview and application in radiology," *Insights into imaging*, vol. 9, pp. 611–629, 2018.

[189] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-iiotset : A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.

[190] A. Khacha, R. Saadouni, Y. Harbi, and Z. Aliouat, "Hybrid deep learning-based intrusion detection system for industrial internet of things," in *2022 5th International Symposium on Informatics and its Applications (ISIA)*. IEEE, 2022, pp. 1–6.

[191] I. Tareq, B. M. Elbagoury, S. El-Regaily, and E.-S. M. El-Horbaty, "Analysis of ton-iot, unw-nb15, and edge-iiot datasets using dl in cybersecurity for iot," *Applied Sciences*, vol. 12, no. 19, p. 9572, 2022.

[192] M. Chertoff, "A public policy perspective of the dark web," *Journal of Cyber Policy*, vol. 2, no. 1, pp. 26–38, 2017.

[193] S. Nazah, S. Huda, J. Abawajy, and M. M. Hassan, "Evolution of dark web threat analysis and detection : A systematic approach," *Ieee Access*, vol. 8, pp. 171 796–171 819, 2020.

[194] M. S. P. dos Santos Horta, "Tor k-anonymity against deep learning watermarking attacks," Master's thesis, Universidade NOVA de Lisboa (Portugal), 2022.

[195] A. Alharbi, M. Faizan, W. Alosaimi, H. Alyami, A. Agrawal, R. Kumar, and R. A. Khan, "Exploring the topological properties of the tor dark web," *IEEE Access*, vol. 9, pp. 21 746–21 758, 2021.

[196] A. Averin, A. Samartsev, and N. Sachenko, "Review of methods for ensuring anonymity and de-anonymization in blockchain," in *2020 International Conference Quality Management, Transport and Information Security, Information Technologies (IT&QM&IS)*. IEEE, 2020, pp. 82–87.

[197] A. A. AlQahtani and E.-S. M. El-Alfy, "Anonymous connections based on onion routing : A review and a visualization tool," *Procedia Computer Science*, vol. 52, pp. 121–128, 2015.

[198] P. J. Ezra, S. Misra, A. Agrawal, J. Oluranti, R. Maskeliunas, and R. Damasevicius, "Secured communication using virtual private network (vpn)," *Cyber security and digital forensics : proceedings of ICCSDF 2021*, pp. 309–319, 2021.

[199] P. Kisanga, I. Woungang, I. Traore, and G. H. Carvalho, "Network anomaly detection using a graph neural network," in *2023 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2023, pp. 61–65.

[200] S. R. Purnama, J. E. Istiyanto, M. A. Amrizal, V. Handika, S. Rochman, and A. Dharmawan, "Inductive graph neural network with causal sampling for iot network intrusion detection system," in *2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*. IEEE, 2022, pp. 241–246.

[201] H.-C. Lin, P. Wang, W.-H. Lin, Y.-H. Lin, and J.-H. Chen, "Graph neural network for malware detection and classification on renewable energy management platform," in *2023 IEEE 5th Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS)*. IEEE, 2023, pp. 164–166.

[202] Z. Zhai, P. Li, and S. Feng, "State of the art on adversarial attacks and defenses in graphs," *Neural Computing and Applications*, vol. 35, no. 26, pp. 18 851–18 872, 2023.

[203] A. Habibi Lashkari, G. Kaur, and A. Rahali, "Didarknet : A contemporary approach to detect and characterize the darknet traffic using deep image learning," in *Proceedings of the 2020 10th international conference on communication and network security*, 2020, pp. 1–13.

[204] D. Sarkar, P. Vinod, and S. Y. Yerima, "Detection of tor traffic using deep learning," in *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2020, pp. 1–8.

[205] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *International conference on information systems security and privacy*, vol. 2. SciTePress, 2017, pp. 253–262.

[206] L. A. Iliadis and T. Kaifas, "Darknet traffic classification using machine learning techniques," in *2021 10th international conference on modern circuits and systems technologies (MOCAST)*. IEEE, 2021, pp. 1–4.

[207] K. Demertzis, K. Tsiknas, D. Takezis, C. Skianis, and L. Iliadis, "Darknet traffic big-data analysis and network management for real-time automating of the malicious intent detection process by a weight agnostic neural networks framework," *Electronics*, vol. 10, no. 7, p. 781, 2021.

[208] M. B. Sarwar, M. K. Hanif, R. Talib, M. Younas, and M. U. Sarwar, "Darkdetect : Darknet traffic detection and categorization using modified convolution-long short-term memory," *IEEE Access*, vol. 9, pp. 113 705–113 713, 2021.

[209] M. C. Marim, P. V. B. Ramos, A. B. Vieira, A. Galletta, M. Villari, R. M. de Oliveira, and E. F. Silva, "Darknet traffic detection and characterization with models based on decision trees and neural networks," *Intelligent Systems with Applications*, vol. 18, p. 200199, 2023.

[210] M. Alimoradi, M. Zabihimayvan, A. Daliri, R. Sledzik, and R. Sadeghi, "Deep neural classification of darknet traffic," in *Artificial intelligence research and development*. IOS press, 2022, pp. 105–114.

[211] Y. Zhu, J. Tao, H. Wang, L. Yu, Y. Luo, T. Qi, Z. Wang, and Y. Xu, "Dgnn : Accurate darknet application classification adopting attention graph neural network," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 1660–1671, 2023.

[212] K. Saheed and S. Henna, "Heterogeneous graph transformer for advanced persistent threat classification in wireless networks," in *2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2023, pp. 15–20.

[213] G. P. Mika, A. Bouzeghoub, K. Węgrzyn-Wolska, and Y. M. Neggaz, "Hgexplainer : explainable heterogeneous graph neural network," in *2023 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. IEEE, 2023, pp. 221–229.

[214] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn : Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proceedings of the web conference 2020*, 2020, pp. 2331–2341.

[215] M. Labonne, *Hands-on graph neural networks using Python*. Packt Publishing Birmingham, UK, 2023.

[216] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European semantic web conference*. Springer, 2018, pp. 593–607.

[217] P. Yu, C. Fu, Y. Yu, C. Huang, Z. Zhao, and J. Dong, "Multiplex heterogeneous graph convolutional network," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 2377–2387.

[218] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[219] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks ?" *arXiv preprint arXiv :1810.00826*, 2018.

[220] T. Kipf, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv :1609.02907*, 2016.

[221] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv :1710.10903*, 2017.

[222] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[223] G. Li, M. Müller, B. Ghanem, and V. Koltun, "Training graph neural networks with 1000 layers," in *International conference on machine learning*. PMLR, 2021, pp. 6437–6449.

[224] N. Rust-Nguyen, S. Sharma, and M. Stamp, "Darknet traffic classification and adversarial attacks using machine learning," *Computers & Security*, vol. 127, p. 103098, 2023.

[225] H. Mohanty, A. H. Roudsari, and A. H. Lashkari, "Robust stacking ensemble model for darknet traffic classification under adversarial settings," *Computers & Security*, vol. 120, p. 102830, 2022.

[226] L. Chang and P. Branco, "Graph-based solutions with residuals for intrusion detection : The modified e-graphsage and e-resgat algorithms," *arXiv preprint arXiv :2111.13597*, 2021.

[227] J. Lan, X. Liu, B. Li, Y. Li, and T. Geng, "Darknetsec : A novel self-attentive deep learning method for darknet traffic classification and application identification," *Computers & Security*, vol. 116, p. 102663, 2022.

[228] H. Karagöl, O. Erdem, B. Akbas, and T. Soylu, "Darknet traffic classification with machine learning algorithms and smote method," in *2022 7th International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2022, pp. 374–378.