الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de L'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE BLIDA 1
Faculté de Technologie
Département d'Électronique

MEMOIRE DE MASTER
En Télécommunication

Spécialité : Réseaux & Télécommunications

THÈME :

Détection de l'utilisation des réseaux anonymes par le biais de l'intelligence artificielle(Darknet/Darkweb).

Réalisé par
Moshe Dayan
Diarisso Dama

Jury de Soutenance le 04/juin/2025
Mr ZAIR - Président
Mme AIT MESSAOUD - Examinateur
Mr MEHDI Merouane - Promoteur

Juin 2025

Remerciement

Nous souhaitons exprimer notre profonde gratitude à Allah, le Tout-Puissant, pour nous avoir accordé la santé, la force et la persévérance nécessaires à l'accomplissement de ce projet.

Nous sommes également infiniment reconnaissants envers nos parents pour leur amour inconditionnel, leur soutien constant et leurs encouragements. Leur confiance en nous et leurs sacrifices nous ont motivés tout au long de notre parcours académique.

Nos sincères remerciements vont à notre promoteur, Mehdi Marouane, pour son accompagnement attentif, ses conseils précieux et son soutien indéfectible tout au long de ce projet. Sa grande expertise et sa patience ont été des atouts essentiels dans la concrétisation de ce travail.

Nous tenons également à remercier chaleureusement les membres du jury pour avoir pris le temps d'évaluer et de commenter ce mémoire, apportant des perspectives enrichissantes à notre réflexion.

Enfin, nous adressons notre gratitude à toute l'équipe du département d'électronique pour leur appui précieux. Leur engagement et leurs compétences ont grandement contribué à la réussite de notre parcours universitaire.

À tous ceux qui ont, de près ou de loin, contribué à la réalisation de ce projet, nous exprimons notre profonde reconnaissance. Vos efforts et votre soutien nous ont été inestimables, et nous vous en remercions du fond du cœur

ملخص

تمثل عملية كشف حركة المرور المشفرة في الشبكة العميقة تحدياًا متزايداً في مجال الأمن السيبراني. تبحث هذه الدراسة في تطبيق التعلم العميق، باستخدام الشبكات العصبية الالتفافية (ر) والشبكات العصبية التغذوية (س) للكشف عن الحالات الشاذة. بعد مرحلة معالجة البيانات التي تضمنت استخلاص الميزات وتطبيعها، حقق نموذج س **دقة بلغت ١٢.٩٩٪** و**معدل دقة ١١.٩٩٪**، مما يؤكد فعاليته في التعرف على الأنماط المخفية. على الرغم من تحديات مثل قلة البيانات المصنفة والمطالب الحاسوبية العالية، تعزز هذه الدراسة تكامل التعلم العميق في أنظمة كشف التسلل (يض) لتحسين أمن الشبكات.

الكلمات المفتاحية: التعلم العميق، كشف حركة مرور الشبكة العميقة، أنظمة كشف التسلل، معالجة البيانات، استخلاص الميزات، ر، س، الأمن السيبراني، كشف الحالات الشاذة.

Abstract

Detecting encrypted darknet traffic is a major cybersecurity challenge. This study applies deep learning, using CNN and FNN to identify anomalies. After rigorous preprocessing, including feature extraction and normalization, FNN achieved **99.21% accuracy** and **99.11% precision**, proving its effectiveness in recognizing hidden patterns. Despite challenges such as limited labeled data and high computational demands, this research enhances IDS integration for improved network security.

Keywords: Deep learning, darknet traffic detection, IDS, dataset preprocessing, feature extraction, CNN, FNN, cybersecurity, anomaly detection.

Résumé

La détection du trafic chiffré du web profond représente un défi majeur en cybersécurité. Cette étude applique l'apprentissage profond, utilisant les réseaux de neurones convolutionnels (CNN) et feedforward (FNN) pour identifier les anomalies. Après un prétraitement rigoureux, incluant l'extraction et la normalisation des caractéristiques, FNN a atteint une **exactitude de 99,21%** et une **précision de 99,11%**, prouvant son efficacité à reconnaître des motifs cachés. Malgré des défis tels que la rareté des données étiquetées et les exigences computationnelles élevées, cette recherche améliore l'intégration des IDS pour une sécurité réseau renforcée.

Mots-clés : Apprentissage profond, détection du trafic du web profond, IDS, prétraitement des données, extraction des caractéristiques, CNN, FNN, cybersécurité, détection d'anomalies.

Liste des Acronymes et Abréviations

1D 1 Dimension (la couche de convolution 1D)

API Application Programming Interface

CNN Convolutional Neural Network

CPU Central Processing Unit

CSV Comma Separated Values

DL Deep Learning

DHT Distributed Hash Table

DT Decision Tree

FN Faux Négatif

FNN Feed-Forward Neural Network

FP Faux Positif

FPR False Positive Rate

GPU Graphic Processsing Unit

HTTP Hypertext Transfer Protocol

IA Intelligence Artificielle

I2P Invisible Internet Project

IDS Intrusion Detection System

Inf Infinite

IP Internet Protocol

IPS Intrusion Prevention System

ML Machine Learning

MLP Multiple-Layer Perceptron

NaN Not a Number

PR Precision

RAM Random Access Memory

RC Recall

ReLU Rectified Linear Unit

RF Random Forest

SMOTE Synthetic Minority Oversampling Technique

SVM Support Vector Machine

TCP Transmission Control Protocol

TPR True Positive Rate

TOR The Onion Router

URL Uniform Resource Locator

VP Vrai Positif

VPN Virtual Private Network

VN Vrai Négatif

Web World Wide Web

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

L'Internet et le World Wide Web (WWW), aussi appelé **clearnet**, regroupent les sites et services accessibles publiquement et indexés par les moteurs de recherche. À l'opposé, le **web profond** inclut des données non indexées mais accessibles, telles que les bases de données internes et les services nécessitant authentification. Parmi ces espaces, les réseaux privés appelés **darknets** constituent le **web sombre**, accessibles uniquement via des protocoles spécifiques tels que **Tor** et **I2P**, garantissant l'anonymat des utilisateurs.

Bien que les darknets offrent des usages légitimes, notamment la protection de la vie privée et la liberté d'expression, ils sont souvent exploités pour des activités illégales telles que le piratage, le commerce illicite et la cybercriminalité. Cet anonymat complique la surveillance et la détection des cybermenaces, rendant les solutions conventionnelles inefficaces face aux méthodes avancées de dissimulation de trafic utilisées par les acteurs malveillants.

L'intelligence artificielle, notamment le **Machine Learning (ML)** et l'**apprentissage profond (Deep Learning)**, constitue une réponse innovante à ces défis. Ces technologies permettent de classifier le trafic réseau et de détecter les anomalies, facilitant ainsi une identification plus précise des connexions suspectes. Pour cette étude, nous exploitons le jeu de données **CIC-Darknet2020**, conçu par le **Canadian Institute for Cybersecurity (CIC)**, qui contient plus de **80 GB** de trafic réseau capturé, incluant des millions de connexions. Cependant, le fichier **CSV** utilisé pour nos analyses est beaucoup plus compact, avec une taille d'environ **76 MB**, car il contient des données prétraitées et des caractéristiques extraites plutôt que les traces brutes du trafic réseau.

Bien que plusieurs darknets existent, tels que **I2P, Freenet et Zeronet**, notre étude se concentre exclusivement sur le trafic de **Tor et VPN**, car ces deux protocoles sont les plus largement utilisés pour l'anonymisation et la protection de la vie privée en ligne. **Tor** permet d'accéder à des services cachés et de garantir l'anonymat des communications, tandis que les **VPN** sont fréquemment employés pour masquer l'origine du trafic et contourner les restrictions géographiques. Leur adoption massive et leur impact sur la cybersécurité font de ces deux réseaux les choix les plus pertinents pour une analyse

approfondie du trafic darknet. Toutefois, le dataset inclut également du trafic **non anonymisé** provenant du **clearnet**, qui couvre des activités courantes comme la navigation sur des sites classiques, le streaming, l'envoi d'e-mails et le transfert de fichiers. Cette diversité permet d'avoir une vision globale du trafic réseau et de distinguer efficacement les connexions anonymes des connexions classiques.

Ce travail de recherche s'articule autour de quatre chapitres.

1. Le premier présente les réseaux anonymes et les défis en cybersécurité, détaillant les principes du darknet, les protocoles d'anonymisation et les enjeux sécuritaires liés à leur utilisation.
2. Le deuxième aborde les fondements théoriques de l'intelligence artificielle et les techniques de détection, explorant les bases du **Machine Learning** et du **Deep Learning**, ainsi que leurs applications à l'analyse du trafic réseau.
3. Le troisième explique la méthodologie adoptée, incluant la préparation des données, la conception des modèles et l'environnement de développement. Il décrit les étapes du prétraitement du jeu de données **CIC-Darknet2020** et l'implémentation des algorithmes dans des environnements tels que **Python, ANACONDA, TensorFlow**.
4. Enfin, le dernier chapitre analyse les résultats obtenus, évalue les performances des modèles et propose des améliorations pour optimiser la détection du trafic darknet.

Afin d'améliorer les performances des modèles et d'adapter les solutions aux besoins évolutifs de la cybersécurité, plusieurs axes sont envisagés. Une optimisation des modèles est explorée à travers des techniques avancées, telles que le **Transfer Learning**, qui permet de réutiliser des connaissances préalablement acquises pour améliorer les prédictions. L'extension des jeux de données est également envisagée, intégrant de nouveaux ensembles pour une meilleure généralisation des modèles. La détection en temps réel constitue un enjeu crucial, visant à mettre en place une approche dynamique qui permette une identification immédiate des connexions suspectes. Enfin, l'automatisation et le déploiement des technologies développées sont étudiés, avec pour objectif leur intégration dans des **systèmes de cybersécurité industriels** et des solutions de **détection des intrusions (IDS)**.

Ce projet vise à **améliorer la détection du trafic darknet** grâce à **l'intelligence artificielle**. En exploitant des techniques avancées de **Machine Learning** et de **Deep Learning**, cette recherche optimise les stratégies de cybersécurité et renforce l'identification des connexions anonymes. L'analyse du jeu de données **CIC-Darknet2020** permet d'établir un cadre méthodologique efficace pour classifier les flux réseau et prévenir les cybermenaces.

Chapitre 1

Anonymat et vie privée

1.1 Introduction

Internet est un réseau informatique mondial offrant un large éventail de services, tels que le courrier électronique et le World Wide Web (couramment appelé Web). À chaque navigation, un internaute laisse des traces numériques, permettant à d'autres de collecter des informations, ce qui soulève des risques pour les données personnelles et la vie privée.

De nombreuses entités collectent ces données, souvent à l'insu des utilisateurs. L'anonymat devient donc essentiel, permettant aux individus de s'exprimer librement, notamment dans des contextes de censure, et de naviguer sans ciblage publicitaire.

Cependant, l'anonymat complique l'identification des internautes, compromettant la transparence et facilitant des comportements criminels comme le piratage ou le harcèlement. Ce dilemme entre protection de la vie privée et sécurité soulève un débat crucial.

Pour garantir une navigation anonyme et sécurisée, des outils comme les VPN, serveurs proxy, Freenet, I2P et TOR permettent de masquer l'identité des utilisateurs. Dans ce projet, les trafics des réseaux TOR et VPN ont été exploités pour la simulation.

L'objectif principal est d'analyser et classifier ces trafics afin de mieux comprendre leurs caractéristiques et de développer des solutions équilibrant confidentialité et cybersécurité..

1.2 Couches du Web

Les sites Web que nous consultons tous les jours ne représentent qu'une infime partie de l'ensemble d'Internet. Cette petite portion est appelée le Web de surface.
Il existe une partie beaucoup plus vaste d'Internet que la plupart des gens ne rencontrent pas, appelée le Web profond et le Darknet. Imaginez qu'Internet est comme un iceberg. La partie visible au-dessus de l'eau représente le Web de surface, celui que nous utilisons quotidiennement. Mais sous l'eau se cache une immense partie invisible : le Web profond et le Darknet, bien plus vastes et inaccessibles au grand public. [13].

1.2.1 Le Web surfacique ou indexable

C'est la partie visible et accessible d'Internet, regroupant les ressources indexées par les moteurs de recherche comme Google et Bing. Elle comprend des sites web, blogs, articles et vidéos facilement trouvables. Cependant, le web surfacique ne constitue qu'une petite portion des données disponibles, souvent limité aux contenus rendus publics par leurs créateurs. Ces ressources sont généralement accessibles sans restrictions particulières [13].

1.2.2 Le Web profond ou DeepWeb

Cette partie invisible pour les moteurs de recherche représente plus de 90% d'Internet. Elle contient des bases de données privées, des archives gouvernementales, et d'autres informations sensibles utilisées par des entreprises et chercheurs pour protéger des données confidentielles. Contrairement au web surfacique, l'accès au web profond nécessite des permissions spécifiques ou des informations d'identification pour naviguer dans ces ressources non indexées [13].

1.2.3 Le darknet

Partie du web profond, le darknet regroupe des données délibérément dissimulées accessibles via des outils comme Tor, I2P ou Freenet. Ces outils garantissent l'anonymat des utilisateurs en masquant leurs adresses IP et en chiffrant leurs communications. En plus de son utilisation illégale, le darknet est également exploité à des fins légitimes, telles que la protection des droits numériques et la communication sécurisée dans les pays soumis à des restrictions gouvernementales [13].

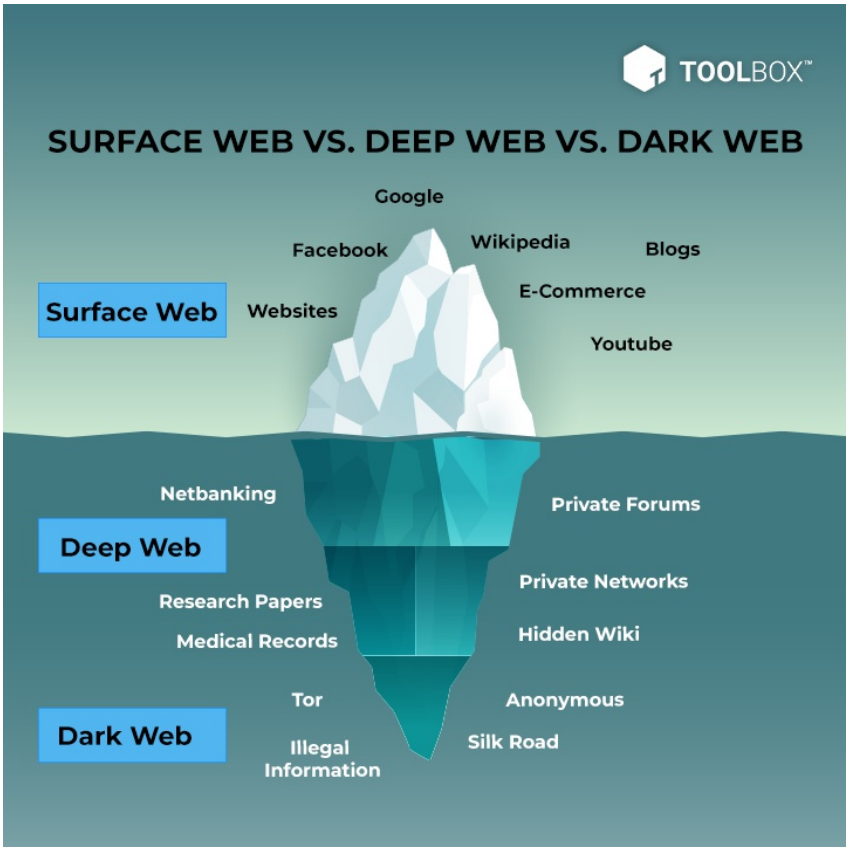

FIGURE 1.1 – Les trois couches du Web [1]

Critères	**Open Web**	**Deep Web**	**Dark Web**
Accessibilité	Accessible à tous gratuitement	Nécessite des identifiants et une URL exacte	Nécessite un navigateur spécifique (Tor) et une URL précise
Compatibilité des navigateurs	Tous les navigateurs (Chrome, Firefox, etc.)	Tous les navigateurs	Tor uniquement
Compatibilité avec les moteurs de recherche	Fonctionne avec Google, Bing, Yahoo	Fermé aux moteurs de recherche, accès uniquement par URL directe	Fermé aux moteurs de recherche, accès uniquement par Tor

TABLEAU 1.1 – Différences entre Open Web, Deep Web et Dark Web

La figure 1.2 illustre la circulation des données au sein du réseau TOR c'est à dire l'acheminement d'une requête du nœud émetteur au nœud destinataire, tandis que le tableau 1.1. met en lumière les points forts du réseau TOR et certaines de ces limites malgré qu'il soit un outil couramment utilisé dans le cadre de l'anonymat sur Web.

1.3 La vie privée sur internet

Il s'agit avant tout d'un partage volontaire d'informations que l'on souhaite préserver dans un cadre restreint, en les transmettant uniquement à des personnes de confiance. Ce type d'échange repose sur un principe de confidentialité, où l'objectif principal n'est pas forcément de dissimuler l'identité des interlocuteurs, mais plutôt d'assurer que le contenu de la conversation reste privé et protégé contre toute divulgation non souhaitée.
Dans un monde hyperconnecté où les données circulent rapidement, la protection de ces échanges devient un enjeu majeur. Que ce soit pour des discussions personnelles, des échanges professionnels ou des communications sensibles, garantir la confidentialité est essentiel afin d'éviter toute exploitation malveillante des informations partagées.
Il est important de noter que la confidentialité ne signifie pas forcément l'anonymat total. Même si l'environnement extérieur peut identifier les participants à la conversation, ce qui importe est que le contenu de leurs échanges ne soit accessible qu'aux personnes concernées. C'est pourquoi l'usage de technologies comme le chiffrement des messages ou l'emploi de plateformes sécurisées devient de plus en plus courant pour préserver la vie privée des utilisateurs. [14]

1.4 L'anonymat

1.4.1 L'anonymat sur internet

L'anonymat désigne l'état d'une personne qui choisit de rester inconnue, difficile à identifier ou à suivre. Être anonyme sur Internet ne se limite pas à utiliser un simple pseudonyme, car la véritable anonymat implique l'absence de traçabilité et le droit de contrôler la collecte des informations personnelles. Cependant, la nature des réseaux informatiques rend cette tâche particulièrement complexe.
Pour contourner cette difficulté, de nombreux internautes ont recours à des solutions techniques appelées réseaux anonymes, spécialement conçues pour garantir l'anonymat des utilisateurs. Parmi les plus connus, on trouve le VPN (Virtual Private Network), TOR (The Onion Router) et FreeNet (Invisible Internet Project).
Cependant, l'anonymat comporte aussi un côté sombre, car il peut être exploité pour mener des activités criminelles, effectuer des téléchargements illégaux, harceler ou intimider d'autres personnes en ligne. [14]

1.5 Les outils d'anonymats

Il existe de multiples solutions pour une navigation sécurisée et anonyme. C'est à dire l'utilisateur individuel de de déterminer la configuration la mieux adaptée à ses habitudes de navigation.

1.5.1 Réseau TOR

TOR, acronyme de "The Onion Routing", est un logiciel conçu pour garantir l'anonymat des utilisateurs sur Internet. Il fonctionne en faisant transiter les connexions à travers un réseau de serveurs relais chiffrés, rendant ainsi leur localisation et leur activité extrêmement difficiles à tracer. En masquant l'empreinte numérique laissée lors de la navigation, TOR permet aux internautes de rester pratiquement introuvables, protégeant ainsi leur vie privée contre la surveillance et le suivi en ligne. [15]

1.5.1.1 Principe de fonctionnement

Le réseau TOR garantit l'anonymat de ses utilisateurs grâce à un système de chiffrement multicouche, connu sous le nom de routage en oignon. Lorsqu'un utilisateur envoie des données via TOR, celles-ci transitent par plusieurs relais (des ordinateurs appartenant à d'autres utilisateurs du réseau), et sont chiffrées à chaque étape.

Son fonctionnement repose sur un processus en plusieurs étapes : le proxy TOR détermine un itinéraire, récupère les clés publiques des nœuds intermédiaires et chiffre les données successivement avec ces clés. Lorsque les données circulent dans le réseau, elles sont déchiffrées progressivement.

Le premier nœud (nœud d'entrée) enlève la première couche de chiffrement avant de transmettre les données au suivant, qui retire à son tour sa propre couche, et ainsi de suite. Ce processus se poursuit jusqu'au nœud de sortie, qui retire la dernière couche de chiffrement avant d'envoyer les données en clair vers leur destination finale.

Par défaut, TOR anonymise uniquement la navigation sur le Web. Toutefois, il peut être configuré pour fonctionner avec des logiciels tiers afin d'assurer l'anonymat dans d'autres types d'activités en ligne, comme l'envoi d'e-mails. [15]

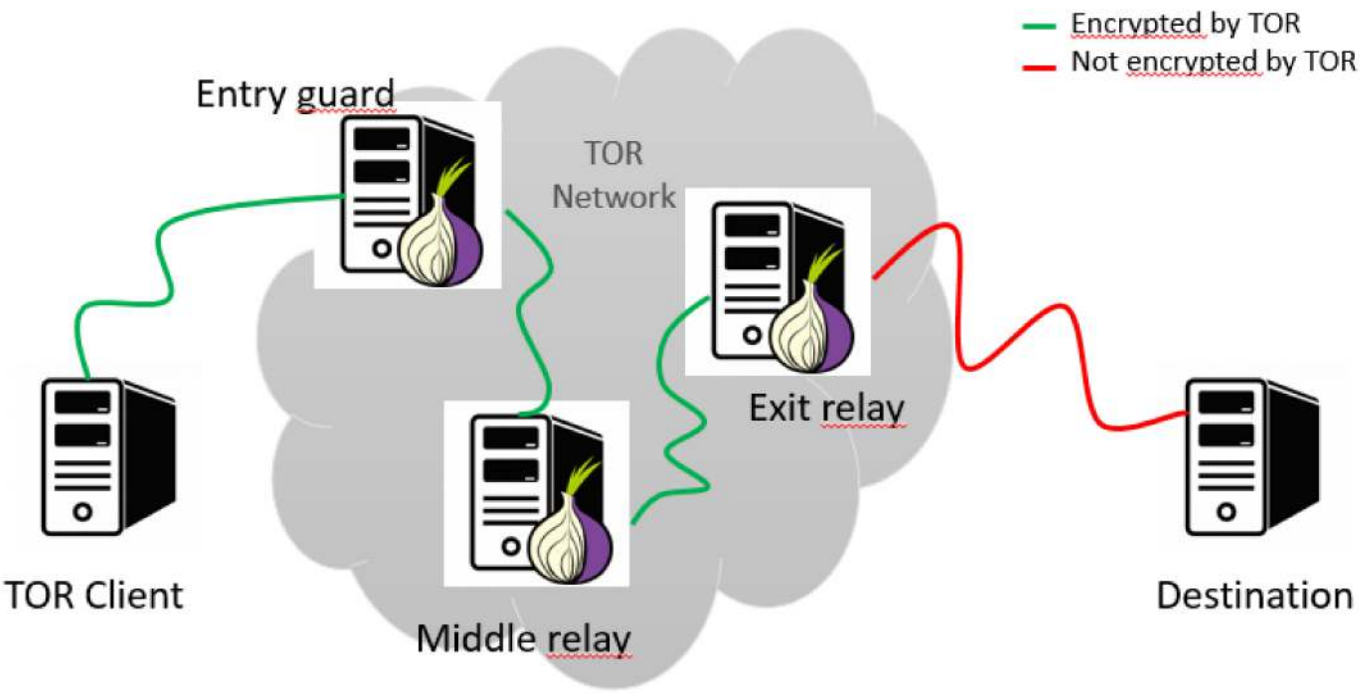

FIGURE 1.2 – Le principe de fonctionnement du réseau TOR [16]

1.5.1.2 Avantages et Inconvénients

Tor est utile pour la sécurité et la confidentialité, mais son efficacité et ses inconvénients doivent être bien compris avant adoption. [17]

Les avantages	Les inconvénients
— Personne ne peut tracer les sites que vous visitez. — Le trafic des utilisateurs est soumis à un triple chiffrement, très difficile à intercepter. — Accès facile aux pages non indexées, notamment par le biais de moteurs de recherche.	— Très lent, car les données sont acheminées via une série de relais. — Certains FAI recherchent et bloquent activement les relais Tor, ce qui rend difficile la connexion pour certains utilisateurs. — Tor étant un programme libre, son code source est accessible à tous ceux qui souhaitent le voir.

TABLEAU 1.2 – Avantages et Inconvénients du réseau TOR.

La figure 1.2 illustre la circulation des données au sein du réseau TOR c'est à dire l'acheminement d'une requête du nœud émetteur au nœud destinataire, tandis que le tableau 1.2 met en lumière les points forts du réseau TOR et certaines de ces limites malgré qu'il soit un outil couramment utilisé dans le cadre de l'anonymat sur Web.

1.6 Réseau I2P

I2P signifie « **le projet internet invisible** » dont l'objectif principal est de construire, déployer, et maintenir un réseau fournissant des communications sécurisées et anonyme. Dans I2P, les utilisateurs peuvent contrôler le niveau de sécurité de l'anonymat et la bande passante qui répond à leur besoin spécifique. L'anonymat d'I2P est assuré du fait que l'expéditeur et le destinataire ne communiquent jamais directement mais passent par plusieurs routeurs appelées **tunnels**. [18]

1.6.0.1 Principe de fonctionnement

I2P (*Invisible Internet Project*) est un réseau de communication anonyme qui permet aux utilisateurs d'échanger des données sans révéler leur identité. Son fonctionnement repose sur un acheminement du trafic via des chaînes de serveurs proxy, créant un réseau isolé où chaque ordinateur agit comme un nœud intermédiaire. Contrairement à un accès classique à Internet, toutes les connexions passent par des tunnels chiffrés utilisant des clés asymétriques, garantissant la confidentialité et empêchant toute identification directe des utilisateurs. Chaque ordinateur participe à un carnet d'adresses distribué contenant des **eepsites**, des services accessibles exclusivement via I2P, qui ne sont pas indexés par les moteurs de recherche classiques.

Le principe de routage d'I2P repose sur une technique avancée appelée **routage en ail (garlic routing)**, qui est une extension du **routage en oignon** utilisé par Tor. Ce modèle permet de regrouper plusieurs messages dans un même paquet, rendant l'analyse de trafic plus difficile et renforçant la protection des communications. Contrairement à Tor, qui permet un accès anonyme aux sites Internet classiques via des relais publics, I2P est conçu principalement pour l'anonymisation des échanges internes au réseau, facilitant ainsi les communications peer-to-peer, le partage de fichiers et l'hébergement de services cachés. Toutefois, bien que ses mécanismes renforcent l'anonymat, des analyses avancées de trafic peuvent parfois révéler la présence d'un utilisateur sur le réseau, sans pour autant compromettre son identité ou le contenu de ses échanges. [18]

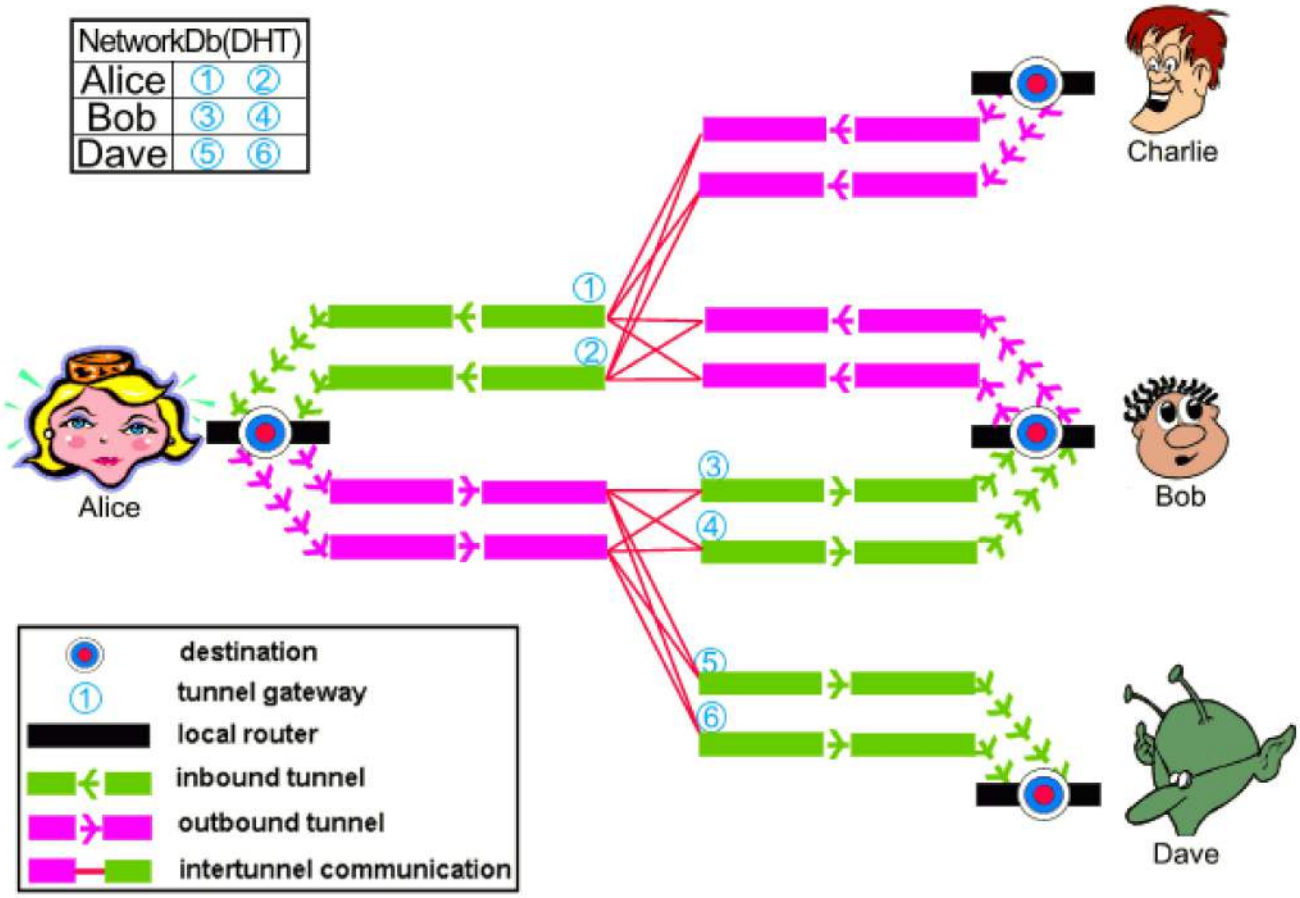

FIGURE 1.3 – Le principe de fonctionnement du réseau I2P [2]

La **figure 1.3** illustre un système de routage sécurisé utilisé dans les réseaux anonymes. **Les tunnels entrants et sortants** masquent le trafic en le chiffrant à plusieurs niveaux avant qu'il n'atteigne sa destination. **Une base de données décentralisée (DHT)** gère l'enregistrement et la recherche des nœuds sans dépendre d'une autorité centrale. **Les passerelles de tunnel** contrôlent l'entrée et la sortie des paquets chiffrés, garantissant une transmission sécurisée. **Les routeurs** dirigent le trafic de manière anonyme, empêchant l'identification des utilisateurs. La **légende** explique les éléments clés, tels que les nœuds de confiance et les chemins chiffrés, mettant en évidence un système conçu pour la confidentialité et la résistance à la surveillance. Le tableau 1.3. montre les atouts et les incommodité de ce mécanisme

1.6.0.2 Avantages et Inconvénients

L'I2P est un excellent choix pour ceux qui recherchent un haut niveau de sécurité et d'anonymat en ligne. Il offre une protection efficace contre la surveillance et le suivi des communications. Cependant, comme toute technologie, il n'est pas exempt de limitations et de défis.

Avant d'opter pour ce réseau, il est essentiel d'en comprendre les forces et les faiblesses. Voici donc un aperçu des principaux avantages qu'il propose, ainsi que des inconvénients à prendre en compte pour une utilisation optimale. [19]

Les avantages	Les inconvénients
— Rendre l'analyse du trafic très difficile. — Compatibilité avec les navigateurs les plus importants. — Utilisation possible dans toutes les activités effectuées sur le Web.	— Pas d'anonymat parfait lors de l'utilisation du Web de surface pour visiter des sites indexés. — Processus d'installation et paramètres complexes pour les utilisateurs novices.

Tableau 1.3 – Avantages et Inconvénients du réseau I2P.

La figure 1.3 démontre le procédé d'échange entre l'expéditeur et le récepteur au milieu du réseau I2P et le tableau 1.3. montre les atouts et les incommodité de ce mécanisme.

1.7 VPN

1.7.0.1 Définition du VPN

Les réseaux privés virtuels, souvent appelés VPN pour Virtual Privat Network, offrent à un client VPN une extension du réseau privé de l'entreprise à travers un support public comme internet.
Après activation du VPN, un tunnel sécurisé est créé entre vous et le réseau internet. Par conséquent, les informations qui y transitent seront cryptées et que l'activation se fait en se connectant à un serveur VPN distant. Ainsi, vous obtiendrez une nouvelle adresse IP usurpée et votre adresse IP sera bloquée. [15]

.

1.7.0.2 Principe de fonctionnement

L'utilisateur se connecte au fournisseur VPN à travers une connexion Internet. Ce dernier crypte toutes les informations stockées ou envoyées sur le réseau. Les connexions VPN permettent également aux internautes d'accéder à du contenu qui ne serait autrement pas accessible à leur emplacement. Les connexions VPN aident les utilisateurs à masquer leurs adresses IP.

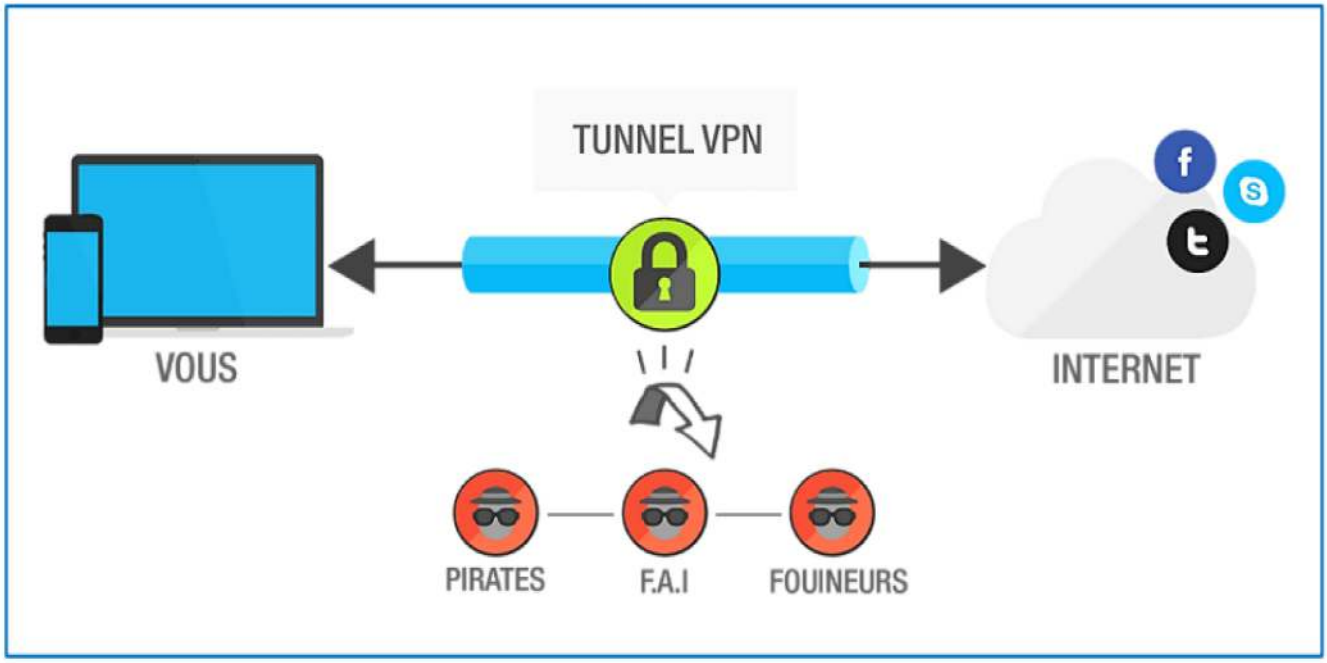

FIGURE 1.4 – Le principe de fonctionnement d'un VPN [3]

1.7.0.3 Avantages et Inconvénients

Un réseau privé virtuel (VPN) est un outil clé pour renforcer la sécurité en ligne et protéger les communications et données sensibles d'une organisation. Bien configuré, il améliore la confidentialité et sécurise les échanges numériques. Voici une synthèse des principaux avantages et limites du VPN pour évaluer son utilité et décider de son intégration pour votre entreprise. [20]

La figure 1.4 schématise les échanges entre expéditeur et récepteur via un réseau VPN, tandis que le tableau 1.4. montre les points forts et les limites d'un VPN

Tableau 1.4 – Avantages et Inconvénients du VPN.

Les avantages	Les inconvénients
— Maintient l'anonymat sur Internet. — Protège contre toute forme de surveillance et d'espionnage en ligne.	— Ralentie la connexion Internet et la rend instable. — Sous-traiter ses données à un acteur externe inconnu.

1.8 Navigation privée

La navigation privée empêche simplement que votre poste conserve les traces de vos activités en ligne. Elle supprime automatiquement les fichiers temporaires, les downloades, les mots de passe et l'historique des plateformes visitées. Tous les navigateurs proposent un mode de surf privé. Leur fonctionnement varie souvent d'une structure à une autre [21]

Figure 1.5 – Navigation privée

1.9 FreeNet

Freenet est un réseau peer-to-peer (P2P) anonyme et décentralisé qui fonctionne sur Internet. Freenet n'a pas de serveurs réguliers ni de sites Web hébergés. En fait, l'ordinateur de chaque utilisateur du projet Freenet est un serveur qui stocke certaines informations du réseau.
Essentiellement, Freenet est un grand référentiel de données et une partie du Dark Web où les utilisateurs téléchargent leurs données, qui deviennent ensuite disponibles pour tous les autres utilisateurs du réseau. Il n'est pas possible de se connecter à des services comme Facebook ou Google avec FreeNet. [22]

1.10 Difficultés techniques pour surveiller et détecter ces activités

Identifier les activités illégales sur le Darknet est une tâche ardue à cause de divers éléments techniques et fonctionnels.

L'analyse des activités illicites sur le Darknet est entravée par plusieurs défis majeurs : un **anonymat renforcé** grâce aux réseaux overlay (Tor, I2P, Freenet) et à la cryptographie de bout en bout ; **une évolution rapide des techniques** employées par les cybercriminels, incluant les marchés éphémères et l'obfuscation du trafic ; **un manque de données labellisées** et des biais dans les datasets disponibles ; **la complexité du trafic réseau**, caractérisé par un volume élevé et un chiffrement rendant l'analyse de contenu impossible sans décryptage.

De plus, la détection des activités illicites est confrontée au problème des **faux positifs** et **faux négatifs** : des activités légitimes peuvent être identifiées à tort comme illicites (faux positifs), tandis que des activités illicites sophistiquées peuvent passer inaperçues (faux négatifs), ce qui complique considérablement l'efficacité des systèmes de surveillance. Enfin, des aspects juridiques et éthiques complexes liés au respect de la vie privée et aux juridictions multiples viennent s'ajouter aux difficultés rencontrées.

1.11 Perspectives prometteuses de l'intelligence artificielle

L'intelligence artificielle (IA) pourrait apporter des solutions pour mieux détecter les activités illégales sur le Darknet, malgré les difficultés rencontrées.
L'intelligence artificielle (IA) révolutionne la cybersécurité grâce à sa capacité d'analyse de grandes quantités de données en temps réel, permettant la détection de patterns complexes, même dans les données chiffrées, grâce aux modèles de machine learning et de deep learning, son adaptabilité face aux nouvelles techniques de cybercriminalité, et son automatisation de la surveillance et de la détection, allégeant ainsi la charge de travail des analystes.

1.12 Conclusion

L'expansion des activités illicites sur le Darknet constitue une menace croissante pour la cybersécurité. L'anonymat renforcé, le chiffrement des communications et la complexité du trafic réseau posent des obstacles considérables à la détection et à la lutte contre ces activités. Cependant, l'intelligence artificielle offre des perspectives prometteuses pour surmonter ces défis. Grâce à ses capacités d'analyse avancée et d'apprentissage automatique, l'IA permet de traiter et d'interpréter des volumes massifs de données, d'identifier des schémas cachés et de détecter des anomalies subtiles. Le chapitre suivant explorera en détail comment ces techniques d'IA peuvent être déployées de manière concrète pour améliorer la détection et la prévention des activités illicites sur le Darknet.

Chapitre 2

Fondements de l'intelligence artificielle

2.1 Introduction

Le Darknet, espace numérique volontairement dissimulé et difficile d'accès, constitue aujourd'hui un défi majeur en cybersécurité. Son architecture décentralisée et son chiffre avancé assurent une protection légitime à certains utilisateurs, mais permettent également des activités illicites telles que la cybercriminalité, les trafics ou d'autres menaces exploitant dans l'ombre. Face à cette dualité, les méthodes classiques de surveillance peinent à faire face, dépassées par l'ampleur des données et la sophistication des techniques d'anonymisation des employés.

Dans ce contexte, l'intelligence artificielle (IA) apparaît comme une solution prometteuse. Grâce à sa capacité à identifier des schémas complexes, à analyser des données variées et à s'adapter en temps réel, elle ouvre de nouvelles perspectives pour détecter et contrer les activités malveillantes sur le Darknet. Ce chapitre examine comment l'IA peut être mobilisée pour relever ce défi.

Nous commençons par exposer les bases théoriques de l'intelligence artificielle applicables à cette problématique dans la section « Fondements de l'intelligence artificielle ». Ensuite, nous aborderons les méthodes de collecte et de prétraitement des données spécifiques au Darknet dans la partie « Collecte et prétraitement des données ». Enfin, nous présentons les modèles d'IA les plus adaptés, tels que les réseaux de neurones et le traitement du langage naturel, en illustrant leur application concrète à la détection d'activités suspectes dans la section « Modèles d'IA adaptés ».

L'objectif de ce chapitre est de démontrer comment l'IA, en alliant puissance analytique et adaptabilité, peut devenir un outil clé pour lever l'anonymat du Darknet, tout en mettant en lumière ses limites et les précautions indispensables à son utilisation.

2.2 Définition de l'intelligence artificielle

L'intelligence artificielle est un domaine scientifique qui cherche à développer des ordinateurs et des machines capables de raisonner, d'apprendre et d'agir de manière autonome, en réalisant des tâches qui nécessiteraient normalement l'intelligence humaine ou en traitant des volumes de données bien au-delà des capacités d'analyse humaine.
Ce champ d'étude est vaste et multidisciplinaire, englobant des domaines tels que l'informatique, l'analyse de données, les statistiques, l'ingénierie matérielle et logicielle, la linguistique, les neurosciences, ainsi que des disciplines plus théoriques comme la philosophie et la psychologie.
Sur le plan opérationnel, notamment en entreprise, l'IA regroupe un ensemble de technologies reposant principalement sur le machine learning et le deep learning. Ces technologies sont utilisées pour l'analyse et la modélisation de données, la prédiction et la prévision, la classification d'objets, le traitement du langage naturel, la recommandation de contenus et la récupération intelligente d'informations. [23]

2.3 Fonctionnement

Bien que les approches varient selon les techniques d'IA, elles reposent toutes sur un principe fondamental : l'exploitation des données. Les systèmes d'IA se perfectionnent en traitant d'importants volumes de données, en détectant des tendances et des corrélations souvent imperceptibles pour les humains. Ce processus d'apprentissage repose généralement sur des algorithmes, qui sont des ensembles de règles ou d'instructions guidant l'analyse et la prise de décision. Dans le domaine du machine learning (apprentissage automatique), une branche majeure de l'IA, ces algorithmes sont entraînés sur des données, qu'elles soient étiquetées ou non, afin de prédire des résultats ou de classer des informations.
Le deep learning, une approche plus avancée, s'appuie sur des réseaux de neurones artificiels à plusieurs couches, reproduisant en partie le fonctionnement du cerveau humain pour traiter et interpréter des données complexes. Grâce à un apprentissage continu et une adaptation progressive, les systèmes d'IA deviennent de plus en plus performants dans diverses tâches, allant de la reconnaissance d'images à la traduction automatique, et bien au-delà. [23]

2.4 Méthodes d'apprentissage d'une IA

De la même manière qu'il existe différents modèles d'intelligence artificielle, les méthodes d'apprentissage se diversifient en fonction des besoins et des objectifs à atteindre.

2.4.1 Apprentissage supervisé

Également appelé machine learning classique, l'apprentissage supervisé repose sur l'intervention d'un expert humain pour étiqueter les données d'entraînement. Par exemple, un data scientist qui développe un modèle de reconnaissance d'images destiné à distinguer les chiens des chats doit attribuer à chaque image une étiquette correspondante (« chien » ou « chat ») ainsi que des caractéristiques clés, comme la taille, la forme ou la texture de la fourrure. Au cours de l'entraînement, le modèle s'appuie sur ces étiquettes pour identifier et apprendre les traits visuels distinctifs de chaque catégorie. [24]

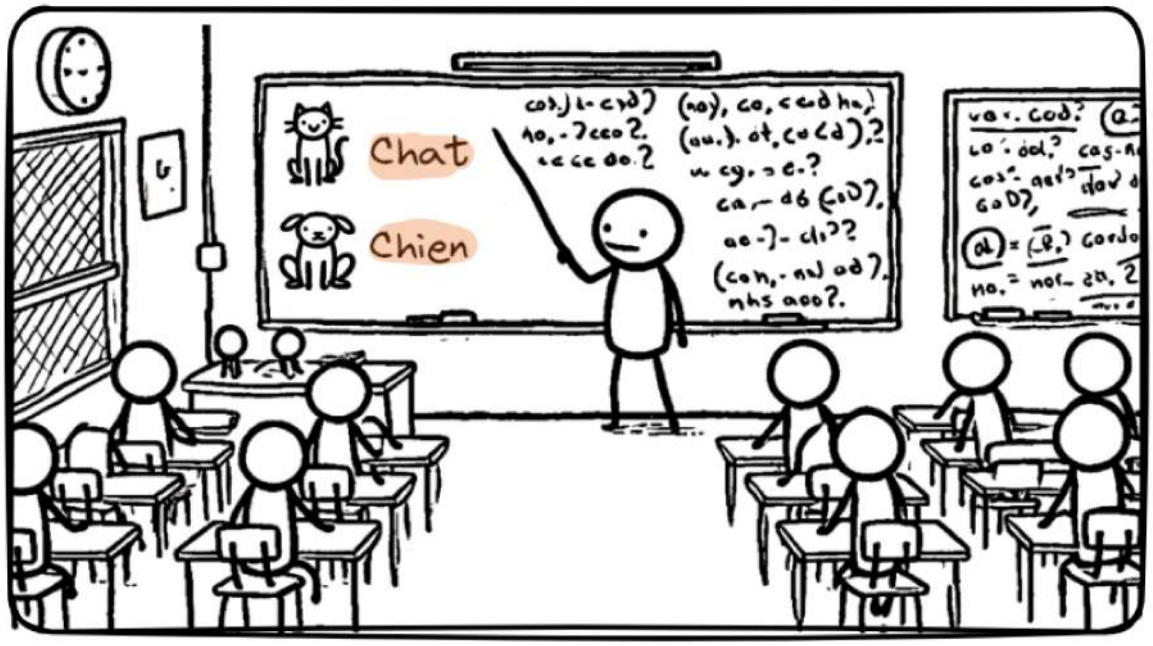

Figure 2.1 – apprentissage-supervisé [4]

La machine analyse les exemples qui lui sont fournis et, grâce à un mécanisme d'auto-évaluation et d'amélioration continue que nous découvrirons dans ce mémoire, elle apprend progressivement à exécuter la tâche qui lui est assignée.
En fonction de la nature de la tâche à accomplir, on distingue deux types de problèmes :

2.4.1.1 Les régressions

La régression estime les paramètres d'un modèle mathématique reliant une variable cible Y à des prédicteurs X1, X2..., permettant de prédire des quantités comme prix d'une maison, durée d'un trajet, température, poids, etc.

FIGURE 2.2 – régression [4]

2.4.1.2 Les classifications

La classification est une technique qui associe une observation à une classe selon ses caractéristiques, permettant de prédire des catégories comme un animal, état de santé, couleur, type de véhicule, etc.

FIGURE 2.3 – classification [4]

2.4.2 Apprentissage non-supervisé

L'apprentissage non supervisé est une autre approche majeure du machine learning. Contrairement à l'apprentissage supervisé, cette méthode permet à la machine d'identifier uniquement des structures et des motifs dans les données, sans lui imposer une correspondance prédéfinie entre les entrées et les sorties (X → y).
Par exemple, si nous lui présentons un ensemble d'images d'animaux sans préciser s'il s'agit de chats ou de chiens, l'algorithme devra regrouper ces images en fonction de leurs similitudes, sans instruction explicite sur les catégories à attribuer. Autrement dit, seuls les attributs X sont fournis, sans indication de la sortie y attendue.

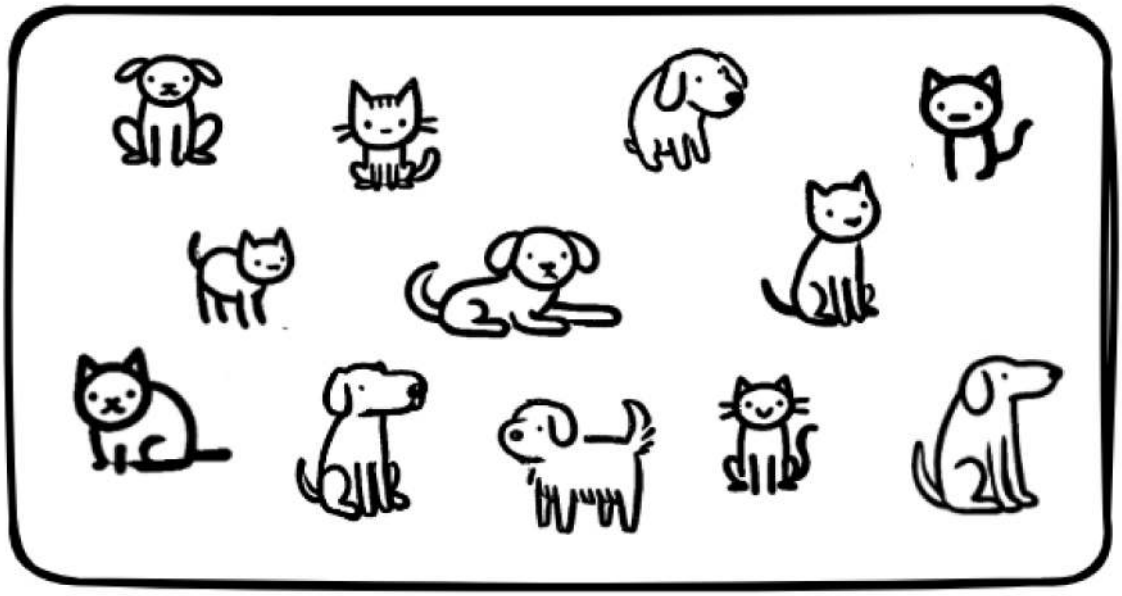

Figure 2.4 – Apprentissage non-supervisé-1 [4]

En analysant uniquement les attributs X (comme la taille, le poids ou l'apparence), la machine identifiera d'elle-même des similitudes entre certains animaux en fonction de leurs caractéristiques communes.

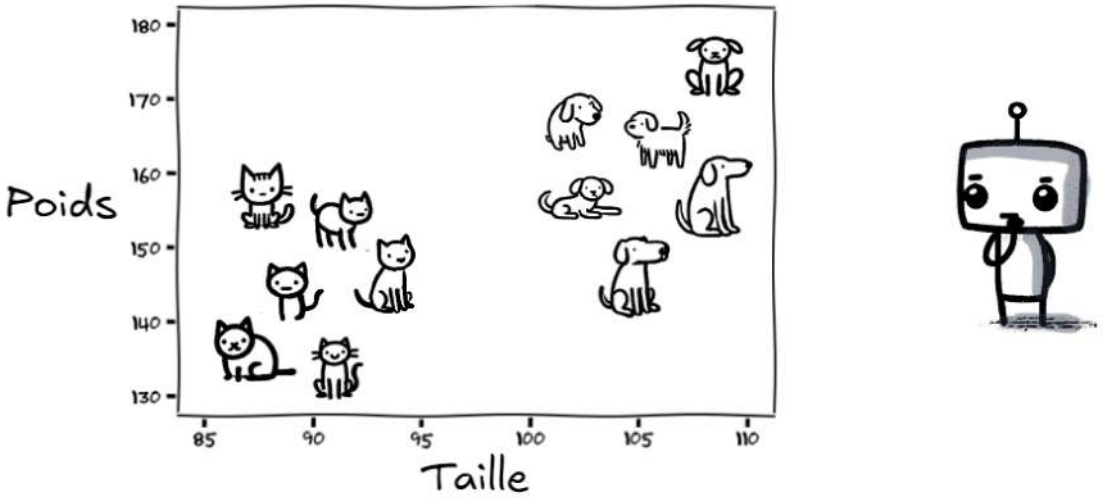

FIGURE 2.5 – Apprentissage non-supervisé-2 [4]

Ainsi, elle pourra former des groupes d'animaux aux caractéristiques similaires, sans pour autant savoir s'il s'agit de chats ou de chiens. En réalité, elle ne disposera d'aucune information à ce sujet, puisqu'elle n'aura pas été entraînée à reconnaître ces catégories spécifiques.

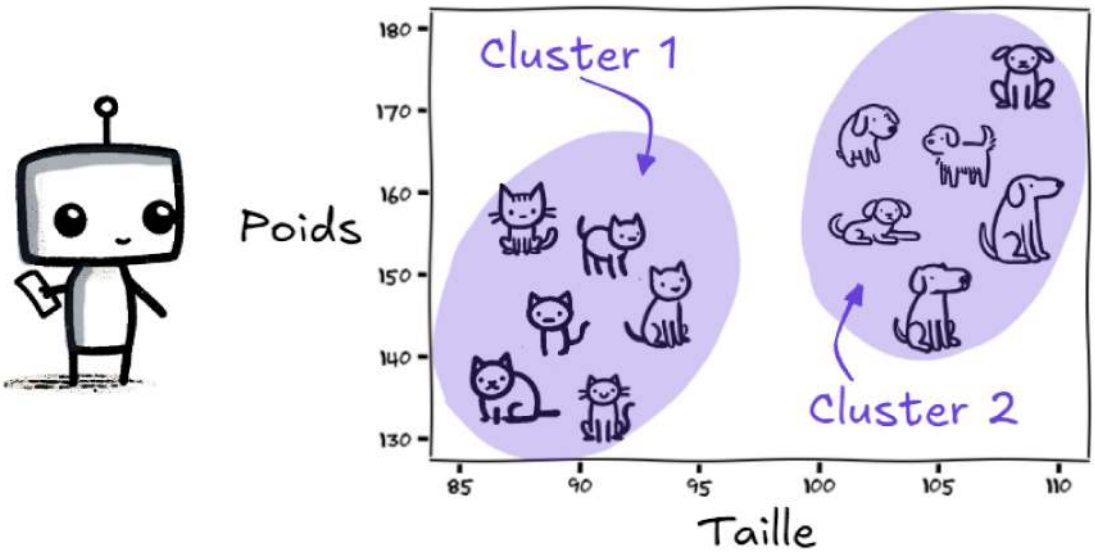

FIGURE 2.6 – Apprentissage non-supervisé-3 [4]

En outre, il y a de nombreuses méthodes d'apprentissage non supervisé en particulier **le clustering**.

2.4.2.1 Clustering

Le clustering est l'une des méthodes d'apprentissage non supervisé les plus répandues, permettant d'identifier la structure sous-jacente ou les regroupements naturels au sein d'un jeu de données. Il est couramment utilisé pour l'analyse exploratoire, la reconnaissance de formes, la détection d'anomalies et la segmentation d'images. Les algorithmes de clustering, comme le k-means ou le clustering hiérarchique, organisent les données en groupes de sorte que les éléments d'un même cluster soient plus similaires entre eux qu'avec ceux des autres groupes. [5]

Par exemple, une entreprise de téléphonie mobile qui souhaite installer de nouvelles antennes peut utiliser l'intelligence artificielle pour analyser où se regroupent le plus souvent ses utilisateurs. Comme un téléphone ne peut se connecter qu'à une seule antenne à la fois, il est essentiel de bien positionner ces antennes pour garantir une bonne couverture réseau. Grâce à des algorithmes spécialisés, l'entreprise peut identifier les zones où se trouvent le plus grand nombre d'utilisateurs et placer ses antennes aux endroits les plus stratégiques afin d'optimiser la qualité du signal.

FIGURE 2.7 – Utilisation du clustering pour trouver des patterns cachés dans les données [5]

Il y a deux principaux types de clusters :

Le hard clustering

Également appelé clustering exclusif, est une approche dans laquelle chaque point de données est attribué à un seul et unique groupe (cluster). Un exemple courant de cette méthode est l'algorithme k-means.

Le soft clustering

Aussi appelé clustering avec chevauchement, permet à un même point de données d'appartenir à plusieurs groupes simultanément, avec différents degrés d'appartenance. Une méthode couramment utilisée pour cela est le modèle de mélange gaussien. [5]

Parmi les algorithmes de clustering les plus couramment utilisés, on retrouve :

Le clustering hiérarchique

Organise les données en une structure arborescente, permettant de regrouper les éléments à différents niveaux de précision pour former une hiérarchie de clusters. [5]

Le modèle de mélange gaussien

Regroupe les données en combinant plusieurs distributions normales multivariées pour former des clusters. [5]

Le clustering k-means

Divise les données en k groupes distincts en attribuant chaque point au groupe dont le centre est le plus proche.

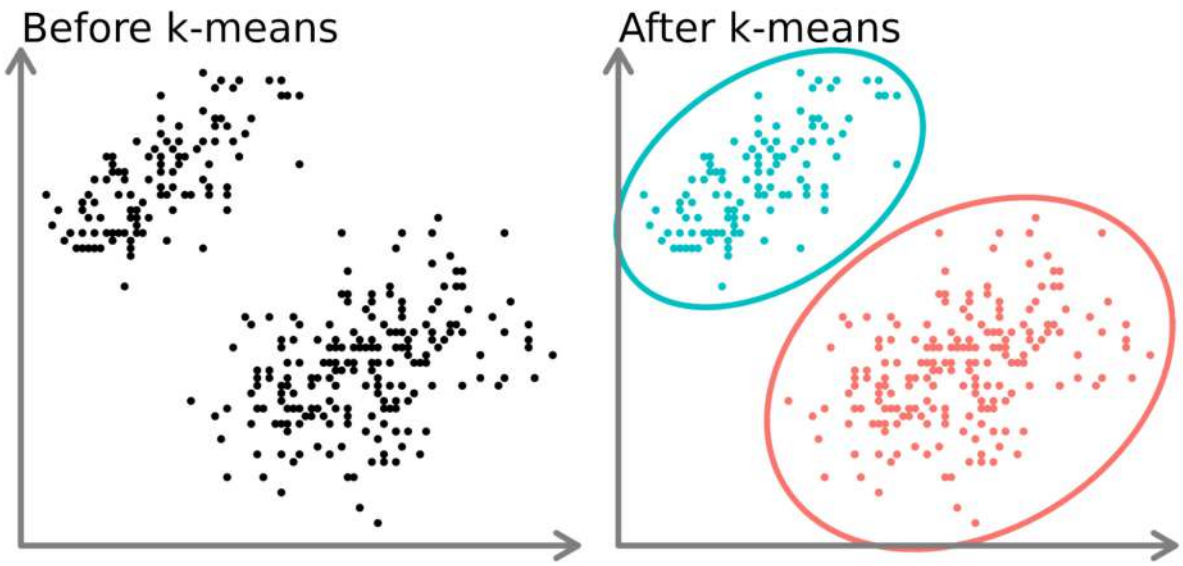

FIGURE 2.8 – k means [6]

Sur la partie gauche du diagramme ci-dessus, on peut voir deux groupes de points distincts, sans étiquettes, mais colorés pour indiquer leur similarité. En appliquant l'algorithme k-means à ces données (partie droite), on obtient deux groupes bien définis,

représentés par des couleurs et des cercles différents.
Lorsque les données sont en deux dimensions, il est simple pour un humain d'identifier ces groupes. Cependant, avec un plus grand nombre de dimensions, un modèle devient indispensable pour les distinguer correctement. [6]

2.5 Comparaison entre l'apprentissage supervisé et l'apprentissage non supervisé

Bien que ces deux approches fassent partie de l'intelligence artificielle, l'apprentissage supervisé repose sur l'entraînement d'un modèle à partir de données étiquetées pour apprendre à prédire un résultat.
Il fonctionne en guidant la machine à travers des exemples concrets de ce qu'elle doit accomplir. L'objectif est de développer des algorithmes capables d'analyser des ensembles de données et d'effectuer des prédictions grâce à une analyse statistique. [25]

Critères	**Apprentissage supervisé**	**Apprentissage non-supervisé**
Données d'entrées	Données connue en entrée	Données inconnue en entrée
Complexité informatique	Complexe	Moins complexe
Domaine d'activités	Classification et régression	Exploitation de règles de clustering et d'association
Précision	Produit des résultats précis	Génère des résultats modérés

Tableau 2.1 – Comparaison entre l'apprentissage supervisé et l'apprentissage non supervisé

2.6 Deep learning et réseaux de neurones

2.6.1 Réseau neuronal

Un réseau neuronal est une technique d'intelligence artificielle qui permet aux ordinateurs d'analyser des données en s'inspirant du fonctionnement du cerveau humain. Il s'agit d'une forme d'apprentissage automatique, plus précisément de deep learning, reposant sur des couches de nœuds interconnectés, appelés neurones, organisés de manière similaire à celle du cerveau. Ce système adaptatif permet aux machines d'apprendre de leurs erreurs

et de s'améliorer progressivement. Les réseaux neuronaux artificiels sont utilisés pour résoudre des problèmes complexes, comme la reconnaissance faciale, la classification des objets ou le résumé automatique de documents, avec une précision accrue. [26]

2.6.2 Fonctionnement

L'architecture des réseaux neuronaux s'inspire directement du cerveau humain. Les neurones, cellules du cerveau, forment un réseau dense et interconnecté, échangeant des signaux électriques afin de permettre le traitement de l'information. De manière similaire, un réseau neuronal artificiel est composé de neurones artificiels qui collaborent pour résoudre un problème. Ces neurones artificiels, ou nœuds, sont en réalité des modules logiciels, et l'ensemble du réseau neuronal constitue un algorithme ou un programme informatique conçu pour effectuer des calculs mathématiques à l'aide de systèmes informatiques. [26]

2.6.2.1 Architecture de réseau neuronal simple

Un réseau neuronal simple se compose de neurones artificiels [26] reliés entre eux et organisés en trois couches :

Couche d'entrée

Les données provenant du monde extérieur sont introduites dans le réseau neuronal artificiel par la couche d'entrée. Les nœuds de cette couche traitent, analysent ou classifient ces informations, puis les transmettent à la couche suivante.
Par exemple, dans un système de reconnaissance d'image, une photo de chat est convertie en données numériques (comme les valeurs de couleur des pixels) et introduite par la couche d'entrée. Les nœuds de cette couche identifient des caractéristiques simples, comme des contours ou des motifs, qu'ils transmettent ensuite aux couches suivantes pour une analyse plus approfondie.

Couche cachée

Les couches cachées reçoivent leurs données en provenance de la couche d'entrée ou d'autres couches cachées situées en amont. Un réseau neuronal artificiel peut contenir de nombreuses couches cachées, chacune jouant un rôle essentiel dans le traitement progressif de l'information. Chaque couche interprète les résultats de la couche précédente, les affine,

puis transmet les données transformées à la couche suivante.
Par exemple, dans le cas d'une image de chat, après que la couche d'entrée a détecté des contours et des motifs simples, les premières couches cachées peuvent repérer des formes plus complexes comme des yeux ou des oreilles. Les couches suivantes combineront ces éléments pour reconnaître des structures complètes, comme la tête ou le corps du chat.

Couche de sortie

La couche de sortie fournit le résultat final issu de l'ensemble des traitements effectués par le réseau neuronal artificiel. Elle peut être composée d'un seul nœud ou de plusieurs, selon la nature du problème à résoudre. Par exemple, dans un cas de classification binaire (oui/non), la couche de sortie comporte un seul nœud qui donne une réponse sous forme de 0 ou 1. En revanche, pour un problème de classification multi-classes, plusieurs nœuds peuvent être présents, chacun correspondant à une catégorie possible.
Dans notre exemple de reconnaissance d'image, si le réseau doit simplement déterminer si l'image représente un chat ou non, un seul nœud de sortie indiquera « 1 » pour oui ou « 0 » pour non. En revanche, si le système doit distinguer entre plusieurs animaux comme un chat, un chien ou un lapin la couche de sortie comportera plusieurs nœuds, chacun représentant une de ces classes, et celui avec la valeur la plus élevée indiquera la prédiction finale. [26]

2.6.3 Architecture de réseau neuronal profond

Les réseaux neuronaux profonds, également appelés réseaux de deep learning, se caractérisent par la présence de nombreuses couches cachées contenant des millions de neurones artificiels interconnectés. Les connexions entre les nœuds sont représentées par des valeurs numériques appelées poids. Un poids positif indique qu'un nœud stimule un autre, tandis qu'un poids négatif signifie qu'il le freine. Plus la valeur du poids est élevée, plus l'influence du nœud sur les autres est importante.
En théorie, les réseaux neuronaux profonds sont capables d'associer n'importe quel type d'entrée à n'importe quel type de sortie. Toutefois, cette puissance a un coût : ils nécessitent un entraînement beaucoup plus intensif que les méthodes classiques de machine learning. Alors qu'un réseau plus simple peut apprendre à partir de centaines ou de milliers d'exemples, un réseau profond exige souvent des millions de données d'entraînement pour atteindre un bon niveau de performance. [26]

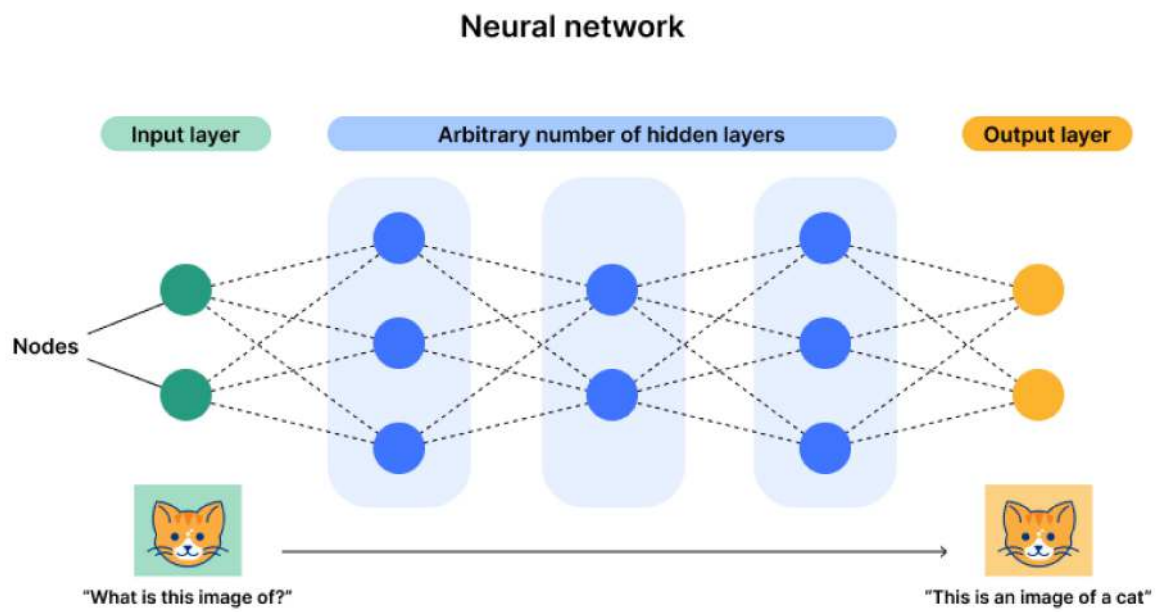

FIGURE 2.9 – architecture de réseau neuronal profond. [7]

2.6.4 Types de réseaux neuronaux

Les réseaux neuronaux artificiels [26] peuvent être catégorisés selon le mode de circulation des données entre les nœuds d'entrée et de sortie. En voici quelques exemples :

2.6.4.1 Réseaux neuronaux à action directe (feedforward neural network)

Les réseaux neuronaux à propagation avant (feedforward) traitent les données de manière unidirectionnelle, en les faisant circuler du nœud d'entrée vers le nœud de sortie. Dans ce type de réseau, chaque nœud d'une couche est relié à tous les nœuds de la couche suivante. Bien que les données circulent dans une seule direction, un réseau à propagation avant peut utiliser des techniques rétroactives, comme la rétropropagation, pour ajuster ses paramètres et améliorer ses prédictions au fur et à mesure de l'entraînement.

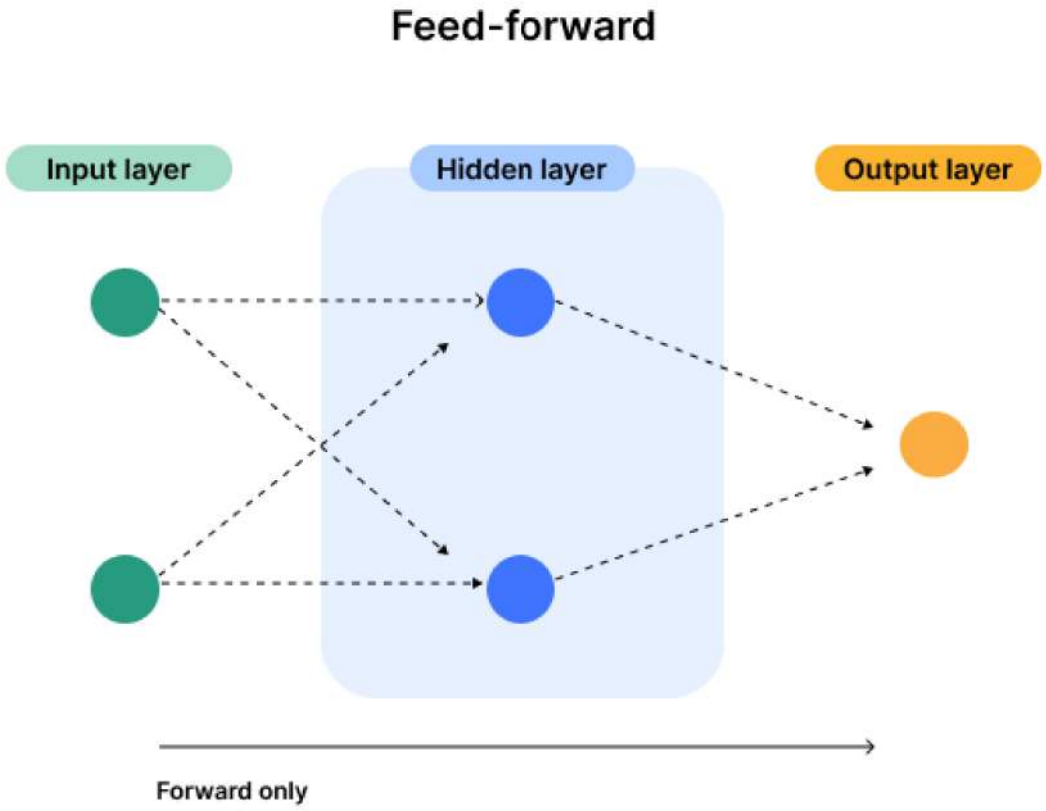

FIGURE 2.10 – Réseau de Neurones à Propagation Avant(FNN) [8]

2.6.4.2 Perceptron multicouche (MLP - Multi-Layer Perceptron)

Les réseaux neuronaux de perceptron multicouches (MLP) augmentent la complexité des réseaux de perceptron en introduisant une ou plusieurs couches cachées entre la couche d'entrée et la couche de sortie. Ces couches cachées permettent au modèle d'apprendre des représentations plus abstraites et complexes des données, ce qui lui permet de résoudre des problèmes non linéaires qui ne peuvent pas être résolus par un simple perceptron à couche unique. Grâce à cette structure, les réseaux de perceptron multicouches sont capables d'effectuer des tâches telles que la classification d'images ou la reconnaissance vocale, en capturant des relations plus profondes dans les données. Les poids associés aux connexions entre les nœuds sont ajustés durant l'entraînement à l'aide de techniques comme la rétropropagation pour minimiser l'erreur du modèle. [27]

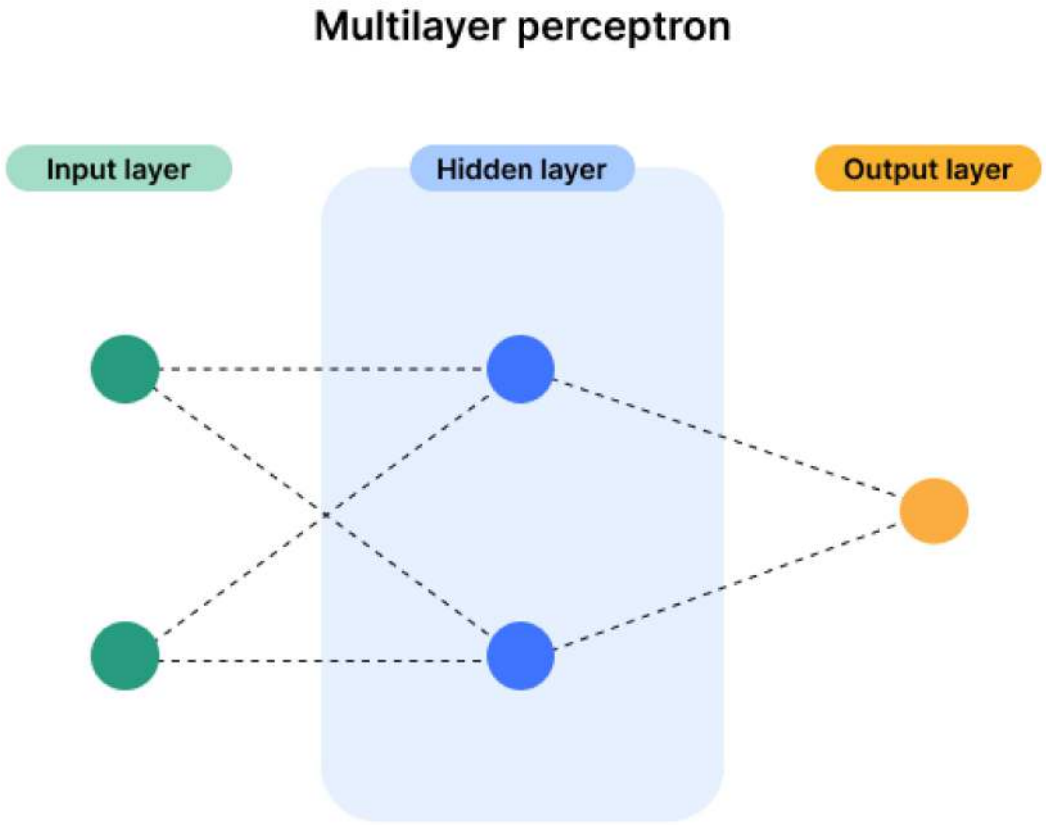

FIGURE 2.11 – Perceptron multicouche [9]

2.6.4.3 Réseaux neuronaux convolutifs (convolutional neuronal network-CNN

Les couches cachées des réseaux neuronaux convolutionnels effectuent des opérations mathématiques spécifiques, telles que la convolution, qui incluent des processus comme le filtrage ou la synthèse. Ces réseaux sont particulièrement efficaces pour la classification d'images, car ils sont capables d'extraire des caractéristiques pertinentes des images, essentielles pour leur reconnaissance et classification. En transformant l'image, ces couches rendent les données plus faciles à traiter tout en préservant les éléments clés nécessaires à une prédiction précise. Chaque couche cachée se spécialise dans l'extraction et le traitement de différentes caractéristiques de l'image, telles que les contours, les couleurs et les textures.

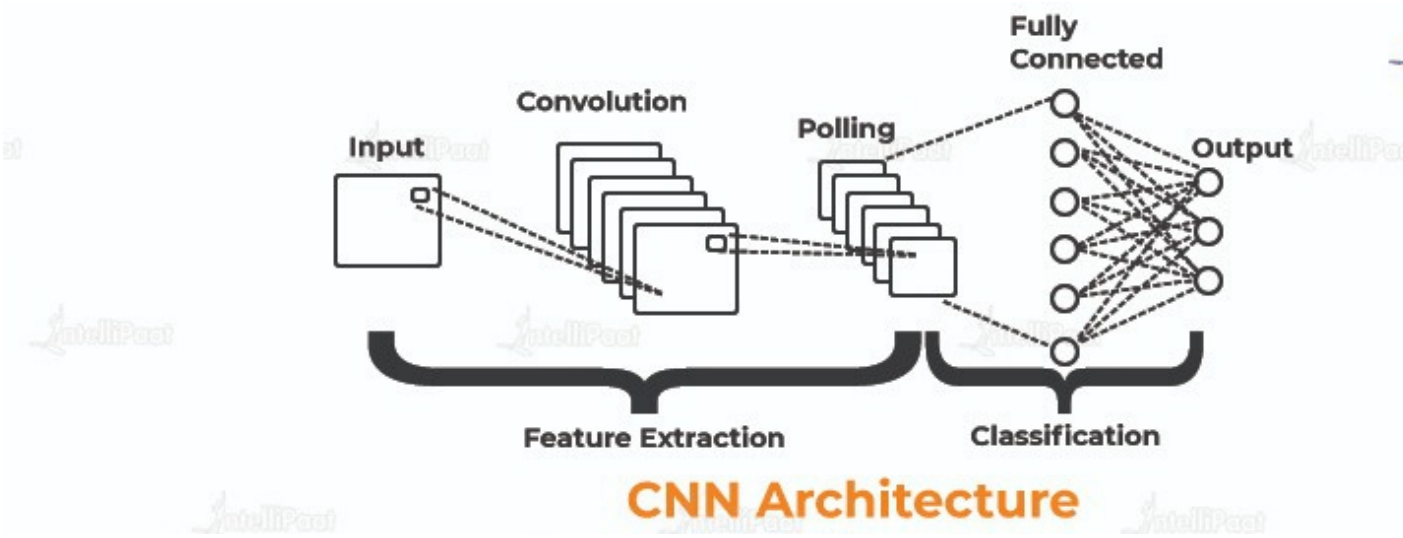

FIGURE 2.12 – Architecture de réseaux neuronaux convolutifs. [10]

2.6.5 Les fonctions d'activations

La fonction d'activation est l'élément du neurone qui lui permet de modifier ou transformer une donnée en sortie.
Elle joue un rôle fondamental dans un réseau de neurones, car elle introduit une transformation non linéaire indispensable au traitement de problèmes complexes.
En mathématiques, il existe une infinité de fonctions non linéaires, mais seule une minorité d'entre elles est réellement utilisée comme fonctions d'activation en deep learning. Bien que les différences entre ces fonctions puissent paraître minimes, le choix de la fonction d'activation est crucial pour assurer le bon fonctionnement d'un réseau neuronal. Certaines doivent être sélectionnées avec soin en fonction de la tâche à accomplir et de la structure du modèle.

2.6.5.1 Fonction ReLU – Rectified Linear Unit

La fonction d'activation Rectified Linear Unit (ReLU) est la plus largement utilisée en deep learning.
Elle retourne la valeur de x si celle-ci est supérieure à 0, et 0 dans le cas contraire. En d'autres termes, elle applique l'opération suivante : le maximum entre x et 0.

$$\text{fonction ReLU}(x) = \max(x, 0). \text{ [28]}$$

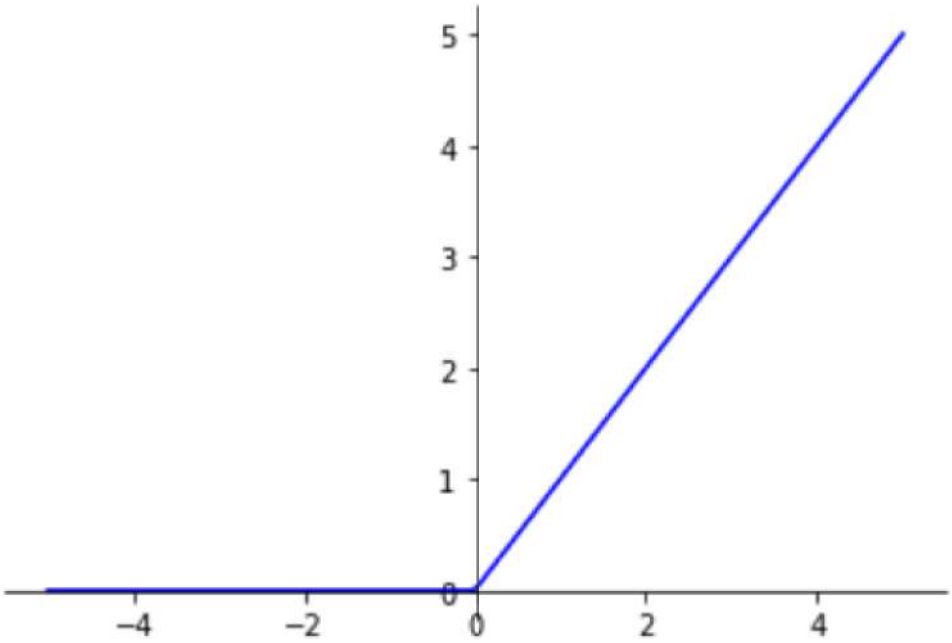

FIGURE 2.13 – Fonction ReLU. [11]

Cette fonction agit comme un filtre en sortie de couche, en transmettant uniquement les valeurs positives aux couches suivantes tout en bloquant les valeurs négatives. Ce mécanisme permet au modèle de se focaliser sur des caractéristiques spécifiques des données, en écartant celles jugées moins pertinentes. [11]

2.6.5.2 Fonction Sigmoide

La fonction sigmoïde est généralement utilisée comme fonction d'activation dans la couche de sortie d'un réseau neuronal conçu pour réaliser une tâche de classification binaire. Elle donne une valeur entre 0 et 1.

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

[29]

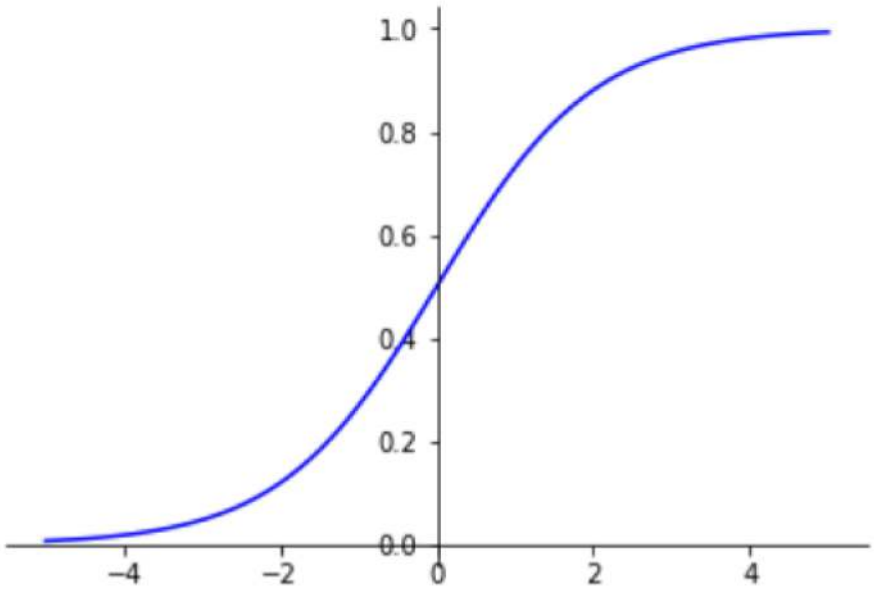

FIGURE 2.14 – Fonction Sigmoide. [11]

2.6.5.3 Softmax

La fonction Softmax est couramment utilisée comme fonction d'activation dans la couche de sortie d'un réseau neuronal destiné à une tâche de classification multi-classes. Elle attribue à chaque sortie une valeur comprise entre 0 et 1, représentant une probabilité. La somme de l'ensemble de ces valeurs est toujours égale à 1, ce qui permet d'interpréter chaque sortie comme la probabilité d'appartenir à une classe donnée.

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

[30]

- x_i est la i-ème composante du vecteur x.
- K est le nombre total d'éléments dans le vecteur.
- La somme au dénominateur garantit que toutes les sorties seront comprises entre 0 et 1, et que leur somme vaudra 1.

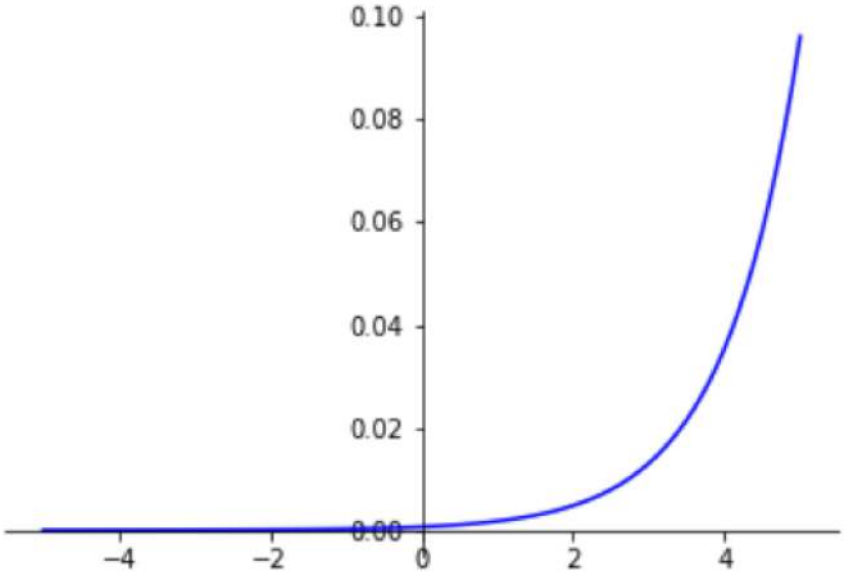

FIGURE 2.15 – Fonction Softmax. [11]

Tout comme la fonction sigmoïde, la fonction Softmax produit des valeurs pouvant être interprétées comme des probabilités. Toutefois, dans le cas de Softmax, chaque valeur correspond à une probabilité associée à une classe précise du jeu de données. Il est essentiel de souligner que la fonction sigmoïde ne convient pas comme fonction d'activation en sortie pour une tâche de classification multi-classes. Bien qu'elle attribue une valeur entre 0 et 1 à chaque sortie, la somme de ces valeurs ne garantit pas un total égal à 1. Par conséquent, les résultats ne peuvent pas être interprétés correctement comme des probabilités, ce qui fausserait l'interprétation du modèle. À l'inverse, la fonction Softmax possède la propriété de normaliser les sorties de manière à ce que leur somme soit exactement égale à 1. Cette caractéristique en fait la fonction idéale pour la couche de sortie d'un réseau neuronal dédié à la classification multi-classes, car elle respecte les principes fondamentaux de la probabilité. [11]

2.6.5.4 Tanh – tangente hyperbolique

La fonction tanh permet de normaliser les valeurs d'entrée en les ramenant dans une plage comprise entre -1 et 1. Elle peut aussi être utilisée comme alternative à la fonction sigmoïde dans la couche de sortie d'un modèle de classification binaire. Elle donne un résultat entre -1 et 1.

$$\begin{aligned}\tanh(x) &= \frac{\sinh(x)}{\cosh(x)} \\ &= \frac{e^x - e^{-x}}{e^x + e^{-x}}\end{aligned}$$

[31]

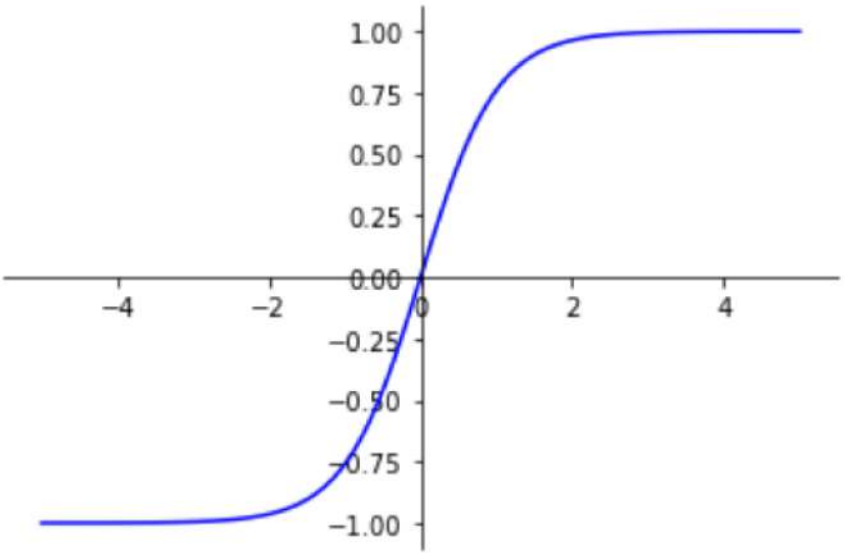

FIGURE 2.16 – Fonction tanh. [11]

Fonctions d'activation communes

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}$$

2.7 Apports et limites du Deep Learning dans l'analyse du trafic réseau

Le deep learning révolutionne l'analyse du trafic réseau en automatisant la détection et la classification des menaces grâce à des réseaux de neurones capables d'identifier des patterns complexes à partir de vastes volumes de données. Son intégration améliore la précision et réduit les faux positifs, permettant aux analystes de se concentrer sur les vraies menaces. Cependant, ces modèles nécessitent une mise à jour constante pour suivre l'évolution des attaques et dépendent de données de qualité, soulevant des défis liés à la confidentialité et aux biais potentiels. Le chapitre suivant détaillera la méthodologie d'implémentation, incluant le prétraitement des données, l'ingénierie des features, l'optimisation des architectures et l'évaluation des performances.

2.8 Conclusion

En résumé, ce chapitre a permis de poser les bases de l'intelligence artificielle, en partant de sa définition jusqu'à son mode de fonctionnement. Nous avons mis en lumière les principales approches d'entraînement des modèles d'IA, notamment les méthodes **d'apprentissage supervisé** telles que **la régression et la classification**, ainsi que **l'apprentissage non supervisé** illustré par **le clustering.**
Nous avons ensuite exploré les concepts liés au **deep learning**, en présentant **l'architecture des réseaux neuronaux**, qu'ils soient **simples ou profonds**, ainsi que **les différents types de réseaux**, notamment **les réseaux convolutifs (CNN)** et les **réseaux à propagation avant** (FNN).
Enfin, nous avons abordé l'importance des **fonctions d'activation**, éléments essentiels des neurones artificiels qui leur permettent de transformer les données en sortie. Le choix de la fonction d'activation, tel que **ReLU**, **Sigmoïde ou Softmax**, dépend de la nature de la tâche et de la structure du modèle, et joue un rôle déterminant dans les performances du réseau.

Chapitre 3

Méthodologie et Mise en œuvre

3.1 Introduction

Cette recherche vise à développer un système d'apprentissage profond pour détecter et classifier le trafic sur le darknet. En utilisant une base de données dédiée, nous avons appliqué un traitement et un nettoyage rigoureux pour garantir la fiabilité des modèles.
Nous avons conçu et entraîné plusieurs modèles, notamment un SVM (*Support Vector Machine*), un RF (*Random Forest*), un CNN (*Convolutional Neural Network*) et un FNN (*Feedforward Neural Network*), pour tester leur capacité à classifier différentes catégories de trafic darknet.
Le choix de ce jeu de données repose sur sa fiabilité et sa pertinence. Il a été utilisé par plusieurs chercheurs dans des études similaires, ce qui confère une grande crédibilité à ses données. De plus, il contient des trafics mixtes, notamment des trafics normaux, VPN et TOR, essentiels pour notre problématique.
Dans ce chapitre, nous présentons les étapes méthodiques allant de la préparation des données à leur organisation en ensembles d'entraînement, test et validation. Nous décrivons également des techniques comme SMOTE pour équilibrer les classes et pour enrichir les classificateurs CNN.
Enfin, des scénarios d'obscurcissement sophistiqués ont été conçus pour tester la robustesse des modèles, permettant une analyse approfondie de leur capacité à classer correctement malgré des tentatives de dissimulation.

3.2 Collecte et Préparation des Données

3.2.1 Collection et Construction du Dataset

La construction d'un jeu de données est essentielle pour évaluer l'efficacité d'un modèle d'apprentissage profond. Le principal défi réside dans la difficulté de trouver des ensembles complets contenant les éléments requis. Nous avons choisi le **CIC-Darknet2020**, compilé par l'Institut canadien de cybersécurité, car il est bien structuré et permet de tester et d'améliorer notre modèle avec fiabilité.

CIC-Darknet2020 [32] combine deux ensembles publics de l'Université du Nouveau-Brunswick : **ISCXTor2016** et **ISCXVPN2016**, qui capturent le trafic en temps réel avec **Wireshark** et **TCPdump** [33], et dont les caractéristiques sont extraites avec **CIC-FlowMeter** [34]. Ce jeu de données contient **158 616 échantillons**, hiérarchiquement étiquetés.

Les catégories principales incluent **Non-Tor**, **Tor**, **Non-VPN** et **VPN**, détaillées dans le Tableau 3.2, avec des sous-catégories comme **streaming audio**, **navigation**, **chat**, **courrier électronique**, **transfert de fichiers**, **P2P**, **streaming vidéo** et **VOIP**, résumées dans le Tableau 3.1.

Classe d'application	Applications utilisées
Streaming audio	Vimeo et YouTube
Navigation	Firefox et Chrome
Chat	ICQ, AIM, Skype, Facebook et Hangouts
E-mail	SMTPS, POP3S et IMAPS
Transfert de fichiers	Skype et FileZilla
P2P	uTorrent et Transmission (BitTorrent)
Streaming vidéo	Vimeo et YouTube
VoIP	Facebook, Skype et Hangouts

Tableau 3.1 – Tableau des applications par classe d'utilisation

Catégorie de trafic	Échantillons
Non-Tor	110 442
Non-VPN	23 863
Tor	1 392
VPN	22 919

Tableau 3.2 – Nombre d'échantillons de données par catégorie de trafic

Catégorie d'application	Échantillons
Streaming audio	21 350
Navigation	46 457
Chat	11 629
Email	6 145
Transfert de fichiers	11 182
P2P	48 520
Streaming vidéo	9 767
VOIP	3 566

Tableau 3.3 – Nombre d'échantillons de jeux de données par catégorie d'application

Le tableau ci-dessous présente les principales caractéristiques du modèle d'apprentissage profond, utilisées pour différencier les trafics **Tor**, **Non-Tor**, **VPN** et **Non-VPN**. Basées sur des mesures temporelles et des données des paquets, ces caractéristiques capturent les comportements spécifiques de chaque type de trafic. Les intervalles entre les paquets (*Fwd IAT*) aident à détecter les trafics masqués comme **VPN** et **Tor**. Les longueurs des paquets (*Fwd Packet Length*, *Bwd Packet Length*) reflètent les particularités des applications et des protocoles. Enfin, les statistiques globales (*Flow Duration*, *Total Packet Length*) améliorent la classification des catégories après l'entraînement.

Fonctionnalité	Description
Fwd IAT	Intervalle entre deux paquets en avant (moy., max, min, écart-type)
Bwd IAT	Intervalle entre deux paquets en arrière (moy., max, min, écart-type)
Flow IAT	Intervalle entre deux paquets dans toutes directions
Active	Durée d'activité avant inactivité (moy., max, min, écart-type)
Idle	Durée d'inactivité avant activité (moy., max, min, écart-type)
Fwd Packet Length	Longueur des paquets en avant (moy., max, min, écart-type)
Bwd Packet Length	Longueur des paquets en arrière (moy., max, min, écart-type)
Packet Length	Longueur totale des paquets (moy., max, min, var.)
Header Length	Longueur des en-têtes
Flag Counts	Nombre de drapeaux (e.g., FIN, SYN, ACK)
Subflow	Paquets/octets par sous-flux

Fonctionnalité	Description
Rates	Flux octets/s, paquets/s
Bulk Statistics	Statistiques sur les groupes d'octets et de paquets
Total Packets	Nombre total de paquets
Segment Size	Taille de segment moyenne/minimum
Init Win Bytes	Octets de fenêtre initiale
Flow Duration	Durée totale du flux
Avg Packet Size	Taille moyenne des paquets
Down/Up Ratio	Rapport descendant/montant

Tableau 3.4 – Résumé des principales fonctionnalités du jeu de données.

3.2.2 Aperçu du Jeu de Données Original

Une analyse du jeu de données original est essentielle pour identifier les valeurs manquantes et établir une méthodologie de nettoyage adaptée. Cette étape prépare les données pour la construction d'un modèle capable de détecter et classifier les trafics réseau (TOR, NON-TOR, VPN, NON-VPN).

Le jeu de données présenté sur la figure 3.1 contient **158 616 entrées** réparties sur **85 colonnes**, offrant une **analyse détaillée du trafic réseau**. Il comprend des attributs essentiels tels que **l'ID du flux, les adresses IP source et destination, les ports, le protocole, l'horodatage, la durée du flux**, ainsi que diverses **statistiques liées aux paquets**. Ces informations nous permettent de mieux comprendre le fonctionnement des échanges de données et leur impact sur la sécurité de nos réseaux informatiques.

La **structure du jeu de données** combine des **données numériques et textuelles**, ce qui nous facilite une **étude approfondie des comportements réseau**. En analysant ces différentes variables, nous pouvons identifier **des tendances, des anomalies et des menaces potentielles**. Grâce à la richesse des informations disponibles, nous sommes en mesure d'exploiter ce jeu de données pour nos analyses **statistiques et nos modèles prédictifs**, renforçant ainsi la **surveillance et la gestion de nos réseaux**.

Avec une **empreinte mémoire de plus de 102,9 Mo**, ce jeu de données constitue une base solide pour **nos applications en cybersécurité**, notamment la **détection d'anomalies et la classification du trafic**. En utilisant des techniques avancées comme **l'intelligence artificielle (IA) et l'apprentissage profond (DL)**, nous pouvons ex-

ploiter ces données pour développer **des modèles de détection des cyberattaques** et améliorer la **résilience de nos infrastructures numériques**.

```
   data.info()
   #plt(samples.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 158616 entries, 0 to 158615
Data columns (total 85 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   Flow ID                     158616 non-null  object
 1   Src IP                      158616 non-null  object
 2   Src Port                    158616 non-null  int64
 3   Dst IP                      158616 non-null  object
 4   Dst Port                    158616 non-null  int64
 5   Protocol                    158616 non-null  int64
 6   Timestamp                   158616 non-null  object
 7   Flow Duration               158616 non-null  int64
 8   Total Fwd Packet            158616 non-null  int64
 9   Total Bwd packets           158616 non-null  int64
 10  Total Length of Fwd Packet  158616 non-null  int64
 11  Total Length of Bwd Packet  158616 non-null  int64
 12  Fwd Packet Length Max       158616 non-null  int64
 13  Fwd Packet Length Min       158616 non-null  int64
 14  Fwd Packet Length Mean      158616 non-null  float64
 15  Fwd Packet Length Std       158616 non-null  float64
 16  Bwd Packet Length Max       158616 non-null  int64
 17  Bwd Packet Length Min       158616 non-null  int64
 18  Bwd Packet Length Mean      158616 non-null  float64
 19  Bwd Packet Length Std       158616 non-null  float64
...
 83  Label                       158616 non-null  object
 84  Label.1                     158616 non-null  object
dtypes: float64(24), int64(55), object(6)
memory usage: 102.9+ MB
```

FIGURE 3.1 – Analyse des caractéristiques du jeu de données

L'analyse des fonctionnalités permet d'identifier les caractéristiques pertinentes pour l'entraînement du modèle, offrant une vue claire de la structure des données et de leurs interrelations (voir Figure 3.1).

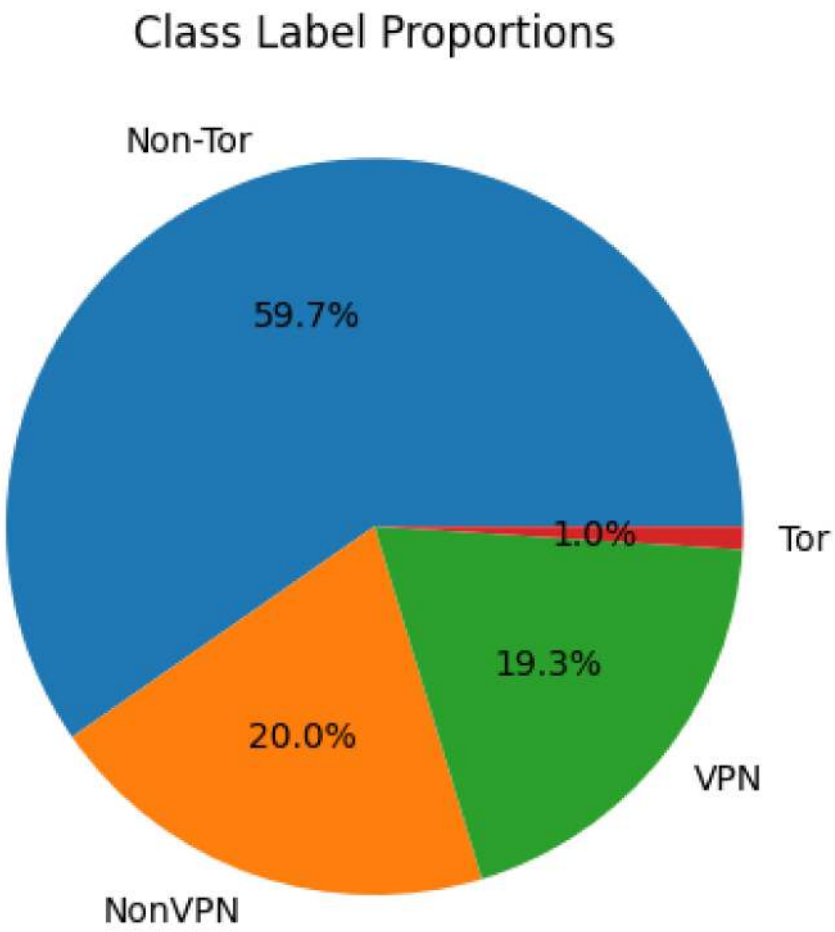

FIGURE 3.2 – Distribution des Étiquettes de Classe

En utilisant diverses fonctions de la bibliothèque pandas, nous pouvons examiner la composition exacte du jeu de données que nous souhaitons analyser, Comme illustré dans la **Figure 3.1** et 3.2, nous avons effectué une analyse des caractéristiques du jeu de données afin de mieux comprendre sa composition.

- `import pandas as pd` : Importe la bibliothèque pandas pour manipuler et analyser les données.
- `data =pd.read_csv(r"Darknet1.csv")` :Charge le jeu de données dans un DataFrame pandas. Le chemin correspond à l'emplacement du fichier.
- `data.info()` : Fournit un résumé des colonnes, des valeurs non nulles et des types de données.

Au fur et à mesure de l'avancement du travail, d'autres fonctions de pandas seront utilisées pour explorer et analyser le jeu de données.

Pour visualiser la répartition des données entre le trafic bénigne et le trafic dark-

net,**Figure 3.3** et **Figure 3.4** présente le nombre et le décompte des données disponibles dans le jeu de données

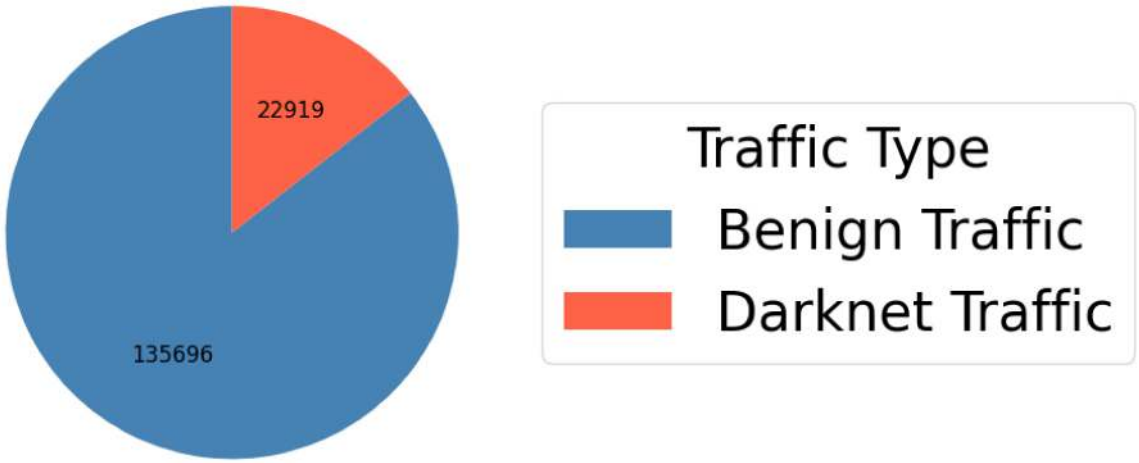

FIGURE 3.3 – Proportion d'étiquette de classe

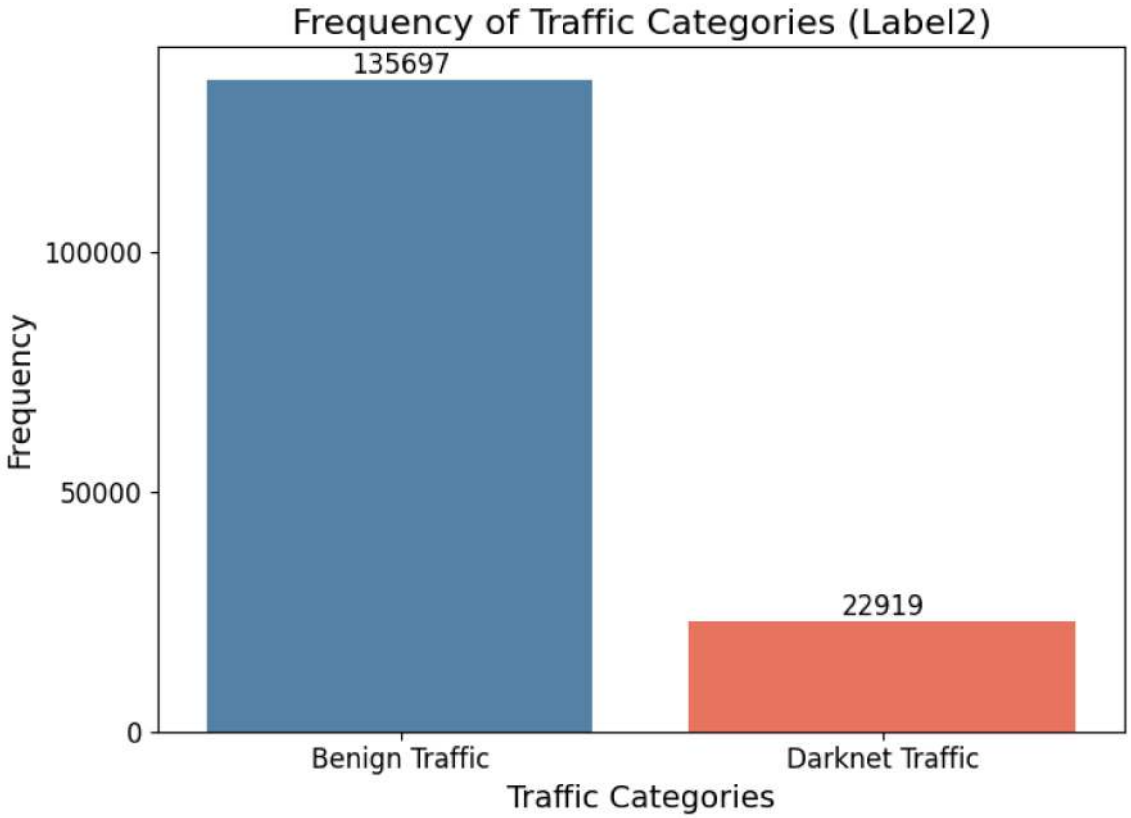

FIGURE 3.4 – Fréquence des catégories de trafic

Ayant une idée claire de la répartition entre les deux types de trafic, il est nécessaire d'examiner leur division en sous-catégories telles que TOR, NON-TOR, VPN et NON-VPN comme **Figure 3.5**.

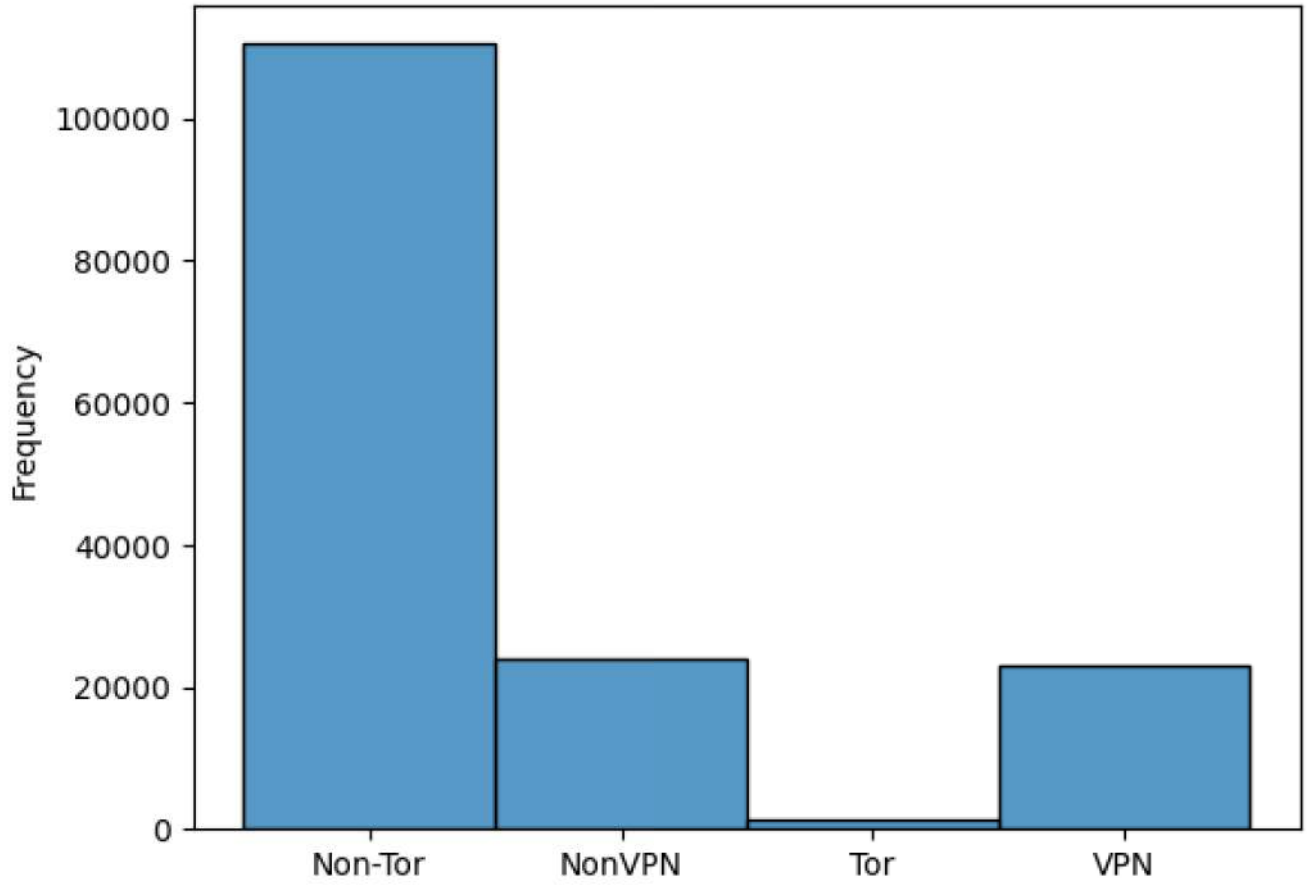

FIGURE 3.5 – Répartition des catégories de trafic en sous-catégories

Le graphique dans la **Figure 3.6** présente les **échantillons pour chaque répartition des catégories de trafic en sous-catégories**, distinguant entre **trafic bénin** et **trafic du darknet**. On observe que le trafic **P2P** est le plus fréquent parmi le trafic bénin, suivi de la **navigation web**, tandis que l'**audio-streaming** domine dans le trafic du darknet. Chaque sous-catégorie est représentée par une couleur spécifique, permettant une visualisation claire des tendances. Cette analyse est essentielle pour la **surveillance réseau et la cybersécurité**, car elle aide à identifier les habitudes de trafic et d'éventuelles anomalies pouvant indiquer une activité suspecte.

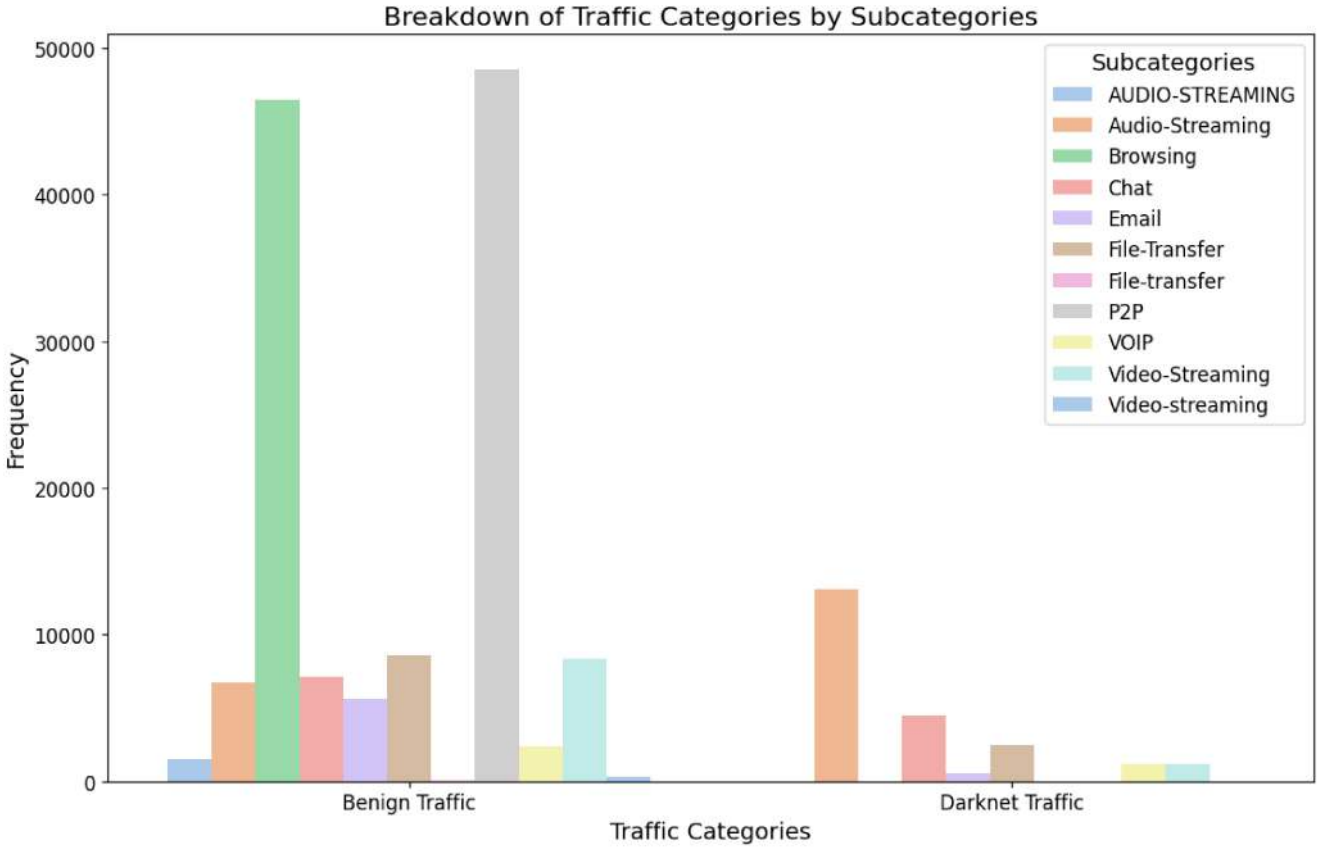

FIGURE 3.6 – Échantillons pour chaque répartition des catégories de trafic en sous-catégories

Après analyse du jeu de données, nous avons constaté des valeurs manquantes, des doublons et un déséquilibre dans la répartition. Ces problèmes peuvent entraîner un modèle biaisé incapable de bien classifier les classes minoritaires. Il est donc essentiel de nettoyer et d'équilibrer le jeu de données avant de le diviser pour l'entraînement du modèle.

3.2.3 Prétraitement des Données

Le jeu de données CIC-Darknet2020 contient des échantillons présentant des valeurs manquantes (`NaN`), des valeurs infinies, et un déséquilibre marqué entre les classes de trafic et d'application. Le prétraitement est une étape essentielle pour nettoyer les données, conserver les caractéristiques pertinentes et équilibrer les classes afin d'améliorer les performances des modèles de classification.

1. **Nettoyage des données** :
   — Suppression des colonnes non pertinentes (`Flow ID`, `Src IP`, `Dst IP`, `Label.1`).
   — Conversion de `Timestamp` en format datetime pour une meilleure gestion temporelle.

2. **Extraction de caractéristiques** :
   — Création de variables temporelles (`Hour`, `DayOfWeek`) sous forme cyclique.
   — Détection des ports Tor courants (`Is_Common_Tor_Port`) pour indiquer une potentielle activité darknet.
   — Calcul de mesures comme `Packet_Length_Ratio`, `Duration_Per_Packet` et `Timing_Variability`.
   — Indicateurs supplémentaires pour identifier le chiffrement potentiel (`Potential_Encrypted`).
3. **Gestion des données manquantes** :
   — Remplacement des valeurs infinies (`+inf`, `-inf`) par des `NaN`.
   — Imputation par la médiane pour toutes les colonnes numériques.
4. **Encodage des labels** :
   — Transformation des catégories (`Non-Tor`, `Tor`, etc.) en valeurs numériques pour faciliter la classification.
5. **Équilibrage des classes** :
   — Rééquilibrage des classes via SMOTE pour suréchantillonner les minoritaires et sous-échantillonnage aléatoire des majoritaires, aboutissant à une distribution équilibrée (~18,700-19,100 échantillons par classe)..
6. **Normalisation et partitionnement** :
   — Standardisation des données via z-score : $z = \frac{x-\mu}{\sigma}$.
   — Partitionnement stratifié (80%-20%) avec conservation de la distribution des classes.
7. **Préparation finale** :
   — Création de vecteurs de caractéristiques pour FNN.
   — Transformation en tenseurs 3D (`échantillons × caractéristiques × 1`) pour CNN.

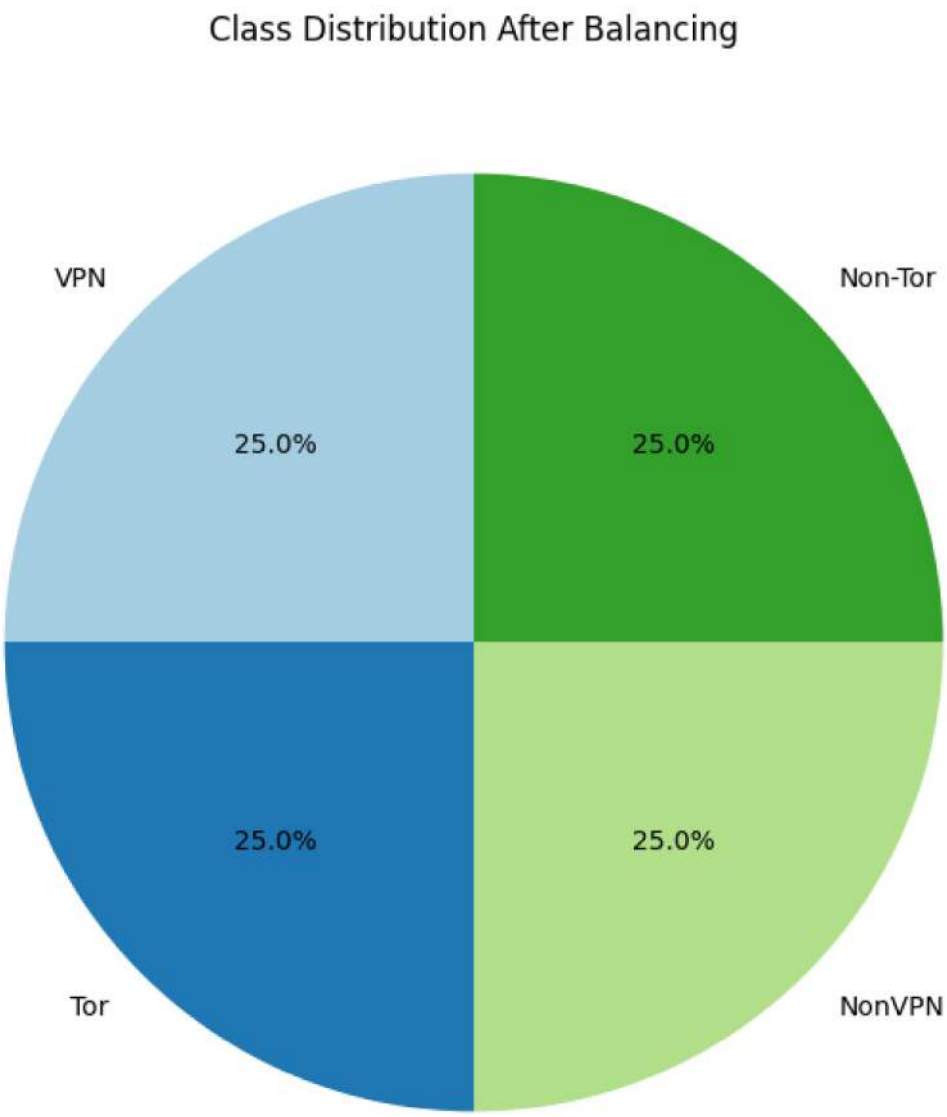

FIGURE 3.7 – Distribution équilibré des classes de trafic

La figure 3.7 ci-dessus représente un **diagramme circulaire** illustrant la **répartition équilibrée des classes** après traitement des données. Les quatre catégories—**VPN, Non-Tor, Tor et NonVPN**—sont distribuées de manière égale, chacune représentant **25,0%** du total. Cette visualisation met en évidence l'équilibre dans la classification, garantissant une **représentation uniforme des différentes classes** dans l'analyse des données.

3.3 Environnement Technique et Intégration du Flux de Travail

3.3.1 Description de l'environnement de développement

La création d'un modèle d'apprentissage profond, surtout pour détecter et classifier le trafic du darknet, est une tâche complexe et exigeante. Elle nécessite des ressources informatiques substantielles et un équipement spécialisé. Ce domaine coûteux demande des investissements significatifs pour garantir des résultats précis et efficaces.

Dans le cadre de notre projet, le système utilisé intègre un **processeur Intel i5**, une **mémoire vive de 64 GB**, et une carte graphique **NVIDIA GeForce RTX 3070**. La **RTX 3070** est une puissante carte graphique, équipée de **8 GB de mémoire vidéo dédiée**, qui permet d'accélérer considérablement les processus d'entraînement des modèles d'apprentissage profond grâce à son architecture **CUDA** et ses **cœurs Tensor**. Ce **GPU** est idéal pour traiter des ensembles de données complexes et pour exécuter des calculs intensifs requis dans la classification du trafic.

En plus de cet environnement matériel performant, notre projet s'appuie sur des concepts clés de l'intelligence artificielle (IA). Nous utilisons des **réseaux neuronaux profonds**, qui imitent la structure du cerveau humain, pour apprendre à partir de grands ensembles de données et identifier des schémas dans le trafic du darknet. Ces modèles d'apprentissage supervisé et non supervisé sont au cœur de notre approche.

Par ailleurs, **Anaconda** est une plateforme clé dans notre développement. Cette distribution open-source de **Python** simplifie la gestion des bibliothèques et des environnements. Grâce à **Anaconda**, nous accédons facilement à des bibliothèques essentielles comme **NumPy** pour la manipulation de données, **Pandas** pour l'analyse des données, **Matplotlib** et **Seaborn** pour la visualisation, et **TensorFlow** ou **PyTorch** pour la création et l'entraînement des modèles d'apprentissage profond. Ces outils offrent une intégration fluide et permettent de développer des solutions robustes et efficaces. [35]

Grâce à cette configuration matérielle et logicielle, nous sommes en mesure de traiter efficacement de grands ensembles de données, d'accélérer l'entraînement des modèles, et de garantir des performances optimales dans la classification du trafic du darknet.

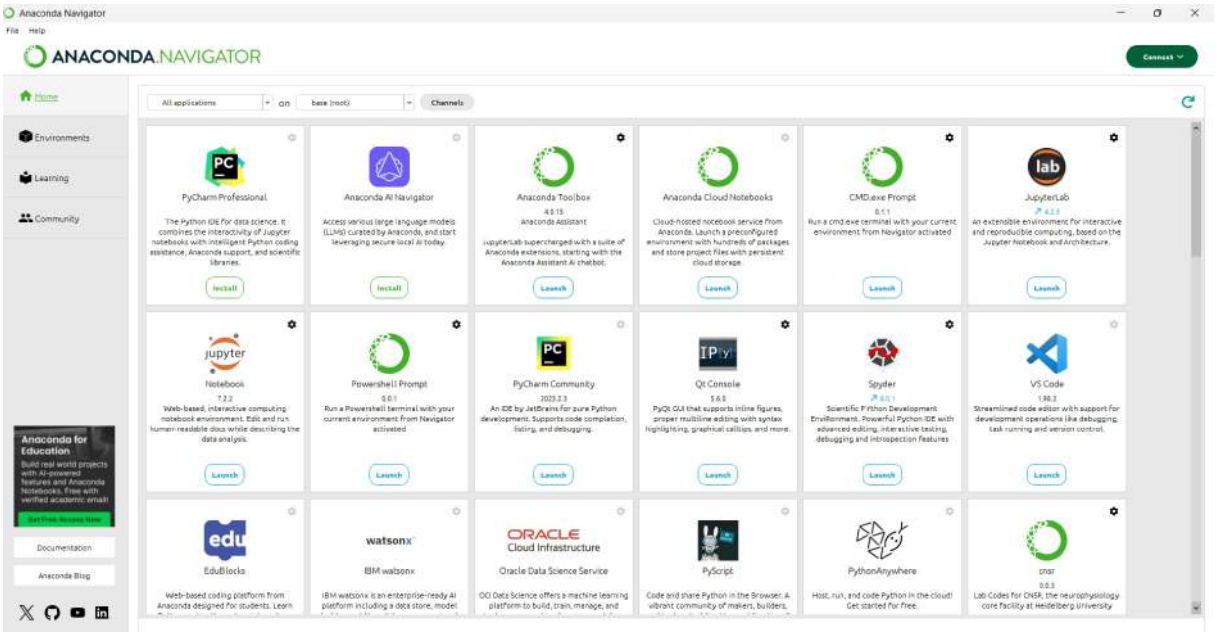

FIGURE 3.8 – Navigateur Anaconda

Avant d'utiliser Jupyter Lab, notre environnement de développement, nous devons lancer Anaconda Navigator, créer un environnement virtuel nommé Py39 avec Python 3.9, compatible avec des bibliothèques comme TensorFlow. Ensuite, via l'invite de commandes, installer les bibliothèques nécessaires

3.3.1.1 Numpy

NumPy est une bibliothèque essentielle en Python, particulièrement adaptée aux calculs scientifiques et à la gestion des tableaux multidimensionnels [36]. C'est un pilier fondamental pour des outils comme Pandas et TensorFlow, offrant une grande efficacité dans le traitement des données complexes et volumineuses.

3.3.1.2 Pandas

Pandas est une bibliothèque incontournable en Python, spécialement conçue pour manipuler et analyser des données. Avec ses deux structures principales, les Series (listes unidimensionnelles) et les DataFrames (tableaux organisés en lignes et colonnes), Pandas simplifie des tâches complexes comme le nettoyage, la transformation, le filtrage ou encore le regroupement des données [37]. C'est un outil essentiel, particulièrement apprécié dans les domaines de la science des données et de l'apprentissage automatique, pour travailler efficacement avec des données structurées.

3.3.1.3 Matplotlib

Matplotlib est une bibliothèque Python permettant de créer une grande variété de visualisations de données, des graphiques simples aux représentations complexes. Elle est polyvalente, facile à utiliser et idéale pour produire des graphiques de qualité, que ce soit pour l'analyse de données, la recherche scientifique ou des présentations. [38]

3.3.1.4 Keras

C'est une bibliothèque open-source conçue pour rendre l'apprentissage profond plus accessible et efficace. Elle s'appuie sur des frameworks comme TensorFlow, utilisé en backend, pour exécuter les calculs complexes. Particulièrement utile pour les modèles CNN, elle simplifie l'ajout de couches convolutionnelles, la régularisation et l'optimisation, tout en tirant parti des GPU pour accélérer les processus. [39]

3.3.1.5 TensorFlow

TensorFlow est une bibliothèque open-source développée par Google pour l'apprentissage automatique et profond. Elle offre une plateforme complète pour concevoir, entraîner et déployer des modèles, avec prise en charge des calculs sur GPU et TPU pour des performances accrues.
Grâce à son architecture flexible et son API conviviale comme Keras, TensorFlow facilite la création de réseaux neuronaux, même pour les débutants. Polyvalent, il est utilisé pour la reconnaissance d'images, le traitement du langage naturel et les systèmes de recommandation, en faisant un outil de référence dans la recherche et l'industrie. [40]

3.3.1.6 Scikit-learn

Scikit-learn est une bibliothèque Python open-source dédiée au machine learning. Elle propose des outils simples et efficaces pour la classification, la régression, le clustering et la réduction de dimension. Facile à utiliser, elle est idéale pour les débutants comme les experts dans le traitement de données et la création de modèles prédictifs. [41]

3.3.1.7 Seaborn

Seaborn est une bibliothèque Python spécialisée dans la visualisation statistique. Elle simplifie la création de graphiques élégants et informatifs en s'appuyant sur Matplotlib, tout en offrant des fonctions avancées pour explorer et représenter des données complexes de manière intuitive. [42]

3.3.2 Flux de travail

1. **Prétraitement des données :** Nettoyage, normalisation et extraction des caractéristiques clés.
2. **Rééquilibrage des classes :** Suréchantillonnage des classes minoritaires avec SMOTE.
3. **Division des données :** Séparation en ensembles d'entraînement, validation et test.
4. **Entraînement des modèles :** Test des architectures CNN, FNN, SVM et RF.
5. **Évaluation et sauvegarde :** Analyse des performances (précision, rappel, F1-score) et stockage des modèles.
6. **Déploiement :** Validation en environnement simulé et tests de robustesse.

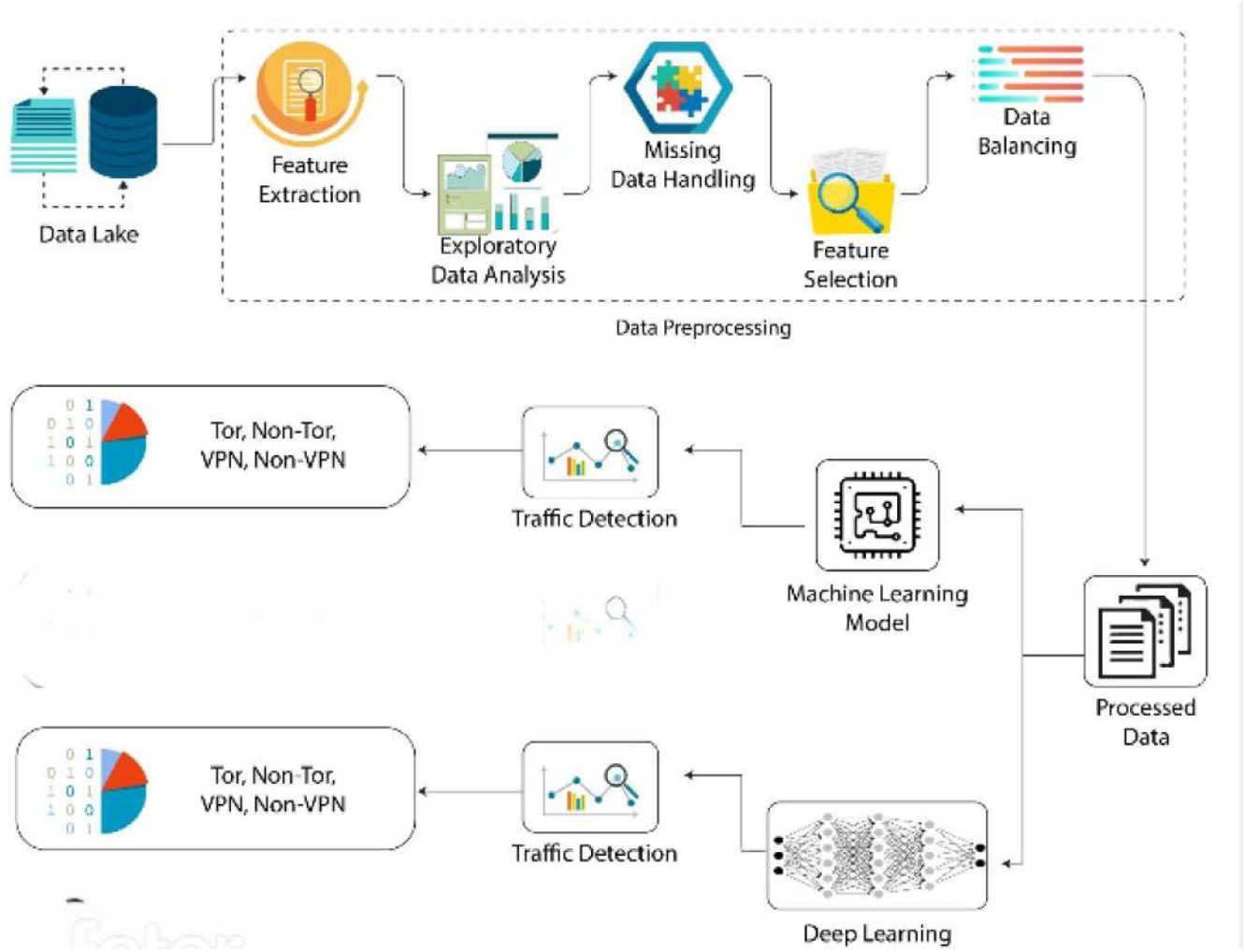

Figure 3.9 – Schéma du Flux de Travail

3.4 Conception et Entraînement des Modèles

3.4.1 Architectures de Réseaux Neuronaux pour la Détection de Trafic

Deux architectures de réseaux neuronaux sont couramment utilisées pour la classification du trafic réseau : le réseau neuronal entièrement connecté (FNN) et le réseau convolutionnel (CNN). Ces architectures permettent de traiter différents types de données et d'extraire des caractéristiques pertinentes pour des tâches de classification complexes.

3.4.1.1 Réseau Neuronal Entièrement Connecté (FNN)

Le Réseau Neuronal Entièrement Connecté (FNN) est adapté au traitement de données tabulaires structurées, telles que les caractéristiques numériques des sessions réseau. Sa structure entièrement connectée permet d'établir des relations complexes entre les caractéristiques d'entrée et de fournir des prédictions précises.

Caractéristiques de l'architecture du FNN :

- **Couche d'entrée :** Reçoit les données tabulaires et transmet les caractéristiques brutes aux couches suivantes.
- **Couches cachées :** Transforme les données en représentations complexes à l'aide de la fonction d'activation ReLU et applique dropout pour éviter le surapprentissage.
- **Couche de sortie :** Utilise une activation softmax pour fournir des probabilités de classification adaptées aux catégories à prédire.

La figure **3.10** illustre l'organisation hiérarchique des couches du FNN, assurant un flux efficace des données.

Structure du modèle FNN employé :

Le résumé de notre modèle de Réseau Neuronal Entièrement Connecté (FNN), conçu pour la classification des données tabulaires issues du trafic darknet, met en évidence les dimensions des couches, les fonctions d'activation (ReLU pour les couches cachées et softmax pour la sortie), ainsi que la régularisation par dropout pour limiter le surapprentissage. En traitant les caractéristiques tabulaires essentielles, le modèle effectue des transformations complexes dans les couches cachées pour fournir des probabilités précises et robustes pour une classification multi-classes adaptée à des données variées, comme illustré dans la figure 3.11.

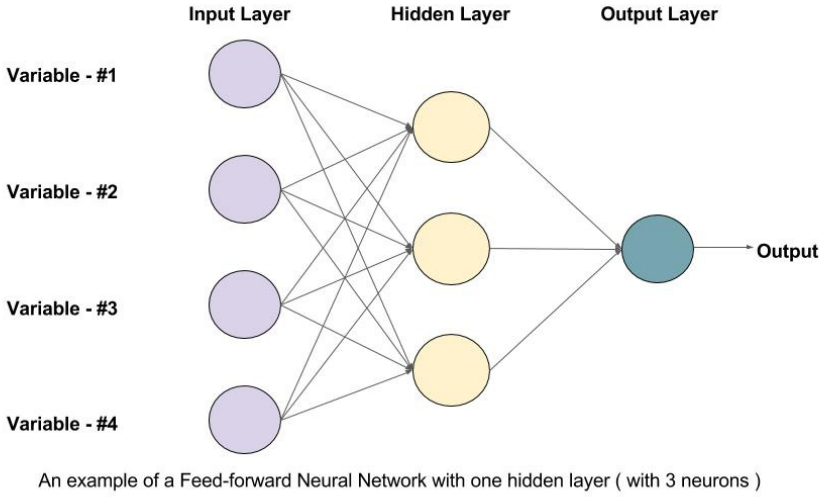

FIGURE 3.10 – Architecture du Réseau Neuronal Entièrement Connecté (FNN) [12]

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 256)	22272
batch_normalization (BatchNormalization)	(None, 256)	1024
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
batch_normalization_1 (BatchNormalization)	(None, 128)	512
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
batch_normalization_2 (BatchNormalization)	(None, 64)	256
dropout_2 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 4)	260

Total params: 65,476
Trainable params: 64,580
Non-trainable params: 896

FIGURE 3.11 – Structure de ce modèle FNN

La **Figure 3.11** ci-dessus montre les paramètres entraînables .Ce réseau neuronal utilise des couches denses, la normalisation par lots et le dropout pour optimiser l'apprentissage et éviter le sur-ajustement. La réduction progressive du nombre de neurones permet une meilleure extraction des caractéristiques, tandis que la normalisation stabilise l'entraînement et accélère la convergence. Le dropout, quant à lui, renforce la généralisation du modèle en évitant la dépendance excessive à certaines connexions.

Avec 64 580 paramètres entraînables, ce modèle trouve un bon équilibre entre complexité et efficacité, garantissant des performances optimales sans nécessiter une puissance de calcul excessive. Des améliorations pourraient inclure l'expérimentation de nouvelles fonctions d'activation ou l'ajustement des hyperparamètres pour affiner la robustesse et la précision des prédictions.
En exploitant pleinement la structure des données tabulaires, le FNN peut fournir des résultats précis pour des tâches de classification. Cependant, il présente des limites dans la capture des motifs localisés et des schémas complexes. Ces aspects nécessitent des architectures spécialisées, comme les réseaux convolutionnels (CNN), qui seront discutés dans la section suivante.

3.4.1.2 L'architecture d'un Réseau de Neurones Convolutionnel (CNN)

Les Réseaux de Neurones Convolutionnels (CNN) sont particulièrement adaptés à la classification du trafic darknet grâce à leur capacité à détecter des schémas complexes dans les données. Leur structure repose sur des couches spécialisées qui transforment les données brutes en informations exploitables.

1. **Couches convolutionnelles :** Ces couches appliquent des filtres pour extraire des motifs et des caractéristiques locales, tout en réduisant la complexité dimensionnelle des données.
2. **Couches de pooling :** Elles simplifient les données en réduisant leur taille avec des techniques comme le *max-pooling*, tout en conservant les informations essentielles et en améliorant la robustesse.
3. **Couches entièrement connectées :** Situées à la fin du réseau, elles interprètent les caractéristiques extraites pour produire les résultats finaux nécessaires à la classification.

Cette architecture hiérarchique permet aux CNN de gérer efficacement de grandes quantités de données et de capturer des motifs complexes et discrets dans le trafic réseau.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 84, 64)	256
batch_normalization (BatchNormalization)	(None, 84, 64)	256
max_pooling1d (MaxPooling1D)	(None, 42, 64)	0
dropout (Dropout)	(None, 42, 64)	0
conv1d_1 (Conv1D)	(None, 40, 128)	24704
batch_normalization_1 (BatchNormalization)	(None, 40, 128)	512
max_pooling1d_1 (MaxPooling1D)	(None, 20, 128)	0
dropout_1 (Dropout)	(None, 20, 128)	0
conv1d_2 (Conv1D)	(None, 18, 256)	98560
batch_normalization_2 (BatchNormalization)	(None, 18, 256)	1024
max_pooling1d_2 (MaxPooling	(None, 9, 256)	0

Figure 3.12 – Structure du modèle CNN employé (partie 1)

Notre étude met en évidence le rôle essentiel des paramètres entraînables dans la structure et l'efficacité du modèle CNN. Les poids et les biais des couches convolutionnelles, de pooling et entièrement connectées permettent au réseau d'extraire des motifs significatifs et d'améliorer la précision des classifications. En optimisant ces paramètres, il est possible d'affiner la capacité du modèle à reconnaître des schémas complexes dans le trafic darknet et à renforcer sa robustesse face aux variations des données.

Le nombre total de paramètres est un indicateur clé de la complexité et de l'adaptabilité du modèle. Une calibration adéquate garantit un équilibre entre puissance de calcul et

efficacité d'apprentissage, limitant le surajustement et favorisant la généralisation. Notre étude souligne également l'impact des ajustements des hyperparamètres et des stratégies d'optimisation sur la précision des prédictions, ouvrant la voie à des améliorations potentielles pour des applications spécifiques. Les paramètres entraînables illustrés dans les figures 3.12 et 3.13 jouent un rôle crucial dans cette analyse, offrant une vision détaillée de la configuration du modèle.

```
dropout_1 (Dropout)          (None, 20, 128)           0

conv1d_2 (Conv1D)            (None, 18, 256)           98560

batch_normalization_2 (Batc  (None, 18, 256)           1024
hNormalization)

max_pooling1d_2 (MaxPooling  (None, 9, 256)            0
1D)

dropout_2 (Dropout)          (None, 9, 256)            0

flatten (Flatten)            (None, 2304)              0

dense (Dense)                (None, 128)               295040

dropout_3 (Dropout)          (None, 128)               0

dense_1 (Dense)              (None, 4)                 516

=================================================================
Total params: 420,868
Trainable params: 419,972
Non-trainable params: 896
```

Figure 3.13 – Structure du modèle CNN employé (partie 2)

Aspect	FNN (Réseau Entièrement Connecté)	CNN (Réseau Convolutionnel)
Points forts	- Idéal pour les données tabulaires structurées. - Moins coûteux en calculs, avec un temps d'entraînement plus rapide.	- Capable de détecter des schémas complexes et relations spatiales. - Convient aux données volumineuses transformées en images.
Limitations	- Moins performant pour capturer des motifs localisés ou complexes. - Plus sujet au surapprentissage sur des ensembles de données complexes.	- Nécessite des prétraitements d'images sophistiqués. - Forte demande en ressources informatiques et temps d'entraînement.
Prétraitement	- Normalisation des données tabulaires.	- Transformation des données en images uniformes (grayscale).
Application	- Analyse des caractéristiques tabulaires des sessions réseau.	- Classification des caractéristiques représentées visuellement sous forme d'images.

Tableau 3.5 – Comparaison entre FNN et CNN pour la Classification du Trafic Réseau

Convolutional Neural Networks (CNN) et Feedforward Neural Networks (FNN) se complètent dans les tâches d'apprentissage profond, chaque modèle étant adapté à des objectifs spécifiques mais interconnectés. Les FNN excellent dans la capture des motifs globaux grâce à leurs couches entièrement connectées, tandis que les CNN se spécialisent dans la détection des relations localisées et des hiérarchies spatiales dans les données structurées.

Dans la classification du trafic darknet, les CNN identifient des motifs locaux complexes, tels que des variations subtiles dans le trafic, que les FNN intègrent ensuite en abstractions globales pour une classification précise. Cette synergie relie l'extraction de caractéristiques détaillées à la reconnaissance des motifs globaux, offrant une compréhension complète du jeu de données. **Le tableau 3.5** ci-dessus compare leurs rôles et forces uniques.

3.4.2 Entraînement de Modèle de Détection Darknet

L'entraînement des modèles de détection, basé sur les réseaux neuronaux CNN et FNN, vise à classifier le trafic du darknet. Cette section présente les étapes essentielles, depuis la préparation des données jusqu'à l'optimisation des modèles.Afin d'entraîner un modèle performant pour distinguer le trafic anonyme du trafic normal, une série d'étapes spécifiques a été suivie. Certaines étapes, comme le nettoyage et le prétraitement des données, étaient communes aux CNN et FNN, tandis que d'autres étaient adaptées aux particularités de chaque modèle. La procédure complète est détaillée ci-après :

3.4.2.1 Préparation des données

Le processus de préparation des données, commun aux architectures FNN et CNN, commence par l'acquisition de données brutes issues de flux de trafic réseau au format CSV, suivie du nettoyage initial **3.15** pour supprimer les colonnes non prédictives, convertir les horodatages et corriger les valeurs manquantes. L'ingénierie des caractéristiques **3.14** inclut l'extraction de fonctionnalités temporelles, des ratios de trafic et des indicateurs spécifiques à la sécurité. Après la normalisation des données et l'encodage des labels, un partitionnement stratifié divise le jeu de données en ensembles d'entraînement et de test tout en préservant la distribution des classes. Enfin, le déséquilibre des classes est traité par un resampling hybride combinant SMOTE et réduction aléatoire, aboutissant à un jeu de données équilibré et prêt pour l'entraînement des modèles.

```python
## 1. Data Loading and Initial Processing
def load_data(file_path):
    """Load and preprocess the dataset"""
    data = pd.read_csv(file_path)

    # Handle timestamps properly
    data['Timestamp'] = pd.to_datetime(data['Timestamp'], format='%d/%m/%Y %I:%M:%S %p', errors='coerce')

    # Drop redundant columns
    data = data.drop(['Flow ID', 'Src IP', 'Dst IP', 'Label.1'], axis=1)

    # Extract temporal features
    data['Hour'] = data['Timestamp'].dt.hour
    data['DayOfWeek'] = data['Timestamp'].dt.dayofweek
    data = data.drop('Timestamp', axis=1)

    return data

## 2. Feature Engineering
def engineer_features(data):
    """Create additional network traffic features"""
    # Basic traffic features
    data['Packet_Length_Ratio'] = data['Total Fwd Packet'] / (data['Total Bwd packets'] + 1)
    data['Duration_Per_Packet'] = data['Flow Duration'] / (data['Total Fwd Packet'] + data['Total Bwd packets'] + 1)

    # Tor-specific features
    tor_ports = [9001, 9030, 9040, 9050, 9051, 9150]
    data['Is_Common_Tor_Port'] = data['Dst Port'].isin(tor_ports).astype(int)

    # Encrypted traffic indicators
    data['Potential_Encrypted'] = ((data['Total Length of Fwd Packet'] == data['Total Length of Bwd Packet']) &
                                   (data['Total Fwd Packet'] == data['Total Bwd packets'])).astype(int)

    # Timing variability
    data['Timing_Variability'] = data['Flow IAT Std'] / (data['Flow IAT Mean'] + 1e-6)
```

Figure 3.14 – L'ingénierie des caractéristiques.

```python
## 3. Data Cleaning
def clean_data(data):
    """Handle missing and infinite values"""
    # Replace infinite values with NaN
    data = data.replace([np.inf, -np.inf], np.nan)

    # Fill missing values with median (for numerical features)
    numerical_cols = data.select_dtypes(include=['int64','float64']).columns
    data[numerical_cols] = data[numerical_cols].fillna(data[numerical_cols].median())

    return data

## 4. Data Preparation
def prepare_data(data):
    """Prepare data for modeling"""
    # Encode labels
    le = LabelEncoder()
    data['Label'] = le.fit_transform(data['Label'])

    # Split features and target
    X = data.drop('Label', axis=1)
    y = data['Label']

    # Split train/test
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42, stratify=y)

    # Scale features
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)
```

FIGURE 3.15 – Code en Python pour illustrer le nettoyage d'un jeu de données

3.4.2.2 Adaptation spécifique au modèle

a. Entraînement du FNN

Le processus d'entraînement du réseau de neurones FNN commence par l'intégration d'un jeu de données prétraité, composé de 42 caractéristiques de trafic réseau normalisées et équilibrées. Le modèle est structuré en trois couches cachées, avec 256, 128 et 64 neurones activés par ReLU, accompagnés de batch normalization et de dropout (30% initialement, 20% en couche finale) pour prévenir le surapprentissage. L'entraînement est réalisé avec l'optimiseur Adam à un taux d'apprentissage de 0,001, sur des mini-batches de 256 échantillons, avec un maximum de 100 époques et un arrêt anticipé après 10 époques sans amélioration du score de validation. Une réduction du taux d'apprentissage (facteur 0,1) est appliquée en cas de stagnation des performances sur 5 époques. Cette configuration permet une optimisation efficace des interactions globales des caractéristiques tout

en limitant le surajustement, malgré les 65 000 paramètres du modèle

b. Entraînement du CNN

L'entraînement du réseau CNN débute par la transformation des caractéristiques en tenseurs tridimensionnels, structurant les 42 variables sous forme de séquence 1D pour exploiter les relations spatiales. Trois blocs convolutionnels successifs (64→128→256 filtres, noyau=3) intègrent la normalisation par lots, le max pooling (taille=2) et un dropout spatial de 30%. Un pooling global moyen réduit le nombre de paramètres avant la classification. L'optimisation suit les réglages du FNN, mais chaque époque d'entraînement prend neuf fois plus de temps en raison des calculs convolutionnels. Enfin, la structure du CNN capture les corrélations locales entre les caractéristiques, notamment les relations temporelles des flux, améliorant ainsi la détection des anomalies du trafic réseau.

c. Évaluation du modèle

L'ensemble de données a été divisé en ensembles de validation et de test afin d'évaluer les performances du modèle en termes de précision, de rappel et de score F1, garantissant ainsi une mesure fiable de sa capacité à classifier correctement les flux de trafic réseau

d. Optimisation et ajustement

L'ensemble des hyperparamètres a été ajusté de manière progressive afin d'optimiser les performances du modèle. Une réduction adaptative du taux d'apprentissage a été appliquée pour éviter la stagnation, tandis que des techniques de régularisation, telles que le dropout et la normalisation par lots, ont été intégrées pour limiter le surajustement et améliorer la généralisation sur des données inconnues

e. Déploiement et utilisation

Le modèle entraîné visait une détection en temps réel du trafic darknet, mais dans cette étude, son évaluation s'est limitée à des tests sur 20% des données, assurant une validation avant une éventuelle intégration en production.

3.5 Configuration des Expériences et des Paramètres

Afin d'assurer la fiabilité et la reproductibilité des résultats, l'entraînement des modèles a été conçu suivant une configuration expérimentale rigoureuse. Chaque aspect, de la sélection des données à l'évaluation des performances, a été structuré pour garantir une

optimisation efficace du modèle tout en évitant les biais statistiques. Cette section détaille les choix méthodologiques adoptés pour entraîner et évaluer les architectures FNN et CNN dans le cadre de la classification du trafic réseau.

3.5.1 Choix des données et partitionnement

Les données utilisées proviennent de flux de trafic réseau, prétraitées pour assurer leur qualité. Un partitionnement stratifié (80% entraînement, 20% test) a été réalisé afin de préserver la distribution des classes. Une validation croisée a été intégrée pour ajuster les hyperparamètres et améliorer la généralisation des modèles.

3.5.2 Prétraitement des données

Le prétraitement a impliqué l'élimination des valeurs aberrantes, la normalisation des caractéristiques et l'encodage des labels catégoriques. Des techniques d'équilibrage de classes telles que SMOTE ont été appliquées pour corriger les déséquilibres du jeu de données et renforcer la robustesse du modèle.La figure **3.15** et **3.14**montre quelque étape suivi ici.

3.5.3 Architecture du modèle

Deux architectures ont été conçues : un réseau de neurones FNN et un CNN. Le FNN est structuré en trois couches cachées avec des activations ReLU et régularisation par dropout, tandis que le CNN exploite des couches convolutionnelles 1D et un pooling global moyen pour capturer les relations spatiales entre les caractéristiques du trafic réseau.

3.5.4 Hyperparamètres d'entraînement

Les hyperparamètres ont été optimisés manuellement . Le taux d'apprentissage initial a été fixé à 0,001 avec ajustements manuel en cas de stagnation. La taille des mini-batches et les taux de dropout ont été ajustés pour maximiser la convergence sans surajustement.

3.5.5 Fonction de perte et optimiseur

Les modèles ont été entraînés avec la fonction de perte *sparse_ categorical_ entropy*, adaptée aux tâches de classification multi-classes. L'optimisation a été réalisée via l'algorithme Adam, sélectionné pour sa capacité à ajuster dynamiquement les gradients et accélérer la convergence.

3.5.6 Métriques d'évaluation

La performance des modèles a été évaluée à l'aide de métriques clés : précision, perte et score F1. Le suivi de ces métriques après chaque époque a permis d'ajuster les paramètres et d'assurer une robustesse optimale du modèle face aux variations des données.

3.6 Méthodes d'évaluation de la performances du modèle

Pour garantir une mesure fiable des performances du modèle et ajuster ses hyperparamètres, plusieurs métriques ont été suivies durant l'entraînement et les tests. Cette section détaille les indicateurs utilisés ainsi que leur rôle dans l'analyse des résultats.

3.6.1 Matrice de Confusion

La matrice de confusion est essentielle pour évaluer les performances du modèle dans la classification du trafic darknet. Elle permet de visualiser les prédictions correctes et incorrectes, facilitant l'extraction de métriques clés telles que la précision, le rappel et le score F1.

1. **Vrai Positif (TP)** : Instances correctement classées dans leur catégorie réelle (ex. trafic *Tor VPN* bien identifié).
2. **Faux Positif (FP)** : Instances incorrectement attribuées à une classe (ex. *Non-Tor VPN* classé comme *Tor VPN*).
3. **Vrai Négatif (TN)** : Exclusions correctes des classes non correspondantes (ex. *Non-Tor Non-VPN* bien identifié).
4. **Faux Négatif (FN)** : Instances mal classées dans une autre catégorie (ex. *Tor Non-VPN* classé comme *Non-Tor Non-VPN*).

L'analyse de ces composantes est fondamentale pour mesurer la robustesse du modèle et affiner les métriques d'évaluation.

3.6.2 Exactitude (Accuracy)

L'exactitude mesure le pourcentage de prédictions correctes du modèle, calculé comme suit :

$$\text{Exactitude} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Elle évalue l'efficacité globale du modèle sur les classes de trafic darknet (*Tor VPN*, *Non-Tor VPN*, *Tor Non-VPN*, *Non-Tor Non-VPN*). Cependant, une exactitude élevée peut être trompeuse en cas de déséquilibre des classes, nécessitant l'usage de métriques comme le rappel et le score F1 pour une évaluation plus fine.

3.6.3 Précision

La précision évalue la proportion de prédictions positives correctes, calculée par :

$$\text{Précision} = \frac{TP}{TP + FP} \tag{3.2}$$

Elle mesure la capacité du modèle à minimiser les fausses alarmes dans la classification du trafic darknet.

3.6.4 Rappel

Le rappel mesure la capacité du modèle à identifier correctement les instances positives, calculé par :

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

Il est essentiel pour évaluer la détection des flux de trafic darknet sans omettre de véritables cas.

3.6.5 Score F1

Le score F1 est une mesure équilibrée entre la précision et le rappel, calculé par :

$$\text{Score F1} = \frac{2 \times \text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \tag{3.4}$$

Il est particulièrement utile lorsque les classes sont déséquilibrées, assurant une meilleure évaluation de la performance globale du modèle.

3.6.6 Taux de Faux Positifs (FPR)

Le taux de faux positifs (*False Positive Rate* - FPR) mesure la proportion de classifications incorrectes parmi les instances négatives. Il est défini par :

$$\text{FPR} = \frac{FP}{FP + TN} \tag{3.5}$$

Un FPR élevé indique que le modèle produit trop de fausses alertes, ce qui peut compromettre la fiabilité de la détection du trafic darknet.

Les bases méthodologiques détaillées dans cette étude ont structuré l'approche adoptée, en couvrant les étapes clés de l'entraînement, de l'évaluation et de l'optimisation du modèle. Ces fondations méthodologiques ont permis d'assurer la précision et la robustesse du modèle dans le cadre spécifique de la classification du trafic darknet. Le prochain chapitre se focalisera sur une analyse approfondie des résultats obtenus. En s'appuyant sur les métriques introduites ici, il explorera l'efficacité pratique du modèle et son potentiel d'application dans des scénarios réels.

3.7 Conclusion

Ce chapitre a posé les bases méthodologiques essentielles pour la conception et l'évaluation du modèle de classification du trafic darknet. Il a couvert la préparation des données, l'entraînement et l'évaluation, garantissant la fiabilité et la reproductibilité des résultats. Les critères d'évaluation, comme la matrice de confusion, l'exactitude, le rappel, le score F1 et le taux de faux positifs, constituent les fondations analytiques nécessaires pour interpréter les performances du modèle.
Le chapitre suivant approfondira cette analyse afin d'évaluer l'efficacité pratique du modèle pour la détection en temps réel du trafic darknet.

Chapitre 4

Résultats et analyses

4.1 Introduction

Ce chapitre analyse les résultats obtenus à partir du modèle de classification développé, en évaluant sa performance dans la détection et la classification des différents types de trafic réseau : anonymisé (Tor et VPN/Darknet) et non anonymisé (trafic normal).
Nous examinerons les performances du modèle de deep learning à travers des métriques clés, notamment l'exactitude, la précision, le rappel, le score F1 et l'analyse des matrices de confusion. Ces matrices permettent d'identifier en détail les classifications correctes, les faux positifs, les faux négatifs, et leur impact sur les résultats globaux.
Enfin, nous proposerons des solutions futures basées sur cette analyse pour intégrer ces résultats dans un système de détection d'intrusion (IDS) ou tout autre environnement adapté, afin de renforcer la sécurité réseau et répondre aux besoins spécifiques de ce domaine.

4.2 Évaluation des performances

Après avoir présenté les objectifs, nous évaluons les performances des modèles CNN et FNN dans la classification du trafic réseau. Cette évaluation utilise des métriques clés telles que l'exactitude, la précision, le rappel, le score F1 et les matrices de confusion, afin d'examiner leur efficacité dans la détection des différents types de trafic.
Les modèles ont été entraînés sur le dataset CIC-Darknet, avec un taux d'apprentissage fixé à 0.001 sur 100 époques. Les données ont été réparties en 80 % pour l'entraînement et la validation, et 20 % pour le test. Le processus d'entraînement a duré 733 secondes pour FNN et 7306 secondes pour le CNN , permettant d'étudier les tendances observées et de comparer les erreurs identifiées.

4.2.1 Performance au Cours de l'Entraînement

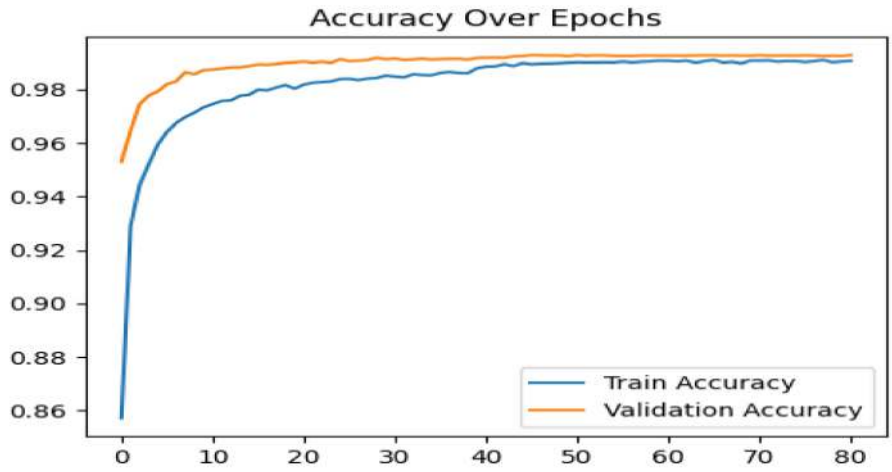

FIGURE 4.1 – Précision globale du modèle FNN

La figure 4.1 montre la précision globale de l'entraînement pour le modèle FNN progresse de 0.86 à un plateau de 0.98 après 40 époques. La précision de validation, quant à elle, se stabilise à 0.98 dès la 20ème époque, montrant une bonne généralisation sur des données non vues. L'alignement des deux courbes indique une absence de surapprentissage et une performance fiable.

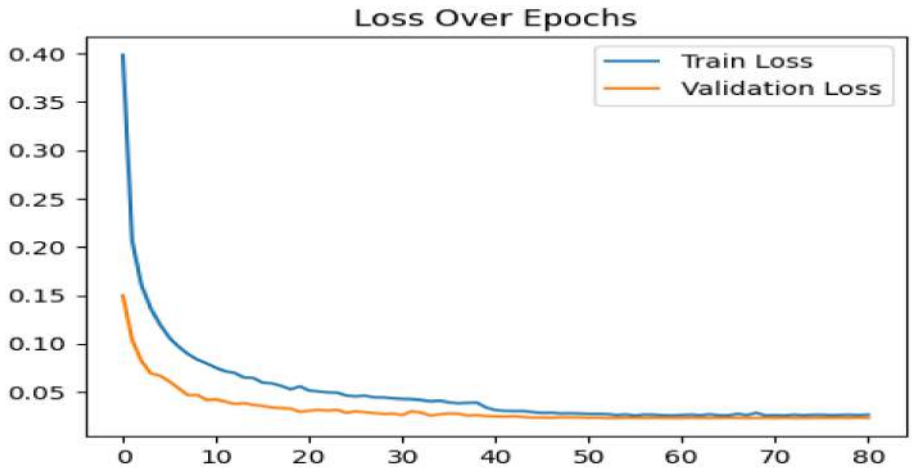

FIGURE 4.2 – Perte du modèle FNN

La figure 4.2 montre la perte de l'entraînement pour le modèle FNN diminue rapidement de 0.40 à un plateau de 0.02 après 30 époques. La perte de validation commence plus bas et se stabilise également à 0.02 après 20 époques. La diminution parallèle des deux courbes reflète une optimisation efficace et l'absence de surapprentissage ou de sous-apprentissage.

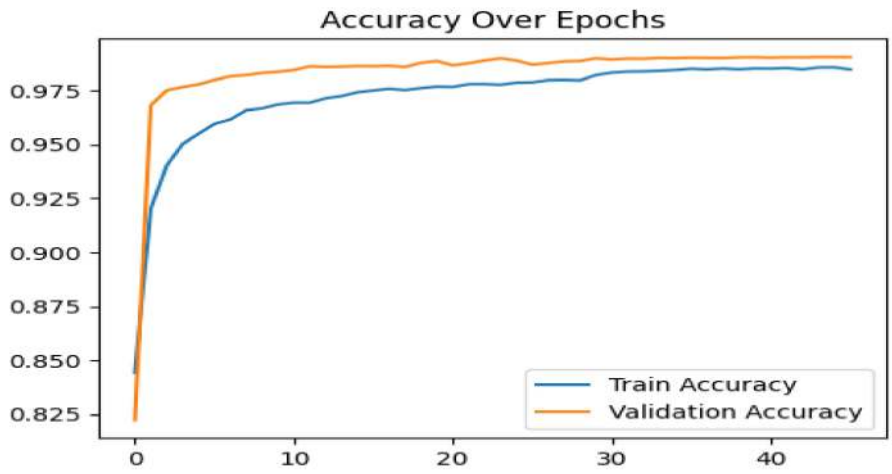

FIGURE 4.3 – Précision globale du modèle CNN

La figure 4.3 montre la précision globale de l'entraînement pour le modèle CNN progresse de 0.825 à 0.975 sur 45 époques. La précision de validation suit une trajectoire similaire, atteignant rapidement un plateau à 0.975, légèrement au-dessus de celle de l'entraînement. Cette convergence montre une bonne généralisation aux données non vues et une absence de surapprentissage.

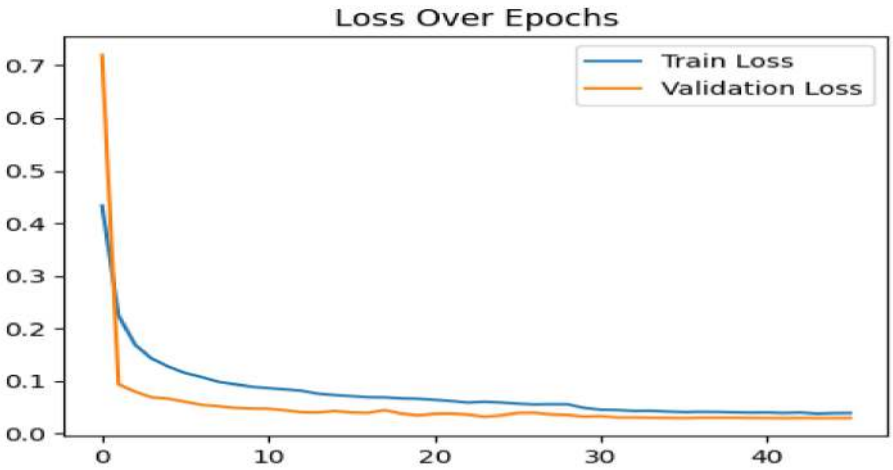

FIGURE 4.4 – Perte du modèle CNN

La figure 4.4 montre la perte d'entraînement pour le modèle CNN diminue rapidement de 0.4 à moins de 0.05 après 30 époques. La perte de validation, initialement plus élevée à 0.7, suit la même tendance et converge vers 0.05. L'alignement des courbes reflète une optimisation efficace et un apprentissage équilibré, sans surapprentissage ni sous-apprentissage.

4.2.2 Performance Globale

Métrique	CNN	FNN
Précision globale	0.9915	0.9921
Score F1	0.9915	0.9921
Précision Macro	0.98	0.98
Rappel Macro	0.99	0.99
Score F1 Macro	0.99	0.98
Précision pondérée	0.99	0.99
Rappel pondéré	0.99	0.99
Score F1 pondéré	0.99	0.99

Tableau 4.1 – Comparaison des résultats d'évaluation entre FNN et CNN

Comme le montre le tableau 4.1, la comparaison entre CNN et FNN met en lumière leurs performances remarquables en classification du trafic réseau. Bien que le CNN soit performant, le FNN le dépasse légèrement avec une précision (0.9921 contre 0.9915) et un score F1 supérieurs, grâce à une architecture mieux adaptée aux relations numériques et catégoriques des données. Les deux modèles présentent des performances équilibrées dans toutes les catégories de trafic, avec des moyennes macro et pondérées élevées. Ainsi, la meilleure généralisation du FNN en fait un choix optimal pour cette tâche spécifique.

4.2.3 Performance par catégorie de Trafic

Catégorie de trafic	Métrique	CNN	FNN
Non-Tor	Précision	1.00	1.00
	Rappel	1.00	1.00
	Score F1	1.00	1.00
NonVPN	Précision	0.98	0.98
	Rappel	0.97	0.98
	Score F1	0.98	0.98
Tor	Précision	0.98	0.97
	Rappel	1.00	1.00
	Score F1	0.99	0.98
VPN	Précision	0.97	0.98
	Rappel	0.99	0.98
	Score F1	0.98	0.98

Tableau 4.2 – Comparaison des performances par catégorie de trafic entre CNN et FNN

Le tableau 4.2 illustre les performances exceptionnelles des modèles CNN et FNN dans la classification du trafic réseau. Les deux modèles obtiennent des scores parfaits en précision,

rappel et score F1 (1.00) pour la catégorie Non-Tor, montrant une classification sans erreurs. Pour NonVPN, le FNN se distingue avec un rappel légèrement supérieur (0.98 contre 0.97) tout en conservant une précision et un score F1 de 0.98. Dans la catégorie Tor, le CNN offre une meilleure précision (0.98 contre 0.97), avec un rappel parfait (1.00) pour les deux modèles, et un score F1 légèrement supérieur (0.99 contre 0.98). Enfin, pour la catégorie VPN, le FNN excelle en précision (0.98 contre 0.97) tandis que le CNN affiche un meilleur rappel (0.99 contre 0.98), conduisant à des scores F1 identiques de 0.98. Bien que les deux modèles soient très performants, le FNN présente une meilleure généralisation dans certaines catégories, le positionnant comme le modèle privilégié, avec un CNN restant compétitif.

4.2.4 Matrices de Confusion

La matrice de confusion est utilisée pour analyser les performances des modèles CNN et FNN dans la classification du trafic réseau en catégories telles que Non-Tor, NonVPN, Tor et VPN. Elle fournit une représentation claire des prédictions correctes et des erreurs de classification pour chaque catégorie, permettant de calculer des métriques essentielles comme la précision, le rappel et le score F1. Ces informations révèlent les forces et faiblesses de chaque modèle, aidant à identifier le modèle le plus adapté à cette tâche tout en orientant les améliorations futures.

Figure 4.5 – Matrice de confusion FNN

La matrice de confusion est composée de quatre parties comme suit :

1. **Vrais Positifs (TP)**
   Les Vrais Positifs représentent les instances correctement classées par le modèle dans leurs catégories réelles. Dans toutes les catégories de trafic, le modèle a obtenu un total de 31 472 Vrais Positifs, incluant 22 012 instances Non-Tor, 4 650 instances NonVPN, 275 instances Tor, et 4 535 instances VPN. Cela reflète la forte capacité du modèle à prédire avec précision les types de trafic, la majorité des instances correctement classées appartenant à la catégorie Non-Tor en raison de sa plus grande représentation dans l'ensemble des données.

2. **Vrais Négatifs (TN)**
   Les Vrais Négatifs correspondent aux instances correctement exclues d'une catégorie et attribuées aux types de trafic corrects. Le modèle totalise 94 416 Vrais Négatifs, répartis comme suit : pour Non-Tor, 4 650 NonVPN, 275 Tor, et 4 535 VPN ont été correctement exclus. Pour NonVPN, 22 012 Non-Tor, 275 Tor, et 4 535 VPN ont été correctement identifiés comme ne lui appartenant pas. Concernant Tor, 22 012 Non-Tor, 4 650 NonVPN, et 4 535 VPN ont été exclus avec précision. Enfin, pour VPN, 22 012 Non-Tor, 4 650 NonVPN, et 275 Tor ont été rejetés correctement. Ce résultat démontre la capacité du modèle à éviter les erreurs de classification.

3. **Faux Négatifs (FN)**
   Les Faux Négatifs désignent les instances où le modèle n'a pas prédit la bonne catégorie, les classant dans des catégories incorrectes. Un total de 109 Faux Négatifs a été enregistré : 21 instances Non-Tor ont été mal classées en NonVPN, Tor ou VPN, 49 instances NonVPN en Non-Tor, Tor ou VPN, 2 instances Tor en Non-Tor, NonVPN ou VPN, et 37 instances VPN en Non-Tor, NonVPN ou Tor. Ces erreurs révèlent des difficultés du modèle à distinguer certaines catégories, notamment NonVPN et VPN.

4. **Faux Positifs (FP)**
   Les Faux Positifs représentent les instances où le modèle a prédit une catégorie incorrecte par rapport au label réel. Au total, 106 Faux Positifs ont été relevés : 77 instances ont été mal classées en Non-Tor, incluant 49 NonVPN, 2 Tor, et 26 VPN. Pour NonVPN, seulement 2 instances d'autres catégories ont été mal classées. Tor compte 1 instance mal prédite. Enfin, la catégorie VPN a 26 Faux Positifs, où le trafic de Non-Tor, NonVPN ou Tor a été incorrectement classé en VPN. Ces erreurs mettent en lumière des chevauchements occasionnels entre les caractéristiques des types de trafic, surtout entre Non-Tor et les autres catégories.

FIGURE 4.6 – Matrice de confusion CNN

L'image 4.6 ci-dessus illustre les composants de la matrice de confusion, fournissant une répartition des éléments clés comme suit :

1. **Vrais Positifs (TP)**
   Les Vrais Positifs représentent les instances correctement classées par le modèle dans leurs catégories réelles. Dans toutes les catégories de trafic, le modèle a obtenu un total de 31 454 Vrais Positifs, incluant 21 989 instances Non-Tor, 4 640 instances NonVPN, 278 instances Tor, et 4 547 instances VPN. Cela reflète la capacité du modèle à prédire avec précision les types de trafic.
2. **Vrais Négatifs (TN)**
   Les Vrais Négatifs correspondent aux instances correctement exclues d'une catégorie et attribuées aux types de trafic corrects. Le modèle totalise 94 416 Vrais Négatifs, répartis comme suit : pour Non-Tor, 4 640 NonVPN, 278 Tor, et 4 547 VPN ont été correctement exclus. Pour NonVPN, 21 989 Non-Tor, 278 Tor, et 4 547 VPN ont été correctement identifiés comme ne lui appartenant pas. Concernant Tor, 21 989 Non-Tor, 4 640 NonVPN, et 4 547 VPN ont été exclus avec précision. Enfin, pour VPN, 21 989 Non-Tor, 4 640 NonVPN, et 278 Tor ont été rejetés correctement.
3. **Faux Négatifs (FN)**
   Les Faux Négatifs désignent les instances où le modèle n'a pas prédit la bonne catégorie, les classant dans des catégories incorrectes. Un total de 118 Faux Négatifs a été enregistré : 71 instances Non-Tor ont été mal classées en NonVPN, Tor ou VPN, 115 instances NonVPN en Non-Tor, Tor ou VPN, 0 instances Tor mal classées en Non-Tor, NonVPN ou VPN, et 37 instances VPN en Non-Tor, NonVPN ou Tor.

4. **Faux Positifs (FP)**
   Les Faux Positifs représentent les instances où le modèle a prédit une catégorie incorrecte par rapport au label réel. Au total, 48 Faux Positifs ont été relevés : 26 instances ont été mal classées en Non-Tor, incluant 14 NonVPN, 0 Tor, et 8 VPN. Pour NonVPN, 2 instances d'autres catégories ont été mal classées. Tor compte 0 instance mal prédite. Enfin, la catégorie VPN a 20 Faux Positifs, où le trafic de Non-Tor, NonVPN ou Tor a été incorrectement classé en VPN.

4.3 Analyse comparative des modèles d'apprentissage automatique et profond

Pour comprendre pourquoi les modèles sélectionnés ont été choisis pour cette tâche, nous avons entraîné et évalué plusieurs autres modèles en utilisant des techniques classiques d'apprentissage automatique, telles que les machines à vecteurs de support (SVM), les arbres de décision (DCT) et les forêts aléatoires (RF), ainsi que des techniques d'apprentissage profond comme les réseaux neuronaux convolutionnels (CNN) et les réseaux neuronaux feedforward (FNN). Cette analyse comparative met en évidence les forces et les limites de chaque approche, offrant des éclaircissements sur le choix final des modèles

4.3.0.1 Aperçu des performances

La Forêt Aléatoire (RF) s'est démarquée comme le modèle le plus efficace avec une précision de 0,9962 et un F1-score de 0,9963, offrant une classification fiable pour toutes les catégories de trafic : Non-Tor, NonVPN, Tor et VPN. Le CNN suit de près avec une précision et un F1-score dépassant 0,99, soulignant sa capacité à extraire des caractéristiques complexes.

Le FNN affiche des résultats solides, comparables au CNN, ce qui en fait une option compétitive. L'Arbre de Décision (DCT), avec une précision de 0,9834 et un F1-score de 0,9835, offre des performances correctes mais reste légèrement en retrait par rapport aux modèles basés sur les réseaux neuronaux. Enfin, le SVM, avec une précision de 0,9580, montre des difficultés, notamment avec les trafics Tor et NonVPN, révélant ses limites face aux ensembles de données complexes ou déséquilibrés.
Le **tableau 4.3** ci-dessous présente un résumé comparatif des performances des modèles entraînés.

Modèle	Précision	F1-Score	Temps d'entraînement (secondes)	Temps d'inférence (secondes)
SVM	0,9580	0,9584	2489,87	165,45
Arbre de Décision (DCT)	0,9834	0,9835	4,35	0,023
Forêt Aléatoire (RF)	0,9962	0,9963	43,11	0,70
FNN	0,9921	0,9921	733,90	11,79
CNN	0,9915	0,9921	7306,85	23,13

Tableau 4.3 – Comparaison des performances des modèles de classification de trafic.

4.3.0.2 Efficacité des ressources

Comme le montre la **figure 4.7**, l'efficacité et l'utilisation des ressources des modèles peuvent être évaluées en tenant compte de facteurs tels que le temps nécessaire à leur entraînement.

En termes d'exigences computationnelles, l'Arbre de Décision s'est révélé le plus efficace, avec un temps d'entraînement de 4,35 secondes et un temps d'inférence de 0,023 seconde, ce qui le rend adapté aux systèmes en temps réel et aux scénarios nécessitant un déploiement rapide.

La Forêt Aléatoire (RF) offre un bon équilibre entre performance et efficacité, avec un temps d'entraînement modéré de 43,11 secondes et un temps d'inférence de 0,70 seconde, garantissant un compromis raisonnable. Les CNN et FNN, bien qu'affichant une précision exceptionnelle, présentent des exigences computationnelles significatives : le CNN nécessite 7306,85 secondes pour l'entraînement et 23,13 secondes pour l'inférence, tandis que le FNN demande 733,90 secondes pour l'entraînement et 11,79 secondes pour l'inférence.

Ces contraintes peuvent limiter leur application dans des environnements aux ressources limitées. Le SVM, avec un temps d'inférence de 165,45 secondes, s'est avéré le moins efficace, rendant son utilisation moins pratique pour les tâches de classification de trafic en temps réel.

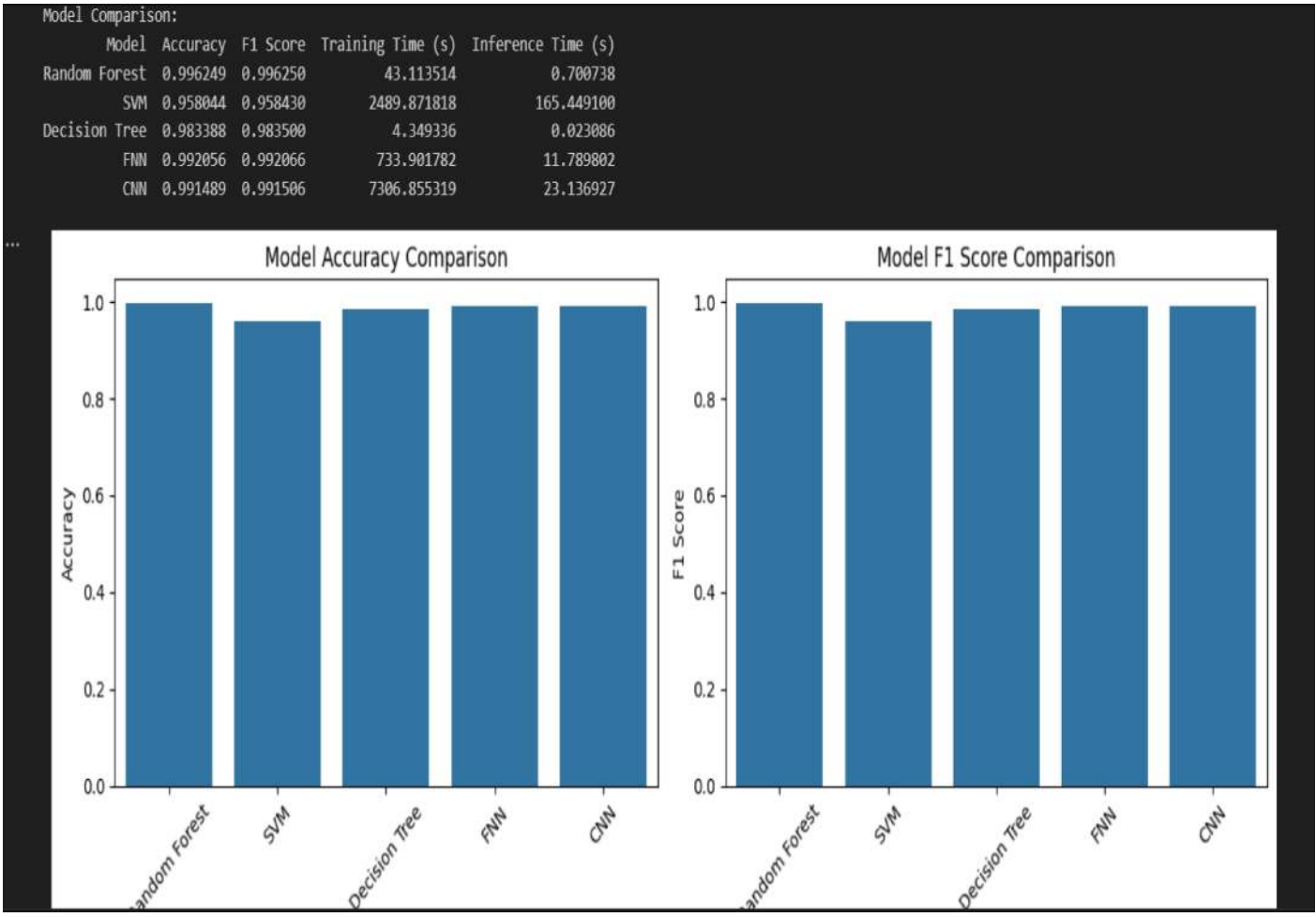

FIGURE 4.7 – efficacité de ressource

4.3.0.3 Analyse spécifique par catégorie

En examinant les catégories de trafic présentés dans le **Tableau 4.4**, les modèles RF, CNN et FNN ont excellé, en particulier pour la détection des trafics Non-Tor et NonVPN, avec des performances parfaites en précision et rappel dans certains cas pour RF. Le CNN a démontré des forces similaires, mais nécessite une optimisation pour accélérer son inférence et améliorer son applicabilité en temps réel. L'Arbre de Décision a montré des performances constantes et efficaces, en faisant un bon choix pour des applications légères. Le SVM a rencontré des difficultés en termes de précision et rappel pour les trafics Tor et VPN, limitant sa fiabilité dans ces catégories. Ces observations suggèrent que RF est le modèle le plus adapté pour des systèmes comme les Systèmes de Détection d'Intrusion (IDS) grâce à sa haute précision et son équilibre computationnel. Les CNN et FNN sont recommandés pour les tâches nécessitant une extraction de caractéristiques complexes, tandis que l'Arbre de Décision offre une solution légère pour des classifications

rapides dans des environnements contraints. Les améliorations futures pourraient inclure des modèles hybrides combinant RF avec CNN ou FNN, des architectures CNN optimisées, ou un enrichissement des caractéristiques pour les types de trafic présentant des taux de détection plus faibles.

Catégorie de trafic	Modèle	Précision	Rappel	F1-Score
Non-Tor	Random Forest	1.00	1.00	1.00
	CNN	1.00	1.00	1.00
	FNN	1.00	1.00	1.00
	Arbre de Décision (DCT)	1.00	0.99	0.99
	SVM	0.99	0.98	0.99
NonVPN	Random Forest	0.99	0.99	0.99
	CNN	0.99	0.98	0.99
	FNN	0.99	0.98	0.99
	Arbre de Décision (DCT)	0.94	0.96	0.95
	SVM	0.91	0.87	0.89
Tor	Random Forest	0.98	0.97	0.98
	CNN	0.98	0.97	0.98
	FNN	0.97	0.96	0.97
	Arbre de Décision (DCT)	0.97	0.98	0.97
	SVM	0.63	0.98	0.77
VPN	Random Forest	0.99	0.99	0.99
	CNN	0.98	0.98	0.98
	FNN	0.98	0.97	0.98
	Arbre de Décision (DCT)	0.96	0.98	0.97
	SVM	0.90	0.92	0.91

Tableau 4.4 – Performance spécifique des modèles par catégorie de trafic.

4.4 Orientations futures et applications pratiques

À l'avenir, l'intégration de notre modèle entraîné CNN et FNN dans un Système de Détection d'Intrusion (IDS) se fera via une pipeline agile et adaptée aux environnements modernes. Les modèles seront préparés pour le déploiement dans des formats tels que ONNX ou TensorFlow SavedModel, garantissant leur compatibilité avec les infrastructures existantes. Ces modèles traiteront des flux de données en temps réel grâce à des solutions comme Apache Kafka, permettant une analyse rapide des journaux de trafic et une identification précoce des anomalies. Intégrés dans des frameworks IDS comme Snort++ ou Zeek, nos modèles offriront une détection avancée des menaces, renforçant les systèmes traditionnels et s'adaptant aux attaques émergentes. Une phase de validation rigoureuse, incluant des tests sur des données réelles et historiques, assurera leur précision

tout en minimisant les erreurs de détection.
Pour le déploiement, notre modèle pourra être utilisé comme service hébergé dans le cloud ou intégré localement dans des dispositifs réseau. Hébergé dans le cloud, il sera accessible via des APIs REST développées avec FastAPI ou Flask et déployées sur des plateformes comme AWS Lambda ou Azure Functions, garantissant une évolutivité facile pour gérer des charges de trafic variables. En mode local, notre modèle sera optimisé par des techniques telles que la quantification pour réduire les exigences en ressources et intégré dans des dispositifs comme les pare-feux ou passerelles IoT. Cette double approche garantit une classification rapide et efficace des trafics en temps réel, même dans des environnements à ressources limitées, tout en offrant une flexibilité pour répondre à des menaces complexes et adaptatives. Avec des boucles de rétroaction intégrées, le modèle pourra évoluer face aux nouveaux schémas d'attaques, garantissant une protection durable et proactive.

4.5 Conclusion

En résumé, les résultats obtenus démontrent la robustesse et l'efficacité de notre modèle entraîné CNN et FNN pour la détection des anomalies et la classification des trafics en temps réel. Ces modèles se distinguent par leur capacité à s'adapter à des schémas de trafic complexes, tout en assurant une grande précision et une latence réduite dans des environnements à ressources limitées. Les implications pratiques de ces résultats sont significatives, offrant des solutions applicables aux Systèmes de Détection d'Intrusion et à d'autres domaines critiques, tels que la cybersécurité et la surveillance des réseaux. En perspective, ce travail ouvre la voie à des recherches futures sur l'intégration de modèles hybrides, l'optimisation des frameworks de déploiement, et le traitement de nouvelles menaces adaptatives. Ces avancées renforceront la capacité des systèmes à anticiper et à répondre aux défis émergents de manière proactive

Conclusion Générale

Dans ce mémoire, nous avons démontré le rôle transformateur de l'intelligence artificielle (IA) dans la résolution des défis complexes en cybersécurité, en particulier pour la détection du trafic darknet. En utilisant nos modèles entraînés, notamment le CNN et le FNN, nous avons prouvé leur capacité à classifier les schémas de trafic, détecter les anomalies et distinguer les activités légitimes des activités malveillantes sur divers environnements réseau. Ces modèles ont atteint une grande précision dans l'identification du trafic darknet et se sont montrés adaptables pour traiter les données chiffrées et gérer les menaces émergentes, s'affirmant ainsi comme des outils clés pour renforcer les cadres de sécurité réseau modernes.
Les résultats de notre travail soulignent l'importance des approches basées sur l'IA pour améliorer les Systèmes de Détection d'Intrusion (IDS) et d'autres solutions de cybersécurité. Grâce à un prétraitement rigoureux, une analyse en temps réel et un déploiement flexible, nos modèles peuvent être intégrés dans des systèmes pratiques pour faire face aux tactiques sophistiquées employées sur le darknet. Les perspectives futures de cette recherche incluent le perfectionnement des modèles pour une efficacité accrue, l'exploration de modèles hybrides et l'examen des implications éthiques liées à la surveillance du trafic chiffré. Ce travail ouvre la voie à des systèmes robustes et évolutifs capables de protéger les infrastructures numériques contre les menaces dissimulées.

Cependant, nous avons également rencontré des défis majeurs dans ce mémoire. La faible quantité de données pour certaines catégories de trafic, comme le trafic Tor et VPN, a limité la généralisation des modèles dans ces cas. Nous avons dû entreprendre un processus de nettoyage et d'équilibrage des données, ce qui a nécessité un effort considérable pour garantir leur fiabilité. En outre, les modèles de deep learning, bien qu'efficaces, requièrent d'importantes ressources matérielles, une quantité significative de données annotées et un temps conséquent pour leur entraînement et leur optimisation. Ces difficultés mettent en évidence les défis futurs à relever pour améliorer l'efficacité et l'applicabilité des modèles IA dans la détection du trafic darknet.

Bibliographie

[1] ZDNet. Surface Web vs Deep Web vs Dark Web. https://zd-brightspot.s3.us-east-1.amazonaws.com/wp-content/uploads/2022/05/05041108/surface-web-vs-deep-web-vs-dark-web.png, May 2022.

[2] Unknown. Network Visualization. https://geti2p.net/_static/images/net.png, April 2025. Image from the I2P website showing a network visualization.

[3] Unknown. Fiber Optic Network Infrastructure. https://media.fs.com/images/community/erp/B5RpN_20230129105001EhCyj.jpg, April 2025. Image showcasing fiber optic network infrastructure from FS community.

[4] Machine Learnia. Machine Learnia : Apprendre le Machine Learning. https://www.machinelearnia.com. Consulté le 13 avril 2025.

[5] MathWorks. Apprentissage non supervisé - MATLAB & Simulink. https://fr.mathworks.com/discovery/unsupervised-learning.html. Consulté le 14 avril 2025.

[6] DataCamp. Introduction to k-Means Clustering with scikit-learn in Python. https://www.datacamp.com/tutorial/k-means-clustering-python. Consulté le 14 avril 2025.

[7] Cloudflare. Neural Network Diagram. https://cf-assets.www.cloudflare.com/slt3lc6tev37/1wkNx98skWwkKAw2XExpQe/33505b0b82e3156fc042bca42a1a2034/neural-network-diagram.png, 2022.

[8] Cloudflare. Feed-Forward Neural Network Diagram. https://cf-assets.www.cloudflare.com/slt3lc6tev37/6o5Um8xfA4q6xehc24q0jJ/fba7c550719cfe51c3ca3fd314e716fb/feed-forward-neural-network-diagram.png, 2022.

[9] Cloudflare. Multilayer Perceptron Neural Network Diagram. https://cf-assets.www.cloudflare.com/slt3lc6tev37/1I459b6ne8fror4e5XNbxJ/6527eabd1e8c0042121242566dbf52ec/multilayer-perceptron-neural-network.png, 2022.

[10] Intellipaat. CNN Architecture Diagram. https://intellipaat.com/blog/wp-content/uploads/2022/02/CNN-Architecture.png, 2022.

[11] Inside Machine Learning. Fonction d'activation : Comment ça marche ? Une explication simple. https://inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/. Consulté le 15 avril 2025.

[12] Scaler. Non-Linear Classification Diagram. https://www.scaler.com/topics/images/non-linear-classification.webp, 2022.

[13] R. Stamboliyska. *La face cachée d'internet : hackers, dark net...* Larousse, 2017.

[14] PrivMX. Privacy vs Anonymity. https://privmx.com/blog/33/privacy-vs-anonymity, June 2022.

[15] Steven Gates. *Réseau Anonyme TOR 101.* BoD – Books on Demand.

[16] Unknown. Tor Network Visualization. https://sysblog.informatique.univ-paris-diderot.fr/wp-content/uploads/2020/03/tor1-768x411.png, April 2025. Image depicting the Tor network from Université Paris Diderot's blog.

[17] Kaspersky. What is the Tor Browser ? https://www.kaspersky.fr/resource-center/definitions/what-is-the-tor-browser.

[18] Daniel Echeverri Montoya. *Deep web : TOR, FreeNET & I2P Privacidad y Anonimato.* ZeroXword Computing.

[19] Ivacy. Détails et fonctionnement d'I2P. https://www.ivacy.com/blog/fr/details-et-fonctionnement-i2p/.

[20] Fortinet. Les avantages des VPN. https://www.fortinet.com/fr/resources/cyberglossary/benefits-of-vpn.

[21] Samuel. Quelles sont les meilleures solutions pour surfer dans l'anonymat ? http://www.calvados-strategie.com/quelles-sont-les-meilleures-solutions-pour-surfer-dans-lanonymat%E2%80%89/, août 2019.

[22] OnlineSim. The Dark Side of the Internet Part 3 : What is Freenet ? https://onlinesim.io/instructions/the-dark-side-of-the-internet-part-3-what-is-freenet?utm_referrer=https://www.google.com/.

[23] Google Cloud. Qu'est-ce que l'intelligence artificielle ? https://cloud.google.com/learn/what-is-artificial-intelligence?hl=fr#artificial-intelligence-defined. Consulté le 13 avril 2025.

[24] IBM. Qu'est-ce qu'un modèle IA ? https://www.ibm.com/fr-fr/topics/ai-model. Consulté le 13 avril 2025.

[25] DataScientest. Apprentissage Non Supervisé : principe et utilisation. https://datascientest.com/apprentissage-non-supervise. Consulté le 19 avril 2025.

[26] AWS. Qu'est-ce qu'un réseau neuronal ? https://aws.amazon.com/fr/what-is/neural-network/. Consulté le 5 avril 2025.

[27] Cloudflare. Qu'est-ce qu'un réseau neuronal ? https://www.cloudflare.com/fr-fr/learning/ai/what-is-neural-network/. Consulté le 14 avril 2025.

[28] DataCamp. Rectified Linear Unit (ReLU). https://www.datacamp.com/fr/blog/rectified-linear-unit-relu, 2022.

[29] JMEGNIDRO. La fonction sigmoïde : un outil puissant pour la prédiction et l'analyse de tendances. https://dev.to/jmegnidro/la-fonction-sigmoide-un-outil-puissant-pour-la-prediction-et-lanalyse-de-tendance~:text=Elle%20peut%20s%27%C3%A9crire%20sous%20la%20forme%20%3A%20%CF%83,pour%20les%20x%20positifs%20qui%20tend%20vers%201, April 2022.

[30] Encyclopédie FR. Fonction Softmax. https://ency.fr/wiki/Fonction_softmax, 2022.

[31] Mathority. Fonction Tangente Hyperbolique. https://mathority.org/fonction-tangente-hyperbolique/#:~:text=La%20fonction%20tangente%20hyperbolique%20est%20l%E2%80%99une%20des%20principales,au%20sinus%20hyperbolique%20divis%C3%A9%20par%20le%20cosinus%20hyperbolique, 2022.

[32] Canadian Institute for Cybersecurity. CIC-Darknet2020 Dataset. https://www.unb.ca/cic/datasets/darknet2020.html, 2020.

[33] Wireshark. A Network Protocol Analyzer. https://www.wireshark.org/, 2025.

[34] Canadian Institute for Cybersecurity. CICFlowMeter. https://www.unb.ca/cic/research/applications.html, 2025.

[35] Anaconda. Anaconda Training : A Learning Path for Data Scientists. https://www.anaconda.com/blog/anaconda-training-a-learning-path-for-data-scientists, 2024.

[36] NumPy Developers. NumPy Documentation. https://numpy.org/doc/. Consulté le 22 mars 2025, 2025.

[37] Pandas Development Team. Documentation officielle de Pandas. https://pandas.pydata.org/docs/. Consulté le 22 mars 2025, 2025.

[38] Matplotlib Development Team. Matplotlib : Visualization with Python. https://matplotlib.org/stable/contents.html. Consulté le 22 mars 2025, 2025.

[39] François Chollet and Keras Team. Keras : Deep Learning for Humans. https://keras.io/, 2025.

[40] Google Brain Team. TensorFlow : End-to-End Machine Learning Platform. https://www.tensorflow.org/. Consulté le 22 mars 2025, 2025.

[41] Scikit learn Developers. Scikit-learn : Machine Learning in Python. https://scikit-learn.org/stable/documentation.html. Consulté le 22 mars 2025, 2025.

[42] Seaborn Developers. Seaborn : Statistical Data Visualization. https://seaborn.pydata.org/. Consulté le 22 mars 2025, 2025.

[43] Vectra AI. Qu'est-ce que l'analyse du trafic réseau ? https://fr.vectra.ai/topics/network-traffic-analysis. Consulté le 16 avril 2025.

[44] Feedforward neural network structure diagram. https://www.researchgate.net/publication/349814755/figure/fig1/AS:1020033140539399@1620206260786/Feedforward-neural-network-structure-diagram.png, 2021. Accessed : [22nd March 2025].

[45] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. Communications of the ACM, vol. 42, no. 2, p. 5., 1999.

[46] Y. Hu, F. Zou, L. Li, and P. Yi. Traffic classification of user behaviors in tor, i2p, zeronet, freenet. 19th International Conference on Trust, Security and Privacy in Computing and Communications., 2020.

[47] M. Wang, X. Wang, J. Shi, Q. Tan, Y. Gao, M. Chen, and X. Jiang. Who are in the darknet ? measurement and analysis of darknet person attributes. Conference on Data Science in Cyberspace (DSC)., 2018.

[48] E. F. Fernandez, R. A. V. Carofilis, F. J. Martino, and P. B. Medina. Classifying suspicious content in tor darknet. arXiv preprint arXiv :2005.10086., 2020.