UQAC
Université du Québec
à Chicoutimi

UTILISATION ET EXPLICABILITÉ DES GRANDS MODÈLES DE LANGAGE (LLM) POUR LA CLASSIFICATION DES ATTAQUES SUR LES DISPOSITIFS MÉDICAUX CONNECTÉS (IOMT)

PAR MAMADOU NIANG

MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI EN VUE DE L'OBTENTION DU GRADE DE MAÎTRE ÈS SCIENCES (M. SC.) EN INFORMATIQUE

QUÉBEC, CANADA

© MAMADOU NIANG, 2025

RÉSUMÉ

L'adoption rapide des dispositifs de l'internet des objets médicaux (IoMT) a amélioré les soins aux patients, mais a également introduit des vulnérabilités et a exposé les systèmes à des menaces cybernétiques, notamment les attaques par déni de service (DoS). Ce mémoire explore le potentiel des grands modèles de langage (LLM) pour détecter et classer les attaques réseau IoMT, en mettant l'accent sur les techniques d'explicabilité pour surmonter la nature boîte noire de ces modèles. En utilisant le jeu de données CICIoMT2024, nous avons évalué des modèles tels que BERT, RoBERTa et DistilBERT affinés (fine-tuned) et spécialisés dans la cybersécurité IoMT.

DistilBERT a obtenu la meilleure performance en termes de temps et d'efficacité avec un score de 96,94 %, surpassant de manière significative BERT et RoBERTa, notamment pour la détection des attaques DoS. Des méthodes d'explicabilité telles que LIME, SHAP et la visualisation de l'attention ont été utilisées pour interpréter les principales caractéristiques influençant les prédictions du modèle. Nos résultats montrent que DistilBERT est à la fois adaptable aux données de réseaux, notamment celles issues des fichiers de capture de paquets (PCAP) et performant. De plus, il est interprétable, ce qui ouvre la voie à des solutions de cybersécurité IoMT robustes et applicables dans des scénarios réels.

TABLE DES MATIÈRES

RÉSUMÉ ii

LISTE DES TABLEAUX v

LISTE DES FIGURES vi

LISTE DES ABRÉVIATIONS viii

DÉDICACE ix

REMERCIEMENTS x

AVANT-PROPOS xi

CHAPITRE I – INTRODUCTION 1

1.1 CONTEXTE DE LA RECHERCHE 1

1.2 PROBLÉMATIQUE DE LA RECHERCHE 5

1.3 CONTRIBUTION DE LA RECHERCHE 6

1.4 APPROCHE DE LA RECHERCHE 7

1.5 STRUCTURE DU DOCUMENT 8

CHAPITRE II – CONCEPTS FONDAMENTAUX 10

2.1 LES TRANSFORMERS 10

2.2 BERT 13

2.3 DISTILBERT 15

2.4 ROBERTA 16

2.5 INTELLIGENCE ARTIFICIELLE EXPLICABLE (XAI) 16

2.5.1 LIME (LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS) 18

2.5.2 SHAP (SHAPLEY ADDITIVE EXPLANATIONS) 19

2.5.3 EXPLICABILITÉ BASÉ SUR L'ATTENTION 20

CHAPITRE III – TRAVAUX CONNEXES 22

3.1 APPROCHES BASÉES SUR ML ET DL 22

3.2 APPROCHES BASÉES SUR LES TRANSFORMERS 24
3.3 TECHNIQUES XAI BASÉS SUR DES MODÈLES DE TYPE TRANSFORMERS 25
CHAPITRE IV – MÉTHODOLOGIE 27
4.1 PRÉPARATION DES DONNÉES 27
4.1.1 COLLECTE DES DONNÉES ET DESCRIPTION 28
4.1.2 LE NETTOYAGE DES DONNÉES 30
4.1.3 L'EXPLORATION ET TRAITEMENT DES DONNÉES 35
4.2 ENTRAÎNEMENT DES MODÈLES 37
4.2.1 LLM (BERT, DISTILBERT ET ROBERTA) 38
4.2.2 FORÊTS ALÉATOIRES (RF) 39
4.2.3 RÉSEAU DE NEURONES CONVOLUTIFS (CNN) 39
4.3 MÉTRIQUES DE PERFORMANCE 40
4.4 INTERPRÉTATION ET TRANSPARENCE 42
CHAPITRE V – RÉSULTATS 46
5.1 EFFICACITÉ DES LLM POUR DÉTECTER LES ATTAQUES IOMT 46
5.2 IDENTIFICATION DES CARACTÉRISTIQUES IMPORTANTES EN UTILISANT DISTILBERT POUR LA DÉTECTION DES ATTAQUES DOS . . . 52
CHAPITRE VI – DISCUSSION ET LIMITES 63
6.1 UTILISATION DES MODÈLES LLM POUR DÉTECTER LES ATTAQUES IOMT À PARTIR DE DONNÉES DU TRAFIC RÉSEAUX PCAP 63
6.2 LIMITATIONS ET MENACES À LA VALIDITÉ 68
6.2.1 SÉLECTION DES MODÈLES 69
6.2.2 CHOIX DE PRÉTRAITEMENT DES DONNÉES 69
6.2.3 JEU DE DONNÉES ET CONDITIONS EXPÉRIMENTALES 70
6.2.4 MENACES À LA SCALABILITÉ ET DE GÉNÉRALISATION 70
CONCLUSION 72
BIBLIOGRAPHIE 74

LISTE DES TABLEAUX

TABLEAU 4.1 : DESCRIPTION DES CARACTÉRISTIQUES DE CICIOMT2024 . . 31

TABLEAU 4.2 : DISTRIBUTION DU JEU DE DONNÉES SUIVANT LES CINQ CLASSES. 35

TABLEAU 5.1 : ÉVALUATION DES PERFORMANCES DES MODÈLES. 50

TABLEAU 5.2 : MOYENNES DES PERFORMANCES DES MODÈLES. 50

TABLEAU 5.3 : LES 10 PRINCIPALES CARACTÉRISTIQUES IDENTIFIÉES PAR DIFFÉRENTES TECHNIQUES D'EXPLICABILITÉ. 61

LISTE DES FIGURES

FIGURE 1.1 – NOMBRE DE CONNEXIONS À L'INTERNET DES OBJETS (IOT) DANS LE MONDE DE 2022 À 2033. 2

FIGURE 1.2 – L'INTERNET DES OBJETS MÉDICAUX. 3

FIGURE 2.1 – L'ARCHITECTURE DU TRANSFORMER 11

FIGURE 2.2 – L'ARCHITECTURE DE BERT. 14

FIGURE 2.3 – RAISON D'EXPLICABILITÉ DE L'INTELLIGENCE ARTIFICIELLE (XAI). 17

FIGURE 2.4 – REPRÉSENTATION DE L'ATTENTION PAR GRAPHE BIPARTITE ET CARTE THERMIQUE POUR LA MATRICE D'ATTENTION. . . . 21

FIGURE 4.1 – FLUX DE TRAVAIL GÉNÉRAL DE L'ÉTUDE. 28

FIGURE 4.2 – TOPOLOGIE DE L'ENSEMBLE DE DONNÉES CICIOMT2024. . . . 29

FIGURE 4.3 – MATRICE DE CORRÉLATION DE L'ENSEMBLE DES DONNÉES. . 33

FIGURE 4.4 – DISTRIBUTION DES DONNÉES D'ENTRAINEMENT AVANT ET APRÈS PRÉTRAITEMENT. 34

FIGURE 5.1 – COURBES DE PERTES EN FONCTION DES ITÉRATIONS D'ENTRAINEMENT 47

FIGURE 5.2 – DURÉE D'ENTRAINEMENT DES MODÈLES EN SECONDES. . . . 48

FIGURE 5.3 – L'UTILISATION DE GPU PAR MODÈLE 48

FIGURE 5.4 – MATRICE DE CONFUSION DE DISTILBERT 49

FIGURE 5.5 – MATRICE DE CONFUSION DE BERT 49

FIGURE 5.6 – MATRICE DE CONFUSION DE ROBERTA 49

FIGURE 5.7 – MATRICE DE CONFUSION DE CNN. 49

FIGURE 5.8 – MATRICE DE CONFUSION DE RF 49

FIGURE 5.9 – UTILISATION DE LIME POUR LA PRÉDICTION DES ATTAQUES DOS 54

FIGURE 5.10 – GRAPHE DE RÉSUMÉ GLOBAL DE SHAP 56

FIGURE 5.11 – GRAPHE DE DÉCISION POUR LA PRÉDICTION DE DOS 58

FIGURE 5.12 – CARTE THERMIQUE ILLUSTRANT L'ATTENTION ENTRE LES CARACTÉRISTIQUES (A). 59

FIGURE 5.13 – CARTE THERMIQUE ILLUSTRANT L'ATTENTION ENTRE LES CARACTÉRISTIQUES (B). 60

LISTE DES ABRÉVIATIONS

IA	Intelligence Artificielle
DT	Decision Tree
NB	Naive Bayes
ML	Machine Learning
DL	Deep Learning
RF	Random Forest
NLP	Natural Language Processing
IoT	Internet of Things
DoS	Denial of Service
XAI	Explainable Artificial Intelligence
CSV	Comma-Separated Values
PCC	Pearson's correlation coefficient
LLM	Large Language Model
CNN	Convolutional Neural Network
SVM	Support Vector Machine
DNN	Deep Neural Network
KNN	K-Nearest Neighbors
MQTT	Message Queuing Telemetry Transport
DDoS	Distributed Denial of Service
IoMT	Internet of Medical Things
LIME	Local Interpretable Model-agnostic Explanations
LSTM	Long Short-Term Memory
SHAP	SHapley Additive exPlanations
PCAP	Packet Capture

DÉDICACE

À mon père et à ma mère.

Je vous dédie ce travail en espérant que vous serez toujours fiers de moi.

Je dédie également ce travail à tous les membres de la famille, pour leurs encouragements, soutiens, affectation et confiance pour continuer mes études.

Aux Professeurs Haifa Nakouri et Fehmi Jaafar qui m'ont donné confiance et espoir et qui m'ont soutenu durant toutes mes recherches.

À tous ceux qui de près ou de loin ont contribué de quelques manières à la réalisation de ce document.

Qu'ALLAH leur accorde la santé, prospérité et leur gratifie de sa miséricorde.

REMERCIEMENTS

Je remercie, en premier lieu, ALLAH pour le courage, la patience et la santé qu'il m'a donné pour poursuivre mes études.

Je tiens à remercier sincèrement ma codirectrice de recherche, Madame Haifa Nakouri, dont les conseils avisés et le soutien ont grandement contribué à enrichir ce travail. Sa patience et son enthousiasme à m'apprendre de nouvelles choses, en particulier dans le domaine de l'intelligence artificielle, ont été inestimables. Grâce à elle, j'ai pu acquérir de nombreuses bonnes pratiques et des astuces précieuses qui m'ont permis d'améliorer mes compétences rapidement.

Je souhaite également exprimer ma sincère reconnaissance à mon directeur de recherche, professeur Fehmi Jaafar, pour son soutien constant et précieux. Son expertise, ses conseils avisés et sa disponibilité ont été des atouts majeurs dans la réalisation de ce projet. Je lui suis particulièrement reconnaissant pour les bourses d'études qu'il m'a attribuées, lesquelles ont grandement facilité le bon déroulement de mes recherches. Sa capacité à m'inspirer et à me challenger intellectuellement m'a permis de repousser mes limites.

Je souhaite également exprimer ma reconnaissance envers tous les professeurs que j'ai eus la chance de croiser tout au long de mon parcours académique. Leurs enseignements, riches et éclairants, ont joué un rôle clé dans la construction de mes connaissances et compétences. Plus particulièrement, je tiens à remercier Monsieur Alain Ploix et Monsieur Pascal Fortin, dont les enseignements et orientations ont grandement contribué à mon développement académique.

Mes remerciements s'adressent aussi à l'ensemble du personnel du département de mathématiques et informatique (DIM) de l'UQAC.

Enfin, merci à ma famille, mes amis et mes proches pour m'avoir toujours aidé à trouver un équilibre entre les études et vie personnelle.

AVANT-PROPOS

Ce mémoire de maîtrise marque l'aboutissement d'un parcours académique à la fois exigeant et enrichissant, au cours duquel j'ai eu la chance d'explorer des domaines passionnants, notamment la cybersécurité et l'intelligence artificielle. Grâce à des cours magistraux et à des recherches approfondies, j'ai pu acquérir de solides compétences dans des secteurs tels que le développement d'applications, la cybersécurité et, plus particulièrement, l'intelligence artificielle. J'ai également eu l'opportunité d'étudier l'impact de l'IA, et plus spécifiquement des grands modèles de langage (LLM).

L'objectif de ce travail est de démontrer comment les modèles d'intelligence artificielle, associés à des techniques d'explicabilité, peuvent non seulement améliorer la détection des menaces, mais également offrir une transparence cruciale pour les experts en cybersécurité, dans un environnement où la compréhension des décisions des modèles est essentielle.

Mon parcours de recherche a été jalonné de défis, de stress et d'obstacles, mais aussi de découvertes marquantes et de réussites significatives. Chaque page de ce mémoire témoigne des nombreuses heures consacrées à la recherche, à la réflexion et à l'analyse, dans un effort constant pour comprendre et maîtriser le sujet.

Je prends l'entière responsabilité des propos tenus dans ce document. Cependant, conscient que toute œuvre humaine peut être améliorée, je vous serai reconnaissant de toute remarque, critique ou suggestion visant à enrichir cette étude.

CHAPITRE I
INTRODUCTION

L'internet des objets (IoT) est un paradigme qui décrit la connexion d'objets et de dispositifs du quotidien à Internet, permettant ainsi la collecte et l'échange de données sans intervention humaine directe (Khanna & Kaur, 2020; Khan, 2019). Selon un récent rapport de Statista, le nombre d'appareils connectés (IoT) dans le monde augmente de plus en plus et devrait atteindre 39,6 milliards d'ici à 2033 (Vailshery, 2024). La figure 1.1 montre le nombre d'appareils connectés dans le monde de 2022 à 2024 avec des prévisions jusqu'à 2033 (Vailshery, 2024). Divers domaines, tels que la santé, le transport, l'agriculture et la fabrication, utilisent des appareils connectés pour faciliter leurs activités quotidiennes (Khanna & Kaur, 2020).

1.1 CONTEXTE DE LA RECHERCHE

Selon un autre rapport de Statista, le secteur de la santé est le troisième secteur le plus ciblé par les cyberattaques mondiales en 2022 et 2023 (Petrosyan, 2024), mettant en évidence l'urgence de sécuriser les dispositifs IoMT. L'Internet des objets médicaux (IoMT), a révolutionné les soins de santé grâce à des appareils et des capteurs connectés, améliorant ainsi les soins aux patients et la rentabilité (Vishnu *et al.*, 2020; Gatouillat *et al.*, 2018). Ce dernier, l'IoMT, prend en charge la surveillance à distance des hôpitaux intelligents, intégrant des dispositifs portables, des applications et la télémédecine pour la remise en forme, l'éducation, la gestion des maladies et autres (Dimitrov, 2016). En outre, il facilite la surveillance de la santé en temps réel et l'administration automatisée des traitements (Dimitrov, 2016). La Figure 1.2 montre un exemple de topologie de l'Internet des Objets Médicaux.

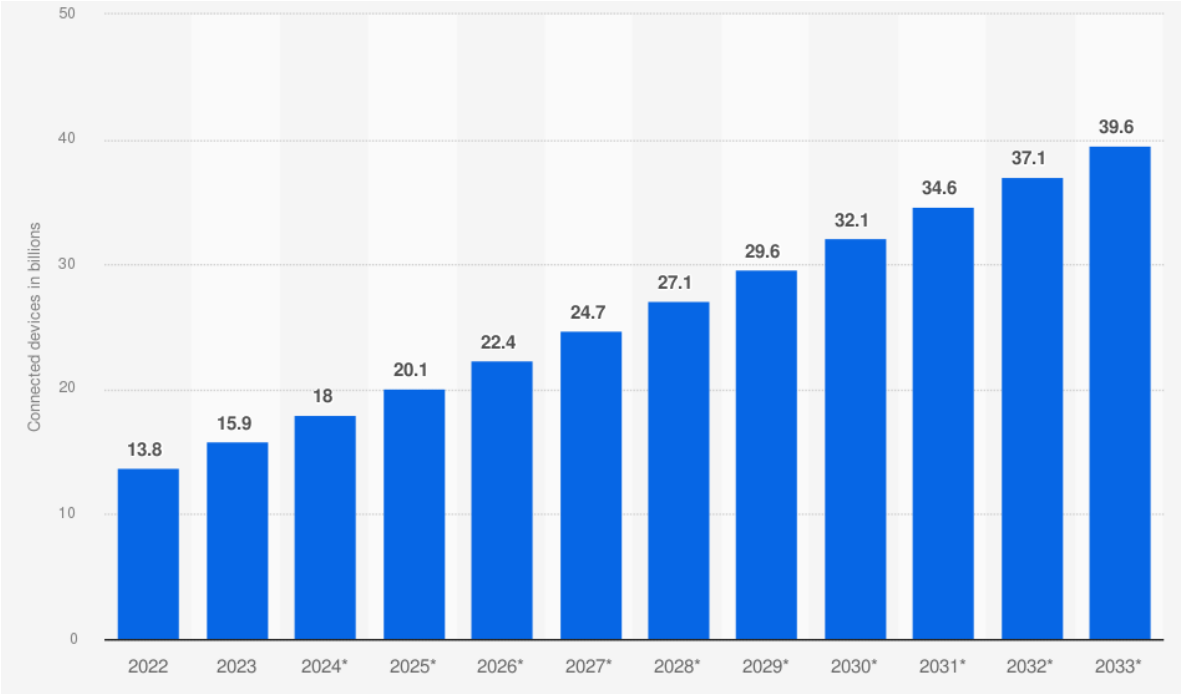

FIGURE 1.1 : Nombre de connexions à l'internet des objets (IoT) dans le monde de 2022 à 2033. (Vailshery, 2024)

Cependant, le nombre croissant de dispositifs IoT a entraîné une augmentation des cybermenaces, ce qui pose de véritables défis pour la sécurité des différents systèmes et réseaux (Ferrag *et al.*, 2024; Moustafa *et al.*, 2023), notamment dans le domaine de la santé. Bien que l'IoMT soit bénéfique dans ledit domaine (santé), elle est confrontée à des défis en termes de fiabilité, de sûreté et de sécurité (Kagita *et al.*, 2022; Gatouillat *et al.*, 2018).

Plusieurs vecteurs d'attaque ciblent les dispositifs de l'internet des objets médicaux (IoMT), mettant en évidence les menaces croissantes qui pèsent sur ces technologies, comme l'ont souligné Allouzi & Khan (2021). Les vecteurs d'attaque font référence aux différentes méthodes ou chemins par lesquels un attaquant peut accéder à un système ou compromettre ses fonctionnalités. Dans le cas de l'IoMT, ces vecteurs peuvent exploiter des failles dans les dispositifs médicaux connectés, mettant ainsi en péril la sécurité des données sensibles.

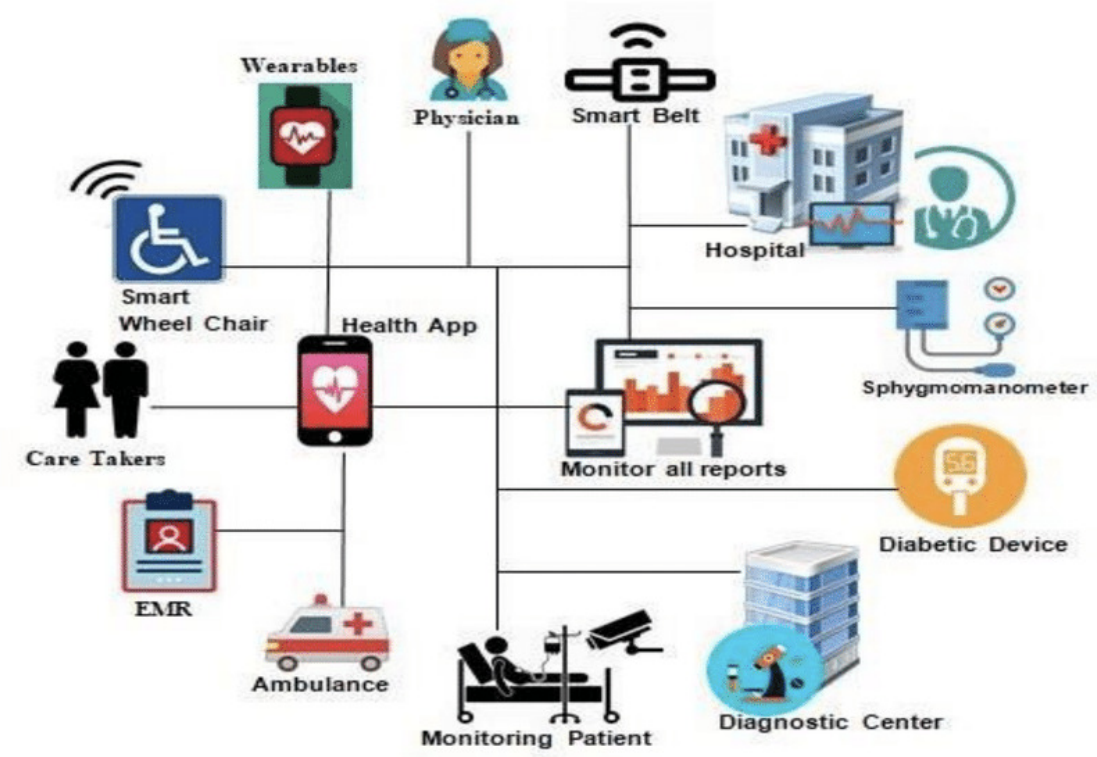

FIGURE 1.2 : l'Internet des Objets Médicaux

(Ayadi, 2019)

Étant des objets médicaux interconnectés qui collectent, stockent et transmettent des données relatives à la santé des patients. Les dispositifs IoMT sont particulièrement vulnérables à un large éventail d'attaques, telles que le déni de service (DoS), où un attaquant surcharge un système afin de le rendre inopérant, ou l'injection de logiciels malveillants qui peuvent endommager les systèmes ou voler des informations sensibles.

Parmi les autres attaques fréquentes figurent l'écoute clandestine, où des informations sont interceptées sans autorisation, l'usurpation d'identité, où un attaquant se fait passer pour une personne ou un dispositif légitime, ainsi que l'injection de commande, où un attaquant exploite des vulnérabilités dans le logiciel du dispositif pour exécuter des commandes non autorisées. Par exemple, il peut reprogrammer un dispositif médical pour altérer son

fonctionnement normal, ce qui peut avoir des conséquences graves pour le patient. (Hasan *et al.*, 2022; Allouzi & Khan, 2021) ont détaillé ces types de menaces, ainsi que d'autres pouvant compromettre la sécurité de l'IoMT.

Ces attaques peuvent avoir des conséquences dramatiques, compromettant la confidentialité des données des patients et, dans certains cas, mettant directement en danger des vies humaines, comme l'ont souligné Rasool *et al.* (2022). Les données des patients, qui comprennent des informations médicales sensibles, peuvent être modifiées, volées ou utilisées à des fins malveillantes, ce qui peut nuire à la sécurité des soins et à la confiance dans les systèmes de santé connectés.

Cependant, de nombreuses solutions ont été proposées par les chercheurs pour atténuer les attaques identifiées et répondre aux exigences de sécurité spécifiques à l'IoMT. Par exemple, Ghubaish *et al.* (2020) proposent un framework de sécurité intégrant de multiples techniques pour atténuer les différents vecteurs d'attaque ciblant les appareils IoMT. Des techniques basées sur l'intelligence artificielle, notamment l'apprentissage machine (ML) et l'apprentissage profond (DL) ont été proposées pour identifier ces attaques et réduire également ces différents types d'attaques dans l'écosystème IoT (AboulEla *et al.*, 2024; Abdullahi *et al.*, 2022). Dans ce mémoire, nous nous concentrons sur l'utilisation des grandes modèles de langage (LLM) pour la détection et la classification des attaques IoMT. Plus précisément, nous traiterons et adapterons les données de trafic réseau générées par les dispositifs de santé à ces types de modèle afin d'en développer un modèle spécifique capable de détecter et de catégoriser différentes attaques contre ces dispositifs. En effet, les modèles larges tels que BERT (Devlin *et al.*, 2019), GPT-4 (Achiam *et al.*, 2023) et Llama-3 (Touvron *et al.*, 2023) ont montré des performances impressionnantes dans le traitement du langage naturel (NLP), une branche de l'intelligence artificielle (IA) qui se concentre sur l'interaction entre les ordinateurs et les langues humaines. Il vise à permettre aux machines de comprendre, générer et

interpréter le langage humain, sous forme écrite ou orale. Ces modèles (LLM) apprennent de manière auto-supervisée, une approche qui leur permet de tirer parti de grandes quantités de données textuelles non étiquetées pour s'entraîner. Ils exploitent des relations implicites dans ces données pour développer une compréhension contextuelle approfondie. Par exemple, ils peuvent identifier la signification d'un mot en fonction de son contexte dans une phrase.

En outre, l'objectif de ce mémoire est d'abord d'exploiter le potentiel des LLM afin de développer un modèle performant pour la classification des attaques IoMT en utilisant le nouvel ensemble de données sur l'IoMT, proposé par Dadkhah *et al.* (2024) de l'Institut canadien de la cybersécurité. Ensuite, d'analyser et d'extraire les caractéristiques ayant le plus d'impact sur les décisions du modèle, à l'aide de diverses techniques d'explicabilité.

1.2 PROBLÉMATIQUE DE LA RECHERCHE

Bien que les LLM puissent effectuer plusieurs tâches de traitement du langage naturel (NLP), telles que l'analyse des sentiments, la classification des textes, la traduction automatique et d'autres, l'utilisation de ces modèles pour traiter les données liées au trafic réseau représente un véritable défi pour les chercheurs en cybersécurité. Ces types de données sont généralement extraits à l'aide d'outils d'analyse de paquets réseau, tels que Wireshark ou tcpdump, et sont stockés sous forme de fichiers PCAP (Packet Capture). Ces fichiers contiennent des informations détaillées sur le trafic réseau, représentées principalement par des caractéristiques (telles que les adresses IP, les ports, les horodatages) et des valeurs numériques (comme la taille des paquets ou le temps entre les transmissions). Contrairement aux données textuelles classiques, ces données nécessitent un prétraitement spécifique, comme la normalisation ou la transformation en formats compatibles, pour pouvoir être efficacement utilisées et interprétées par les grands modèles de langage (LLM). De plus, il est crucial d'interpréter les prédictions des modèles d'apprentissage automatique, car comprendre les décisions qu'ils

prennent est essentiel pour renforcer la sécurité des dispositifs, en particulier ceux utilisés dans des domaines critiques comme la médecine. Dans ce contexte, les réseaux de neurones modernes, tels que les Transformers (Vaswani *et al.*, 2017), qui incluent des modèles comme BERT et ses variantes, traitent les données en les décomposant en plusieurs jetons ou "tokens" (unités de données plus petites) (Kokalj *et al.*, 2021). Bien que cette approche permette une meilleure représentation contextuelle des données, elle complique l'identification des caractéristiques spécifiques ayant influencé les décisions du modèle, rendant ainsi le processus d'interprétabilité plus complexe.

Par conséquent, cette recherche vise à répondre aux questions suivantes :

Q1 Les LLM pré-entraînés sur des données de fichier PCAP sont-ils efficaces pour détecter et classifier les attaques IoMT ?

Q2 Ces modèles peuvent-ils identifier efficacement les caractéristiques les plus importantes contribuant à leurs décisions, et comment les techniques d'intelligence artificielle explicable (XAI) peuvent-elles être utilisées pour renforcer leur efficacité et leur pertinence dans le domaine de la cybersécurité ?

L'hypothèse de cette recherche est qu'un grand modèle de langage (LLM) affiné et spécialisé sur des données PCAP (Packet Capture) sera capable de surpasser les modèles séquentiels traditionnels. Non seulement il offrira de meilleures performances en termes de détection et de classification des attaques, mais il permettra également de fournir des informations plus approfondies sur l'importance des différentes caractéristiques du trafic réseau ainsi que sur les motifs sous-jacents des attaques.

1.3 CONTRIBUTION DE LA RECHERCHE

Cette recherche a pour objectif d'apporter plusieurs contributions au domaine de la cybersécurité des objets médicaux. Plus précisément :

- Classification efficace des attaques IoMT : L'étude montre que les LLM affinés (fine-tuned) peuvent classer avec précision les attaques IoMT, un aspect crucial pour garantir la sécurité et la fiabilité des systèmes de santé.
- Interprétabilité des prédictions des modèles LLM : Grâce aux techniques d'intelligence artificielle explicable (XAI), cette recherche offre une meilleure compréhension du processus de décision des LLM, renforçant ainsi la confiance dans les systèmes de sécurité basés sur l'IA.
- Contribution en cybersécurité pour la santé : Ce travail contribue à l'élaboration de solutions de cybersécurité plus performantes pour les environnements de santé, améliorant ainsi la sécurité des patients et la qualité des soins.

1.4 APPROCHE DE LA RECHERCHE

Pour répondre à nos questions de recherche, nous commençons par un pré-traitement et la normalisation de l'ensemble des données CICIoMT2024 (Dadkhah *et al.*, 2024), suivi d'une réduction des caractéristiques en nous inspirant de la méthodologie décrite par (Khan & Alkhathami, 2024). Cette étape vise à optimiser le temps d'entraînement et de test. Ensuite, nous appliquons une stratégie de sous-échantillonnage des données d'entraînement afin d'équilibrer les classes et de prévenir les biais liés à la sur-représentation.

Par la suite, nous utilisons des modèles LLM tels que BERT, DistilBERT et RoBERTa pour détecter et classifier les attaques ciblant les dispositifs IoMT. Ces modèles sont évalués et comparés sur le même ensemble de données pré-traité afin d'évaluer leur performance dans la classification de différentes attaques contre l'IoMT. Cette comparaison nous permettra de déterminer le modèle le plus efficace pour assurer la sécurité de ces dispositifs médicaux.

Ensuite, nous analyserons et expliquerons le modèle le plus performant en utilisant des techniques d'explicabilité comme LIME, SHAP et la visualisation de l'attention pour comprendre l'importance des caractéristiques. Enfin, nous comparerons ce modèle aux approches traditionnelles des recherches précédentes afin d'évaluer leur efficacité en termes de performance et de déterminer leurs avantages et limites dans un environnement réel.

1.5 STRUCTURE DU DOCUMENT

Le reste de ce mémoire est organisé en cinq chapitres principaux afin d'analyser et d'expliquer efficacement les grands modèles de langage pour la classification des attaques IoMT.

- **Chapitre 2 : Concepts fondamentaux.** Ce chapitre présente les modèles LLM utilisés (BERT, DistilBERT, et RoBERTa) et leurs diverses applications. Il explore également le rôle des techniques d'explicabilité utilisées dans ce mémoire pour déterminer les caractéristiques les plus importantes qui contribuent aux décisions du modèle.
- **Chapitre 3 : Travaux Connexes.** Dans ce chapitre, nous analyserons les travaux existants sur la classification des attaques IoT et examinerons les approches actuelles d'explication des modèles basées sur les Transformers.
- **Chapitre 4 : Méthodologie.** Ce chapitre décrit la méthodologie adoptée pour cette recherche, en présentant le processus détaillé et structuré, depuis la préparation des données jusqu'à la classification des attaques IoMT, suivi de l'application des techniques d'explicabilité (XAI) pour interpréter les résultats.
- **Chapitre 5 : Résultats.** Ce chapitre détaille les résultats obtenus, en offrant une analyse critique de la performance et de l'explication de notre modèle.
- **Chapitre 6 : Discussion et Limites.** Ce chapitre discute les résultats, ses limitations ainsi que certaines améliorations possibles.

- **Conclusion et Recherche Future.** Ce chapitre résume les principales contributions de ce travail. Il conclut l'étude par une brève discussion sur les orientations et les applications futures potentielles afin d'améliorer la sécurité des appareils médicaux.

CHAPITRE II

CONCEPTS FONDAMENTAUX

Dans ce chapitre, nous explorons le concept du Transformer et son importance dans le traitement du langage naturel (NLP). Le chapitre comprend également une présentation des modèles utilisés dans cette étude et explore diverses techniques d'explicabilité permettant d'améliorer la transparence et la clarté de ces modèles complexes.

2.1 LES TRANSFORMERS

Le Transformer, introduit par Vaswani *et al.* (2017), a montré des capacités impressionnantes dans les tâches de traitement du langage naturel (NLP) et d'apprentissage automatique, surpassant les modèles séquentiels traditionnels tels que les réseaux de neurones récurrents (RNN) et les réseaux à mémoire long terme (LSTM). Son architecture est devenue une architecture d'apprentissage profond obtenant des résultats satisfaisants dans divers domaines (Andrecut, 2023). La figure 2.1 montre l'architecture du Transformer de Vaswani *et al.* (2017). Cette architecture utilise le mécanisme d'attention (Bahdanau *et al.*, 2014) pour capturer les relations et dépendances entre les mots d'une phrase. Cette approche permet une compréhension plus profonde du texte, produisant des résultats cohérents, significatifs et de manière plus efficace que les modèles séquentiels traditionnels.

Les composants principaux du Transformer sont l'**encodeur** et le **décodeur**. L'encodeur lit et encode les données d'entrée, tandis que le décodeur génère la sortie. Chaque composant est constitué de plusieurs couches, comprenant des sous-couches d'auto-attention et de réseaux Feed-Forward.

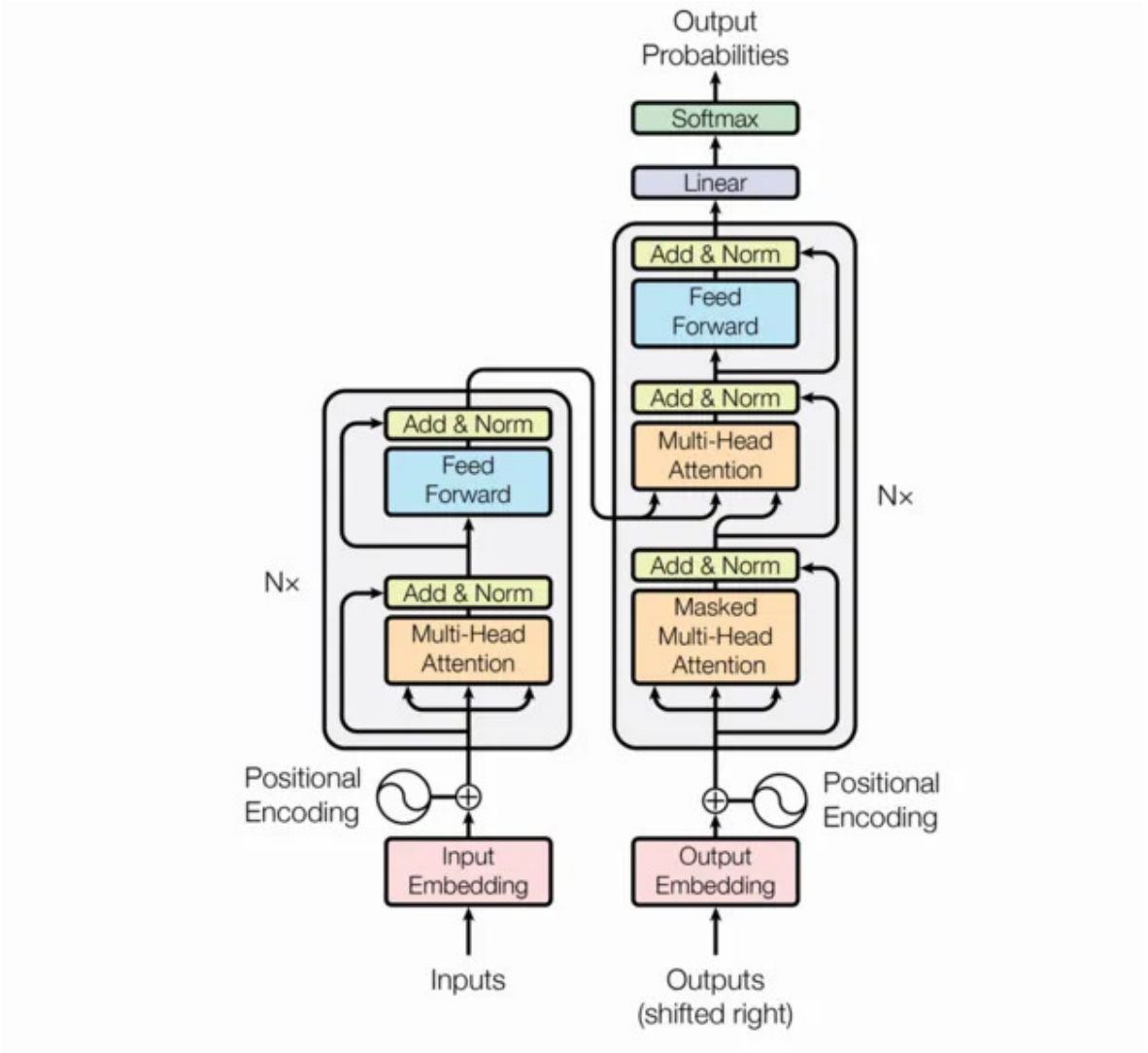

FIGURE 2.1 : l'architecture du Transformer

(Vaswani *et al.*, 2017)

Le mécanisme d' **auto-attention** est l'innovation clé, permettant au modèle de se concentrer sur les parties pertinentes de la séquence d'entrée à travers les couches, quelle que soit leur position. Pour chaque mot, l'auto-attention intègre des informations de tous les autres mots de la séquence, créant une nouvelle représentation enrichie. Par exemple, prenons la phrase suivante "Je viens regarder la pièce, mais je dois d'abord réparer la pièce

de la machine." Dans cette phrase, le mot "pièce" apparaît deux fois, mais dans des sens très différents. La première occurrence désigne une œuvre théâtrale, tandis que la seconde fait référence à une partie d'une machine. Les modèles traditionnels pourraient être confrontés à l'ambiguïté du mot, alors que les modèles basés sur l'architecture Transformer sont capables de discerner le contexte et d'attribuer la bonne signification à chaque occurrence.

Les réseaux **Feed-Forward**, quant à eux, jouent également un rôle essentiel dans l'architecture Transformer. Ils complètent le travail de la sous-couche d'auto-attention en appliquant des transformations supplémentaires aux représentations vectorielles des jetons générées. Ces réseaux agissent comme des mémoires associatives, où des schémas d'entrée spécifiques sont mis en correspondance avec des distributions de sortie particulières (Geva *et al.*, 2020). Cela leur permet de capturer et d'amplifier les relations contextuelles pertinentes identifiées par l'auto-attention. De plus, ils modifient les représentations en mettant l'accent sur des compositions linguistiques spécifiques, telles que des syntaxes ou des structures sémantiques, en fonction de l'objectif de la tâche en aval, telles que la classification ou la génération de texte (Kobayashi *et al.*, 2023).

Dans la phase d'encodage, chaque élément de l'entrée est converti en une représentation numérique (Input Embedding) et combiné avec un encodage de position (Position Encoding). Le décodeur combine la représentation numérique de la sortie (Output Embedding) et l'encodage de position. Il utilise l'auto-attention pour se concentrer sur les jetons précédents de la sortie et intégrer des informations de l'encodeur. Grâce à ces mécanismes, le décodeur prédit l'élément suivant en tenant compte du contexte des séquences d'entrée et de sortie. Ce processus se poursuit jusqu'à la génération complète de la séquence.

Au-delà des applications comme la traduction automatique ou bien l'analyse de sentiment, l'efficacité du Transformer s'étend à d'autres domaines, tels que la détection de

vulnérabilités dans le développement de logiciels. Par exemple, Hou *et al.* (2022) ont proposé un algorithme basé sur l'architecture Transformer pour identifier les vulnérabilités dans les structures de code complexes, obtenant une précision globale de 95,04% et un taux de rappel de 88,89%. Cette approche consiste à extraire les corps des méthodes du code source, à construire des arbres syntaxiques abstraits et à modéliser les informations de séquence pour détecter les vulnérabilités potentielles, ce qui met en évidence la polyvalence et la puissance du Transformer dans diverses applications. De plus, en tant que modèle de langage génératif, l'architecture Transformer peut-être utilisée pour d'autres tâches, telles que la vision par ordinateur (Jamil *et al.*, 2023). Cette architecture a conduit au développement de modèles pré-entraînés, tels que le Generative Pre-trained Transformer (GPT) et Bidirectional Encoder Representations from Transformers (BERT) (Ghojogh & Ghodsi, 2020).

2.2 BERT

BERT (Devlin *et al.*, 2019), qui signifie Bidirectional Encoder Representations from Transformers, est devenu une référence omniprésente dans le traitement du langage naturel (NLP). C'est un modèle de représentation linguistique contextuelle avancé introduit par des chercheurs de Google. Sa spécificité réside dans sa capacité à apprendre des représentations textuelles non étiquetées de manière bidirectionnelle, en prenant en compte simultanément les contextes gauche et droit de chaque mot à travers toutes les couches de son architecture. Contrairement aux modèles de langage classiques qui examinaient les mots de manière unidimensionnelle (de gauche à droite, or de droite à gauche), BERT offre une nouvelle approche permettant d'acquérir une compréhension plus approfondie du langage. Les procédures générales de pré-entraînement et de fine-tuning de BERT sont présentées dans la Figure 2.2.

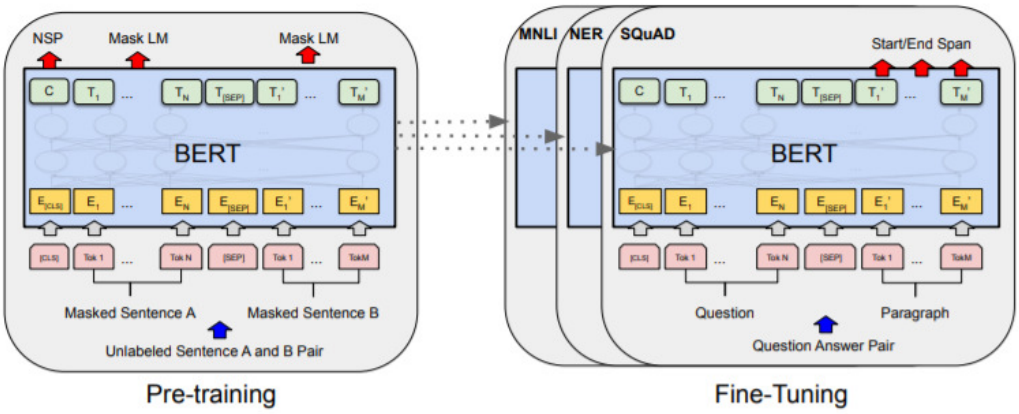

FIGURE 2.2 : l'architecture de BERT.

(Devlin *et al.*, 2019)

Il utilise le mécanisme d'auto-attention mentionné dans la section 2.1, qui lui permet d'évaluer et de capturer l'importance de chaque mot dans une phrase en fonction de son contexte. BERT a été conçu avec deux versions du modèle, offrant différentes capacités en termes de profondeur et de complexité (base et large).

- BERT-base : Cette version est conçue pour les environnements aux ressources informatiques limitées et comprend 110 millions de paramètres avec 12 couches, chacune contenant 12 têtes d'attention et 768 unités cachées.
- BERT-large : Ce modèle est plus grand, avec 24 couches et 16 têtes d'attention et 1024 unités cachées pour 340 millions de paramètres. Grâce à sa taille plus importante, il peut capturer des relations plus complexes dans le texte, offrant ainsi une meilleure performance pour des tâches plus exigeantes, mais au prix d'une plus grande consommation de ressources et de temps d'entraînement.

Toutes les deux versions sont pré-entraînés sur Toronto BookCorpus (800M mots) et English Wikipedia (2,500M mots), en utilisant deux objectifs distincts : la prédiction de mots

masqués (Masked Language Modeling) et la prédiction de la prochaine phrase (Next Sentence Prediction) (Devlin *et al.*, 2019).

2.3 DISTILBERT

Le modèle DistilBERT est une version allégée et optimisée de BERT, présentée dans l'article de blog "Smaller, faster, cheaper, lighter : Introducing DistilBERT, a distilled version of BERT", et le papier (Sanh, 2019). Il est plus petit, rapide et économique, avec 40% de paramètres en moins que la version de base de BERT. Il a été pré-entraîné sur le même ensemble de données que BERT et comprend 6 couches, 12 têtes d'attention et 768 unités cachées, pour un total d'environ 66 millions de paramètres.

Sa taille réduite et ses temps d'apprentissage et d'inférence plus rapides en font un choix populaire parmi les chercheurs et la communauté de traitement du langage naturel(NLP). Des études récentes ont exploré les applications DistilBERT, dans diverses tâches NLP. Récemment, Diwakar & Raj (2024) présentent une méthode de classification de textes utilisant DistilBERT pour le diagnostic automatique des troubles mentaux. Leur approche atteint une précision de 96% en analysant des données textuelles de communautés de santé mentale en ligne, ciblant spécifiquement l'anxiété, le trouble de la personnalité limite et l'autisme. Également, DistilBERT a montré des résultats prometteurs dans la détection de personnalité sur les médias sociaux (Julianda *et al.*, 2023), la classification des sentiments des nouvelles en ligne (Akpatsa *et al.*, 2022), la classification des tweets liés aux catastrophes (Alharbi & Haq, 2024), la réponse aux questions (Mozafari *et al.*, 2020), et la reconnaissance d'entités nommées (NER) (Silalahi *et al.*, 2022). Ces études mettent en évidence la capacité de DistilBERT à maintenir des performances élevées tout en réduisant les exigences de calcul, ce qui le rend adapté aux environnements à ressources limitées.

2.4 ROBERTA

RoBERTa, proposé par Liu *et al.* (2019) dans RoBERTa : A Robustly Optimized BERT Pretraining Approach est une optimisation robuste du modèle BERT qui a prouvé que BERT était sous-entraîné et pouvait atteindre des résultats supérieurs avec un réglage minutieux des hyperparamètres et une augmentation des données d'entraînement (Liu *et al.*, 2019).

La principale différence entre RoBERTa et BERT réside dans la taille du corpus de pré-entraînement. Alors que BERT utilise un ensemble de données de 16 Go, RoBERTa a été entraîné sur un corpus beaucoup plus grand, dépassant 160 Go de texte non compressé. Ce corpus élargi comprend des données issues de Wikipédia et de livres en anglais (16 Go), ainsi que des textes web divers, des articles de CommonCrawl News et des récits de Common Crawl, totalisant 160 Go (Liu *et al.*, 2019).

Grâce à ces améliorations, RoBERTa a obtenu des performances supérieures sur des benchmarks tels que GLUE, RACE et SQuAD, sans nécessiter de réglages fins multi-tâches pour GLUE ni de données supplémentaires pour SQuAD (Liu *et al.*, 2019). Comme BERT, le modèle est disponible en deux versions : base et large. La version de base comprend 12 couches, 12 têtes d'attention et 768 unités cachées, totalisant environ 125 millions de paramètres. La version large, quant à elle, comporte 24 couches, 16 têtes d'attention et 1024 unités cachées, pour un total de 335 millions de paramètres.

2.5 INTELLIGENCE ARTIFICIELLE EXPLICABLE (XAI)

L'intelligence artificielle (IA) a connu des avancées significatives dans divers domaines, grâce à l'apprentissage automatique (ML). Cependant, certains algorithmes, comme l'apprentissage profond (DL), sont considérés comme des systèmes de boite noire ou "black-box", rendant leur fonctionnement et la compréhension des décisions qu'ils prennent difficiles. Des

modèles comme XGBoost, utilisés pour surmonter l'overfitting, illustrent cette complexité. Pour répondre aux défis de compréhension des modèles, des techniques d'explicabilité (XAI) ont été développées afin d'éclaircir le degré de confiance dans les résultats et d'améliorer les modèles. Des recherches menées par l'agence américaine DARPA soulignent l'importance de ces méthodes (Gunning & Aha, 2019). Ces techniques se concentrent sur la compréhension par l'homme des décisions de l'IA, ce qui est crucial pour établir la confiance des modèles d'intelligence artificielle dans des applications critiques telles que les soins de santé, les véhicules autonomes et la défense (Gohel *et al.*, 2021).

Plusieurs techniques XAI sont couramment utilisées pour les algorithmes d'apprentissage automatique et d'apprentissage profond. Ces méthodes incluent des approches globales pour expliquer les décisions du modèle ainsi que des explications spécifiques pour des entrées individuelles. Elles permettent de répondre aux préoccupations illustrées dans la figure 2.3 concernant la prise de décision par l'IA (Choudhary *et al.*, 2022; Reddy & Kumar, 2023).

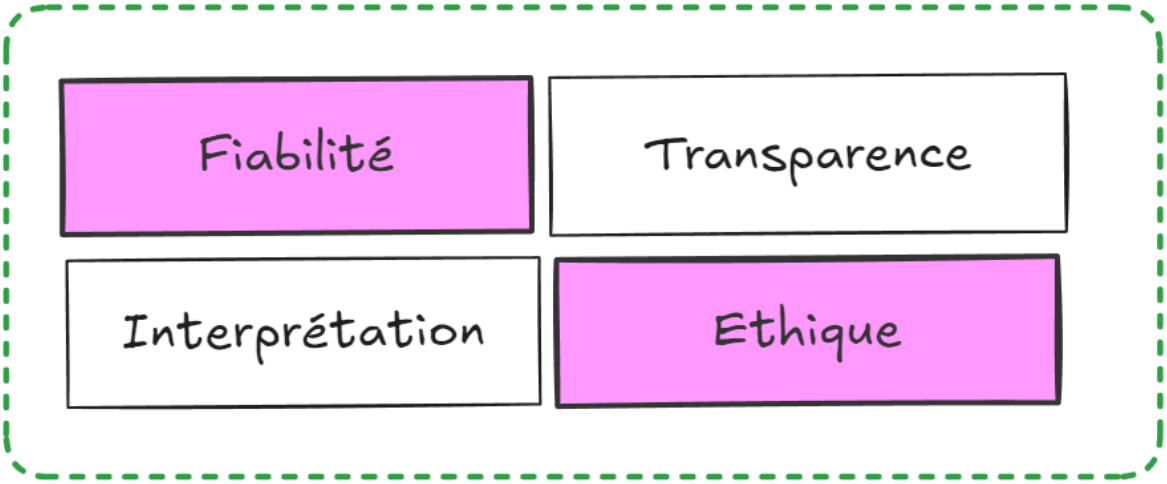

FIGURE 2.3 : Raison d'explicabilité de l'intelligence artificielle (XAI).

Dans cette optique, notre étude se concentre sur des techniques telles que LIME (Local Interpretable Model-Agnostic Explanations), SHAP (SHapley Additive exPlanations) et la méthode de visualisation basée sur l'attention.

2.5.1 LIME (LOCAL INTERPRETABLE MODEL-AGNOSTIC EXPLANATIONS)

LIME, ou Explications Locales Interprétables Indépendantes du Modèle en français, est un framework populaire pour expliquer les modèles d'apprentissage automatique (ML) et d'apprentissage profond (DL) introduit par Ribeiro *et al.* (2016). Cette technique permet d'extraire l'importance des différentes caractéristiques dans une variété de types de données, y compris les images, du texte, ainsi que les données tabulaires pour des tâches de classification. L'idée centrale de LIME est de générer un modèle de substitution local, pour fournir des explications sur les résultats des décisions. Il crée un jeu de données avec des entrées perturbées et les prédictions correspondantes du modèle en boîte noire (black-box). Sur ce nouveau jeu de données, LIME entraîne un modèle interprétable, qui est pondéré en fonction de la proximité des instances échantillonnées avec l'instance d'intérêt (Choudhary *et al.*, 2022). Cependant, LIME ne se concentre pas sur l'analyse des prédictions du modèle dans sa globalité, mais examine plutôt une petite partie, comme un échantillon unique ou un groupe d'échantillons similaires appartenant à la même classe. L'algorithme LIME part du principe qu'en analysant un échantillon isolé, on se concentre sur une zone locale où la régression linéaire peut être appliquée avec précision (Ribeiro *et al.*, 2016). L'explication générée par LIME utilise la formule mathématique présentée dans l'équation 1 ci-dessous.

$$\xi(x) = \underset{g \in G}{\arg\max}\, L(f, g, \pi_x) + \Omega(g) \tag{1}$$

Où :

- f est le modèle réel (boîte noire) ;

- g le modèle de substitution appartenant à l'ensemble G ;
- G est la classe des modèles interprétables potentiels ;
- π_x définit la taille du voisinage ;
- L est la fonction de perte pondérée qui mesurera la proximité de l'explication ;
- $\Omega(g)$représente la complexité du modèle de substitution.

2.5.2 SHAP (SHAPLEY ADDITIVE EXPLANATIONS)

SHAP, ou Explications Additives de Shapley en français, est une méthode proposée par Lundberg (2017) pour l'interprétation des prédictions des modèles, attribuant des valeurs d'importance aux caractéristiques pour des prédictions spécifiques. C'est une méthode puissante et largement utilisée pour l'interprétation de modèles complexes d'apprentissage automatique, y compris les forêts aléatoires (RF), les machines à vecteurs de support (SVM) et les réseaux neuronaux profonds (DNN) (Rodríguez-Pérez & Bajorath, 2019). Développé sur la base de la théorie des jeux coopératifs (Shapley, 1953), il offre une méthode claire et théorique pour attribuer à chaque caractéristique un score pour sa contribution à la prédiction finale du modèle. C'est pourquoi il a gagné en importance dans diverses études en raison de son solide fondement théorique et de son applicabilité générale (Mosca *et al.*, 2022). Plusieurs variantes basées sur SHAP ont été développées, adaptant les hypothèses de base et ciblant des applications spécifiques, en particulier dans le domaine du NLP (Mosca *et al.*, 2022). SHAP calcule la valeur de Shapley et la présente sous la forme d'un modèle linéaire ou d'une attribution additive des caractéristiques. La formule principale de SHAP est illustré dans l'équation 2 ci-dessous.

$$g(x_0) = \theta_0 + \sum_{j=1}^{M} \theta_j z_{0j} \tag{2}$$

Où $g(x_0)$ est un modèle d'explication, M la taille maximale de la coalition, ou le nombre total de caractéristiques, θ_j est la contribution pour la caractéristique j et z_{0j} est la valeur ou le vecteur binaire de l'entrée x_0. En plus de fournir des explications locales telles que LIME, SHAP propose également des explications globales. Chaque caractéristique est décrite de façon additive et optimale. Par contre, SHAP est plus coûteux en ressources et de calcul, car il nécessite l'évaluation de toutes les permutations possibles des caractéristiques pour parvenir à une attribution précise (Molnar, 2023).

2.5.3 EXPLICABILITÉ BASÉ SUR L'ATTENTION

Le mécanisme d'attention de Bahdanau *et al.* (2014) est devenu populaire dans les tâches de NLP. Les poids d'attention, générés par les modèles basés sur l'attention, indiquent l'importance des différentes parties de l'entrée pour une prédiction donnée et peuvent être utilisés pour l'interprétation locale, comme SHAP et LIME (Zhao *et al.*, 2024). Dans le contexte du NLP, les jetons (tokens) ayant des poids plus élevés sont considérés comme importants (Choudhary *et al.*, 2022). En utilisant des techniques de visualisation telles que les cartes thermiques ou les graphes bipartites (voir Figure 2.4), comme le souligne Zhao *et al.* (2024), il est possible de représenter les poids d'attention de manière graphique. Cela aide à mieux comprendre l'impact des caractéristiques sur les prédictions et rend les décisions du modèle plus transparentes. Diverses recherches ont exploré ces techniques de visualisation de l'attention pour l'explicabilité des modèles. Par exemple, BertViz développé par Vig (2019) est un outil open-source permettant de visualiser l'auto-attention multi-têtes dans les modèles basés sur Transformer, en particulier BERT et GPT-2. Cet outil peut être utilisé pour détecter les biais du modèle, identifier des schémas récurrents et établir des liens entre les neurones et le comportement du modèle (Vig, 2019). Ainsi, La visualisation de l'attention permet de

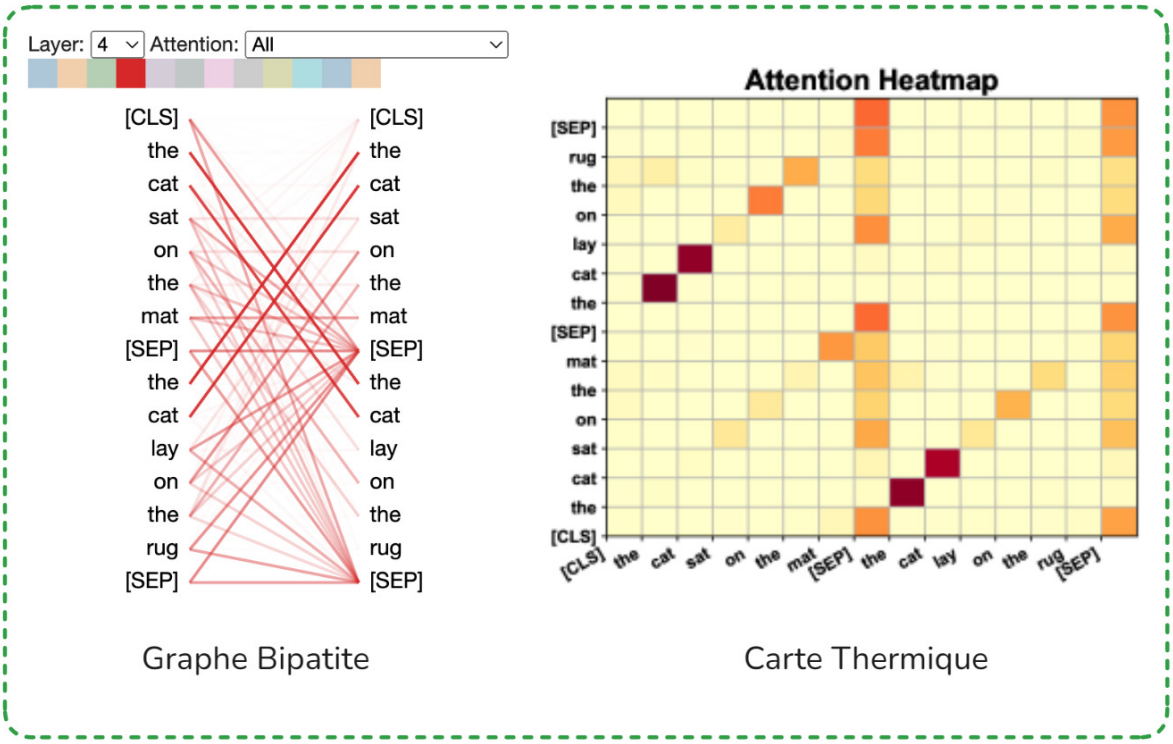

FIGURE 2.4 : Représentation de l'attention par graphe bipartite et carte thermique pour la matrice d'attention.

(Zhao *et al.*, 2024)

diagnostiquer les biais et erreurs, d'évaluer les règles de décision, et aide à formuler des hypothèses explicatives (Zhao *et al.*, 2024).

CHAPITRE III

TRAVAUX CONNEXES

L'expansion rapide de l'IoT a exposé une vaste surface d'attaque, nécessitant des mesures de sécurité avancées (Srinivasan, 2024). Ce chapitre donne un aperçu des recherches antérieures dans le sujet. Nous discuterons des différentes techniques utilisées pour détecter et classifier les cyberattaques sur les appareils IoT en général, ainsi que les dispositifs utilisés dans le domaine médical.

3.1 APPROCHES BASÉES SUR ML ET DL

Des recherches récentes se concentrent sur l'amélioration de la sécurité et la classification des attaques IoT en utilisant les algorithmes de ML et de DL. Wang *et al.* (2024) ont proposé une méthode de classification des appareils basée sur la sensibilité aux attaques. Ils ont développé un modèle de classification multi-classes, basé sur des réseaux neuronaux convolutifs (CNN), catégorisant les appareils dans des groupes prédéfinis et atteignant une précision globale de 99,91%. Aldhaheri *et al.* (2024) explorent également diverses méthodes de DL pour la détection des intrusions dans les réseaux IoT en abordant les défis de sécurité, les algorithmes, les ensembles de données, les types d'attaques, et les mesures d'évaluation pour une détection efficace des cybermenaces. Une autre étude se concentre particulièrement sur DNN, LSTM, et CNN, pour classer les attaques de cybersécurité dans les réseaux IoT (Becerra-Suarez *et al.*, 2024). Les auteurs démontrent que CNN est supérieur en termes de performance et d'efficacité avec une précision globale de 99,10% pour la classification multi-classes et de 99,40% pour la classification binaire. Pendant que ces différentes recherches se concentrent sur l'utilisation des algorithmes DL, d'autres chercheurs mettent en évidence l'utilisation des méthodes de ML. Par exemple, Churcher *et al.* (2021) comparent divers algorithmes ML pour

la classification des attaques binaires et multi-classes et ont constaté que Random Forest (RF) était le plus performant pour la classification binaire, tandis que K-Nearest Neighbors (KNN) excellait dans les scénarios multi-classes. Xu *et al.* (2023) proposent une approche axée sur les données pour la détection des intrusions et des anomalies, utilisant divers algorithmes de traitement et de filtrage des données. Leur méthode atteint une précision de globale de 99,7% pour la classification multi-classes.

Pour sécuriser efficacement les dispositifs utilisés dans le domaine de la santé, des applications de sécurité importantes ont été proposées. Khan & Alkhathami (2024) affirment que les techniques de ML peuvent être utilisées pour détecter efficacement le trafic réseau anormal et les attaques contre les appareils médicaux. Récemment, Dhanushkodi *et al.* (2024) ont développé un modèle hybride utilisant l'optimisation PLSTM et OOYO, atteignant une précision de 99% dans la détection du trafic malveillant dans IoMT. De même, Sripriyanka & Mahendran (2024) ont présenté ICDC-Net, qui combine le chiffrement et le DL pour la détection d'attaques DDoS contre IoMT et la classification COVID-19 à partir de radiographies du thorax. Kumar *et al.* (2021) ont proposé un cadre d'apprentissage d'ensemble (Ensemble Learning Framework) à l'aide d'une architecture "Fog-Cloud", utilisant des modèles tels que RF, l'arbre de décision (DT), naive bayes (NB), et XGBoost. Wagan *et al.* (2022) ont présenté un framework "Duo-Secure IoMT" utilisant des données sensorielles multimodales pour identifier les schémas d'attaque et traiter les données médicales en toute sécurité.

Ces études démontrent collectivement l'importance des méthodes de classification avancées et des techniques ML et DL pour relever les défis de sécurité croissants posés par la prolifération des dispositifs IoT.

3.2 APPROCHES BASÉES SUR LES TRANSFORMERS

Plusieurs recherches se sont concentrées sur le développement de modèles basé sur le mécanisme d'auto-attention, spécialisés pour la sécurité de IoT. Récemment Sasi *et al.* (2024) présente un nouveau modèle appelé 1D-CNN-LSTM, qui utilise CNN combinés à LSTM amélioré par l'auto-attention pour détecter les attaques de l'IoT. Huang & Wang (2024) ont proposé SecBERT, un protocole d'inférence de réseau neuronal préservant la vie privée pour les tâches de NLP à l'aide d'un partage de secret additif. SecurityBERT, un modèle basé également sur BERT pour le trafic réseaux IoT, intègre une nouvelle technique d'encodage préservant la vie privée et atteint une précision globale de 98,2% dans l'identification de quatorze types d'attaques, surpassant les solutions hybrides précédentes (Ferrag *et al.*, 2024). Ces modèles permettent d'améliorer les performances en matière d'analyse de données des réseaux IoT, de détection des menaces et de préservation de la vie privée. De plus, Rahali & Akhloufi (2021) ont présenté MalBERT, pour l'analyse statique des applications Android, et ont obtenu des performances élevées dans la classification des logiciels malveillants. Dans un contexte différent, Xu *et al.* (2021) ont proposé une méthode de pré-entraînement pour la détection des logiciels malveillants sur les systèmes Windows, démontrant que leur solution peut atteindre un taux de détection de 99,9% grâce à l'analyse dynamique. Ces études mettent également en évidence le potentiel des modèles basés sur l'auto-attention mécanisme pour améliorer la cybersécurité des systèmes IoT et IoMT.

Cependant, ces travaux précédents n'ont pas fourni d'explications sur leurs modèles, bien que la fiabilité et l'efficacité des modèles nécessite transparence et compréhension, surtout pour leur utilisation dans des domaines sensibles tels que la santé. Dans ce contexte, des études récentes soulignent l'importance cruciale de l'explicabilité des grands modèles de langage (LLM), qui sont de plus en plus utilisés dans diverses applications et présentent des difficultés en raison de leur nature "black-box" (Luo & Specia, 2024; Zhao *et al.*, 2024).

3.3 TECHNIQUES XAI BASÉS SUR DES MODÈLES DE TYPE TRANSFORMERS

Les modèles basés sur les Transformers, comme BERT, DistilBERT et RoBERTa, sont très performants dans divers domaines, mais leur complexité les rend difficilement interprétables. Pour remédier à ce problème, plusieurs techniques XAI ont été développées afin d'expliquer les prédictions des modèles. Zhao *et al.* (2024) ont proposé un résumé structuré des méthodes d'explication basées sur les Transformers, en distinguant les explications locales et globales selon les techniques appliquées. Parmi les principales approches XAI appliquées aux modèles Transformers, les cartes de saillance (saliency maps) ont été utilisées par plusieurs chercheurs intéressés par les tâches de vision par ordinateur intégrant les Transformers. Beemelmanns *et al.* (2023) proposent une méthode pour générer des cartes de saillance dans la détection d'objets 3D multi-caméras. Englebert *et al.* (2023) présentent Transformer Input Sampling (TIS), une méthode basée sur la perturbation qui évite d'introduire des caractéristiques aberrantes. Nous avons également l'analyse par attribution en utilisant les gradients intégrés (Integrated Gradients). Les gradients intégrés (IG) sont une méthode XAI de premier plan pour interpréter les réseaux neuronaux profonds, y compris les Transformers (Sundararajan *et al.*, 2017). Des recherches récentes ont étendu l'IG à l'analyse des modèles Transformers dans divers domaines, tels que la prédiction de la fonction des protéines (Wenzel *et al.*, 2024) et le traitement du langage naturel (Zhao *et al.*, 2024; Attanasio *et al.*, 2022). Ensuite, la propagation de la pertinence par couches (LRP), une autre méthode de l'analyse par attribution. LRP est une technique permettant d'expliquer les prédictions des réseaux neuronaux en propageant la pertinence en amont dans le réseau (Montavon *et al.*, 2019). Des recherches récentes ont étendu la LRP au traitement des couches d'attention dans les Transformers (Achtibat *et al.*, 2024). l'auto-attention permet au modèle de se concentrer sur les informations importantes tout en ignorant les détails non pertinents des données d'entrée (Hu, 2020; Zhang & Kim, 2023) et plusieurs travaux analysent les scores d'attention pour fournir des explications. La

méthode la plus populaire est la visualisation, qui permet d'afficher l'attention entre les jetons pour montrer les relations que le modèle considère comme importantes (voir section 2.5.3).

Ce mémoire apporte une contribution significative à l'amélioration et à la spécification d'un modèle de langage large (LLM) plus transparent, interprétable et fiable, spécialement affiné pour la classification des attaques ciblant l'Internet des objets médicaux (IoMT) et aborde les limites des études précédentes en se concentrant sur l'explicabilité et l'interprétabilité des LLM dans la classification des attaques contre IoT. Plus précisément :

- Cette recherche met l'accent sur la transparence et l'interprétabilité des modèles de langage large (LLM), en soulignant leur importance pour renforcer la fiabilité et l'efficacité de ces modèles complexes.
- L'étude utilise diverses techniques, telles que des méthodes d'explicabilité indépendantes du modèle, des outils de visualisation et des mécanismes d'attention, afin de mieux comprendre les processus de prise de décision des LLM et de réduire leur nature de boite noire.

CHAPITRE IV
MÉTHODOLOGIE

Ce chapitre décrit la méthodologie de cette étude, comprenant le prétraitement des données, l'entraînement et l'évaluation des modèles BERT, DistilBERT et RoBERTa, ainsi que l'interprétation du modèle le plus performant pour répondre à notre problématique. L'approche est structurée selon les phases suivantes :

- Prétraitement de l'ensemble de données CICIoMT2024 utilisé, comprenant la réduction des caractéristiques et le rééquilibrage des classes ;
- L'entraînement (fine-tuning) des modèles sur le jeu de données pré-traité ;
- Évaluation et comparaison des performances des modèles ;
- Interprétation des résultats du modèle le plus performant à l'aide de techniques d'explicabilité telles que LIME, SHAP et la visualisation de l'attention ;

Ce processus fournit une méthode systématique pour évaluer les grands modèles de langage (LLM) utilisés dans l'analyse du trafic réseau, en mettant en avant leurs performances, leur transparence et leur explicabilité. La figure 4.1 illustre le flux de travail général, décrivant étape par étape le processus.

4.1 PRÉPARATION DES DONNÉES

Dans cette section, nous abordons des étapes fondamentales du cycle de vie de la science des données, à savoir la collecte des données, leur nettoyage, et leur exploration et traitement.

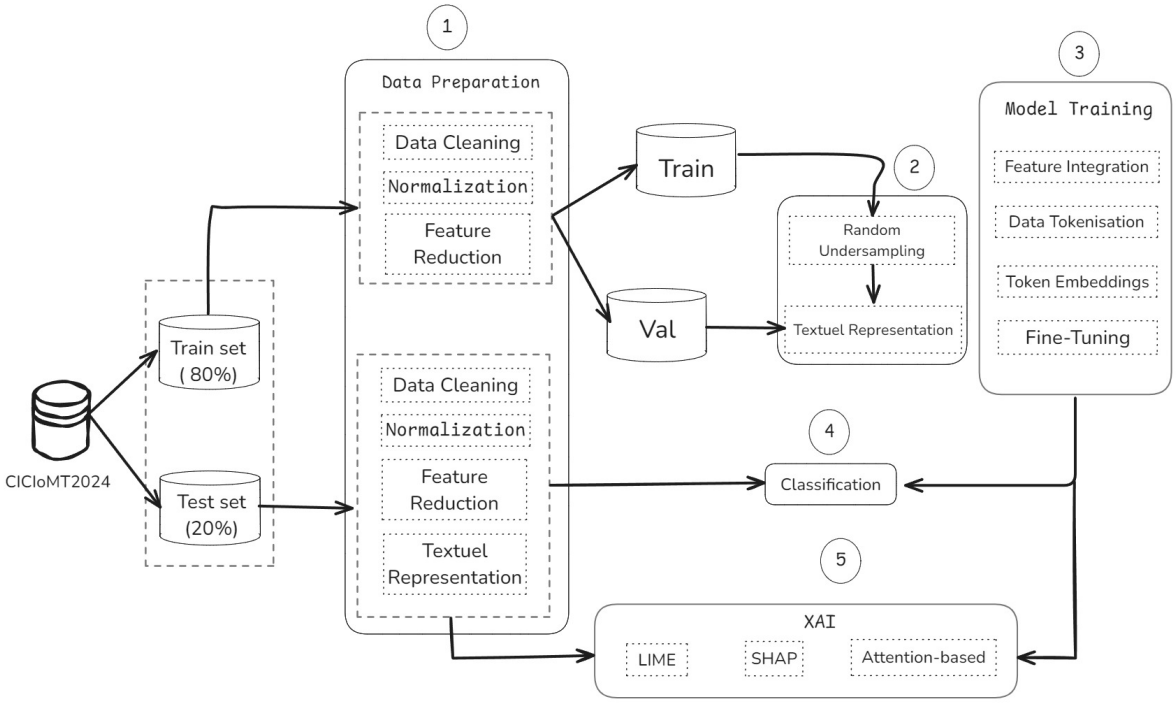

FIGURE 4.1 : Flux de travail général de l'étude.

4.1.1 COLLECTE DES DONNÉES ET DESCRIPTION

COLLECTE DES DONNÉES

Dans la présente étude, nous utilisons un récent ensemble de données d'attaques IoMT disponibles publiquement, à savoir CICIoMT2024 (Dadkhah *et al.*, 2024), qui comprend diverses attaques contre les appareils IoMT. L'objectif de cet ensemble de données est d'aider les chercheurs à développer des solutions créatives pour améliorer la sécurité des systèmes de santé en utilisant des techniques telles que l'apprentissage automatique. Les auteurs ont exécuté 18 attaques basées sur des tests effectués sur 40 appareils IoMT, avec 25 appareils réels et 15 appareils simulés. Ces attaques sont classées en cinq catégories (voir Figure 4.2) : DDoS (déni de service distribué), DoS (déni de service), Recon (reconnaissance), MQTT (protocole

de transport de télémétrie) et Spoofing (usurpation d'identité). En outre, l'ensemble de données contient des activités bénignes associées à des comportements normaux. Il couvre une pluralité de protocoles utilisés dans le secteur de la santé, tels que Wi-Fi, MQTT, et Bluetooth. Cet

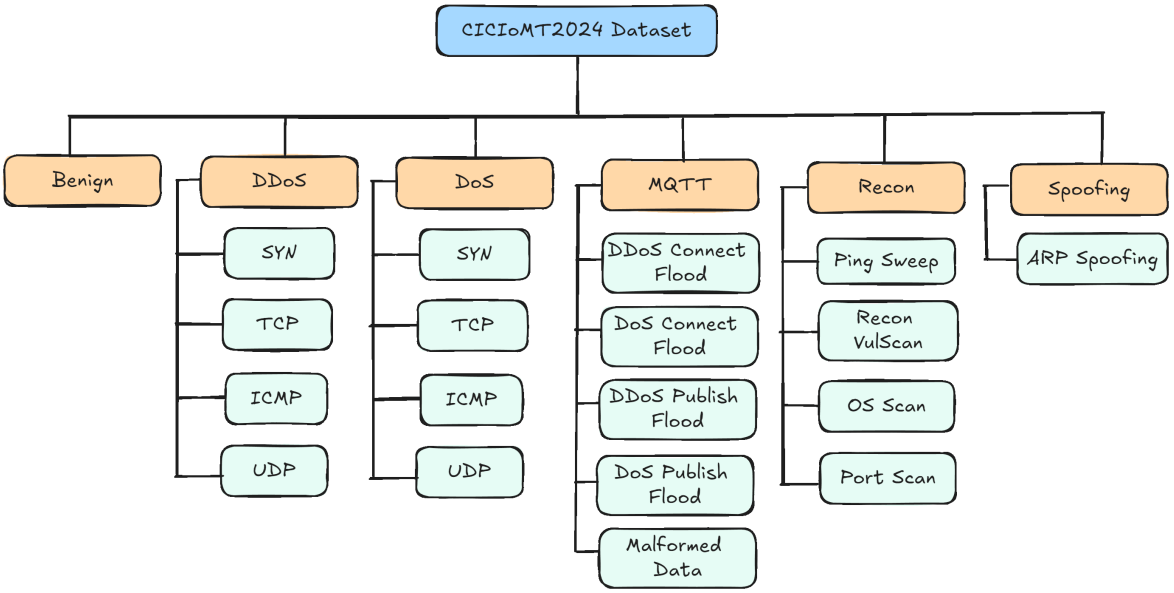

FIGURE 4.2 : Topologie de l'ensemble de données CICIoMT2024.

(Dadkhah *et al.*, 2024)

ensemble de données comprend deux formats de fichiers différents : PCAP (captures de paquets) et CSV (valeurs séparées par des virgules). Les fichiers PCAP contiennent des données brutes et non structurées, tandis que les fichiers CSV contiennent des données traitées et structurées, prêtes à être utilisées. Nous avons suivi une approche méthodique pour la collecte de nos données. Nous avons importé les données d'entraînement et de test dans leurs répertoires respectifs et trié les fichiers CSV afin d'identifier les différents types d'attaques. Nous avons regroupé les attaques DoS et DDoS en une seule classe, "DoS", pour faciliter l'analyse, et classé les attaques en quatre classes au lieu de cinq. En outre, nous avons ajouté une colonne cible pour l'étiquetage. Le jeu de données contenait 7 160 831 échantillons et

46 colonnes pour l'ensemble d'apprentissage et 1 614 182 échantillons et 46 colonnes pour l'ensemble de test.

DESCRIPTION DES DONNÉES

Dans cette sous-section, nous décrivons les principales caractéristiques du jeu de données CICIoMT2024 utilisées dans cette étude (voir Tableau 4.1). Ces caractéristiques capturent des informations essentielles sur les paquets réseau et les protocoles, facilitant ainsi l'analyse approfondie et la classification des flux de trafic réseau. Chaque caractéristique est décrite avec précision pour assurer une compréhension claire de sa contribution à l'analyse.

4.1.2 LE NETTOYAGE DES DONNÉES

Le nettoyage de données est l'ensemble des actions visant à identifier et à rectifier (par le biais de suppression des erreurs, de réduction des caractéristiques et de l'équilibrage) de l'ensemble de données.

Dans cette étape, nous avons tout d'abord identifié et supprimé les caractéristiques de notre jeu de données dont la variance est nulle. Une variance nulle signifie que toutes les valeurs d'une colonne sont identiques, ce qui rend la caractéristique non informative pour toute tâche de modélisation prédictive. En supprimant ces caractéristiques à variance nulle, nous nous assurons que l'ensemble de données utilisé pour l'apprentissage des modèles ne contient pas d'informations redondantes, ce qui peut contribuer à améliorer les performances des modèles et à réduire la complexité et le temps de calcul. En conséquence, sur les 45 caractéristiques, seule **drate** a été supprimée. Bien qu'il fasse partie du jeu de données, cette caractéristique ne figure même pas dans la liste de description des caractéristiques donnée par Dadkhah *et al.* (2024). Ensuite, nous vérifions et traitons les valeurs manquantes et les doublons

TABLEAU 4.1 : Description des caractéristiques de CICIoMT2024

(Dadkhah *et al.*, 2024)

#	Caractéristiques	Description
1	Header Length	Longueur de l'en-tête du paquet
2	Duration	Durée de vie du paquet en transit
3	Rate	Vitesse de transmission des paquets dans un flux
4	Srate	Vitesse de transmission des paquets sortants dans un flux
5	Fin flag number	Valeur du drapeau Fin dans TCP/IP
6	Syn flag number	Valeur du drapeau Syn dans TCP/IP
7	Rst flag number	Valeur du drapeau Rst dans TCP/IP
8	Psh flag number	Valeur du drapeau Psh dans TCP/IP
9	Ack flag number	Valeur du drapeau Ack dans TCP/IP
10	Ece flag number	Valeur du drapeau Ece dans TCP/IP
11	Cwr flag number	Valeur du drapeau Cwr dans TCP/IP
12	Syn count	Nombre d'occurrences du drapeau Syn dans un flux
13	Ack count	Nombre d'occurrences du drapeau Ack dans un flux
14	Fin count	Nombre d'occurrences du drapeau Fin dans un flux
15	Rst count	Nombre d'occurrences du drapeau Rst dans un flux
16	IGMP	Utilisation d'IGMP dans les protocoles de couche application
17	HTTPS	Utilisation de HTTPS dans les protocoles de couche application
18	HTTP	Utilisation de HTTP dans les protocoles de couche application
19	Telnet	Utilisation de Telnet dans les protocoles de couche application
20	DNS	Utilisation de DNS dans les protocoles de couche application
21	SMTP	Utilisation de SMTP dans les protocoles de couche application
22	SSH	Utilisation de SSH dans les protocoles de couche application
23	IRC	Utilisation d'IRC dans les protocoles de couche application
24	TCP	Utilisation de TCP dans les protocoles de transport
25	UDP	Utilisation de UDP dans les protocoles de transport
26	DHCP	Présence de DHCP dans les protocoles de couche application
27	ARP	Utilisation d'ARP dans les protocoles de couche liaison
28	ICMP	Utilisation d'ICMP dans les protocoles de couche réseau
29	IPV	Utilisation d'IP dans les protocoles de couche réseau
30	LLC	Utilisation de LLC dans les protocoles de couche liaison
31	Tot sum	Longueur totale des paquets dans un flux
32	Min	Longueur minimale d'un paquet dans un flux
33	Max	Longueur maximale d'un paquet dans un flux
34	AVG	Longueur moyenne des paquets dans un flux
35	Std	Variabilité de la longueur des paquets dans un flux
36	Tot size	Longueur du paquet
37	IAT	Intervalle entre le paquet courant et le paquet précédent
38	Number	Nombre total de paquets dans le flux
39	Radius	Racine carrée moyenne des variances des longueurs des paquets entrants et sortants dans le flux
40	Magnitude	Racine carrée moyenne des moyennes des longueurs des paquets entrants et sortants dans le flux
41	Variance	Rapport des variances des longueurs des paquets entrants et sortants dans le flux
42	Covariance	Covariance entre les longueurs des paquets entrants et sortants
43	Weight	Produit du nombre de paquets entrants et sortants
44	Protocol Type	Type de protocole utilisé (IP, UDP, TCP, etc.) exprimé en valeurs entières

dans l'ensemble de données. Il s'agit d'une étape cruciale dans le nettoyage des données, car les valeurs manquantes et les doublons peuvent avoir un impact négatif sur la performance et la précision des modèles. Puis, les 44 caractéristiques restantes ont été normalisées à l'aide de

la méthode StandardScaler de la bibliothèque de prétraitement Python sklearn. Cette méthode est une technique de prétraitement courante utilisée pour transformer les données afin qu'elles aient une moyenne de zéro et un écart-type de 1. Ceci est particulièrement utile pour les algorithmes d'apprentissage automatique et d'apprentissage profond qui sont sensibles à l'échelle des caractéristiques d'entrée.

Pour optimiser notre analyse et la préparation des données, nous avons utilisé la matrice de corrélation pour identifier et traiter les relations entre les caractéristiques. Les caractéristiques fortement corrélées, avec des coefficients de corrélation proches de 1 ou -1, fournissent souvent des informations redondantes. En utilisant le coefficient de corrélation de Pearson(PCC), comme indiqué par Khan & Alkhathami (2024), nous avons identifié et supprimé les caractéristiques dont les corrélations dépassaient 0.95 (voir Figure 4.3), à l'exclusion des auto-corrélations.

Les caractéristiques supprimées comprenaient **srate**, **llc**, **udp**, **ipv**, **tot_size**, **number**, **magnitude**, **radius**, **covariance** et **weight**. Cette étape a permis de réduire la complexité de l'ensemble de données et d'atténuer le risque que des caractéristiques redondantes aient un impact négatif sur les performances du modèle. À l'issue de ce processus, les ensembles finaux de formation et de test étaient composés de 34 caractéristiques, garantissant un ensemble de données rationalisé pour améliorer l'efficacité du modèle.

Après avoir achevé la phase d'extraction et de réduction des caractéristiques, notre ensemble de données se compose finalement de 7 155 712 échantillons et de 35 colonnes, y compris l'étiquette, ainsi que de 1 612 117 échantillons destinés à l'évaluation. Cependant, les données sont déséquilibrées avec une représentation importante de la classe DoS. L'équilibrage de l'ensemble des données est essentiel pour améliorer les performances et l'équité

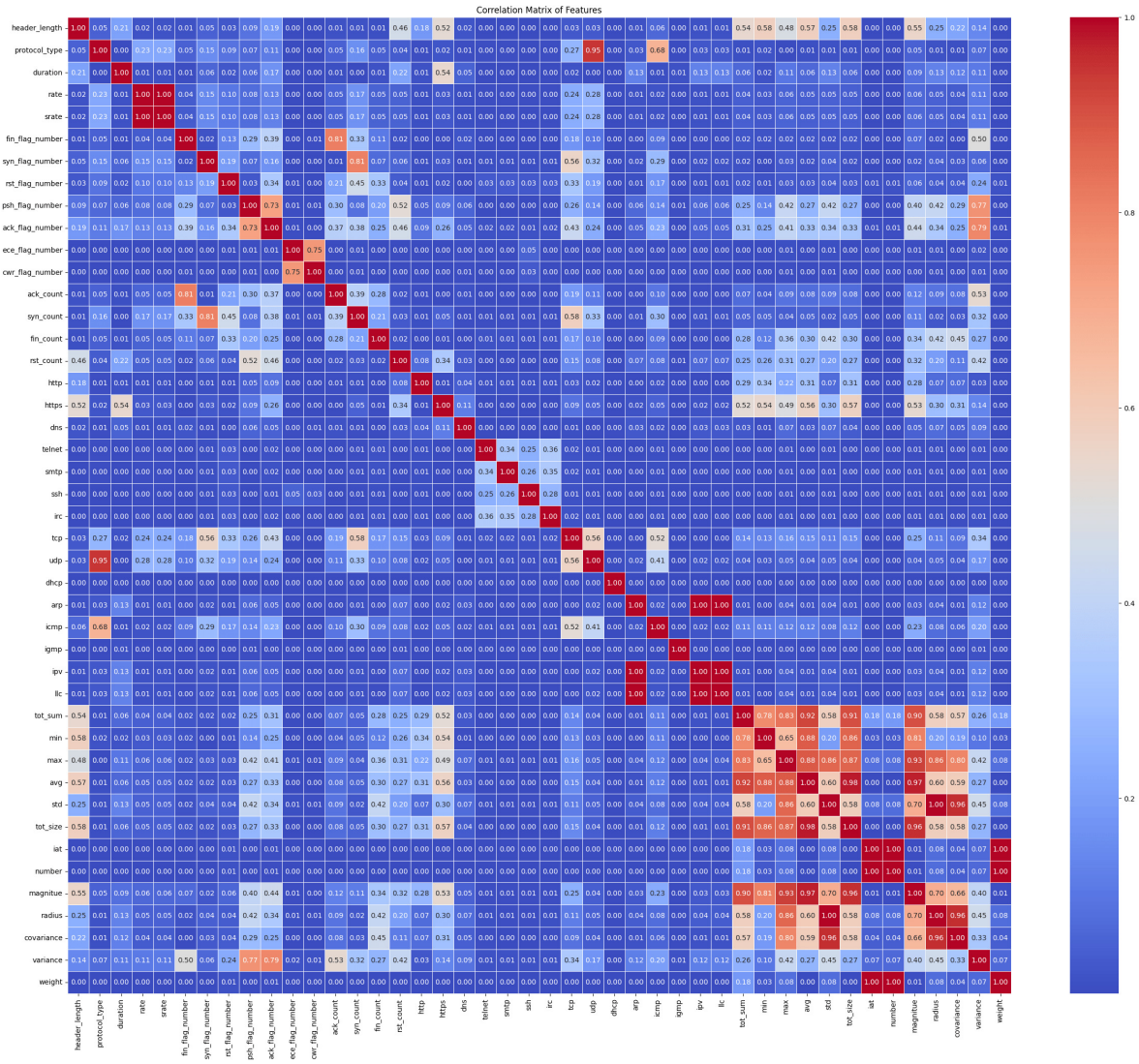

FIGURE 4.3 : Matrice de corrélation de l'ensemble des données.

des modèles d'apprentissage automatique, particulièrement dans les tâches de classification déséquilibrées (Ali *et al.*, 2013; Mohammed *et al.*, 2020). Pour y remédier, nous avons appliqué un sous-échantillonnage aléatoire aux données d'entrainement. La Figure 4.4 montre la distribution des données d'entrainement avant et après la préparation.

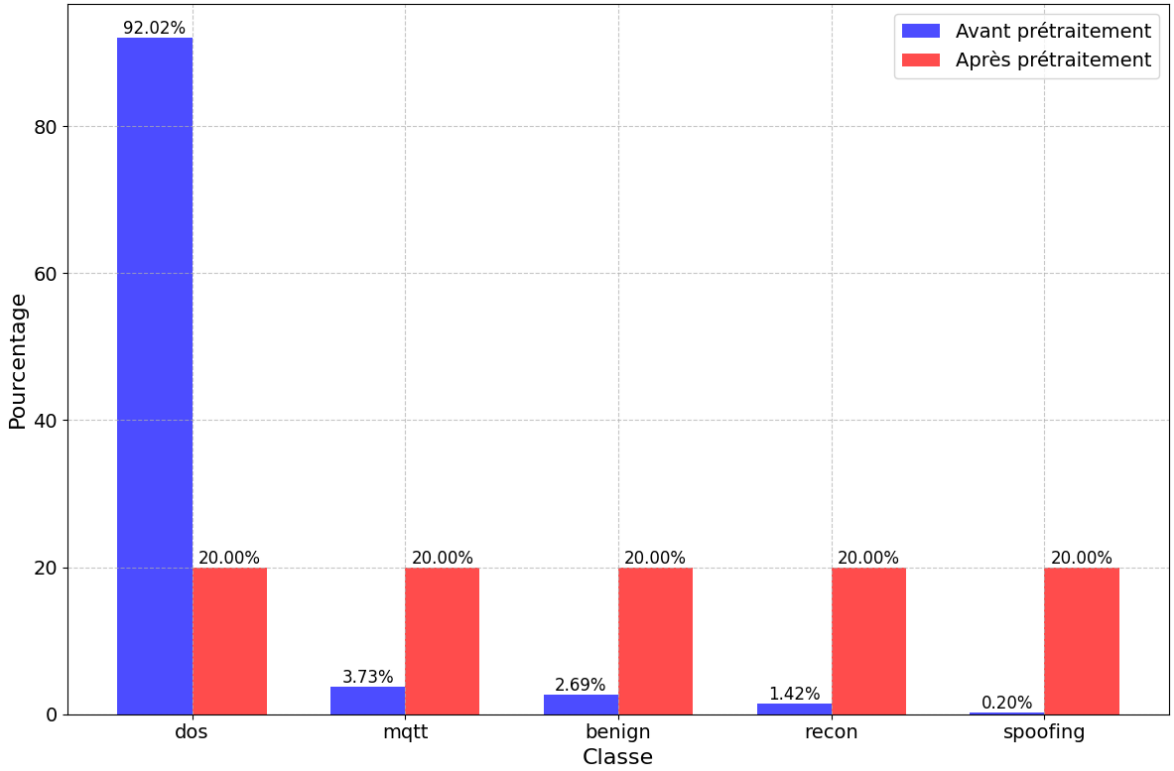

FIGURE 4.4 : Distribution des données d'entrainement avant et après prétraitement.

Le processus a commencé par un échantillonnage stratifié de cet ensemble de données d'entrainement initial, 5% (357 786 instances) ont été alloués à la validation, tandis que les 95% restants (6 797 926 instances) ont été désignés pour la formation du modèle. L'échantillonnage stratifié garantit que la distribution des classes dans les ensembles de formation et de validation reste proportionnelle à celle de l'ensemble de données original. Cette approche préserve l'intégrité des données et aide le modèle à se généraliser efficacement, particulièrement pour les classes minoritaires. La division de l'ensemble de données a été effectuée à l'aide de la méthode **train_ test_split** du module sklearn de Python.

Ensuite, un sous-échantillonnage aléatoire a été appliqué à la classe majoritaire de l'ensemble dédié à la formation pour obtenir une distribution équilibrée des classes (Devi

et al., 2020). Cette méthode réduit le risque de biaiser le modèle en faveur de la classe majoritaire, améliorant ainsi sa capacité à détecter les classes minoritaires. Après équilibrage, l'ensemble d'entrainement final contenait 15 245 instances par classe pour un total de 76 225, garantissant une distribution équitable entre toutes les classes et améliorant les performances du modèle.

Pour les données de validation (357 786 d'instances) et d'évaluation (1 612 117 d'instances), nous avons appliqué un échantillonnage stratifié de ces ensembles afin d'utiliser uniquement 10% des données pour la validation. Cependant, pour les données de test, l'échantillonnage stratifié de 11 % est appliqué uniquement sur les classes majoritaires (DoS, MQTT, Benign et Recon), en conservant toutes les données de Spoofing pour l'évaluation des modèles qui est seulement 0.11% dans l'ensemble des données de test. Cette approche nous permet de minimiser le temps requis pour l'entraînement et l'inférence. Le Tableau 4.2 détaille la distribution finale du jeu de données utilisé pour notre expérimentation.

TABLEAU 4.2 : Distribution du jeu de données suivant les cinq classes.

Étiquettes	Classes	Données d'entrainement	Données de validation	Données d'évaluation
0	Benign	15 245	964	4 136
1	DoS	15 245	32 924	163 178
2	MQTT	15 245	1 315	7 008
3	Recon	15 245	496	2 817
4	Spoofing	15 245	80	1 744
	Total	**76 225**	**35 779**	**178 883**

4.1.3 L'EXPLORATION ET TRAITEMENT DES DONNÉES

Cette étape est cruciale dans la préparation des données, car elle permet d'adapter nos données de manière à ce qu'elles soient efficacement interprétées par les modèles LLM tels que BERT et ses variantes. Tous d'abord, nous avons procédé à l'encodage des étiquettes et

le formatage des données afin de préparer nos ensembles de données personnalisés pour les tâches NLP comme la classification du texte. Les étiquettes cibles ont été converties de valeurs catégorielles en valeurs numériques à l'aide de la classe LabelEncoder du module sklearn de Python.

Ensuite, nous avons formaté l'ensemble de données (Formation, Validation et Test) dans une représentation textuelle spécifique. Pour ce faire, nous avons incorporé des balises telles que **<val>** et **</val>** pour distinguer clairement les caractéristiques de leurs valeurs correspondantes. En conséquence, les ensembles de données personnalisés ont été structurés avec seulement deux colonnes : **text** et **label**. Cette approche garantit que les données sont correctement formatées et prêtes à être traitées par les LLM, facilitant ainsi une classification précise.

Dans les tâches de traitement du langage naturel (NLP) qui utilisent des modèles basés sur l'attention comme BERT, l'entrée est divisée en divers sous-mots (tokens). Cependant, cette segmentation (tokenisation) ne permet pas de discerner distinctement les caractéristiques influençant la décision du modèle. Ainsi, il est crucial de ne pas fragmenter ces caractéristiques pendant la tokenisation. Ceci permet au modèle candidat d'assurer des performances optimales et une explicabilité adéquate. Ferrag *et al.* (2024) ont utilisé la classe **ByteLevelBPETokenizer** de la bibliothèque HuggingFace Transformer (Wolf, 2019) dans leur approche. ByteLevelBPETokenizer utilise l'algorithme BPE (Byte-Pair Encoding) (Shibata *et al.*, 1999), qui remplace les paires d'octets consécutifs les plus fréquentes d'une séquence par un seul jeton (token). Bien qu'efficace pour leur cas d'utilisation, cette méthode n'est pas idéale pour préserver toutes les caractéristiques de notre ensemble de données.

Pour résoudre ce problème, nous avons utilisé la méthode **add_tokens** de la classe Tokenizer de la bibliothèque HuggingFace Transformer. Cette méthode nous permet d'étendre

le vocabulaire du Tokenizer en ajoutant des tokens personnalisés, à savoir les caractéristiques spécifiques au domaine (colonne) ou des tokens spéciaux (<val></val>). En personnalisant le Tokenizer, nous nous assurons qu'il est mieux adapté au traitement de notre ensemble de données tout en maintenant la compatibilité avec les grands modèles de langage pour les tâches de traitement du langage naturel (NLP).

Enfin, les échantillons de formation et validation ont été transformés dans un format compatible avec DistilBERT, BERT et RoBERTa. Cette étape de traitement est essentielle pour préparer les données textuelles en vue d'entrainement et d'évaluation efficaces des modèles. Lors de la tokenisation, les colonnes nécessaires suivantes ont été créées :

- **input_ids** : Identifiants de jetons générés par le Tokenizer.
- **attention_mask** : Indicateurs binaires spécifiant quelles parties de la séquence doivent être prises en compte et celles qui doivent être ignorées.
- **label** : Les étiquettes cibles associées à chaque échantillon.

Ces composants garantissent que les données s'alignent sur le format d'entrée requis par les modèles, permettant une formation et une evaluation efficace.

4.2 ENTRAÎNEMENT DES MODÈLES

Cette section décrit le processus d'entrainement de nos différents modèles pour la classification des attaques contre les dispositifs de l'IoMT. BERT, DistilBERT et RoBERTa ont été entraînés et validés respectivement à l'aide de données d'entrainement et de validation tokenisées, en utilisant la classe Trainer de la bibliothèque Transformers de Hugging Face. Les modèles ont été affinés et spécialisés pour classer cinq catégories (DoS, MQTT, Spoofing, Recon, Benign) et de distinguer avec précision ces différents types d'attaques et l'activité normale dans les environnements IoMT.

Nous avons mené des expériences en utilisant un système Ubuntu équipé de 24 cœurs physiques et 48 cœurs logiques, ainsi que de deux GPU NVIDIA RTX A6000. Voici les détails des paramètres et de l'architecture employés :

4.2.1 LLM (BERT, DISTILBERT ET ROBERTA)

Pour chaque modèle, nous avons fixé un taux d'apprentissage de 3e-5 et une taille de lot de 128, avec des étapes d'accumulation du gradient fixées à 2. L'optimisation a été réalisée à l'aide de l'optimiseur adamw_8bit avec une décroissance des poids de 0,01, et le paramètre de perte a été fixé à CrossEntropyLoss.

Pour mieux gérer les déséquilibres de classes dans les ensembles de données, nous avons utilisé une approche basée sur le calcul de poids spécifiques pour chaque classe, une méthode largement reconnue en classification supervisée. Par exemple, He & Garcia (2009) mettent en avant l'efficacité des poids de classe pour accorder une importance particulière aux classes minoritaires. Récemment, des études ont également appliqué cette approche aux grands modèles de langage (LLM), montrant son impact positif sur leurs performances (Jung & van der Plas, 2024; Jiang *et al.*, 2024).

La procédure consiste à mesurer les fréquences des classes dans le jeu de données d'entraînement, puis à calculer les poids en inversant ces fréquences, ce qui favorise les classes sous-représentées. Ces poids sont ensuite normalisés pour obtenir une distribution équilibrée. Enfin, les poids sont intégrés dans la fonction de perte du modèle pour ajuster l'apprentissage. Cette méthode renforce la capacité du modèle à gérer les déséquilibres, garantissant une classification plus équitable et robuste, en particulier pour les classes minoritaires.

Le processus d'affinement a été effectué en 10 itérations. Enfin, un rappel EarlyStoppingCallback avec une patience de 5 a été utilisé pour éviter un surajustement pendant la formation.

Ensuite, nous avons ajusté les embeddings de tokens du modèle pour refléter la taille mise à jour du vocabulaire. Les embeddings de tokens sont des représentations numériques qui capturent la signification sémantique des jetons ou tokens, tels que les mots, les sous-mots ou, dans notre cas, les caractéristiques. Pour ce faire, nous avons utilisé la méthode **resize_-token_embeddings** intégrée par le modèle. Cette étape garantit que la couche d'embedding est élargie pour inclure des vecteurs correspondant aux nouveaux tokens ajoutés, permettant ainsi au modèle de les traiter correctement. L'ajustement des embeddings est essentiel pour maintenir la compatibilité entre le tokeniser et le modèle, car chaque token du vocabulaire doit avoir un embedding correspondant. Cette adaptation préserve l'intégrité du traitement des entrées par le modèle et assure une prise en charge fluide du vocabulaire étendu.

4.2.2 FORÊTS ALÉATOIRES (RF)

Dadkhah *et al.* (2024) ont appliqué leur jeu de données à divers modèles d'apprentissage automatique (ML) et d'apprentissage profond (DL). Leurs résultats montrent que les Forêts Aléatoires (RF) obtiennent les meilleures performances. Nous avons utilisé le modèle RF avec un critère Gini, un nombre d'estimateurs égal à 100 et un random state fixé à 42.

4.2.3 RÉSEAU DE NEURONES CONVOLUTIFS (CNN)

Plus récemment, Torre *et al.* (2024) ont utilisé le même ensemble de données pour évaluer les performances de leur modèle. Dans leur étude, l'architecture employée repose sur un modèle CNN de TensorFlow Keras Sequential comprenant diverses couches. Elle

inclut une couche d'entrée, une couche convolutionnelle 1D suivie d'une couche de max pooling, puis une couche Flatten. Ensuite, une couche dense avec 128 unités et une fonction d'activation ReLU est utilisée, suivie d'une couche de sortie avec une activation softmax pour la classification. Le modèle est compilé avec la fonction de perte « categorical cross-entropy » et l'optimiseur Adam. L'architecture inclut également une couche Dropout, une couche de normalisation par lot (Batch Normalization).

En comparant notre étude à ces deux approches traditionnelles, nous pouvons démontrer la capacité de LLM à traiter les données de trafic collectées sur les appareils IoMT.

4.3 MÉTRIQUES DE PERFORMANCE

Dans les problèmes de classification en apprentissage automatique, il est essentiel de disposer de métriques fiables pour évaluer les performances d'un modèle. Parmi les plus couramment utilisées figurent l'exactitude (accuracy), la précision (precision), le rappel (recall) et le F1-score (Mariano, 2021).

- **Exactitude (Accuracy)** : Elle mesure la proportion des prédictions correctes parmi l'ensemble des prédictions effectuées. Bien qu'elle soit intuitive, cette métrique peut être trompeuse dans les cas de déséquilibre des classes, car elle privilégie les classes majoritaires. Elle est calculée mathématiquement à l'aide de l'Équation (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- **Précision (Precision)** : Elle évalue la capacité du modèle à éviter les faux positifs. Elle est calculée en utilisant la formule de l'équation (2). Cette métrique est particulièrement importante dans les scénarios dans lesquels les faux positifs ont un coût élevé, comme

dans les systèmes de détection d'anomalie ou d'attaque comme le nôtre.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

- **Rappel (Recall)** : Également appelé sensibilité, il mesure la capacité du modèle à identifier correctement les cas positifs. Il est calculé avec l'équation (3). Cette métrique est cruciale dans des contextes dans lesquels il est essentiel de minimiser les faux négatifs.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

- **F1-Score** : Elle combine la précision et le rappel en une métrique harmonisée, offrant un équilibre entre les deux. Calculée utilisant l'équation (4), elle est particulièrement utile lorsque les classes sont déséquilibrées.

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

où

— TP (Vrai Positif) est le nombre d'instances positives correctement prédites.

— TN (Vrai Négatif) est le nombre d'instances négatives correctement prédites.

— FP (Faux Positif) est le nombre d'instances qui sont en fait négatives, mais qui ont été prédites à tort comme positives. qui étaient en fait négatives, mais qui ont été incorrectement prédites comme positives.

— FN (Faux Négatif) est le nombre d'instances qui étaient positives, mais qui ont été incorrectement prédites comme négatives.

4.4 INTERPRÉTATION ET TRANSPARENCE

Pour interpréter la prise de décision du modèle, nous avons utilisé les techniques d'explicabilité présentées dans la section 2.5. Ces techniques facilitent une compréhension approfondie du comportement du modèle, applicables aux instances individuelles et à l'ensemble des données. Tout d'abord, pour expliquer le comportement du modèle à l'aide de LIME, il est nécessaire d'adapter la fonction de prédiction afin qu'elle soit compatible avec ce dernier, pour que ce dernier puisse interpréter correctement un modèle, il faut que la fonction de prédiction retourne les résultats dans un format compréhensible, tel qu'un tableau contenant les probabilités des différentes classes pour chaque échantillon (voir Algorithme 4.1).

Entrées : textes : liste de textes, batch_size : taille du lot

Output : probabilités : tableau des probabilités prédites

```
1  Fonction predict_proba(textes, batch_size) :
2      Initialiser une liste vide probs;
3      pour i de 0 à longueur(textes) par pas de batch_size faire
4          Extraire un lot de textes;
5          Encoder le lot avec le tokenizer pour générer des entrées compatibles;
6          Passer les entrées dans le modèle pour obtenir les outputs;
7          Appliquer softmax sur les outputs pour calculer les probabilités;
8          Ajouter les probabilités calculées à probs;
9      fin
10     Concaténer les éléments de probs en un seul tableau NumPy;
11     retourner probs;
```

Algorithme 4.1 : Prédiction des probabilités avec LIME

Cet algorithme est conçu pour traiter efficacement de grands ensembles de données en traitant les textes par lots. Il tokenise les textes, génère des sorties de modèle, applique la fonction softmax pour obtenir les probabilités, et renvoie les probabilités concaténées pour tous les textes d'entrée.

Pour la deuxième approche basée sur SHAP, nous avons utilisé l'algorithme 4.2, qui est une légère amélioration de l'algorithme 4.1. Cet algorithme est conçu pour prédire les probabilités des différentes classes d'étiquettes pour un lot de données d'entrée sous le format NumPy.

Entrées : data : données d'entrée, batch_size : taille du lot

Output : probabilités : tableau des probabilités prédites

```
Fonction predict_proba_shap(data, batch_size) :
    Initialiser une liste vide probs;
    pour i de 0 à longueur(data) par pas de batch_size faire
        Extraire un lot batch;
        Formater chaque échantillon du lot en textes comprenant chaque caractéristique et
          sa valeur;
        Encoder les textes avec le tokenizer;
        Passer les entrées formatées dans le modèle pour obtenir les outputs;
        Appliquer softmax sur les outputs pour calculer les probabilités;
        Ajouter les probabilités calculées à probs;
    fin
    Concaténer les éléments de probs en un seul tableau NumPy;
    retourner probs;
```

Algorithme 4.2 : Prédiction des probabilités pour SHAP

Cette fonction prend en entrée des données, les transforme en format texte (voir la section 4.1), puis les tokenise. Elle génère ensuite des sorties à partir du modèle et applique la fonction softmax pour obtenir les probabilités. Enfin, elle renvoie ces probabilités sous forme concaténée pour toutes les données d'entrée. Cette fonction est particulièrement utile pour générer des prédictions que nous pouvons utiliser avec SHAP afin d'expliquer les décisions du modèle.

Pour l'interprétation, nous avons utilisé le KernelExplainer de SHAP. Le KernelExplainer est un outil agnostique qui mesure l'importance de chaque caractéristique dans la prédiction. Cependant, cette méthode est coûteuse en termes de temps de calcul. Pour y remédier, nous avons utilisé l'algorithme K-Means, via la fonction `kmeans` de SHAP, pour regrouper les données en 50 clusters. Ce résumé des données permet de réduire la complexité des calculs tout en préservant la distribution originale des données. Enfin, les valeurs SHAP sont calculées pour les 100 premières instances du jeu de test. Elles fournissent une explication détaillée de la façon dont chaque caractéristique influence les prédictions du modèle pour ces instances. Cela permet de mieux comprendre le comportement du modèle, d'identifier les caractéristiques importantes et d'obtenir des informations sur son processus de décision.

Notre dernière approche repose sur l'utilisation des mécanismes d'attention pour identifier les caractéristiques les plus influentes dans les prédictions du modèle. L'objectif est d'analyser les poids d'attention associés à chaque caractéristique afin de mieux comprendre le processus décisionnel du modèle.

Les jetons ou tokens du texte d'entrée sont d'abord extraits, puis filtrés pour ne conserver que ceux correspondant aux caractéristiques d'intérêt. Ensuite, les poids d'attention associés à ces tokens sont également filtrés, en excluant ceux des autres tokens. Le résultat est ensuite analysé pour déterminer quelles caractéristiques ont le plus grand impact sur la prédiction.

Les valeurs d'attention sont ensuite normalisées et visualisées sous forme de heatmap (carte thermique), permettant de mettre en évidence les relations les plus influentes entre les caractéristiques. Cette visualisation améliore la transparence du modèle et facilite son interprétation en identifiant les caractéristiques clés dans le processus de prise de décision.

CHAPITRE V
RÉSULTATS

Dans ce chapitre, nous analysons les performances des modèles étudiés en tenant compte de plusieurs critères clés, notamment le temps d'entrainement, la performance globale des modèles et leurs capacités de classification face à différents types d'attaques tout en garantissant la transparence du modèle le plus performant en offrant des explications détaillées sur les facteurs influençant ses prédictions. Cette analyse approfondie met également en lumière les défis rencontrés lors des expériences. L'objectif de ce chapitre est de répondre de manière claire et détaillée aux deux questions de recherche posées dans le chapitre 1, tout en identifiant les limitations et les opportunités d'amélioration dans le contexte de la cybersécurité de l'Internet des objets médicaux (IoMT).

5.1 EFFICACITÉ DES LLM POUR DÉTECTER LES ATTAQUES IOMT

Dans cette section, nous répondons à la question suivante : les LLM pré-entraînés sur des données de trafic issues de fichiers PCAP sont-ils efficaces pour détecter et classifier les attaques contre les dispositifs IoMT ?

La motivation principale derrière l'utilisation des LLM dans la détection des attaques IoMT réside dans leur capacité à capturer des relations complexes dans de grandes quantités de données. Contrairement aux méthodes classiques comme les forets aléatoires (Random Forest), qui se basent souvent sur des règles et des heuristiques fixes, les LLM, en particulier ceux pré-entraînés sur des données de trafic réseau, peuvent offrir une compréhension plus profonde des schémas et des anomalies dans les données. Leur capacité à intégrer de manière flexible des informations contextuelles et à s'adapter aux évolutions des attaques pourrait

améliorer la précision des modèles de détection tout en nécessitant potentiellement moins d'interventions humaines et de réglages de paramètres.

Pour répondre à cette première question, nous avons commencé par préparer notre ensemble de données en suivant le processus décrit à la section 4.1. Cette étape a permis de s'assurer que les données étaient correctement préparées et structurées pour une formation et une évaluation efficaces. Ensuite, nous avons entraîné nos différents modèles (voir section 4.2), à l'aide de l'ensemble de données préparé. Enfin, nous avons procédé à une évaluation complète et une comparaison des performances des modèles à l'aide de métriques d'évaluation standard (voir la section 4.3). La perte d'entraînement est enregistrée à chaque étape pour surveiller la performance du modèle (voir la figure 5.1), montrant les courbes de perte des différents modèles LLM et de CNN.

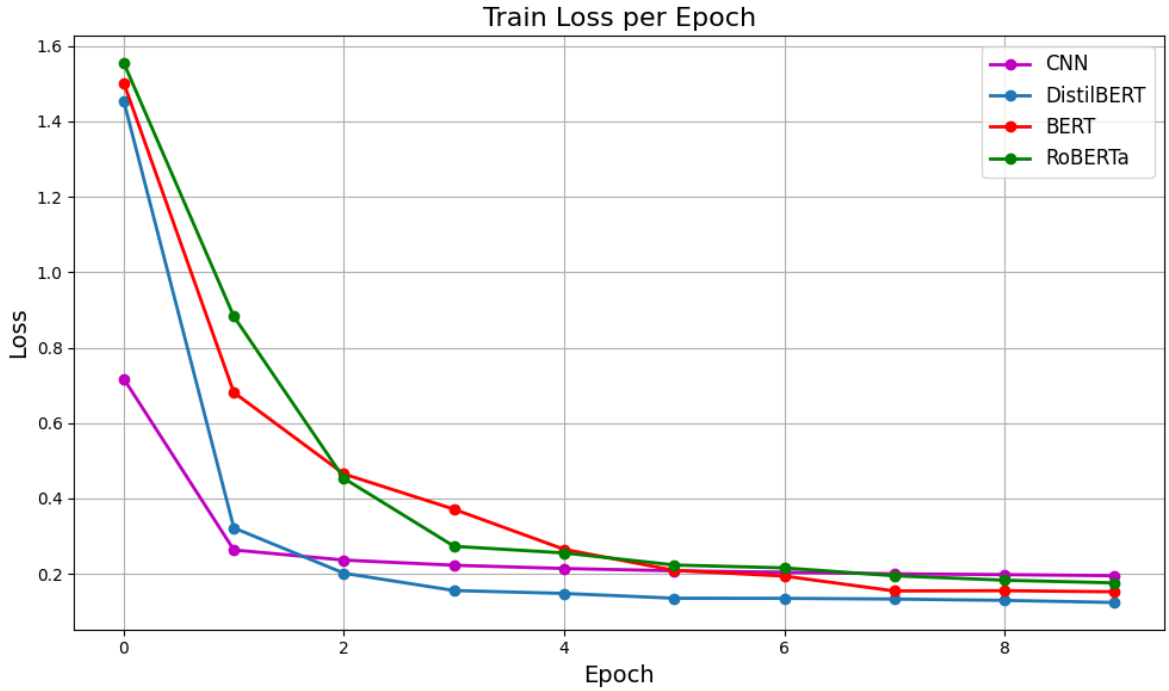

FIGURE 5.1 : Courbes de pertes en fonction des itérations d'entrainement

On peut remarquer que la perte diminuait constamment pendant de l'entraînement pour chaque modèle, et cette tendance semble se poursuivre même dans les dernières itérations. On peut soutenir que l'utilisation de plus d'époques au préalable aurait probablement conduit à une perte qui aurait continué à diminuer.

La Figure 5.2 illustre le temps d'entrainement pour chaque modèle en secondes et la Figure 5.3 montre l'utilisation du GPU par nos modèles LLM. Ces résultats fournissent des indications précieuses sur la convergence des modèles au fil des itérations.

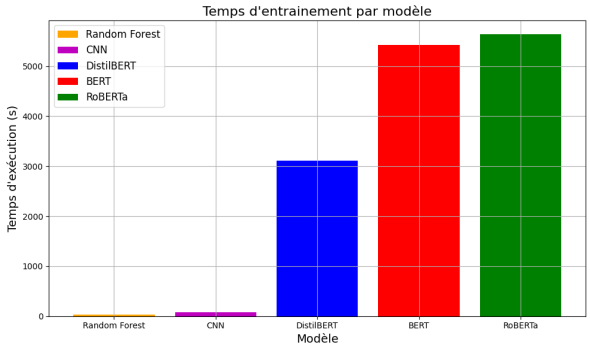

FIGURE 5.2 : Durée d'entrainement des modèles en secondes

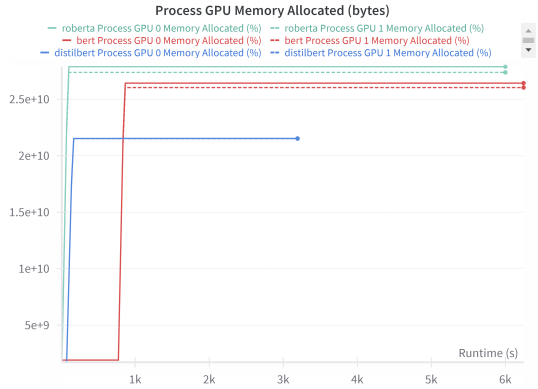

FIGURE 5.3 : l'utilisation de GPU par modèle

Les matrices de confusion des modèles BERT, DistilBERT et RoBERTa, RF et CNN sont respectivement présentées dans les figures 5.4, 5.6, 5.5, 5.7 et 5.8, offrant une vue détaillée des performances en termes de classifications correctes et erronées. Les tableaux 5.1 et 5.2 proposent un rapport complet des performances de classification pour chaque modèle, permettant une comparaison précise de leurs forces et faiblesses.

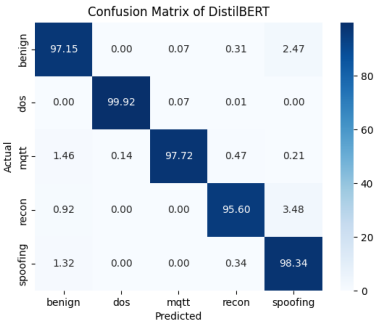

FIGURE 5.4 : Matrice de confusion de DistilBERT

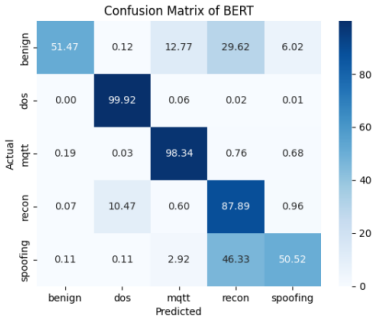

FIGURE 5.5 : Matrice de confusion de BERT

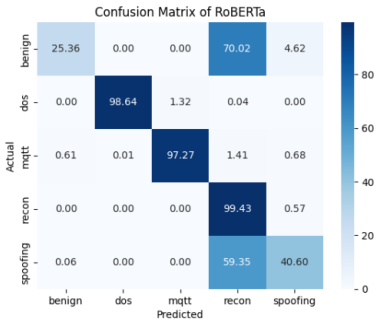

FIGURE 5.6 : Matrice de confusion de RoBERTa

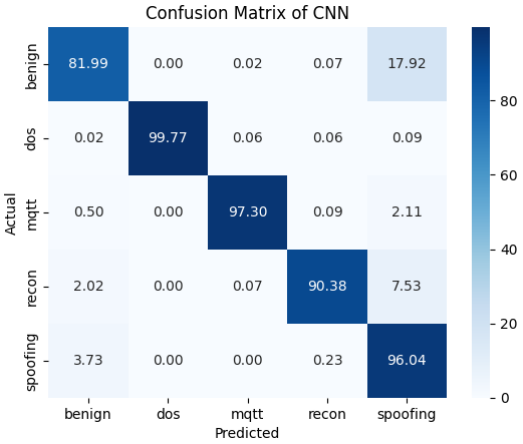

FIGURE 5.7 : Matrice de confusion de CNN

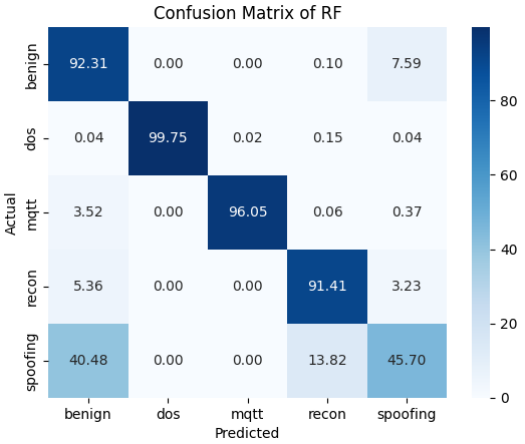

FIGURE 5.8 : Matrice de confusion de RF

Résultats : Nos résultats montrent que DistilBERT présente des performances remarquables pour toutes les catégories. La moyenne pondérée des métriques est la plus élevée parmi tous les modèles (F1 = 96,94 %). Notamment, il excelle dans la détection des attaques `DoS` avec de meilleures métriques. Un score faible est observé dans la catégorie `Spoofing` (F1 = 93,33 %). Bien que ce score reste élevé, il est significativement inférieur par rapport à d'autres catégories. Cela peut indiquer une difficulté particulière dans la distinction de cette catégorie. De son côté, BERT offre aussi d'excellents résultats pour certaines catégories comme `DoS`

TABLEAU 5.1 : Évaluation des Performances des Modèles

Model	Metric	Benign	DoS	MQTT	Recon	Spoofing
DistilBERT	Precision (%)	96.35	99.99	98.32	97.79	88.81
	Recall (%)	97.15	99.92	97.72	95.60	98.34
	F1-score (%)	96.75	99.96	98.02	96.68	93.33
BERT	Precision (%)	99.21	99.81	90.90	53.96	72.21
	Recall (%)	51.47	99.92	98.34	87.89	50.52
	F1-score (%)	67.78	99.87	94.48	66.86	59.45
RoBERTa	Precision (%)	95.89	100.00	76.04	40.61	73.37
	Recall (%)	25.36	98.64	97.27	99.43	40.60
	F1-score (%)	40.11	99.32	85.36	57.66	52.27
CNN (Torre *et al.*, 2024)	Precision (%)	94.59	100.00	98.60	95.57	57.32
	Recall (%)	81.99	99.77	97.30	90.38	96.04
	F1-score (%)	87.84	99.88	97.95	92.90	71.80
RF (Dadkhah *et al.*, 2024)	Precision (%)	76.47	100.00	99.62	83.90	61.40
	Recall (%)	92.31	99.75	96.05	91.41	45.70
	F1-score (%)	83.65	99.87	97.80	87.50	52.40

TABLEAU 5.2 : Moyennes des Performances des Modèles

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
DistilBERT	99.68	96.25	97.74	96.94
BERT	98.06	83.62	77.62	77.68
RoBERTa	96.32	77.18	72.26	66.74
CNN (Torre *et al.*, 2024)	99.07	89.72	92.56	90.47
RF (Dadkhah *et al.*, 2024)	98.77	84.28	85.04	84.24

(F1 = 99,87 %) et `MQTT` (F1 = 94,48 %), mais les performances sur les catégories `Benign`, `Recon` et `Spoofing` (67 % < F1 < 59,45 %) sont extrêmement faibles, ce qui entraîne une précision moyenne significativement plus basse que celle de DistilBERT. RoBERTa, montre des performances solides sur certaines catégories (`DoS` et `MQTT` avec F1 > 85 %). Cependant, une variabilité importante est observée entre les catégories. Les performances sur `Benign` (F1 = 40,11 %) et `Spoofing` (F1 = 52,27 %) sont faibles, ce qui réduit également sa moyenne globale.

Cette recherche montre que, bien que tous les modèles LLM gèrent efficacement la classe **DoS**, DistilBERT se distingue par ses performances globales, notamment grâce à un équilibre plus homogène entre les différentes classes. Cependant, tous les modèles indiquent une difficulté particulière à détecter les attaques **Spoofing**, un défi également souligné par les auteurs du jeu de données (Dadkhah *et al.*, 2024). En utilisant leur approche avec RF, l'expérimentation montre que le modèle se démarque par sa robustesse sur des catégories comme `DoS` et `MQTT`. La catégorie `Spoofing` est particulièrement mal classifiée (F1 = 52,40 %), ce qui réduit la performance moyenne. Ce modèle présente également une performance plus modérée sur `Benign` comparé à BERT et RoBERTa. CNN, la méthode utilisée par Torre *et al.* (2024), offre une bonne performance globale s'approchant de DistilBERT des scores F1 élevés pour les catégories `DoS`, `MQTT`, et `Recon` (F1 > 92 %). Pareil avec ce dernier, la catégorie `Spoofing` reste un point faible (F1 = 71,80 %), bien qu'elle soit meilleure que pour certains modèles comme BERT, RoBERTa ou RF.

Nos expériences révèlent également que DistilBERT est le plus rapide en termes de temps d'entraînement et d'inférence parmi les LLM, mais nécessite plus de temps que les modèles traditionnels comme illustré dans la Figure 5.2. Ainsi, DistilBERT, étant le modèle LLM le plus léger et le plus compact, surpasse le modèle BERT et sa version optimisée RoBERTa, à la fois en termes de performances et de rapidité, dans cette tâche de classification spécifique. Ces résultats soulignent l'efficacité des LLMs pour relever les défis critiques de sécurité liés à l'IoMT, en particulier pour atténuer les attaques **DoS**. DistilBERT se positionne ainsi comme une solution prometteuse pour renforcer la cybersécurité dans ce domaine.

Ainsi, la réponse à la première question est :

Oui, les LLM, en particulier DistilBERT, se révèlent efficaces pour classifier les attaques IoMT, avec de solides performances sur les attaques **DoS** et une rapidité supérieure à BERT et RoBERTa. Toutefois, la détection des attaques **Spoofing** reste un défi commun. DistilBERT se positionne comme une solution prometteuse pour améliorer la cybersécurité dans ce domaine.

5.2 IDENTIFICATION DES CARACTÉRISTIQUES IMPORTANTES EN UTILISANT DISTILBERT POUR LA DÉTECTION DES ATTAQUES DOS

Dans cette section, nous répondons à la question suivante : Ces modèles peuvent-ils identifier efficacement les caractéristiques les plus importantes contribuant à leurs décisions, et comment les techniques d'intelligence artificielle explicable (XAI) peuvent-elles être utilisées pour renforcer leur efficacité et leur pertinence dans le domaine de la cybersécurité ?

Comprendre quelles caractéristiques contribuent de manière significative aux décisions d'un modèle est essentiel pour garantir la transparence, instaurer la confiance et faciliter un déploiement efficace dans des applications concrètes. Dans le cadre de la sécurité des dispositifs IoMT, les données provenant des fichiers PCAP contiennent de nombreuses caractéristiques, mais ne jouent pas toutes un rôle égal dans l'identification et la classification des attaques. En utilisant les techniques explicables décrites dans la section 4.4, cette question explore si DistilBERT, notre modèle le plus performant, est capable d'identifier les caractéristiques les plus importantes. Cela permet non seulement de mieux comprendre ses décisions, mais aussi d'aider les administrateurs réseau à prioriser les mesures de sécurité de manière plus efficace.

Pour répondre à cette question, nous nous sommes concentrés sur l'identification des k-meilleures caractéristiques qui influencent le processus décisionnel de DistilBERT en guise de démonstration, en mettant l'accent sur la détection des attaques DoS à titre d'exemple.

Cette analyse permet de mieux comprendre comment le modèle parvient à ses prédictions et améliore son interprétabilité. Nous avons utilisé des techniques d'explicabilité (XAI) telles que LIME et SHAP pour identifier, dans un premier temps, les caractéristiques les plus importantes contribuant à la détection des attaques DoS. Ensuite, nous avons calculé la moyenne des poids d'attention générés par le modèle, puis filtré les poids correspondant aux caractéristiques pour visualiser l'attention, offrant ainsi des informations supplémentaires. Cette approche complète d'interprétabilité renforce la transparence et la confiance envers ces modèles boite noire (black box) comme DistilBERT dans les tâches de détection des attaques contre les dispositifs médicaux.

Notre première technique a consisté à expliquer la sortie d'une instance pour prédire les attaques DoS à l'aide de LIME, comme le montre la Figure 5.9. La présentation des explications fournies par LIME est organisée comme suit :

- Sur la gauche, les différentes classes (Benign, DoS, MQTT, Recon et Spoofing) sont affichées avec leurs probabilités de prédiction associées. L'explication montre que DistilBERT a classé cet échantillon comme DoS avec une probabilité de 1 (100%), tandis que toutes les autres classes ont une probabilité de 0.
- La section droite met en évidence l'impact de diverses caractéristiques sur cette classification "DoS". Le terme "not dos" indique les caractéristiques ayant un impact négatif sur la prédiction, tandis que la section "dos" montre celles qui influencent le modèle à prédire une attaque "DoS".
- Enfin, une liste des caractéristiques avec leurs valeurs correspondantes est affichée, indiquant leurs forces et directions. En orange, les caractéristiques qui favorisent la classification "DoS" sont mises en avant, tandis qu'en bleu, celles qui s'y opposent sont indiquées. Cependant, dans ce cas précis, les influences non liées au "DoS" sont presque

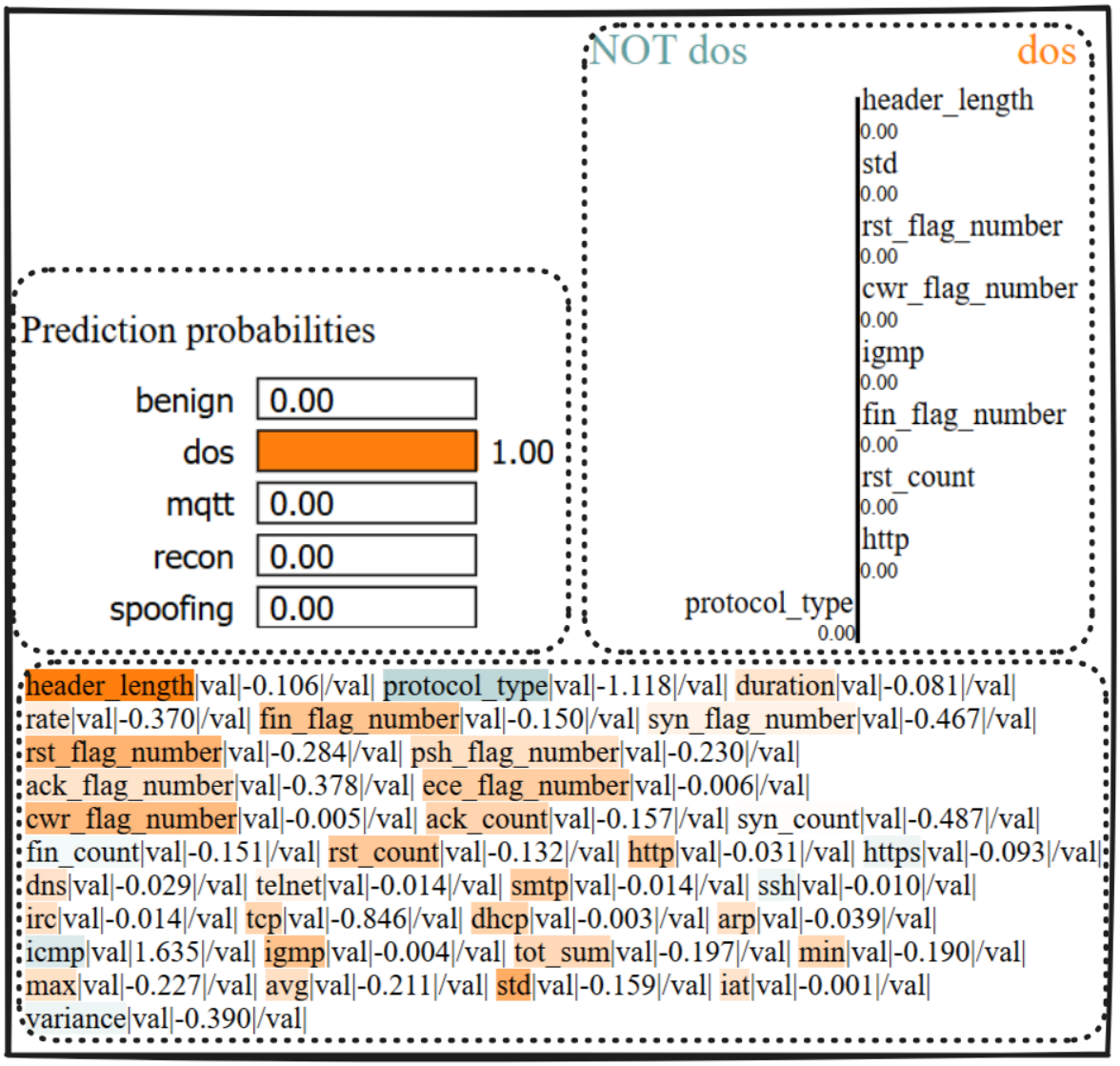

FIGURE 5.9 : Utilisation de LIME pour la prédiction des attaques DoS

nulles, ce qui montre que les caractéristiques correspondent fortement à celles d'une attaque DoS.

Résultats : Les résultats de LIME montrent que DistilBERT est très confiant dans la classification de cet échantillon comme "DoS". De plus, on observe que des caractéristiques telles que "header length", "std", "protocol type", ainsi que diverses propriétés contenant

"flag number", influencent directement le résultat de la prédiction. Cette approche permet aux professionnels de la cybersécurité de comprendre quelles caractéristiques des données de trafic contribuent aux menaces potentielles, et de se concentrer sur ces dernières pour améliorer la précision de la détection. Cette transparence renforce la confiance dans les décisions du modèle et peut orienter les ajustements des politiques de sécurité. Cependant, LIME fournit une explication locale pour des instances individuelles plutôt qu'une explication globale, ce qui peut poser un problème dans des environnements d'attaques dynamiques.

La Figure 5.10 montre un graphique récapitulatif global des décisions de DistilBERT sur toutes les classes à l'aide de SHAP, illustrant comment les différentes caractéristiques influencent les prédictions du modèle, tandis que la Figure 5.11 montre un graphe de décision pour la prédiction d'une instance d'attaque DoS.

Pour la présentation du graphique récapitulatif (Figure 5.10), l'axe vertical (y) montre le classement des caractéristiques par ordre décroissant d'importance, avec les plus influentes en haut. L'axe horizontal (x) représente les valeurs SHAP moyennes pour chaque caractéristique, qui quantifient leur impact sur les prédictions du modèle. Des valeurs plus élevées indiquent un impact plus important. Chaque classe est représentée par une couleur différente, facilitant la distinction de l'importance des caractéristiques entre les différents types d'attaques. Par exemple, la couleur bleue met en évidence la classe DoS, le violet correspond à MQTT, le rose à Spoofing, le marron au Recon, et le vert représente un comportement normal (Benign).

Résultats : Le graphique résume que la caractéristique "IAT" présente la valeur SHAP moyenne la plus élevée, ce qui signifie qu'elle a l'impact le plus important sur les décisions du modèle, en particulier pour la classe "DoS". Le graphique montre également que des caractéristiques liées aux indicateurs de drapeaux (par exemple, syn flag number, ack flag number, rst flag number) et au comptage de paquets (par exemple, rst count, syn count)

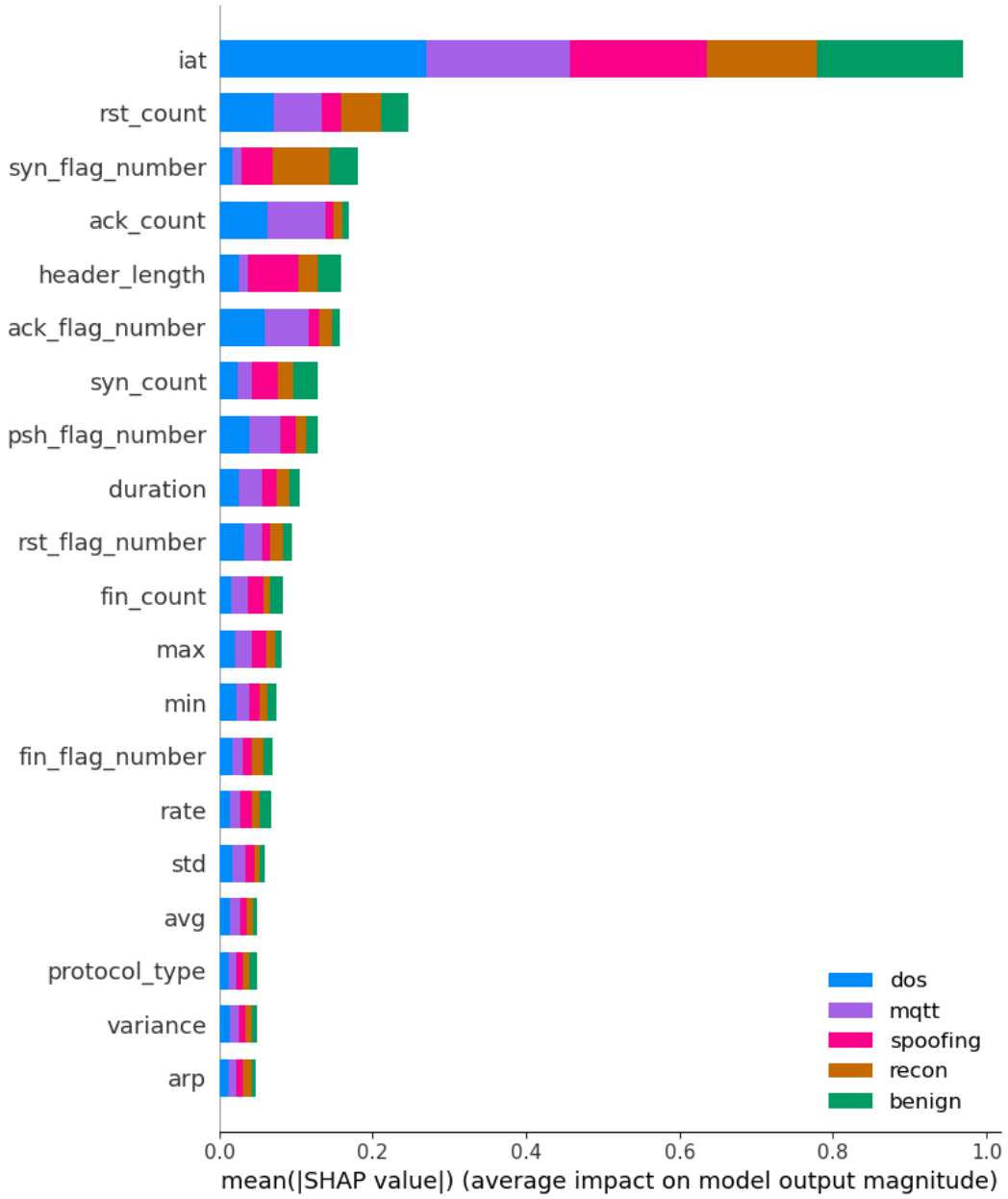

FIGURE 5.10 : Graphe de Résumé global de SHAP

jouent des rôles cruciaux. Ces caractéristiques sont souvent associées à des réinitialisations de connexion réseau et à des drapeaux SYN-ACK, indicateurs courants d'attaques de type DoS ou d'activités de reconnaissance.

Contrairement à LIME, l'utilisation du graphique récapitulatif SHAP met en lumière l'influence des différentes caractéristiques sur chaque classe, offrant une vue transparente du comportement du modèle.

De plus, SHAP propose d'autres visualisations importantes, telles que la décision plot, illustré dans la Figure 5.11. Ce type de graphique représente la contribution de chaque caractéristique à une prédiction spécifique, ici la détection de la classe DoS. Pour ce graphique, la ligne verticale indique la valeur de base générée par SHAP pour le modèle, tandis que l'axe horizontal affiche la sortie du modèle, comprise entre 0 et 1. Une valeur proche de 1 indique l'importance élevée d'une caractéristique pour la prédiction de la classe. L'axe vertical (y) liste les 20 principales caractéristiques du modèle, classées par ordre décroissant d'importance.

Résultats : La décision plot confirme que, tout comme IAT, les caractéristiques liées aux nombres de drapeaux jouent un rôle significatif et influencent fortement la décision du modèle pour détecter les attaques DoS. Ce type de graphique fournit une explication claire et interprétable de la contribution de chaque caractéristique à une classification spécifique. Ces informations sont particulièrement utiles pour comprendre et ajuster les prédictions du modèle, améliorant ainsi sa fiabilité et sa pertinence dans des scénarios concrets.

Enfin, les Figures 5.12 et 5.13 présentent deux exemples de carte thermiques issues de la matrice d'attention. Ces visualisations montrent le poids moyen de l'attention accordée aux différentes caractéristiques. Les couleurs proches du rouge indiquent une importance élevée, tandis que celles proches du bleu signalent une importance moindre. La ligne diagonale représente l'auto-attention (self-attention), c'est-à-dire l'importance qu'une caractéristique accorde à elle-même. Les éléments hors diagonale illustrent les interactions entre différentes caractéristiques, avec des points plus foncés indiquant des relations fortes entre des paires de caractéristiques.

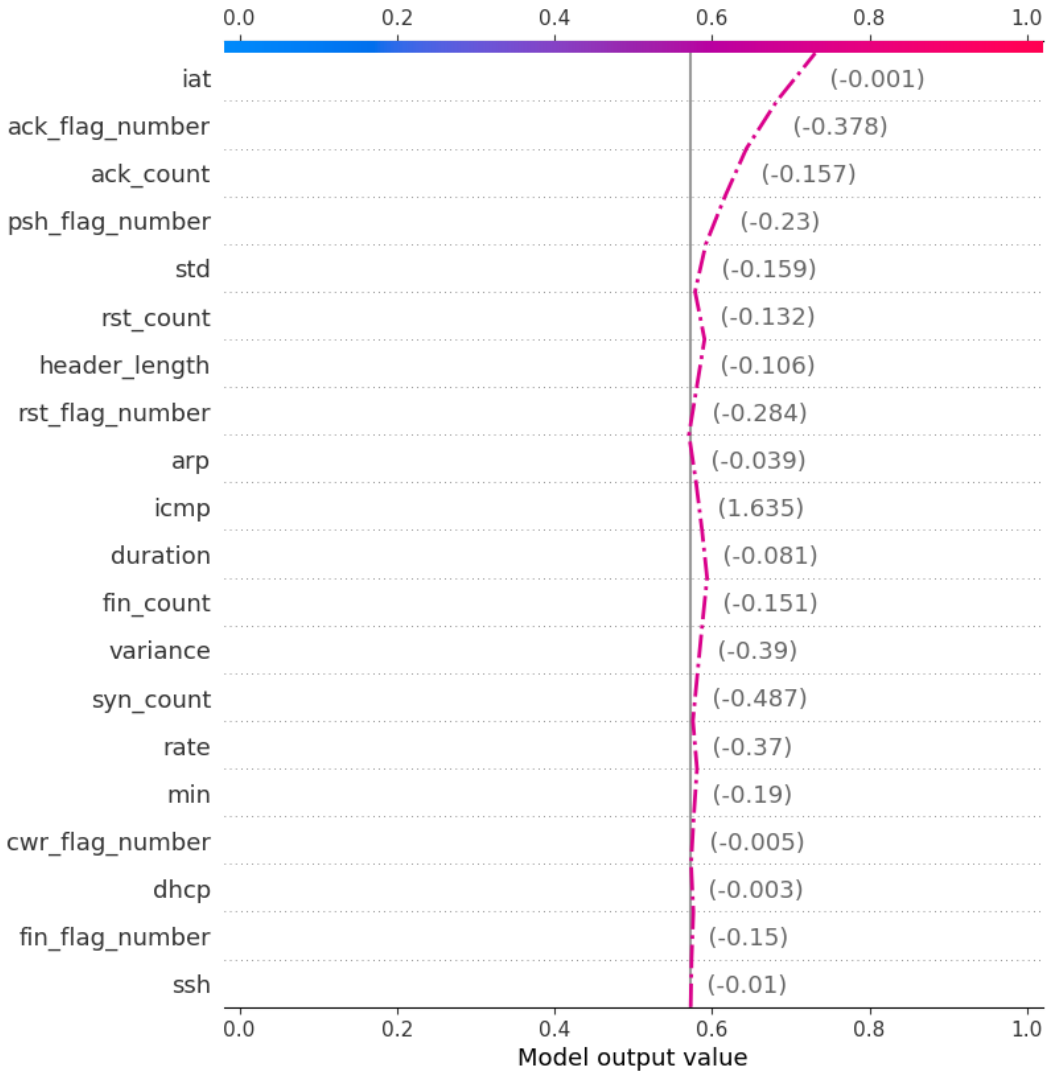

FIGURE 5.11 : Graphe de décision pour la prédiction de DoS

Resultats : Des caractéristiques comme "ack flag number" et "header length" montrent des corrélations significatives. D'autres caractéristiques, telles que "syn count", "rst count", "iat", "ack count" et "igmp", mettent également en évidence des interactions importantes. La caractéristique "syn flag number" se révèle influente en raison de ses interactions avec d'autres caractéristiques comme "fin flag number", ce qui est essentiel dans la prise de décision du modèle. Par ailleurs, des caractéristiques comme "protocol type", "header length" et "duration"

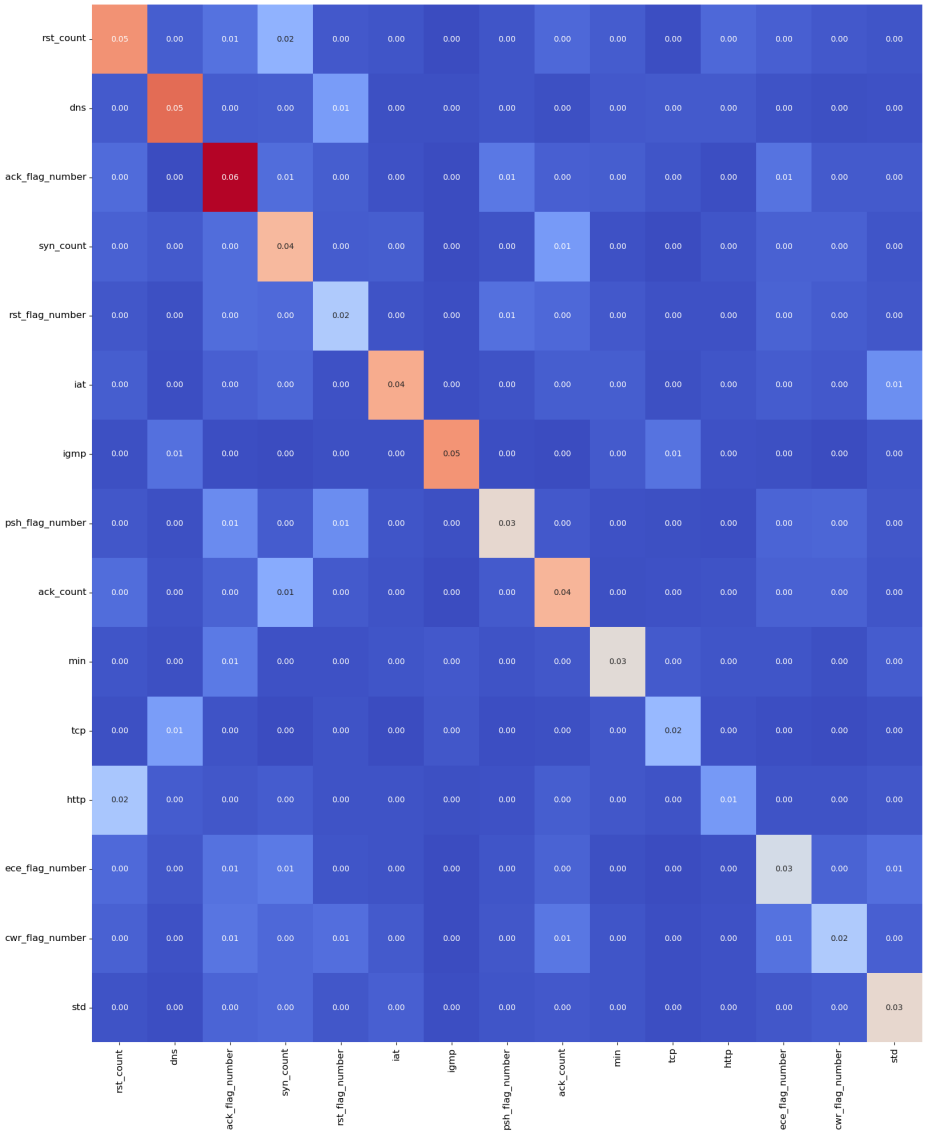

FIGURE 5.12 : Carte thermique illustrant l'attention entre les caractéristiques (a).

interagissent également entre elles, suggérant qu'elles jouent un rôle pertinent dans la décision du modèle. Ces observations permettent d'identifier les relations clés entre les caractéristiques, ce qui améliore la compréhension du fonctionnement interne du modèle et offre des indications précieuses pour renforcer l'efficacité des systèmes de détection.

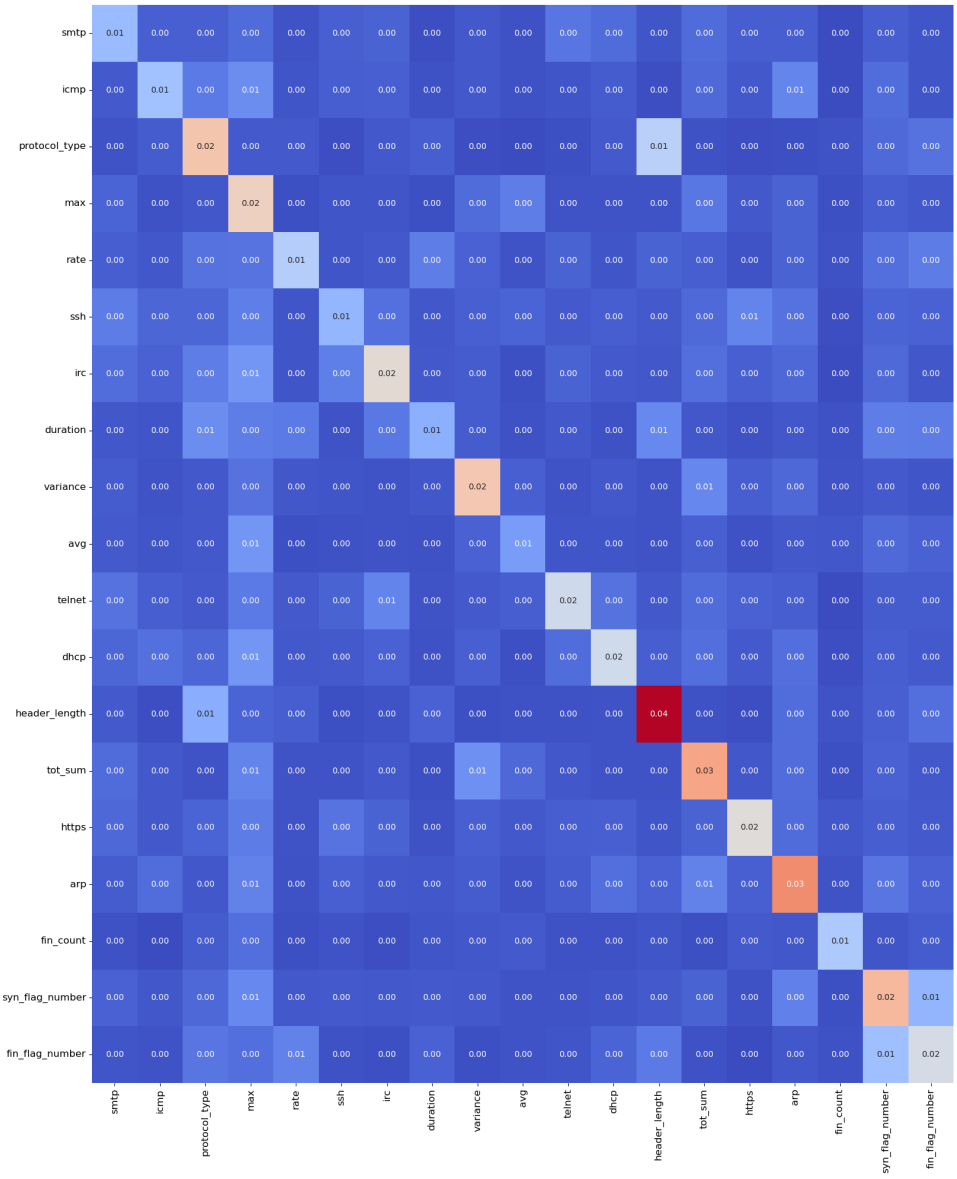

FIGURE 5.13 : Carte thermique illustrant l'attention entre les caractéristiques (b).

Le tableau 5.3 ci-dessous montre un résumé des 10 meilleures caractéristiques identifiées par les trois techniques d'explication pour la prédiction DoS décrites dans la deuxième question de recherche. La description de chaque caractéristique est présentée dans la sous-section 4.1.1.

TABLEAU 5.3 : Les 10 principales caractéristiques identifiées par différentes techniques d'explicabilité.

Rang	LIME	SHAP	Attention Based
1	Header Length	IAT	Header Length
2	Std	Ack flag number	Ack flag number
3	Rst flag number	Ack count	IGMP
4	Cwr flag number	Psh flag number	IAT
5	IGMP	Std	Ack count
6	Fin flag number	Rst count	Syn count
7	Rst count	Header Length	Rst count
8	HTTP	Rst flag number	Rst flag number
9	Protocol Type	ARP	Psh flag number
10	Ack count	ICMP	Std

En se concentrant seulement sur les 10 meilleures caractéristiques, toutes les techniques XAI mettent en avant des points communs tels que "Header Length", "RST flag number", "Std", "RST count" et "Ack count", ce qui confirme leur importance pour la prédiction du modèle. LIME identifie des caractéristiques pertinentes, mais sa nature locale peut parfois entraîner un manque de cohérence globale. En revanche, SHAP offre une analyse plus complète et diversifiée, mettant en évidence des caractéristiques essentielles telles que "IAT" (Inter-Arrival Time), cruciale pour détecter les anomalies temporelles souvent associées aux attaques DoS, ainsi que "Ack flag number", qui signalent des comportements inhabituels dans les protocoles TCP. De plus, SHAP identifie "Psh flag number", qui reflète les occurrences de paquets nécessitant un transfert immédiat des données, souvent associé à des schémas d'attaque inhabituelle. Ces résultats montrent que SHAP excelle dans l'identification des anomalies temporelles et des comportements atypiques liés aux protocoles, en faisant une approche robuste et complémentaire à LIME.

La technique de visualisation d'attention apporte une validation supplémentaire en exploitant directement les poids d'attention du modèle. Les visualisations montrent que "Header Length", "RST flag number", "Std", "RST count" et "Ack count", reçoivent une

attention significative, confirmant leur pertinence pour les attaques DoS. Elles soulignent également l'importance de caractéristiques identifiées par SHAP, telles que "IAT", "Ack flag number" et "Psh flag number", tout en mettant en lumière des éléments supplémentaires comme "Syn count", qui est particulièrement pertinent pour les attaques DoS impliquant des connexions TCP incomplètes. Cette complémentarité entre les techniques XAI et les approches basées sur l'attention renforce l'identification précise et robuste des caractéristiques critiques pour la détection des attaques DoS.

En somme, la réponse à la deuxième question est la suivante :

Oui, les techniques XAI permettent d'identifier des éléments cruciaux comme "Header Length" et "IAT" pour la détection des attaques DoS. SHAP et la visualisation de l'attention améliorent l'efficacité du modèle en fournissant une analyse complète des comportements typiques et temporels dans les protocoles, améliorant ainsi la détection proactive des menaces et la cybersécurité des systèmes IoMT.

CHAPITRE VI
DISCUSSION ET LIMITES

Dans cette étude, nous avons classé les attaques IoMT en utilisant des modèles LLM et fourni une interprétabilité approfondie du modèle le plus performant, DistilBERT, pour garantir la transparence. Dans l'expérience, nous avons utilisé des données de fichiers PCAP, qui ont été pré-traitées et normalisées. Cette section discute de nos résultats ainsi que des différentes approches utilisées pour identifier les caractéristiques contribuant à la prédiction des attaques DoS, validant ainsi notre hypothèse. De plus, nous exposons les limites et menaces potentielles pour la validité de cette recherche afin de garantir sa transparence et sa fiabilité, ainsi que les applications potentielles en cybersécurité.

6.1 UTILISATION DES MODÈLES LLM POUR DÉTECTER LES ATTAQUES IOMT À PARTIR DE DONNÉES DU TRAFIC RÉSEAUX PCAP

Les résultats présentés pour répondre à notre question 1 montrent que les modèles de type Transformers montrent des performances élevées globalement, mais avec des variations significatives entre eux. DistilBERT, en particulier, surpasse les autres grâce à son architecture optimisée et simple. L'étude met en évidence une diminution des performances des modèles LLM à mesure que leur complexité augmente, notamment en termes de nombre de couches ou de paramètres. Par exemple, Ferrag *et al.* (2024) ont utilisé l'architecture de base de BERT avec leurs propres données PCAP. Leurs résultats montrent que leur modèle, SecurityBERT, surpasse la version pré-entraînée de BERT tout en comportant moins de paramètres. Ces observations suggèrent que, pour des données de réseaux issues des fichiers PCAP, souvent bruyantes et structurées, un modèle plus léger, comme DistilBERT, peut offrir des avantages significatifs. Bien que simplifié par rapport à BERT, DistilBERT conserve une capacité robuste

à capturer des relations complexes dans les données. Cette remarque est aussi observée par rapport au modèle conventionnel CNN qui offre des performances très compétitives (F1 moyen de 90%) tout en étant généralement plus efficace en termes de ressources que ces modèles LLM. CNN donc reste une option viable pour des applications à ressources limitées. RF quant à lui, bien que performant sur certaines catégories comme DoS et MQTT, il est plus limité par sa capacité à modéliser les relations complexes dans les données séquentielles. La classe Spoofing est difficile à distinguer dans certains cas, probablement en raison de la similitude du trafic (par exemple, OS, port et recherche de vulnérabilité) avec la classe Benign et Recon comme souligné par Dadkhah *et al.* (2024). Tous les modèles sont performants à la détection d'attaque DoS. Allouzi & Khan (2021) ont analysé les menaces de sécurité IoMT en utilisant des distributions de probabilité basées sur les chaînes de Markov et le système de notation des vulnérabilités communes (CVSS). Leur étude a révélé que les attaques par déni de service (DoS) surpassaient les autres vecteurs d'attaque en fréquence et en impact. Cette étude apporte un éclaircissement sur l'importance des données d'attaque DoS dans le jeu de donnée CICIoMT2024 avec une portion de 92% des données.

Les modèles basés sur l'architecture Transformer, y compris DistilBERT, nécessitent des ressources importantes (CPU/GPU et RAM) comme illustré dans la Figure 5.3, ce qui pourrait limiter leur déploiement en temps réel sur des environnements contraints. Cependant, cette limitation met en évidence les avantages supplémentaires qu'apporte DistilBERT dans les efforts de détection.

PERFORMANCE

DistilBERT utilise le mécanisme d'attention, ce qui le différencie des modèles traditionnels tels que RF et CNN. Ce mécanisme permet à DistilBERT de capturer efficacement

les relations contextuelles entre les caractéristiques des fichiers PCAP. La capacité à détecter des motifs complexes est cruciale pour identifier des attaques sophistiquées, y compris les attaques de type 0-day à venir dans le trafic réseau. En plus, notre modèle déjà entrainé sur des données de trafic réseaux pourra être réutilisé par d'autres chercheurs pour encore améliorer ses performances avec une future base de données d'attaques. Par contre, les modèles statiques comme Random Forest nécessitent une ré-ingénierie pour faire face aux menaces émergentes, comme le souligne Yang *et al.* (2021).

INFRASTRUCTURE ET TEMPS RÉEL

Dans des environnements disposant de ressources GPU ou TPU, DistilBERT peut être utilisé en temps réel ou quasi temps réel, réduisant ainsi les contraintes liées aux ressources. Par exemple, l'utilisation d'un GPU NVIDIA RTX A6000 dans notre expérience a révélé une différence notable dans les temps d'entraînement et d'inférence par rapport au GPU T4 fourni par Kaggle. Avec le GPU T4, l'entraînement a pris environ 4 heures et 9 secondes pour 5 époques, tandis qu'avec le RTX A6000, l'entraînement pour 10 époques a été réalisé en approximativement 52 minutes.

INTERPRÉTABILITÉ ET TRANSPARENCE

Bien que les modèles traditionnels tels que le RF soit explicable grâce à des techniques comme LIME et SHAP, des études ont montré que ces méthodes peuvent parfois produire des explications trompeuses selon le modèle ou les données utilisées, notamment sur des données tabulaires. Garreau & von Luxburg (2020) soulignent que les approximations locales effectuées par LIME peuvent être inadéquates dans certains cas, en particulier lorsqu'il s'agit de modèles non linéaires ou partitionnés, comme les arbres de décision (DT). Cela met en évidence

l'importance d'interpréter les résultats de manière prudente et d'utiliser d'autres techniques pour confirmation. De plus, comme pour toute méthode, une application rigoureuse et adaptée est essentielle pour garantir leur efficacité et leur fiabilité dans des scénarios pratiques, surtout dans le domaine de la santé.

D'autre part, les réseaux de neurones convolutifs (CNN), initialement conçus pour des données structurées comme les images, peuvent être performants sur la classification des données tabulaires (comme le montre le Tableau 5.1). Cependant, cela présente des défis d'interprétabilité et ne garantit pas une explicabilité fiable sur ces types de données. Les techniques d'explicabilité comme LIME et SHAP analysent les sorties d'un modèle pour en déduire des explications locales ou globales. Ces méthodes reposent sur des distributions de probabilités pour chaque classe afin de calculer les contributions des caractéristiques. Donc, si un modèle CNN ne génère qu'une seule prédiction de classe sans probabilités associées, cela complique leur application directe. Une solution envisagée consiste à créer une fonction de sortie artificielle, en attribuant une probabilité de 1 à la classe prédite et 0 aux autres, puis en appliquant une transformation softmax pour obtenir une distribution de probabilités. Cette approche reste néanmoins problématique. Les probabilités générées artificiellement ne reflètent pas les mécanismes internes réels du modèle et risquent de produire des explications trompeuses ou biaisées. Ces ajustements détournent également LIME et SHAP de leur objectif principal, qui est d'interpréter le modèle tel qu'il est.

En revanche, les modèles basés sur les Transformers, tels que DistilBERT, offrent naturellement des probabilités pour chaque classe à la sortie, nécessitant une simple normalisation via la fonction softmax. Ils offrent également des mécanismes d'explicabilité intrinsèques, tels que la visualisation des poids d'attention, qui permettent d'identifier les caractéristiques les plus influentes dans les prédictions. Cette technique, combinée à des techniques externes comme LIME et SHAP, offrent des explications plus transparentes et réduisent le risque de

biais, renforçant ainsi la confiance dans les décisions du modèle. Par exemple, pour prédire les attaques DoS, nous observons que le modèle se concentre principalement sur les drapeaux de paquets, le volume de trafic et l'abus de protocoles. Ce qui suggère que des drapeaux TCP tels que SYN, PUSH et ACK sont essentiels pour détecter ce type d'attaque. Holmes (2013) affirme qu'un attaquant peut submerger un serveur en envoyant un grand nombre de requêtes PUSH et ACK, ce qui mène à une inondation de type PUSH ou ACK. Cette vulnérabilité est également soulignée dans l'étude de Al-Naeem (2021), qui met en évidence le rôle de ces drapeaux TCP dans la perturbation des réseaux. En outre, les résultats montrent l'importance de caractéristiques telles que le type de protocole utilisé, le nombre de SYN, le nombre d'ACK et la longueur d'en-tête. Ces caractéristiques sont particulièrement importantes pour capturer des comportements anormaux du trafic lors des attaques par inondation SYN et ACK (Salunkhe *et al.*, 2017). Le protocole IGMP peut aussi être exploité pour augmenter le trafic réseau vers la victime, permettant ainsi à l'attaquant de forger l'adresse source de la victime et de rediriger les réponses, ce qui permet d'amplifier la bande passante pour des attaques DoS à grande échelle (Sargent *et al.*, 2017). L'étude de Sharafaldin *et al.* (2018) montre également que des caractéristiques liées au flux, telles que les métriques de temps d'inter-arrivée (IAT) et la durée du flux, sont des indicateurs courants pour la détection des attaques DoS. Cette transparence renforce la confiance dans les décisions du modèle et fournit des informations exploitables pour les administrateurs réseau, leur permettant de prioriser les mesures de sécurité. En rendant les caractéristiques influençant la détection des attaques facilement compréhensibles, nous offrons un moyen d'adapter les stratégies de défense aux besoins spécifiques du réseau tout en renforçant la capacité du modèle à s'adapter aux menaces en constante évolution.

VALIDATION DE NOTRE HYPOTHÈSE

Dans l'ensemble, nous constatons que DistilBERT, entrainé sur des données de domaines spécifiques tels que les données de trafic réseau, obtient de meilleures performances tout en offrant plus d'outils d'explicabilité pour des résultats interprétables. Le modèle offre des informations approfondies sur l'importance des caractéristiques et les schémas d'attaques grâce à des techniques d'explicabilité (XAI). Il identifie des caractéristiques critiques telles que le type de protocole, la longueur d'en-tête, le nombre de SYN, le nombre d'ACK, qui sont essentielles pour détecter des comportements anormaux lors des attaques par inondation, SYN et ACK. Ces interprétations permettent aux administrateurs réseau de comprendre les processus décisionnels du modèle et de mieux appréhender les facteurs influençant les détections. Ces résultats mettent en évidence la valeur des modèles LLM dans la cybersécurité IoMT, en offrant non seulement des performances élevées, mais aussi une transparence cruciale pour les environnements de sécurité en évolution.

6.2 LIMITATIONS ET MENACES À LA VALIDITÉ

Globalement, cette recherche montre que les modèles de langage (LLM) peuvent analyser et traiter efficacement les données de trafic réseau lorsqu'elles sont correctement préparées et formatées, tout en mettant en évidence leur capacité à fournir des explications compréhensibles. Cependant, plusieurs facteurs limitants à la validité (Yin, 2012) des résultats doivent être pris en compte afin de garantir la transparence, la fiabilité et la reproductibilité.

L'objectif d'aborder ces limites dans notre étude est d'identifier les éléments susceptibles d'influencer négativement les résultats. L'objectivité d'une recherche est étroitement liée aux choix des chercheurs et aux contextes dans lesquels l'étude est menée. Voici les principaux facteurs qui pourraient influencer l'application et la robustesse de nos conclusions :

6.2.1 SÉLECTION DES MODÈLES

Le choix d'utiliser les modèles basés sur BERT parmi les modèles LLM a été motivé par des contraintes pratiques telles que la disponibilité des ressources, l'efficacité de l'entraînement et l'adaptabilité aux systèmes de détection d'intrusions réseau (NIDS). Bien que des modèles plus récents, comme LLAMA-3 (Touvron *et al.*, 2023) et Mistral (Jiang *et al.*, 2023), soient disponibles, ils nécessitent des ressources de calcul et de mémoire considérablement plus élevées, ce qui limite leur faisabilité dans ce contexte d'application.

Notre approche privilégie notamment un modèle simple, fiable et efficace, capable de fournir une analyse pertinente de l'importance des caractéristiques, malgré les contraintes liées à la tokenisation qui divise les caractéristiques en sous-mots. Par conséquent, il est probable que l'utilisation de modèles concurrents conduirait à des conclusions similaires.

6.2.2 CHOIX DE PRÉTRAITEMENT DES DONNÉES

Dans notre pipeline de prétraitement, nous avons utilisé la sélection de caractéristiques basée sur le coefficient de corrélation de Pearson (PCC) afin d'éliminer les caractéristiques fortement corrélées. Lors de la transformation des données en représentation textuelle, les valeurs ont été arrondies à trois chiffres après la virgule pour simplifier les données, car les modèles basés sur BERT sont limités à un maximum de 512 tokens en entrée. Sans cet arrondi, l'utilisation complète des décimales aurait pu entraîner une augmentation excessive du nombre de tokens après tokenisation. Ce choix visait également à améliorer l'efficacité computationnelle et l'interprétabilité du modèle.

De plus, pour remédier au déséquilibre des classes, nous avons opté pour un sous-échantillonnage aléatoire (Devi *et al.*, 2020). Cette méthode s'est avérée efficace pour équilibrer le jeu de données tout en réduisant le temps d'entraînement. Cependant, un sur-échantillonnage

aurait probablement permis d'améliorer les performances, en particulier pour les modèles comme BERT et RoBERTa, qui probablement nécessitent plus de données pour exploiter pleinement leur capacité d'apprentissage.

6.2.3 JEU DE DONNÉES ET CONDITIONS EXPÉRIMENTALES

Les résultats de cette étude reposent sur un seul jeu de données, limitant ainsi leur capacité de généralisation à d'autres environnements ou scénarios. Dans des réseaux réels, les comportements de trafic, les types d'attaques et les facteurs environnementaux peuvent varier de manière significative. Cela souligne l'importance de valider le modèle sur des jeux de données diversifiés pour capturer une variété de scénarios d'attaque, de protocoles et de comportements réseau.

Les expériences ont été réalisées sur un serveur Ubuntu équipé de deux GPU NVIDIA RTX A6000. Bien que cette configuration soit bénéfique et adaptée à nos tests et entraînements, elle ne reflète pas nécessairement les contraintes des environnements pratiques, tels que les dispositifs IoMT intégrés ou les infrastructures distribuées en temps réel.

Enfin, les performances observées peuvent dépendre des versions spécifiques des bibliothèques logicielles utilisées, telles que TensorFlow, PyTorch et HuggingFace Transformers, ainsi que des configurations des systèmes d'exploitation. Ces éléments pourraient influencer les résultats dans d'autres contextes ou avec des mises à jour logicielles ultérieures.

6.2.4 MENACES À LA SCALABILITÉ ET DE GÉNÉRALISATION

Bien que les LLM aient montré un certain potentiel pour l'analyse de ce type de données de réseau, leur scalabilité et leur capacité à se généraliser à différents environnements réseau restent à évaluer. Des facteurs tels que la topologie du réseau et le volume de trafic peuvent

affecter l'applicabilité de nos résultats à d'autres contextes, soulignant ainsi la nécessité d'études supplémentaires pour valider leur robustesse dans des scénarios variés.

CONCLUSION

Dans cette étude, nous avons exploré le potentiel des modèles LLM pour détecter les attaques ciblant des dispositifs IoMT. À partir du jeu de données CICIoMT2024, composé de données de trafic réseau PCAP, nous avons montré que les LLM, en particulier DistilBERT, peuvent être utilisés efficacement dans la cybersécurité, atteignant une performance globale de 96,94 % et excellent dans la détection des attaques de type DoS.

Bien que DistilBERT soit moins performant que certains modèles traditionnels en termes de temps d'entraînement et de ressources nécessaires, il offre une adaptabilité unique et une capacité remarquable à modéliser les relations contextuelles complexes présentes dans les données de trafic réseau. Cette capacité à identifier des schémas sophistiqués est particulièrement utile pour détecter des attaques avancées, telles que les attaques "zero day", que les modèles séquentiels peuvent ignorer. De plus, les techniques d'explicabilité, comme LIME, SHAP et la visualisation des poids d'attention, ont mis en évidence des caractéristiques critiques, notamment les drapeaux SYN et ACK, essentiels pour la détection des attaques DoS. Ces approches renforcent la transparence et permettent aux administrateurs réseau de prioriser efficacement les mesures de sécurité et de répondre aux menaces émergentes.

Cependant, des défis subsistent. Les performances sur les classes sous-représentées, comme les attaques de type "Spoofing", nécessitent des améliorations. Par ailleurs, l'optimisation du coût de calcul est cruciale pour envisager un déploiement pratique dans des environnements réels, notamment dans des systèmes intégrés à ressources limitées.

Pour l'avenir, les recherches se concentreront sur l'intégration de l'apprentissage fédéré combiné à des techniques de sur-échantillonnage et sur l'utilisation de jeux de données diversifiés. Ces approches visent à résoudre les déséquilibres des données tout en améliorant la performance et l'évolutivité. En outre, les LLM pourraient être appliqués à d'autres aspects de

la cybersécurité, tels que l'analyse des journaux de sécurité pour détecter des comportements anormaux, l'évaluation des rapports d'incidents et l'extraction d'informations exploitables pour fournir des recommandations en temps réel. Cela faciliterait une réponse rapide et efficace aux menaces, en particulier dans des secteurs critiques comme la santé.

En surmontant ces défis, DistilBERT et d'autres modèles similaires pourraient jouer un rôle majeur dans la lutte contre les cyberattaques, contribuant à renforcer la résilience des écosystèmes IoMT face à des menaces de plus en plus sophistiquées.

BIBLIOGRAPHIE

Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L. F. & Abdulkadir, S. J. (2022). Detecting cybersecurity attacks in internet of things using artificial intelligence methods : A systematic literature review. *Electronics*.

AboulEla, S., Ibrahim, N., Shehmir, S., Yadav, A. & Kashef, R. (2024). Navigating the Cyber Threat Landscape : An In-Depth Analysis of Attack Detection within IoT Ecosystems. *AI*.

Achiam, J., Adler, S., Agarwal *et al.* (2023). Gpt-4 technical report. *arXiv preprint arXiv :2303.08774*.

Achtibat, R., Hatefi, S. M. V., Dreyer, M., Jain, A., Wiegand, T., Lapuschkin, S. & Samek, W. (2024). Attnlrp : attention-aware layer-wise relevance propagation for transformers. *arXiv preprint arXiv :2402.05602*.

Akpatsa, S. K., Lei, H., Li, X., Obeng, V.-H. K. S., Martey, E. M., Addo, P. C. & Fiawoo, D. D. (2022). Online News Sentiment Classification Using DistilBERT. *Journal of Quantum Computing*.

Al-Naeem, M. A. (2021). Prediction of re-occurrences of spoofed ACK packets sent to deflate a target wireless sensor network node by DDOS. *IEEE Access*.

Aldhaheri, A., Alwahedi, F., Ferrag, M. A. & Battah, A. (2024). Deep learning for cyber threat detection in IoT networks : A review. *Internet of Things and cyber-physical systems*.

Alharbi, K. & Haq, M. A. (2024). Enhancing Disaster Response and Public Safety with Advanced Social Media Analytics and Natural Language Processing. *Engineering, Technology & Applied Science Research*.

Ali, A., Shamsuddin, S. M. & Ralescu, A. L. (2013). Classification with class imbalance problem. *Int. J. Advance Soft Compu. Appl*.

Allouzi, M. A. & Khan, J. I. (2021). Identifying and modeling security threats for IoMT edge network using Markov chain and common vulnerability scoring system (CVSS). *arXiv*

preprint arXiv :2104.11580.

Andrecut, M. (2023). A Brief Introduction to Transformers as Language Models. *Available at SSRN 4521095*.

Attanasio, G., Pastor, E., Di Bonaventura, C. & Nozza, D. (2022). ferret : a Framework for Benchmarking Explainers on Transformers. *arXiv preprint arXiv :2208.01575*.

Ayadi, O. (2019). *Analyse et étude de la sécurité des données médicales dans l'Internet des objets à partir d'une approche technologique Blockchain*.

Bahdanau, D., Cho, K. & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR*, *abs/1409.0473*.

Becerra-Suarez, F. L., Tuesta-Monteza, V. A., Mejia-Cabrera, H. I. & Arcila-Diaz, J. (2024). Performance Evaluation of Deep Learning Models for Classifying Cybersecurity Attacks in IoT Networks. Dans *Informatics*. MDPI.

Beemelmanns, T., Zahr, W. & Eckstein, L. (2023). Explainable Multi-Camera 3D Object Detection with Transformer-Based Saliency Maps. *arXiv preprint arXiv :2312.14606*.

Choudhary, S., Chatterjee, N. & Saha, S. K. (2022). Interpretation of black box nlp models : A survey. *arXiv preprint arXiv :2203.17081*.

Churcher, A., Ullah, R., Ahmad, J., Ur Rehman, S., Masood, F., Gogate, M., Alqahtani, F., Nour, B. & Buchanan, W. J. (2021). An experimental analysis of attack classification using machine learning in IoT networks. *Sensors*, *21*(2), 446.

Dadkhah, S., Neto, E. C. P., Ferreira, R., Molokwu, R. C., Sadeghi, S. & Ghorbani, A. (2024). CICIoMT2024 : Attack Vectors in Healthcare devices-A Multi-Protocol Dataset for Assessing IoMT Device Security. *preprint,2024020898*.

Devi, D., Biswas, S. K. & Purkayastha, B. (2020). A review on solution to class imbalance problem : Undersampling approaches. Dans *international conference on computational performance evaluation (ComPE)*.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019). BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding. Dans *North American Chapter of the Association for Computational Linguistics*.

Dhanushkodi, K., Shunmugiah, J. *et al.* (2024). An optimal attention PLSTM-based classification model to enhance the performance of IoMT attack detection in healthcare application. *Transactions on Emerging Telecommunications Technologies*.

Dimitrov, D. V. (2016). Medical internet of things and big data in healthcare. *Healthcare informatics research*.

Diwakar & Raj, D. (2024). DistilBERT-based Text Classification for Automated Diagnosis of Mental Health Conditions. Dans *Microbial Data Intelligence and Computational Techniques for Sustainable Computing*. Springer.

Englebert, A., Stassin, S., Nanfack, G., Mahmoudi, S. A., Siebert, X., Cornu, O. & De Vleeschouwer, C. (2023). Explaining through Transformer Input Sampling. Dans *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 806–815.

Ferrag, M. A., Ndhlovu, M., Tihanyi, N., Cordeiro, L. C., Debbah, M., Lestable, T. & Thandi, N. S. (2024). Revolutionizing cyber threat detection with large language models : A privacy-preserving bert-based lightweight model for iot/iiot devices. *IEEE Access*.

Garreau, D. & von Luxburg, U. (2020). Looking deeper into tabular LIME. *arXiv preprint arXiv :2008.11092*.

Gatouillat, A., Badr, Y., Massot, B. & Sejdić, E. (2018). Internet of medical things : A review of recent contributions dealing with cyber-physical systems in medicine. *IEEE internet of things journal*.

Geva, M., Schuster, R., Berant, J. & Levy, O. (2020). Transformer Feed-Forward Layers Are Key-Value Memories. *ArXiv*.

Ghojogh, B. & Ghodsi, A. (2020). Attention mechanism, transformers, BERT, and GPT : tutorial and survey.

Ghubaish, A., Salman, T., Zolanvari, M., Unal, D., Al-Ali, A. & Jain, R. (2020). Recent advances in the internet-of-medical-things (IoMT) systems security. *IEEE Internet of Things Journal*.

Gohel, P., Singh, P. & Mohanty, M. (2021). Explainable AI : current status and future directions. *arXiv preprint arXiv :2107.07045*.

Gunning, D. & Aha, D. (2019). DARPA's explainable artificial intelligence (XAI) program. *AI magazine*.

Hasan, M. K., Ghazal, T. M., Saeed, R. A., Pandey, B., Gohel, H., Eshmawi, A., Abdel-Khalek, S. & Alkhassawneh, H. M. (2022). A review on security threats, vulnerabilities, and counter measures of 5G enabled Internet-of-Medical-Things. *IET communications*.

He, H. & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*.

Holmes, D. (2013). Mitigating DDoS Attacks with F5 Technology.

Hou, F., Zhou, K., Li, L., Tian, Y., Li, J. & Li, J. (2022). A vulnerability detection algorithm based on transformer model. Dans *International Conference on Artificial Intelligence and Security*. Springer.

Hu, D. (2020). An introductory survey on attention mechanisms in NLP problems. Dans *Intelligent Systems and Applications : Proceedings of the 2019 Intelligent Systems Conference (IntelliSys)*.

Huang, H. & Wang, Y. (2024). SecBERT : Privacy-preserving pre-training based neural network inference system. *Neural Networks*.

Jamil, S., Jalil Piran, M. & Kwon, O.-J. (2023). A comprehensive survey of transformers for computer vision. *Drones*.

Jiang, A. Q., Sablayrolles, A., Mensch, A. *et al.* (2023). Mistral 7B. *arXiv preprint arXiv :2310.06825*.

Jiang, C., Chan, C.-m., Xue, W., Liu, Q. & Guo, Y. (2024). Importance weighting can help large language models self-improve. *arXiv preprint arXiv :2408.09849*.

Julianda, A. R., Maharani, W. *et al.* (2023). Personality Detection on Reddit Using DistilBERT. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*.

Jung, V. & van der Plas, L. (2024). Understanding the effects of language-specific class imbalance in multilingual fine-tuning. *arXiv preprint arXiv :2402.13016*.

Kagita, M. K., Thilakarathne, N., Gadekallu, T. R. & Maddikunta, P. K. R. (2022). A review on security and privacy of internet of medical things. *Intelligent internet of things for healthcare and industry*.

Khan, J. Y. (2019). Introduction to IoT systems. Dans *Internet of Things (IoT)*. Jenny Stanford Publishing.

Khan, M. M. & Alkhathami, M. (2024). Anomaly detection in IoT-based healthcare : machine learning for enhanced security. *Scientific Reports*.

Khanna, A. & Kaur, S. (2020). Internet of things (IoT), applications and challenges : a comprehensive review. *Wireless Personal Communications*.

Kobayashi, G., Kuribayashi, T., Yokoi, S. & Inui, K. (2023). Analyzing Feed-Forward Blocks in Transformers through the Lens of Attention Maps. Dans *International Conference on Learning Representations*.

Kokalj, E., Škrlj, B., Lavrač, N., Pollak, S. & Robnik-Šikonja, M. (2021). BERT meets shapley : Extending SHAP explanations to transformer-based classifiers. Dans *Proceedings of the EACL hackashop on news media content analysis and automated report generation*, pp. 16–21.

Kumar, P., Gupta, G. P. & Tripathi, R. (2021). An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Computer Communications*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*.

Lundberg, S. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv :1705.07874*.

Luo, H. & Specia, L. (2024). From understanding to utilization : A survey on explainability for large language models. *arXiv preprint arXiv :2401.12874*.

Mariano, D. (2021). Métricas de avaliação em machine learning : acurácia, sensibilidade, precisão, especificidade e F-score. *BIOINFO - Revista Brasileira de Bioinformática e Biologia Computacional*.

Mohammed, R., Rawashdeh, J. & Abdullah, M. (2020). Machine learning with oversampling and undersampling techniques : overview study and experimental results. Dans *11th international conference on information and communication syst. (ICICS)*.

Molnar, C. (2023). *Interpretable Machine Learning (Second Edition) : A Guide for Making Black Box Models Explainable*. Packt Publishing.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W. & Müller, K.-R. (2019). Layer-wise relevance propagation : an overview. *Explainable AI : interpreting, explaining and visualizing deep learning*.

Mosca, E., Szigeti, F., Tragianni, S., Gallagher, D. & Groh, G. (2022). SHAP-based explanation methods : a review for NLP interpretability. Dans *Proceedings of the 29th international conference on computational linguistics*.

Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A. Y. & Tari, Z. (2023). Explainable intrusion detection for cyber defences in the internet of things : Opportunities and solutions. *IEEE Communications Surveys & Tutorials*.

Mozafari, J., Fatemi, A. & Moradi, P. (2020). A method for answer selection using DistilBERT and important words. Dans *2020 6th International Conference on Web Research (ICWR)*. IEEE.

Petrosyan, A. (2024). Average weekly number of cyberattacks in organizations worldwide in 2022 and 2023, by industry. *Statista Report.*

Rahali, A. & Akhloufi, M. A. (2021). Malbert : Malware detection using bidirectional encoder representations from transformers. Dans *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*.

Rasool, R. U., Ahmad, H. F., Rafique, W., Qayyum, A. & Qadir, J. (2022). Security and privacy of internet of medical things : A contemporary review in the age of surveillance, botnets, and adversarial ML. *Journal of Network and Computer Applications*.

Reddy, G. P. & Kumar, Y. P. (2023). Explainable ai (xai) : Explained. Dans *2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*. IEEE.

Ribeiro, M. T., Singh, S. & Guestrin, C. (2016). " Why should i trust you ?" Explaining the predictions of any classifier. Dans *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

Rodríguez-Pérez, R. & Bajorath, J. (2019). Interpretation of compound activity predictions from complex machine learning models using local approximations and shapley values. *Journal of medicinal chemistry*.

Salunkhe, H. S., Jadhav, S. & Bhosale, V. (2017). Analysis and review of TCP SYN flood attack on network with its detection and performance metrics. *IJERT*.

Sanh, V. (2019). DistilBERT, A Distilled Version of BERT : Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv :1910.01108*.

Sargent, M., Kristoff, J., Paxson, V. & Allman, M. (2017). On the Potential Abuse of IGMP. *SIGCOMM Comput. Commun. Rev.*

Sasi, T., Lashkari, A. H., Lu, R., Xiong, P. & Iqbal, S. (2024). An Efficient Self Attention-Based 1D-CNN-LSTM Network for IoT Attack Detection and Identification Using Network Traffic. *Journal of Information and Intelligence*.

Shapley, L. S. (1953). A value for n-person games. *Contribution to the Theory of Games*, *2*.

Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A. *et al.* (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*.

Shibata, Y., Kida, T., Fukamachi, S., Takeda, M., Shinohara, A., Shinohara, T. & Arikawa, S. (1999). Byte pair encoding : A text compression scheme that accelerates pattern matching. *Springer and Berlin and Heidelberg*.

Silalahi, S., Ahmad, T. & Studiawan, H. (2022). Named entity recognition for drone forensic using BERT and distilbert. Dans *2022 International Conference on Data Science and Its Applications (ICoDSA)*, pp. 53–58. IEEE.

Srinivasan, N. (2024). Artificial Intelligence in IoT Security : Review of Advancements, Challenges, and Future Directions. *Challenges, and Future Directions (May 29, 2024)*.

Sripriyanka, G. & Mahendran, A. (2024). Securing IoMT : A Hybrid Model for DDoS Attack Detection and COVID-19 Classification. *IEEE Access*.

Sundararajan, M., Taly, A. & Yan, Q. (2017). Axiomatic attribution for deep networks. Dans *International conference on machine learning*.

Torre, D., Chennamaneni, A., Jo, J., Vyas, G. & Sabrsula, B. (2024). Towards Enhancing Privacy-Preservation of a Federated Learning CNN Intrusion Detection System in IoT : Method and Empirical Study. *ACM Transactions on Software Engineering and Methodology*.

Touvron, H., Lavril, T., Izacard, G. *et al.* (2023). Llama : Open and efficient foundation language models. *arXiv preprint arXiv :2302.13971*.

Vailshery, L. S. (2024). *Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2033*. Statista Report. Repéré le 12 OCT 2024, à https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. &

Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Vig, J. (2019). Visualizing attention in transformer-based language representation models. *arXiv preprint arXiv :1904.02679*.

Vishnu, S., Ramson, S. J. & Jegan, R. (2020). Internet of medical things (IoMT)-An overview. Dans *5th international conference on devices, circuits and systems (ICDCS)*.

Wagan, S. A., Koo, J., Siddiqui, I. F., Qureshi, N. M. F., Attique, M. & Shin, D. R. (2022). A Fuzzy-Based Duo-Secure Multi-Modal Framework for IoMT Anomaly Detection. *J. King Saud Univ. Comput. Inf. Sci.*

Wang, H., Guo, D., Wei, J. & Li, J. (2024). A method of classifying IoT devices based on attack sensitivity. *Journal of Information Security and Applications*, *82*, 103751.

Wenzel, M., Grüner, E. & Strodthoff, N. (2024). Insights into the inner workings of transformer models for protein function prediction. *Bioinformatics*.

Wolf, T. (2019). Huggingface's transformers : State-of-the-art natural language processing. *arXiv preprint arXiv :1910.03771*.

Xu, H.-C., Sun, Z., Cao, Y. & Bilal, H. (2023). A data-driven approach for intrusion and anomaly detection using automated machine learning for the Internet of Things. *Soft Computing*, *27*, 14469–14481.

Xu, Z., Fang, X. & Yang, G. (2021). Malbert : A novel pre-training method for malware detection. *Computers & Security*, *111*.

Yang, L., Manias, D. M. & Shami, A. (2021). PWPAE : An Ensemble Framework for Concept Drift Adaptation in IoT Data Streams. Dans *2021 IEEE Global Communications Conference (GLOBECOM)*.

Yin, R. K. (2012). *Applications of case study research*. sage.

Zhang, N. & Kim, J. (2023). A Survey on Attention mechanism in NLP. Dans *ICEIC*. IEEE.

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D. & Du, M. (2024). Explainability for large language models : A survey. *ACM Trans. on Intelligent Syst. & Tech.*