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 du trafic Tor avec l'apprentissage profond

Réalisé par
Mr Caid Mohamed Racim

Encadré par
Mr MEHDI Merouane

Juin 2024

Remerciement

Je tiens à exprimer ma profonde gratitude à Allah, le Tout-Puissant, pour m'avoir accordé la santé, la force et la détermination nécessaires à la réalisation de ce projet.

Je suis également profondément reconnaissant envers mes parents pour leur amour, leur soutien et leurs encouragements constants. Leur confiance en moi et leurs sacrifices ont été des sources inestimables de motivation tout au long de mon parcours académique.

J'aimerais exprimer ma profonde gratitude envers mon promoteur, Mehdi Marouane, pour son accompagnement attentif, ses judicieux conseils et son soutien indéfectible durant la réalisation de ce projet. Ses compétences pointues et sa grande patience ont été des piliers majeurs qui ont grandement contribué à la concrétisation de ce travail.

Je tiens à adresser mes sincères remerciements aux membres du jury pour avoir bien voulu accorder de leur temps précieux à la lecture, l'évaluation et la critique de ce mémoire.

Enfin, je suis reconnaissant envers toute l'équipe du département d'électronique pour leur soutien précieux. Leur dévouement et leur expertise ont été d'une aide considérable dans mon parcours universitaire.

Je souhaite exprimer ma sincère gratitude à chacun de vous pour votre contribution précieuse à ce projet. Vos efforts et votre soutien ont été inestimables, et je vous en suis profondément reconnaissant. Merci du fond du cœur pour tout ce que vous avez fait.

ملخص

الهدف من هذا المشروع هو كشف استخدام شبكة تور، التي تستخدم في الأنشطة غير القانونية والهجمات الإلكترونية، والتي تشكل مخاطر كبيرة من حيث الأمن والخصوصية. قمنا بتدريب وتقييم نماذج التعلم العميق، بما في ذلك لـ CNN و لـ DNN، مجموعة بيانات ISCXTor2016. تشير النتائج إلى أن النموذج المحسّن بواسطة التحقق المتقاطع يظهر كفاءة عالية في اكتشاف شبكة تور، مع درجة f1 بنسبة ٩٨٪.

الكلمات المفتاحية: الإنترنت، الويب المظلم، تور، الأمان، الخصوصية، نموذج التعلم العميق

Abstract

The goal of this project is to detect the use of the TOR network, which is used for illegal activities and cyberattacks, presenting significant risks in terms of security and privacy. We trained and evaluated deep learning models, including CNN and DNN, using the ISCXTor2016 dataset. The results indicate that the CNN model, optimized by cross-validation, demonstrates high efficiency in detecting the TOR network, with an F1-score of 98%.

Keywords: Internet, Dark web, TOR, Security, Privacy, Deep learning model

Résumé

Le but de ce projet est de détecter l'utilisation du réseau TOR, utilisé pour des activités illégales et des cyberattaques, présente des risques importants en termes de sécurité et de confidentialité. Nous avons entraîné et évalué des modèles de deep learning, notamment CNN et DNN, en utilisant l'ensemble de données ISCXTor2016. Les résultats indiquent que le modèle CNN optimisé par validation croisée démontre une grande efficacité dans la détection du réseau TOR, avec un F1-score de 98%.

Mots Clée : Internet, Dark web, TOR, Sécurité, Confidentialité, Modèle de deep learning

Liste des Acronymes et Abréviations

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

ADSL Asymmetric Digital Subscriber Line

API Application Programming Interface

BSD Berkeley Software Distribution

CNN Convolutional Neural Network

DNN Deep Neural Network

FAI Fournisseur d'Accès à Internet

FN Faux Négatif

FP Faux Positif

HTTP Hypertext Transfer Protocol

IA Intelligence Artificielle

I2P Invisible Internet Project

IP Internet Protocol

ISCXTor2016 Canadian Institute for Cybersecurity's Tor Network Traffic 2016 Dataset

TCP Transmission Control Protocol

TOR The Onion Router

VP Vrai Positif

VN Vrai Négatif

Web World Wide Web

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

L'internet a révolutionné la communication et le partage d'informations à l'échelle mondiale. Cependant, cette connectivité a également introduit des dangers en termes de sécurité et de confidentialité. Sur Internet, on retrouve d'un côté le web de surface classique, et de l'autre le dark web qui est accessible via des réseaux anonymes tels que I2P et TOR. Ces réseaux permettent aux utilisateurs de naviguer et de communiquer de manière anonyme.

Le dark web pose des risques significatifs. L'anonymat qu'il offre peut être utilisé à des fins malveillantes, telles que la diffusion de contenus illégaux et la réalisation d'activités criminelles en ligne. Un exemple notable est Silk Road, un marché noir en ligne où des transactions illégales étaient effectuées anonymement via TOR. De plus, certains gouvernements utilisent également ces réseaux pour surveiller les communications, soulignant ainsi les failles de sécurité de ces systèmes.

Pour contrer ces risques, nous proposons un modèle de deep learning capable de détecter l'utilisation de réseaux TOR en analysant les caractéristiques du trafic sur ce réseau.

Notre méthodologie implique plusieurs étapes. Tout d'abord, nous avons récupéré un ensemble de données, puis nous avons prétraité ces données pour les rendre compatibles avec les modèles de deep learning. Ensuite, nous avons conçu et entraîné différents modèles, notamment des réseaux de neurones profonds (DNN) et des réseaux de neurones convolutifs (CNN). Nous avons également utilisé la validation croisée pour optimiser les performances de nos modèles. Enfin, nous avons évalué ces modèles à l'aide de la matrice de confusion et le F1-score.

Notre mémoire est structuré en trois chapitres principaux. Dans le chapitre 1, nous fournissons une introduction détaillée sur le réseau TOR, ses avantages et ses risques, ainsi que les concepts fondamentaux du deep learning et leur application potentielle dans la détection du trafic TOR. Le chapitre 2 décrit la méthodologie adoptée pour notre étude, y compris la présentation de l'ensemble de données, les tâches de prétraitement, les architectures des modèles (DNN et CNN), les techniques d'entraînement, et les métriques d'évaluation utilisées. Enfin, le chapitre 3 se concentre sur les expérimentations et les

résultats obtenus, comparant les performances de différents modèles à l'aide de la matrice de confusion et du F1-score, avec une évaluation incluant un DNN, un CNN et un CNN optimisé par validation croisée. Nous concluons avec une discussion sur les limitations de notre étude et les perspectives d'amélioration.

Chapitre 1

Contexte et état de l'art

1.1 Introduction

Avec l'avènement d'Internet, la communication et le partage d'informations à l'échelle mondiale ont connu une révolution, mais cette connectivité pose aussi des problèmes pour la sécurité et la confidentialité. L'un des plus grands enjeux est de protéger l'anonymat et la vie privée des gens, surtout avec la surveillance et le contrôle de l'information qui deviennent de plus en plus courants.

Le réseau TOR (The Onion Router) s'est imposé comme une solution de premier plan pour ceux qui souhaitent naviguer sur Internet de manière anonyme et sécurisée. Grâce à un mécanisme de routage en oignon, TOR crypte et achemine les données via plusieurs nœuds intermédiaires, rendant ainsi extrêmement difficile de trouver qui a envoyé ou reçu les communications.

Toutefois, l'anonymat offert par TOR peut être utilisé à mauvais usage, par exemple pour accéder à des contenus illégaux ou mener des activités criminelles en ligne. De plus, certains gouvernements exploitent également TOR pour surveiller les communications, mettant ainsi en lumière ses failles de sécurité.

Pour relever ces défis, le Deep Learning offre une solution en permettant de détecter l'utilisation du réseau TOR. En utilisant des techniques avancées d'apprentissage automatique, il est possible d'analyser les modèles de trafic sur le réseau et d'identifier les comportements suspects.

Dans ce chapitre, nous allons approfondir le fonctionnement du réseau TOR, les risques associés à son utilisation, ainsi que les principes fondamentaux du Deep Learning et son potentiel application dans la détection de ce réseau.

1.2 Internet et les réseaux anonymes

Pour établir une communication entre deux ordinateurs, il est nécessaire de disposer d'un lien physique entre eux, que ce soit par câble ou sans fil. Lorsque plusieurs ordinateurs sont connectés pour échanger des informations, nous obtenons un réseau informatique.

Internet est un réseau de réseaux, c'est-à-dire une infrastructure qui permet à plusieurs réseaux de se connecter entre eux, autorisant ainsi l'échange d'informations entre ordinateurs connectés à des réseaux différents. Par exemple, les ordinateurs d'une bibliothèque universitaire sont connectés au réseau interne de l'université via un câble Ethernet. Chez un particulier, l'ordinateur familial est connecté au réseau de son fournisseur d'accès à Internet, généralement via ADSL. Les réseaux de l'université et du fournisseur d'accès à Internet sont à leur tour connectés à d'autres réseaux, qui sont eux-mêmes connectés à d'autres réseaux. Ensemble, tous ces réseaux forment Internet.

Pour gérer la transmission de données sur ce réseau de réseaux, deux protocoles clés sont utilisés et constituent les fondements d'Internet : le protocole IP (Internet Protocol) et le protocole TCP (Transmission Control Protocol). [20]

L'Internet est fréquemment comparé à un iceberg, avec ses nombreuses couches dissimulées sous la surface.

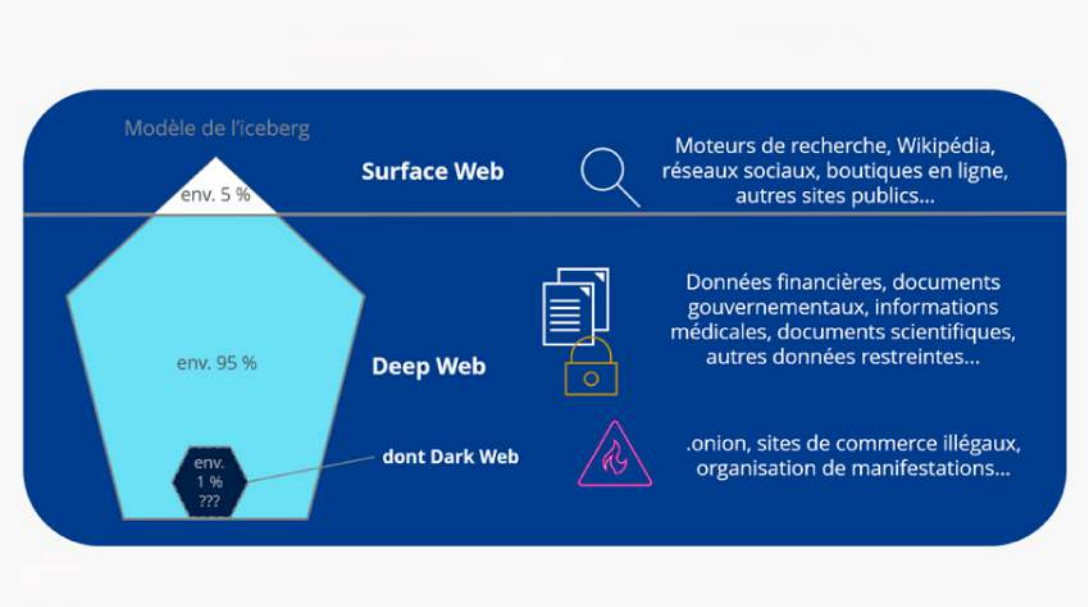

FIGURE 1.1 – Les différentes couches du web [21]

Cette structure à trois niveaux comprend le web visible, où la majorité des utilisateurs naviguent quotidiennement via des moteurs de recherche traditionnels pour accéder à des sites web publics indexés, le deep web, où se trouvent des contenus non indexés et protégés par des mots de passe et enfin, le dark web, une partie sombre souvent associée à des

activités illicites, accessible uniquement via des logiciels spécifiques. Cette segmentation met en évidence la complexité et la diversité d'Internet, offrant des niveaux d'accessibilité et de confidentialité variés pour ses utilisateurs. [19]

1.2.1 Web

Le World Wide Web, également connu sous le nom de Web, est un système d'organisation de documents sur Internet. Ces documents, appelés hypertextes ou pages web.

Les navigateurs web affichent ces pages et interprètent les hyperliens pour faciliter la navigation. Les données échangées sur le Web sont transmises grâce au protocole HTTP (HyperText Transfert Protocol).

Lors de l'échange de données sur le Web via le protocole HTTP, les rôles des machines sont différenciés. L'une d'entre elles, appelée "serveur", a pour mission de fournir des ressources ou des services, tandis que l'autre, le "client", utilise ces ressources et services. [20]

1.2.2 Web visible

Le Web visible, également appelé Web de surface, correspond à la partie superficielle « visible » de l'iceberg qu'est le Web dans son ensemble. Si l'on continue à représenter le Web comme un iceberg, le Web visible serait la partie supérieure qui dépasse de l'eau. Il s'agit de l'ensemble des sites Web accessibles au public via des navigateurs traditionnels tels que Google Chrome, Internet Explorer et Firefox. Ces sites Web sont généralement identifiés par des opérateurs de registre tels que « .com » et « .org », et peuvent être localisés à l'aide des moteurs de recherche couramment utilisés.

1.2.3 Web profond

Le Web profond représente la partie immergée d'un iceberg, qui est beaucoup plus vaste que le Web de surface. En réalité, cette partie invisible du Web est si étendue qu'il est impossible de déterminer avec précision le nombre de pages ou de sites Web qui y sont actifs simultanément.

Ce Web profond inclut également la partie que nous connaissons sous le nom de Dark Web. Bien que de nombreux médias utilisent indifféremment les termes « Deep Web » et « Dark Web », une grande partie du Web profond dans son ensemble est parfaitement légale et sûre.

Parmi les parties les plus importantes du Web profond, on trouve les Bases de données qui sont des collections de fichiers publics et privés protégés qui ne sont pas reliés à d'autres

zones du Web, ainsi que les Intranets qui sont des réseaux internes d'entreprises, de gouvernements et d'établissements d'enseignement utilisés pour communiquer et contrôler certains éléments en privé au sein de leurs organisations.

1.2.4 Dark Web

Le Dark Web désigne des sites qui ne sont pas indexés et qui ne sont accessibles que via des navigateurs Web spécifiquement conçus à cet effet.

Il est considéré comme faisant partie du Web profond. Si l'on reprend notre métaphore de l'océan et des icebergs, le Dark Web correspondrait à la partie inférieure de l'iceberg.

La réputation du Dark Web est souvent associée à des activités criminelles ou à des contenus illégaux, ainsi qu'à des sites de « vente » où les utilisateurs peuvent se procurer des biens ou des services illicites. Toutefois, des parties parfaitement légales ont également recours à cette infrastructure.

1.2.5 Accéder au Dark Web

Accéder au Dark Web nécessite un navigateur anonyme comme Tor , The Onion Routing , permet d'accéder à des sites avec l'extension ".onion". Aujourd'hui, tout le monde peut télécharger gratuitement Tor pour explorer le Dark Web. Ce navigateur fonctionne en utilisant un réseau de serveurs chiffrés appelés "nœuds" pour assurer l'anonymat des utilisateurs.

D'autres logiciels similaires, tels que I2P (Invisible Internet Project), offrent également des fonctionnalités similaires pour préserver l'anonymat en ligne. Ces outils permettent aux utilisateurs de naviguer sur le Dark web sans craindre d'être suivis ou surveillés . [19]

1.3 Présentation du réseau TOR et de son fonctionnement

1.3.1 Introduction

Les messages envoyé sur Internet connaît une trajectoire d'envoi et de réception passant par plusieurs serveurs qui peuvent savoir d'où ils viennent ou où ils vont.

les serveurs sauvegarde les adresses de Provence et de destination, et peuvent être ensuite récupéré pour identifier l'identité des personnes ayant envoyé ou reçu ces paquets.

Puisque les fournisseurs d'accès a l'internet ont la main pour avoir transférer les paquets, ces derniers ont la possibilité de contrôler, et de filtrer ce qui transite sur les réseaux

via l'accès qu'ils offrent. Il n'y a donc aucune garantie qu'un message arrive effectivement à destination sans avoir été préalablement censuré ou du moins lu, intercepté ou conservé par un tiers. [11]

Ces risques en matière de sécurité et de confidentialité des communications sur Internet ont amené plusieurs organisations, à développer un logiciel dont l'objectif était d'assurer la sécurité des communications et l'anonymat des participants. L'une des méthodes a été le routage en oignons, TOR. Les premiers éléments techniques sur le routage en oignon viennent de Paul Syverson , qui dans les années 1995 travaillait pour le labo de recherche de la Marine aux États-Unis. [3]

1.3.2 Définition

Tor est un réseau qui permet aux utilisateurs d'Internet de naviguer sur le Web, de publier des sites et de communiquer avec d'autres en préservant leur anonymat.

FIGURE 1.2 – Illustration du Navigateur Tor [16]

Tor est aussi connu sous le nom d'« onion router », et a été développé à l'origine par un laboratoire de recherche associé à la marine américaine pour sécuriser les communications de l'armée.

Le service est accessible depuis le site Internet de Tor, permet de brouiller les pistes de nos données. Ainsi, il devient plus difficile de savoir aussi rapidement d'où viennent les

données ou où elles se dirigent.

TOR est aujourd'hui un logiciel sous licence libre BSD (Berkeley Software Distribution license) qui a reçu en 2010 le prix du logiciel libre remis par la Free Software Foundation, basé sur un réseau de volontaires du monde entier.

En ce qui concerne la vie privée, cela permet d'empêcher la surveillance des réseaux et l'analyse du trafic Internet par les fournisseurs d'accès, tout en permettant aux utilisateurs de communiquer des informations de manière anonyme. En matière de liberté d'expression, TOR est un outil qui peut être utilisé pour contourner la censure, étant donné qu'il permet aux utilisateurs d'accéder à des serveurs bloqués par des firewalls nationaux ou institutionnels, et de communiquer des informations au public de manière anonyme, sans risque de se faire appréhender. [11]

Le réseau TOR à travers le monde compte 6 000 relais qui sont maintenus par environ 3 000 bénévoles. [3]

1.3.3 Fonctionnement du réseau TOR

Tor est à la fois un réseau de routeurs et un protocole de chiffrement, Basé sur le mécanisme de routage en oignon (onion routing).

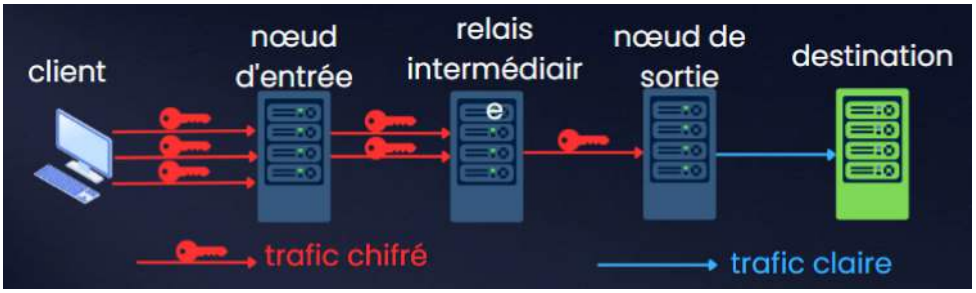

FIGURE 1.3 – Mécanisme de fonctionnement de Tor

Le routage en oignon est un mécanisme par lequel les données sont cryptées à plusieurs reprises avec différentes clés de chiffrement. C'est pourquoi on l'appelle routage en oignon, en référence aux différentes couches de cryptage qui entourent un paquet, similaire aux couches d'un oignon.

Les données sont ensuite envoyées à plusieurs nœuds dans le réseau. Les données sont ainsi protégées par plusieurs couches de cryptographie, qui sont progressivement retirées par chaque nœud jusqu'à ce qu'elles atteignent leur destination finale où elles sont à nouveau déchiffrées et deviennent lisibles.

Afin d'assurer une communication anonyme, les données transférées sur le réseau Tor doivent passer au minimum par trois routeurs différents. L'origine des communications

n'est connue que des routeurs d'entrée et la destination des communications n'est connue que des routeurs de sortie. Les autres nœuds du réseau sont des routeurs intermédiaires qui ne connaissent ni la provenance ni la destination des paquets qu'ils transfèrent.

Les routeurs d'entrée sont les nœuds par lesquels les utilisateurs se connectent au réseau Tor. Étant donné que l'adresse IP des utilisateurs est nécessairement visible à ce stade. Pour éviter les risques, les utilisateurs doivent s'assurer de ne pas se connecter à un nœud malveillant. Pour cela, Tor a mis en place une série d'annuaires officiels qui fournissent aux utilisateurs une liste de nœuds Tor certifiés auxquels ils peuvent faire confiance.

Lorsqu'un utilisateur souhaite communiquer de manière anonyme, le logiciel Tor crée un chemin aléatoire composé de trois nœuds différents chargés de transférer les données de l'adresse d'origine à la destination finale. [11]

1.3.4 Les routeurs ognion

Le réseau TOR est composer de plusieurs relais placé par nombreux bénévoles dans le monde, n'import quel personne peut participer au projet tor en installant un proxy tor sur leur machines. [2]

1.3.4.1 Relais de garde et intermédiaire

Le relais de garde et le relais intermédiaire sont deux types de nœuds dans le réseau Tor, qui constituent les premiers et deuxièmes sauts dans le circuit Tor, le relais de garde doit être stable et rapide (avec une bande passante d'au moins 2 Mo/s dans les deux sens), sinon il restera un relais intermédiaire.

Tous les relais sont répertoriés publiquement dans une liste, ce qui signifie qu'ils pourraient être bloqués par certains services.

1.3.4.2 Relais de sortie

Un relais de sortie est le dernier nœud d'un circuit Tor, il envoie le trafic vers sa destination. Les services que les utilisateurs Tor utilisent (sites Web, services de chat, etc.) verront l'adresse IP du relais de sortie au lieu de la véritable adresse IP de l'utilisateur Tor.

Les relais de sortie sont les plus exposés juridiquement parmi tous les relais. Par exemple, si un utilisateur télécharge du contenu protégé par des droits d'auteur via votre relais de sortie, vous en tant qu'opérateur pourriez recevoir un avis de violation de droits d'auteur. Les plaintes d'abus concernant la sortie vous seront directement adressées.

En raison des risques juridiques, il est déconseillé d'exécuter un relais de sortie Tor depuis chez soi. Les opérateurs de relais de sortie idéaux sont liés à une institution telle qu'une université, une bibliothèque ou une organisation axée sur la confidentialité. Ces institutions peuvent fournir une bande passante plus importante et sont mieux équipées pour gérer les plaintes d'abus ou les enquêtes éventuelles.

1.3.4.3 Pont

Les ponts sont des relais d'entrée discrets utilisés lorsque l'accès habituel au réseau Tor est bloqué par le fournisseur d'accès Internet. Habituellement, les adresses IP des relais Tor sont publiques, ce qui les rend vulnérables au blocage par les gouvernements ou les FAI.

Cependant, les ponts ne sont pas répertoriés dans la liste publique des adresses de nœuds de réseau Tor, ce qui les rend plus difficiles à identifier et à bloquer pour les FAI et les gouvernements. Ainsi, les ponts sont une solution pour contourner la censure et accéder à Tor dans des régions où son utilisation est restreinte.

1.4 Risques et enjeux de sécurité de l'utilisation du réseau TOR

L'anonymat offert par le réseau TOR peut être utilisé de manière abusive. Puisque les utilisateurs ne peuvent être suivis que jusqu'aux routeurs de sortie, certains peuvent potentiellement utiliser le réseau pour accéder à des contenus illégaux, comme des œuvres protégées par le droit d'auteur sans autorisation, des contenus pornographiques ou pédophiles, ou même pour Acheter des choses illégales, comme de la drogue et des armes, sur des sites comme The Silk Road, qui utilisent le réseau TOR et la monnaie virtuelle Bitcoin pour garantir l'anonymat des transactions. aussi Tor est un outil qui peut également être utilisé pour accéder à des serveurs bloqués par des firewalls nationaux ou institutionnels, et de communiquer des informations au public de manière anonyme, sans risque de se faire appréhender. [11]

Certains gouvernements ont choisi d'utiliser Tor pour surveiller les communications au lieu de d'adopter des lois sur sa légalité. Tor, conçu pour garantir l'anonymat mais présente des failles de sécurité exploitables. Les routeurs de sortie peuvent analyser les données non cryptées, exposant les identifiants et mots de passe.

Certains gouvernements, comme les États-Unis, la Chine et la Russie, utilisent le réseau Tor pour surveiller ce que les utilisateurs font en ligne. Ils ont des projets, comme Vigilant en 2014, qui collectent des informations sur les utilisateurs de Tor. Malgré son

but initial de protéger la vie privée, Tor présente des faiblesses que les gouvernements exploitent pour surveiller les utilisateurs. [15]

1.5 Défis actuels de la détection du trafic TOR et l'utilisation du Deep Learning

Les approches actuelles de détection de l'utilisation du réseau Tor souffrent de lacunes importantes. ces méthodes se basent sur une analyse statique du trafic, visant à détecter des schémas ou des signatures spécifiques associés à Tor. Cependant, cette méthode est facilement contournable par les développeurs de Tor, qui peuvent modifier le protocole ou utiliser des techniques pour rendre le trafic moins détectable. En conséquence, les approches basées sur l'analyse statique basé sur les signatures sont souvent inefficaces pour détecter de manière fiable l'utilisation de Tor.

De plus, un autre défi majeur est l'incapacité de ces méthodes à s'adapter aux évolutions du réseau Tor et des logiciels associés. Étant donné que le réseau Tor et les logiciels associés évoluent rapidement, les approches statiques peuvent conduire à des faux positifs (détection erronée de l'utilisation de Tor) ou des faux négatifs (échec à détecter l'utilisation de Tor lorsque celle-ci est présente).

L'utilisation du deep learning offre une opportunité significative pour améliorer la détection de l'utilisation du réseau Tor. En premier lieu, son aptitude à l'apprentissage automatique permet aux modèles de déceler indépendamment des motifs et des caractéristiques complexes dans les données, sans exiger de programmation spécifique. Cette faculté d'adaptation leur permet de suivre efficacement les évolutions du réseau Tor et des logiciels associés, contrairement aux méthodes traditionnelles qui deviennent rapidement dépassées.

En outre, les modèles de deep learning, grâce à leur entraînement sur de grandes quantités de données, sont essentiels pour réduire les taux de faux positifs et de faux négatifs. En analysant en profondeur des données variées et en repérant des motifs subtils, ces modèles améliorent considérablement la fiabilité de la détection de l'utilisation du réseau Tor, offrant ainsi une meilleure sécurité et une meilleure gestion des risques. [12]

1.6 Concept de base de deep learning

1.6.1 L'apprentissage automatique

L'apprentissage automatique représente une sous-discipline de l'intelligence artificielle , ou les programmes peuvent s'améliorer tout seuls en analysant des données et en apprenant de leurs erreurs, sans qu'on doive les reprogrammer à chaque fois.

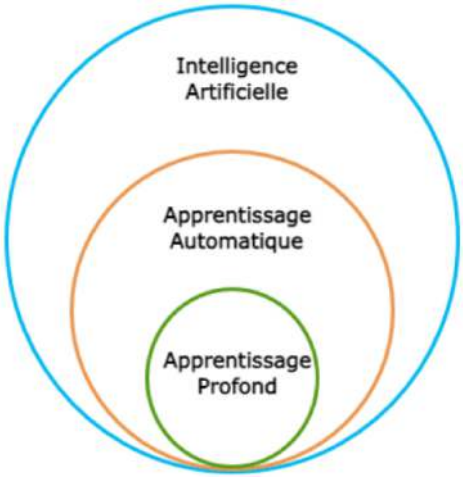

FIGURE 1.4 – Processus d'apprentissage automatique [5]

Dans un mode supervisé, ces programmes apprennent à prendre des décisions en se basant sur des exemples déjà étiquetés par des humains experts. Ils essaient de devenir de plus en plus précis dans leurs prédictions en comparant leurs résultats avec les étiquettes fournies par les experts. [6]

1.6.2 Les données d'apprentissage

Dans tout projet d'IA, travailler avec des ensembles de données est la partie la plus difficile, parfois, cela peut prendre jusqu'à 70 % du temps total .

De plus, constituer un ensemble de données pour L'apprentissage automatique nécessite des professionnels expérimentés et formés, capables d'exploiter efficacement les données collectées.

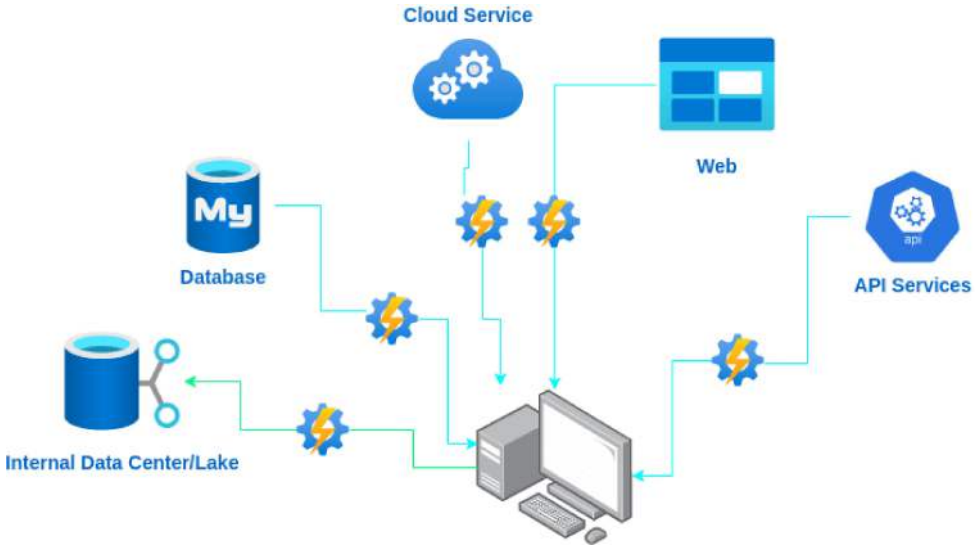

FIGURE 1.5 – Extraction des données [8]

Les données sont importantes pour les modèles d'IA et c'est grâce à elles que l'apprentissage automatique est devenu si populaire.

Des volumes de données suffisants permettent d'analyser et de prendre des décisions. Cependant, même si cela semble simple, travailler avec des données est plus compliqué. Cela nécessite un traitement approprié des données, depuis la préparation des données brutes jusqu'aux finalités d'utilisation d'un ensemble de données, pour qu'elles soient réellement utilisables.

La qualité des données est un élément essentiel lors de la collecte d'un jeu de données pour un projet de maching learning. Tout d'abord, les éléments de données doivent être pertinents par rapport à l'objectif du modèle. Un jeu de données déséquilibré présente le risque de fausser les résultats de prédiction du modèle de machine learning. Ce n'est pas seulement la qualité, mais aussi la quantité qui compte. Il est important de disposer suffisamment de données pour entraîner correctement le modèle. [17]

1.6.3 Apprentissage supervisé et non supervisé

L'apprentissage supervisé et l'apprentissage non supervisé sont deux stratégies distinctes en apprentissage automatique. L'apprentissage supervisé consiste à former un modèle en utilisant des données étiquetées, dans le but de réaliser des prédictions précises. D'autre part, l'apprentissage non supervisé vise à détecter des structures et des modèles cachés dans des données non étiquetées. Chaque méthode a ses points forts et ses limites, et le choix entre les deux dépend du type de problème et des données accessibles.

1.6.4 Surapprentissage

L'overfitting est un problème fréquent en apprentissage automatique supervisé, où un modèle entraîné sur un jeu de données d'entraînement ne fonctionne pas bien sur de nouvelles données, car il ne se généralise pas bien. Cela peut arriver pour plusieurs raisons, mais une cause courante est que le modèle est trop complexe pour le volume de données disponible, ce qui donne de bons résultats sur les données d'apprentissage, mais pas sur les nouvelles données. Pour éviter l'overfitting, on peut utiliser des modèles plus simples, ou disposer de plus de données pour l'apprentissage. De plus, il existe diverses méthodes de régularisation pour limiter le modèle et empêcher l'overfitting.

1.6.5 Sous-apprentissage

L'underfitting, ou sous-apprentissage, est un problème courant en apprentissage automatique. Il se produit lorsqu'un modèle ne parvient pas à capturer les motifs dans les données d'entraînement.

Lorsqu'un modèle est trop simple et ne dispose pas d'une capacité suffisante pour représenter les données de manière adéquate, il peut sous-apprendre les motifs présents dans les données. De même, si nous disposons d'un ensemble de données d'entraînement trop petit, le modèle peut avoir du mal à généraliser les motifs présents dans les données. [1]

1.6.6 Les réseaux de neurones

Les réseaux de neurones, également désignés sous les termes de réseaux de neurones artificiels.

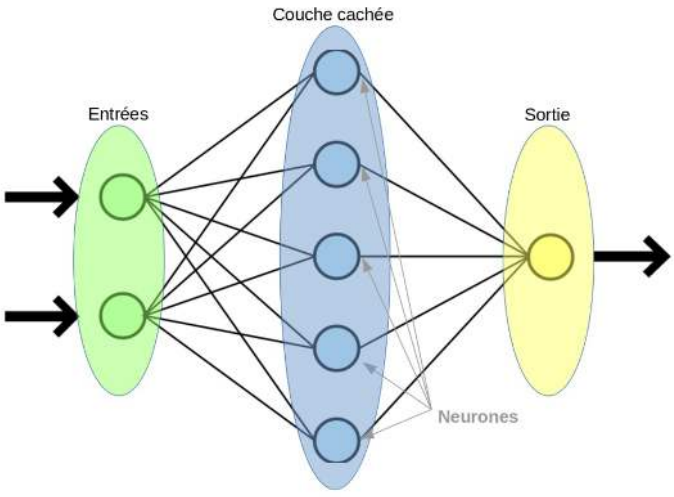

Figure 1.6 – Réseau neuronal artificiel simple [10]

Les réseaux de neurones forment une branche de l'apprentissage automatique et sont essentiels aux algorithmes d'apprentissage profond, leur dénomination et leur conception s'inspirent du cerveau humain, en reproduisant le fonctionnement des neurones biologiques qui s'échangent des signaux. [1]

1.6.7 Les éléments fondamentaux qui constituent un réseau neuronal

Le réseau de neurones est composé des composants principaux suivants :

1.6.7.1 Neurones

Les neurones prennent des données en entrée, les traitent, et donnent des nouvelles infos en sortie.Plusieurs neurones sont regroupés en couches (ou layers), où chaque neurone du même groupe effectue un type de fonctionnement similaire. Les neurones d'entrée reçoivent les données d'entrée, les traitent et les transmettent aux neurones de la couche suivante. Les neurones cachés, quant à eux, reçoivent les sorties des neurones précédents en entrée, effectuent des calculs pour produire de nouvelles sorties et les transmettent aux couches suivantes.

1.6.7.2 Les couches

Les couches contiennent des neurones, aident à faire circuler les données dans un réseau de neurones. Une telle architecture comporte au minimum deux couches : la couche d'entrée et la couche de sortie.

Il est possible d'avoir plusieurs couches dans un réseau de neurones complexe, ce qui le rend "profond" (deep learning), les couches autres que l'entrée et la sortie sont appelées couches cachées.

1.6.7.3 Poids

Les poids dans un réseau de neurones représentent l'intensité des connexions entre les neurones et déterminent l'importance relative des entrées provenant des neurones précédents.

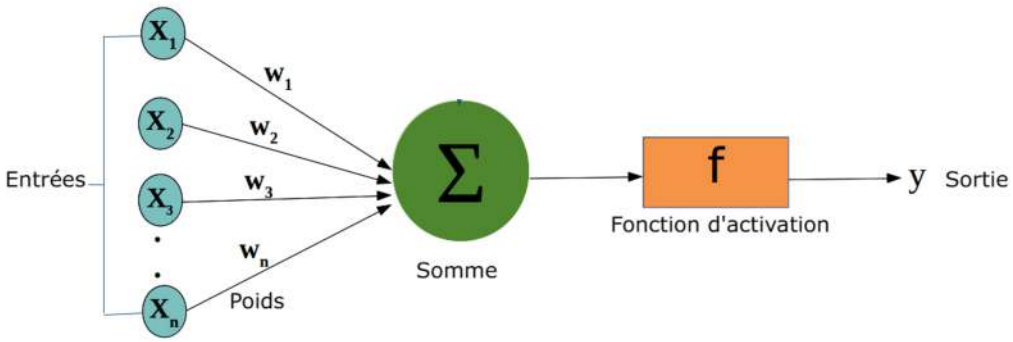

FIGURE 1.7 – Poids dans les réseaux neuronaux [18]

En ajustant ces poids, le réseau peut apprendre à détecter des motifs et des relations complexes dans les données.

1.6.7.4 Fonction d'activation

Les fonctions d'activation normalisent les sorties des neurones avant d'être transmises aux suivantes, contribuant ainsi à l'amélioration et à l'apprentissage des réseaux de neurones.

Ces fonctions sont des formules mathématiques qui s'activent dans certaines conditions. Lorsque les neurones calculent la somme pondérée des entrées et le poids, elles sont transmises à la fonction d'activation, qui vérifie si la valeur calculée dépasse le seuil spécifié.

Si la valeur calculée est supérieure au seuil, la fonction d'activation est activée et une valeur de sortie est calculée. Cette valeur est ensuite transmise aux couches suivantes ou précédentes, selon la complexité du réseau, ce qui peut aider les réseaux de neurones à ajuster les poids de leurs neurones.

1.6.8 Fonctionnement de réseaux de neurones

Un réseau de neurones fonctionne en trois étapes principales :

— **Entrée :** Chaque neurone reçoit des données d'entrée représentant les caractéristiques des données à traiter.

— **Calcul :** Les neurones combinent ces données d'entrée avec des poids pour produire une nouvelle valeur. Ce calcul capture les relations complexes entre les caractéristiques.
— **Activation :** Une fonction d'activation est appliquée sur le résultat du calcul. Cette fonction introduit de la non-linéarité dans le réseau .

Ce processus est répété à travers les différentes couches du réseau jusqu'à obtenir la sortie finale. [14]

1.6.9 L'apprentissage en profondeur

L'apprentissage en profondeur est une sous-catégorie de l'apprentissage automatique qui repose sur un réseau neuronal composé de trois couches ou plus.

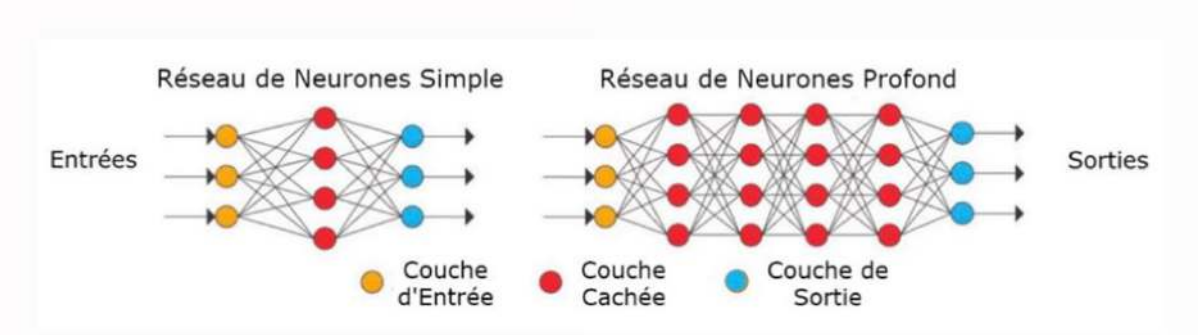

Figure 1.8 – Comparaison des réseaux neuronaux simples et des réseaux neuronaux profonds (Deep Learning) [7]

Bien qu'un réseau de neurones à une seule couche puisse effectuer des prédictions approximatives, l'ajout de couches cachées supplémentaires permet d'optimiser et d'améliorer la précision des résultats. L'apprentissage en profondeur est à la base de nombreuses applications et services d'intelligence artificielle (IA) qui améliorent l'automatisation en exécutant des tâches analytiques et physiques sans intervention humaine.

1.6.10 Architectures courantes

Les architectures fréquemment utilisées en apprentissage profond représentent un ensemble de structures et d'algorithmes servant à entraîner des réseaux de neurones profonds. Avec l'avènement de l'apprentissage profond, diverses architectures ont été mises au point pour résoudre des problèmes spécifiques. Ces architectures se distinguent par leur complexité, leur efficacité et leur aptitude à s'adapter à différents domaines d'application. Dans cette introduction, nous allons passer en revue quelques-unes des architectures

les plus influentes et les plus répandues dans le domaine de l'apprentissage profond, en soulignant leurs caractéristiques, leurs atouts et leurs applications.

1.6.10.1 CNN

Un réseau neuronal convolutionnel (CNN ou ConvNet) est une architecture de réseau dédiée à l'apprentissage en profondeur, qui apprend directement à partir des données.

Ces réseaux sont spécifiquement adaptés pour détecter des motifs dans les images, permettant ainsi la reconnaissance d'objets, de classes et de catégories. Ils se révèlent également très performants pour classer des données audio, des séries temporelles et des signaux.

Un réseau neuronal convolutionnel est généralement constitué de couches de convolution , de pooling, d'activation et entièrement connectées.

A- Couche de convolution Un CNN s'appuie sur des calculs mathématiques appelés convolutions ou produits de convolution, qui constituent la couche la plus importante . L'entrée d'une couche de convolution consiste en une image ou une carte de caractéristiques, et sa sortie est une carte de caractéristiques extraite grâce à des convolutions. À l'origine, la convolution était utilisée sur des données tabulaires, impliquant l'application d'un filtre sur les colonnes de données pour en extraire des motifs locaux ou réduire le bruit, en faisant glisser le filtre le long des vecteurs unidimensionnels représentant les caractéristiques du tableau.

Une couche de convolution correspond à une combinaison de plusieurs convolutions appliquées sur les cartes de caractéristiques. Cette approche permet aux couches d'extraire des caractéristiques de plus en plus complexes.

B- Couche de mise en commun (pooling) Les couches de convolution sont généralement utilisées en série dans un réseau de neurones profond, suivies de couches de pooling telles que Max-Pooling ou Avg-Pooling, qui réduisent la taille des caractéristiques extraites. Le Max-Pooling sélectionne la valeur maximale dans chaque fenêtre de la carte de caractéristiques, tandis que l'Avg-Pooling calcule la moyenne. Il est important de noter que ces opérations ne sont pas des neurones, mais elles contribuent à diminuer la charge de calcul en réduisant les dimensions des données et en régulant le sur-apprentissage.

C- Couche complètement connectée Dans une tâche de classification, un réseau de neurones profond se conclut généralement par une ou plusieurs couches entièrement connectées. Ces couches utilisent toutes les caractéristiques extraites par les couches précédentes pour générer la sortie finale, comme la classification d'une image en différentes

catégories. En outre, ces couches sont souvent suivies d'une couche Dropout, qui désactive certains neurones afin de prévenir le surapprentissage, une technique courante pour maîtriser la complexité du réseau neuronal.

D- Couche d'activation Après chaque couche de convolution ou de neurones entièrement connectés, on ajoute des couches d'activation. Bien que ces couches ne contiennent pas de neurones, elles emploient des fonctions spéciales pour déterminer si un neurone doit être activé ou non. Elles acceptent la sortie de la couche précédente, appliquent une transformation non linéaire, puis transmettent cette valeur modifiée à la couche suivante. En somme, elles permettent d'introduire de la non-linéarité dans le modèle, ce qui est essentiel pour l'apprentissage de modèles complexes. [9]

1.7 Conclusion

Le réseau TOR constitue une ressource précieuse pour préserver l'anonymat et la vie privée des utilisateurs sur Internet, bien qu'il soit également sujet à des risques potentiels, tels que l'utilisation abusive à des fins illégales ou la surveillance gouvernementale. Le Deep Learning représente une solution prometteuse pour détecter et contrer ces abus en examinant les modèles de trafic sur le réseau. Pour développer des outils efficaces permettant de protéger la sécurité et la confidentialité en ligne, il est essentiel de comprendre le fonctionnement du réseau TOR, les risques associés et les principes fondamentaux du Deep Learning. Cela nécessite une connaissance approfondie des concepts d'apprentissage automatique, de l'architecture des réseaux de neurones et des défis spécifiques liés à l'analyse du trafic TOR. En combinant ces connaissances avec des ensembles de données pertinents et de qualité, il est possible de créer des systèmes robustes capables de détecter et de prévenir l'utilisation du réseau TOR .

Chapitre 2

Méthodologie de travail

2.1 Introduction

Afin de résoudre le problème des risques liés à l'utilisation du réseau TOR, nous avons créé plusieurs modèles de deep learning pour comparer leurs performances dans la détection de l'utilisation de ce réseau. Dans ce chapitre, nous détaillons les aspects méthodologiques de notre travail. Nous commençons par présenter l'ensemble de données utilisé, suivi des tâches de prétraitement nécessaires pour préparer les données à l'entraînement des modèles de deep learning. Ensuite, nous présentons les architectures des modèles utilisés ainsi que les méthodes d'entraînement appliquées. Nous abordons également les métriques d'évaluation utilisées pour mesurer la performance des modèles, ainsi que les outils et l'environnement de développement employés. Cette méthodologie permet de garantir une approche rigoureuse et systématique pour l'analyse et la détection du trafic sur le réseau TOR.

2.2 Description des données utilisées "Dataset"

Le jeu de données ISCXTor2016 est un ensemble de données sur le trafic de réseau, collecté par le Centre d'excellence en sécurité de l'information (ISCX) de l'Université du Nouveau-Brunswick au Canada, durant l'année 2016. [13]

Le jeu de données contient à la fois le trafic normal et le trafic Tor et couvre une période de 2 semaines. Les flux de données ont été capturés à partir d'une variété d'interfaces et d'appareils réseau, tels que des ordinateurs portables, des téléphones intelligents, des interfaces câblées et sans fil, ainsi que des serveurs.

L'ensemble de données est étiqueté et comprend à la fois le trafic normal et le trafic Tor. Le trafic normal provient principalement de la navigation web, du transfert de fichiers

et des activités de courrier électronique. Le trafic Tor inclut des activités à la fois légitimes et malveillantes, telles que le partage de fichiers, la communication anonyme, ainsi que le contrôle et la commande de logiciels malveillants. Dans notre travail, nous avons utilisé le fichier 'Scenario-A-merged$_5$s.csv'.

Le scénario A est une tâche de classification binaire qui distingue le trafic entre Tor et non-Tor.

	nonTOR	**TOR**	**Total**
Nombre d'échantillons	69686	14508	84194

Tableau 2.1 – Répartition du trafic nonTOR et TOR

Le scénario A comprend un trafic TCP avec des flux bidirectionnels, à savoir des paquets aller (de la source vers la destination) et des paquets retour (de la destination vers la source) et se compose de 28 colonnes clés décrivant divers caractéristiques des flux de données réseau, et une colonne 'label' pour la classification.

Fonctionnalité	**Description**
Durée du flux	Temps total de la connexion (secondes)
Octets de flux	Total des octets envoyés pendant la connexion
Paquets de flux	Nombre de paquets envoyés
IAT de flux	Délai entre les arrivées du flux de paquets (Délai entre les arrivées successives de paquets [l](max, min, moyenne, écart-type).)
Transfert IAT	Temps entre l'arrivée des paquets envoyés (max, min, moyenne, écart-type).
IAT	Délai entre les arrivées de paquets en retour (max, min, moyenne, écart-type).
Actif	Durée pendant laquelle des données sont échangées (max, min, moyenne, écart-type).
Inactif	Temps pendant lequel la connexion est ouverte sans échange de données (max, min, moyenne, écart-type).

Tableau 2.2 – Description des fonctionnalités [4]

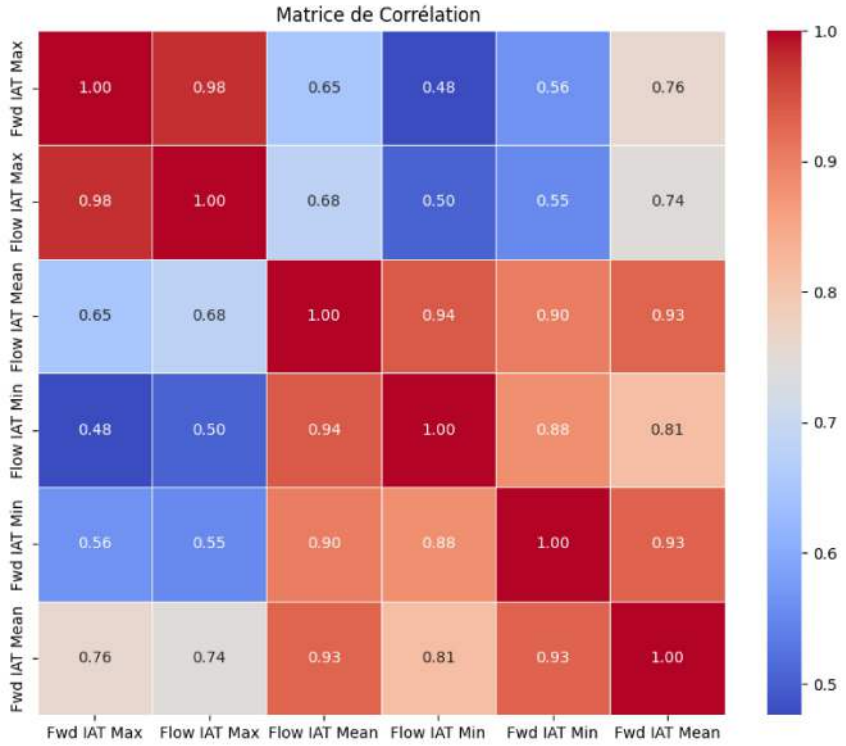

FIGURE 2.1 – Cartographie de la corrélation entre différentes caractéristiques

On observe de fortes corrélations supérieures à 0,9 entre différentes caractéristiques du dataset, ce qui suggère qu'elles sont fortement liées entre elles.

2.3 Prétraitement des données

La préparation des données est une étape obligatoire pour obtenir des résultats précis. Cette étape comprend plusieurs opérations de prétraitement qui peuvent varier en fonction des données brutes.

Dans le cas de l'ensemble de données ISCXTor2016, les étapes de prétraitement sont les suivantes :

2.3.1 Suppression des valeurs NAN et infinies

Dans un ensemble de données, les valeurs NAN (Not a Number) et infinies représentent des données manquantes ou anormales qui peuvent affecter la précision des résultats, Il est donc essentiel de les identifier et de les traiter correctement.

index	Source IP	Source Port	Destination IP	Destination Port	Protocol	Flow Duration	Flow Bytes/s	Flow Packets/s
84187	10.8.8.134	34251	134.170.18.137	443	6	0	Infinity	Infinity
84188	131.202.240.183	7116	239.255.255.250	1900	17	0	Infinity	Infinity
84189	131.202.240.183	7116	239.255.255.250	1900	17	0	Infinity	Infinity
84190	131.202.240.87	11365	31.13.73.1	443	6	0	Infinity	Infinity
84191	10.0.2.15	51024	37.97.149.8	443	6	0	Infinity	Infinity
84192	131.202.6.26	13000	131.202.240.87	64584	6	0	NaN	Infinity
84193	38.124.168.119	80	131.202.240.87	65089	6	0	NaN	Infinity

FIGURE 2.2 – Lignes contenant des valeurs NAN et infinity [13]

Dans le cas de l'ensemble de données ISCXTor2016, où il n'y a pas beaucoup de lignes contenant des valeurs manquantes ou infinies, nous les avons simplement supprimées.

2.3.2 Suppression de source et destination IP

Pendant la phase de prétraitement, nous avons décidé de supprimer les colonnes contenant les adresses IP source et de destination. En effet, ces informations ne sont pas pertinentes pour la classification du trafic Tor ou non-Tor. Leur suppression permet de simplifier le jeu de données en se concentrant sur les caractéristiques les plus pertinentes.

2.3.3 Codage de la colonne 'label'

Le processus de codage de la colonne de classification "label" consiste à assigner des valeurs numériques aux différentes classes pour faciliter l'analyse et la formation du modèle.

Après codage	Avant codage
0	nonTOR
0	nonTOR
1	TOR
1	TOR

FIGURE 2.3 – "Label" avant et aprés le codage

Dans notre cas, la colonne "label" est codée de manière binaire : 0 pour représenter le trafic non-Tor et 1 pour indiquer le trafic Tor. Cette méthode simplifie l'utilisation des algorithmes de classification qui nécessitent des entrées numériques.

2.3.4 Équilibrage des deux classes

Les données présentent un déséquilibre entre les exemples de trafic non-Tor et ceux de trafic Tor, avec une forte majorité de cas de trafic non-Tor. Ce déséquilibre entraîne des erreurs de classification significatives et une mauvaise interprétation des données.

Nous avons utilisé la fonction "RandomUnderSampler" de la bibliothèque "imblearn" pour résoudre ce problème de déséquilibre des classes. Cette méthode diminue le nombre d'exemples de la classe majoritaire (non-Tor) afin d'équilibrer le jeu de données.

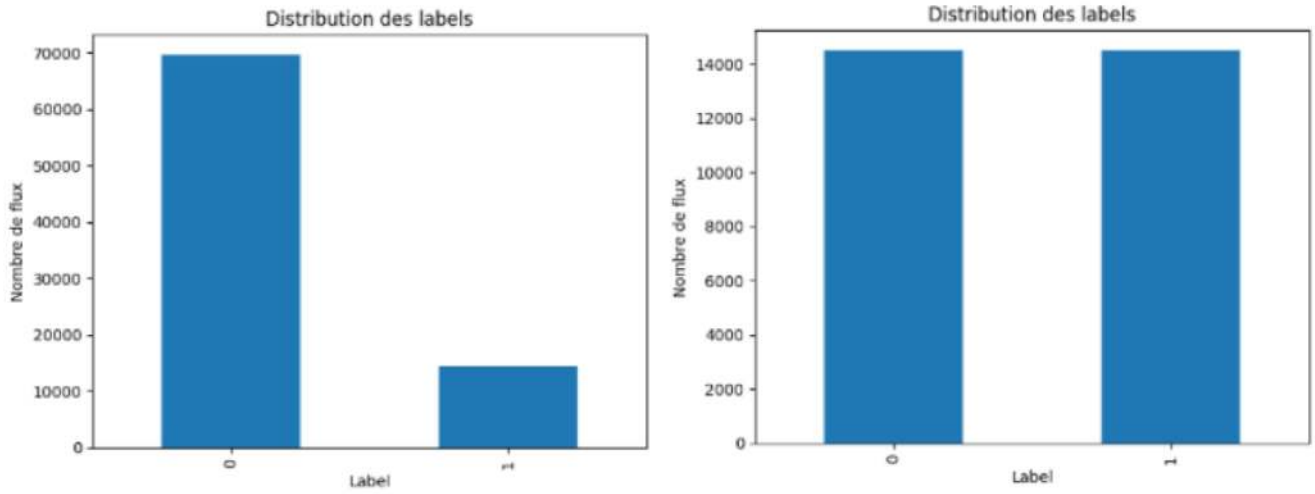

FIGURE 2.4 – Distribution des classes avant et après l'équilibrage

2.3.5 Éliminer certaines fonctionnalités

"Active Mean", "Active Std", "Active Max", "Active Min", "Idle Mean", "Idle Std", "Idle Max", "Idle Min" , ont été retirées de l'ensemble des caractéristiques, car la valeur la plus élevée de ces fonctionnalités est nulle, ce qui introduisait un bruit perturbateur dans la détection du trafic réseau. Leur suppression permet d'améliorer les performances du modèle de classification en éliminant ces sources de confusion.

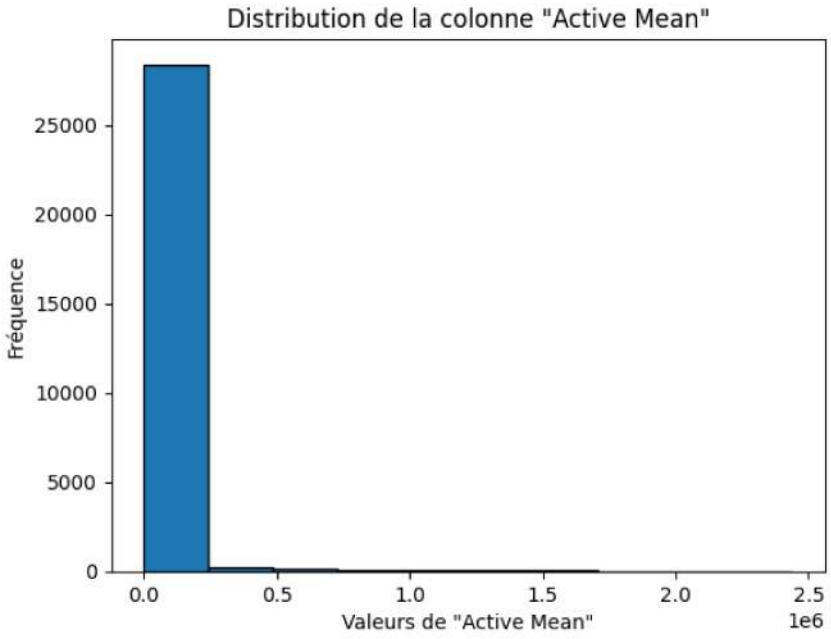

FIGURE 2.5 – Distribution de la colonne Active Mean

2.3.6 La Normalisation

Afin d'homogénéiser les différentes échelles des variables explicatives, une étape de normalisation a été effectuée sur les données. Cette transformation consiste à faire passer les valeurs de chaque variable dans un intervalle commun, typiquement entre 0 et 1. L'outil "MinMaxScaler" de la bibliothèque "scikit-learn" a été utilisé à cet effet. Cette mise à l'échelle uniforme des caractéristiques facilite l'entraînement des modèles de classification et tend à améliorer leurs performances prédictives. La formule mathématique implémentée par MinMaxScaler pour réaliser cette normalisation est la suivante :

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{2.1}$$

où :

— X_{scaled} : la valeur normalisée
— X : la valeur d'entrée
— X_{min} : la valeur minimale de l'ensemble des données
— X_{max} : la valeur maximale de l'ensemble des données

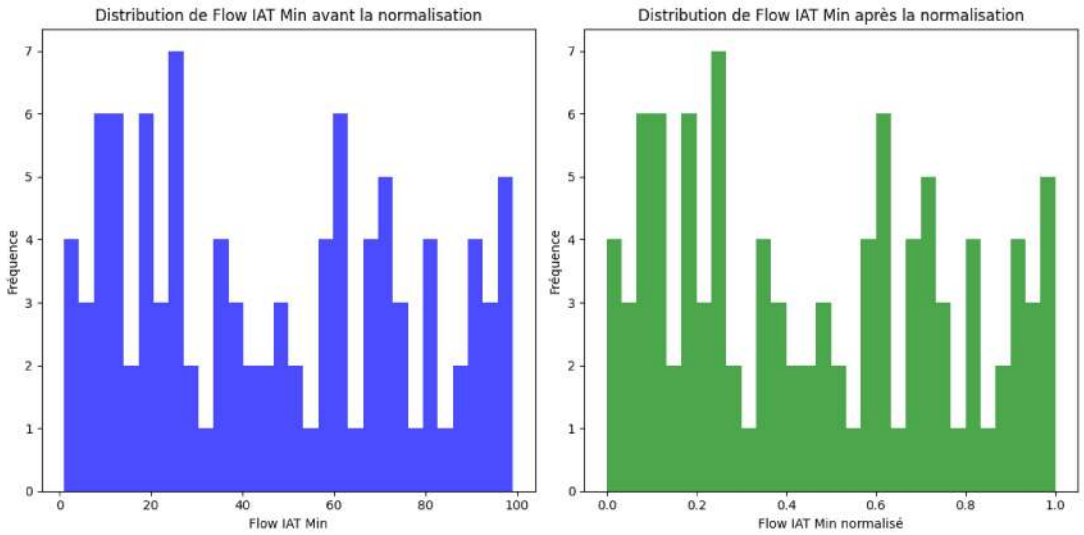

FIGURE 2.6 – Distribution de Flow IAT min avant et après la normalisation

2.3.7 La division des données en test et entraînement

Pour évaluer les performances du modèle et l'apprentissage du modèle, le jeu de données initial a été divisé en deux sous-ensembles : un ensemble d'entraînement et un ensemble de test. On a effectué cette partition de façon aléatoire en utilisant la fonction "train_test_split" fournie par la bibliothèque "scikit-learn". Le but est d'avoir un échantillon représentatif pour modifier les paramètres du modèle (ensemble d'entraînement) et un autre échantillon indépendant pour évaluer la prédictivité du modèle entraîné sur des données inédites (ensemble de test).

```
# Diviser la dataset en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

FIGURE 2.7 – Division du dataset en ensembles d'entraînement et de test

Dans le cadre de cette étude, un échantillonnage stratifié a été mis en œuvre, en allouant 10% des données à l'ensemble test grâce au paramétrage "test_size=0.1". Par ailleurs, afin d'assurer la reproductibilité des résultats, la graine du générateur de nombres aléatoires a été fixée à la valeur 42 via le réglage "random_state=42".

2.3.8 Présentation de l'architecture des modèles de deep learning utilisés

Dans le cadre de cette étude, deux architectures de réseaux de neurones profonds ont été mises en oeuvre et comparées : les réseaux de neurones denses (DNN) et les

réseaux de neurones convolutifs (CNN). Le DNN est constitué de couches entièrement connectées, où chaque neurone est relié à l'ensemble des neurones de la couche précédente. Cette disposition permet d'extraire les caractéristiques discriminantes des données et de réaliser les prédictions. Le CNN adopte une approche différente en intégrant des couches de convolution. Celles-ci permettent de capturer et d'extraire des motifs et structures spécifiques au sein des données, en effectuant des opérations de filtrage local.

2.3.8.1 CNN

Ce modèle a été conçu pour effectuer une classification binaire, c'est-à-dire prédire si une donnée appartient à l'une ou l'autre de deux classes différentes. La structure du modèle est la suivante :

1. La couche d'entrée récupère les données sous forme de vecteurs 1D.
2. Trois couches de convolution 1D extraient successivement des caractéristiques des données à l'aide de filtres. Un mécanisme de *dropout* régularise le modèle après chaque couche.
3. Des couches de *pooling* réduisent la dimension temporelle des caractéristiques extraites.
4. Une couche d'aplatissement transforme les sorties en un vecteur 1D.
5. Deux grandes couches denses entièrement connectées, avec *dropout*, effectuent des traitements supplémentaires.
6. La couche de sortie finale, avec une fonction d'activation sigmoïde, fournit la probabilité que la donnée appartienne à la classe positive.

L'architecture alterne donc des extractions de caractéristiques par convolutions, des réductions de dimension et des couches denses pour obtenir une prédiction binaire en sortie.

2.3.8.2 DNN

Le modèle DNN est composé de plusieurs couches denses entièrement connectées dans le but de réaliser une tâche de classification binaire.

1. La couche d'entrée reçoit les données sous forme de vecteurs de 18 caractéristiques.
2. Quatre couches denses successives transforment ces données d'entrée en une représentation compressée :
   — Une première couche de 23 neurones avec une activation linéaire.
   — Trois autres couches de 256 neurones chacune avec une activation sigmoïde.

   - Un *dropout* de 10% est appliqué après les deux dernières couches pour la régularisation.
3. La couche de sortie finale est une couche dense à un seul neurone avec une activation sigmoïde. Elle fournit la prédiction binaire compressée dans un espace à une dimension, à partir des 18 dimensions initiales.

En résumé, le DNN compresse les données d'entrée en passant par plusieurs couches denses réductrices, avant de réaliser la prédiction binaire finale dans un espace de sortie à une dimension.

2.4 Entraînement des modèles

L'apprentissage des modèles de deep learning se fait par ajustement itératif des poids (paramètres) des neurones du réseau. À chaque itération :

1. Le modèle fait des prédictions avec les poids actuels.
2. Il compare ses prédictions aux véritables valeurs cibles.
3. Il ajuste les poids pour réduire l'erreur de prédiction.

Ce processus vise à minimiser une fonction de perte qui mesure l'écart entre les prédictions et la réalité. L'apprentissage se déroule sur plusieurs époques. Une époque correspond à un passage complet sur l'ensemble des données d'entraînement pour mettre à jour les poids. En parallèle, un jeu de données de validation externe permet de suivre les performances du modèle au fur et à mesure. Cela permet de détecter le sur-apprentissage.

2.4.1 Entraînement du DNN

Pour entraîner le modèle DNN, les étapes suivantes ont été réalisées :

1. Le jeu de données d'entraînement a été divisé en deux sous-ensembles :
   - 70% pour l'apprentissage (entraînement) du modèle
   - 30% pour la validation pendant l'entraînement
2. L'entraînement a été lancé avec la méthode `fit()` de "Keras".
3. Une technique d'arrêt précoce ("Early Stopping") a été mise en place pour éviter le sur-apprentissage :
   - La perte sur les données de validation était surveillée.
   - Si cette perte ne s'améliorait pas pendant 4 époques d'affilée, l'entraînement s'arrêtait.

4. Un nombre maximal de 500 époques a été fixé pour limiter la durée d'entraînement.

```
# Entraîner le modèle
history_DNN = DNN_model.fit(X_train, y_train, validation_split=0.3, callbacks=[early_stopping] , epochs=500)
```

FIGURE 2.8 – La méthode fit() pour l'entraînement de DNN

En résumé, après avoir séparé les données, le modèle a été entraîné de façon contrôlée, avec un mécanisme d'arrêt basé sur les performances de validation, afin d'éviter les problèmes de sur-apprentissage.

2.4.2 Entraînement du CNN

Pour entraîner le modèle CNN (réseau de neurones convolutif), la même approche que pour le modèle DNN a été suivie :

1. Division du jeu de données d'entraînement en deux sous-ensembles :
   — 70% pour l'apprentissage du modèle
   — 30% pour la validation pendant l'entraînement
2. Entraînement lancé avec la méthode `fit()` de "Keras".
3. Mise en place d'un mécanisme d'arrêt précoce ("early stopping") pour éviter le sur-apprentissage :
   — Surveillance de la perte sur les données de validation.
   — Arrêt de l'entraînement si cette perte ne s'améliore pas pendant 4 époques consécutives.
4. Nombre maximal d'époques fixé à 500.

```
# Entraîner le modèle
history = model.fit(X_train, y_train, validation_split=0.3, callbacks=early_stopping , epochs=500)
```

FIGURE 2.9 – La méthode fit() pour l'entraînement de CNN

2.4.3 Entraînement du CNN avec validation croisée

Pour améliorer les performances du modèle CNN, la technique de validation croisée (cross-validation) a été appliquée.

la validation croisée consiste à entraîner et évaluer le modèle sur différents sous-ensembles des données afin d'obtenir une estimation plus fiable de ses performances générales, pour cela :

1. Les données d'entraînement ont été divisées en 5 sous-ensembles ou "plis" ($n\_splits=5$) à l'aide de "StratifiedKFold" de "Scikit-learn".
2. À chaque itération, un pli différent est utilisé comme données de validation, les 4 autres servant à l'entraînement.
3. Cette alternance des plis d'entraînement/validation permet d'évaluer le modèle sur différentes portions des données.
4. Le mélange aléatoire des données dans les plis est fixé par une graine ($random\_state=0$) pour la reproductibilité.
5. Le modèle est entraîné sur les données d'entraînement de chaque pli pendant 10 époques.

2.5 Métriques d'évaluation utilisées pour mesurer la performance des modèles

Après l'apprentissage des modèles, il est essentiel d'évaluer leurs performances afin de les comparer objectivement, cette évaluation permet de déterminer l'efficacité de chaque modèle à généraliser sur des données non observées et de sélectionner le modèle le plus performant pour notre tâche . Pour ce faire, j'ai employé les métriques d'évaluation suivantes :

2.5.1 Matrice de confusion

Pour évaluer les performances d'un modèle de classification, la matrice de confusion représente un outil d'analyse essentiel. Elle permet de comparer visuellement les prédictions réalisées par le modèle aux véritables étiquettes des données, en décomptant les différents cas de prédictions correctes et incorrectes.

La matrice se compose de quatre éléments : les vrais positifs (VP), les vrais négatifs (VN), les faux positifs (FP), et les faux négatifs (FN).

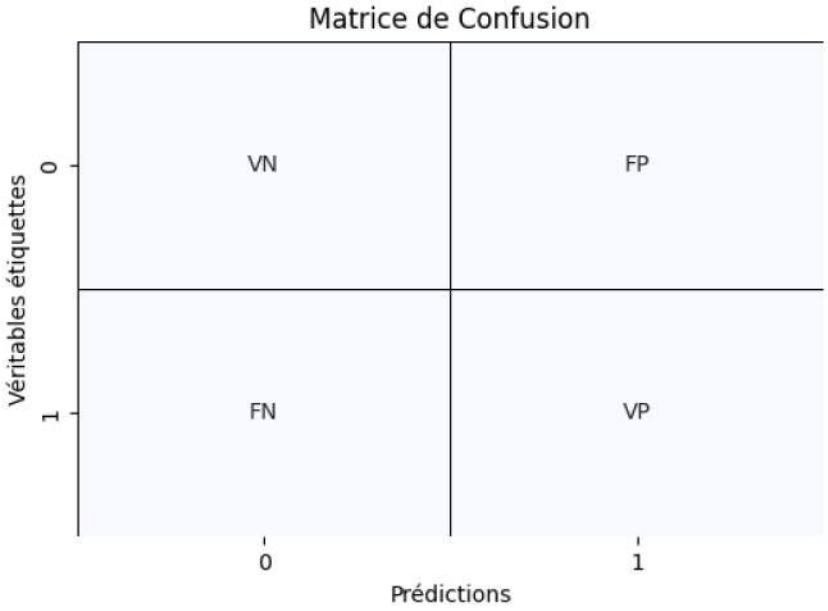

Figure 2.10 – Matrice de confusion

— **Vrais Positifs (VP)** : Exemples correctement prédits comme appartenant à la classe positive (ex : trafic TOR)
— **Vrais Négatifs (VN)** : Exemples correctement prédits comme appartenant à la classe négative (ex : non TOR)
— **Faux Positifs (FP)** : Exemples incorrectement prédits comme appartenant à la classe positive
— **Faux Négatifs (FN)** : Exemples incorrectement prédits comme appartenant à la classe négative

2.5.2 Le F1-score

Le F1-score est une métrique clé pour évaluer les performances d'un modèle de classification binaire. Il combine deux mesures importantes en une seule valeur, fournissant ainsi un bon aperçu global de la qualité du modèle. Ces deux mesures sont :

-La précision, qui indique à quel point le modèle est bon pour identifier correctement les exemples positifs parmi tous ceux qu'il a classés comme positifs. Elle se calcule comme suit :

$$\text{Précision} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs}}$$

-Le rappel, qui indique à quel point le modèle est capable de détecter tous les exemples

réellement positifs. Il se calcule ainsi :

$$\text{Rappel} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}}$$

Le F1-score combine alors la précision et le rappel en une seule valeur grâce à la formule suivante :

$$\text{F1-score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Un F1-score élevé indique que le modèle obtient à la fois une bonne précision et un bon rappel, ce qui en fait un bon modèle de classification binaire dans l'ensemble.

2.6 Outils et environnement de développement utilisés

Lors de la création de modèles de deep learning, plusieurs outils ont été utilisés, notammentt les bibliothèques Python, la plateforme Google Colab ainsi que Google Drive.

2.6.1 Google Drive

Google Drive est un service de stockage en ligne fourni par Google, qui attribue un espace de stockage de 15GB gratuit aux utilisateurs lié à leur compte Google. Ce service permet de conserver, de partager et d'accéder à des fichiers et des dossiers à partir de tout appareil connecté à Internet.

Pour notre projet, nous avons utilisé Google Drive pour stocker notre dataset et de nos modèles entraînés.

2.6.2 Google colab

Google Colaboratory (également connu sous le nom de Colab) est un service cloud basé sur les Jupyter Notebooks, destiné à la diffusion de l'éducation et de la recherche en apprentissage automatique. Il offre un environnement d'exécution entièrement configuré pour l'apprentissage profond.

Google Colab offre un CPU, 12-16 Go de RAM, et un accès limité à des GPU. Il permet jusqu'à 12 heures de session, environ 100 Go de stockage temporaire, et une intégration facile avec Google Drive.

Dans notre projet, nous avons utilisé Google Colab avec un processeur Intel Xeon fonctionnant à 2,20 GHz. Ce processeur possède un cœur physique capable de gérer deux threads grâce à l'Hyper-Threading, et il est équipé de 56 Mo de mémoire cache, et une mémoire RAM de 12 GO. Nous avons également utilisé la version 3 de Python pour le

prétraitement des données, ainsi que pour la construction et l'entraînement des modèles de deep learning.

2.6.3 Bibliothèques Python

Dans le cadre de la réalisation de ce projet, plusieurs bibliothèques Python ont été utilisées. Les bibliothèques permettent de simplifier et d'automatiser de nombreuses tâches complexes, telles que le prétraitement des données, la construction et l'entraînement de réseaux de neurones, ainsi que l'évaluation et la visualisation des performances des modèles. Les bibliothèques utilisées sont les suivantes :

2.6.3.1 Pandas

"Pandas" est une bibliothèque Python puissante et flexible pour la manipulation et l'analyse de données. Elle est utilisée pour différentes tâches, notamment le chargement de datasets et le prétraitement des données. Par exemple, nous avons utilisé Pandas pour charger notre dataset depuis Google Drive .

```
from google.colab import drive
import pandas as pd

# Montage de Google Drive pour accéder aux fichiers
drive.mount('/content/drive')

# Chemin d'accès au fichier CSV
file_path = "/content/drive/My Drive/tor_dataset/tor_nontor_2.csv"

data = pd.read_csv(file_path)
```

FIGURE 2.11 – Chargement du dataset avec "Pandas"

"Pandas" a également été utilisé pour effectuer des tâches de prétraitement, telles que la suppression de certaines colonnes inutiles du dataset.

```
# Liste des colonnes à supprimer
colonnes_a_supprimer = ["Active Mean", "Active Std", "Active Max",
                        "Active Min", "Idle Mean", "Idle Std", "Idle Max", "Idle Min"]

# Supprimer les colonnes spécifiées
data = data.drop(colonnes_a_supprimer, axis=1)
```

FIGURE 2.12 – Suppression de certaines colonnes

2.6.3.2 Scikit-learn

"Scikit-learn" est une bibliothèque Python qui offre une vaste gamme d'outils pour la modélisation et l'évaluation des données. Dans le cadre de ce projet, "Scikit-learn" a été utilisée pour diverses tâches essentielles, notamment la division du dataset, la mise à l'échelle des données avec la fonction "Minmaxscaler" .

```
scaler= MinMaxScaler(feature_range=(0,1))

scaler.fit(X_train)

▾ MinMaxScaler
MinMaxScaler()

X_train = scaler.transform(X_train)
```

Figure 2.13 – Normalisation des données avec MinMaxScaler

Scikit-learn a été utilisée aussi pour l'évaluation des modeles avec plusieur fonction comme la matrice de confusion et le F1-score

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
cm = confusion_matrix(y_test, y_pred_binary)
f1 = f1_score(y_test, y_pred_binary)
```

Figure 2.14 – Calcul de la matrice de confusion et du F1-score

2.6.3.3 TensorFlow

"TensorFlow", et en particulier son API "Keras", a été largement utilisée pour la construction, l'entraînement, et l'optimisation des modèles de deep learning dans ce projet.

2.6.3.4 Seaborn et Matplotlib

"Seaborn" et "Matplotlib" ont été utilisées pour la visualisation des données et des résultats, facilitant ainsi l'interprétation et la présentation des performances des modèles de deep learning dans ce projet.

2.7 Conclusion

Ce chapitre a détaillé la méthodologie pour comparer les performances de modèles de deep learning dans la détection de l'utilisation du réseau TOR. Il a couvert le prétraitement des données, les architectures de modèles (DNN et CNN), les techniques d'entraînement, et les métriques d'évaluation comme la matrice de confusion et le F1-score. L'utilisation d'outils comme Google Colab et diverses bibliothèques Python a facilité le processus.

Chapitre 3

Expérimentations et résultats

3.1 Introduction

Dans ce chapitre, nous détaillons les expérimentations menées pour évaluer les performances du modèle de réseaux de neurones dans la classification du trafic TOR. Nous avons examiné un modèle de réseau dense (DNN), un modèle de réseau convolutif (CNN) et un modèle CNN optimisé par validation croisée. Chaque modèle a été testé et évalué en utilisant la matrice de confusion et le F1-score, pour fournir une mesure précise de leurs performances.

Les modèles ont été testés sur un ensemble de données comportant des étiquettes indiquant le trafic normal et le trafic TOR. Nous avons également effectué des tests de prédiction sur dix exemples réels, comprenant à la fois des exemples positifs (trafic TOR) et des exemples négatifs (trafic non-TOR), pour évaluer la capacité de prédiction des modèles sur des données non vues pendant l'entraînement. Ces tests nous ont permis d'observer comment les modèles se comportent dans des scénarios proches des conditions réelles.

Nous terminons ce chapitre par une discussion sur les limitations de notre étude et des perspectives d'amélioration pour des recherches futures.

3.2 Évaluation des modèles

Dans cette section, nous explorerons l'évaluation des modèles, en examinant l'entraînement du réseau dense (DNN), du réseau convolutif (CNN), et du CNN avec validation croisée

3.2.1 Apprentissage de DNN

L'entraînement du modèle de réseau dense (DNN) a été réalisé en utilisant une stratégie de "early stopping" pour éviter le sur-apprentissage et optimiser les performances. Le modèle a été entraîné sur un maximum de 500 époques, mais l'amélioration de la performance s'est stabilisée, et l'entraînement s'est arrêté automatiquement après 66 époques. Les résultats montrent que le modèle a atteint une précision (accuracy) de 94,8% avec une perte (loss) de validation de 0,14.

Voici un aperçu des résultats obtenus au cours des premières et dernières époques de l'entraînement :

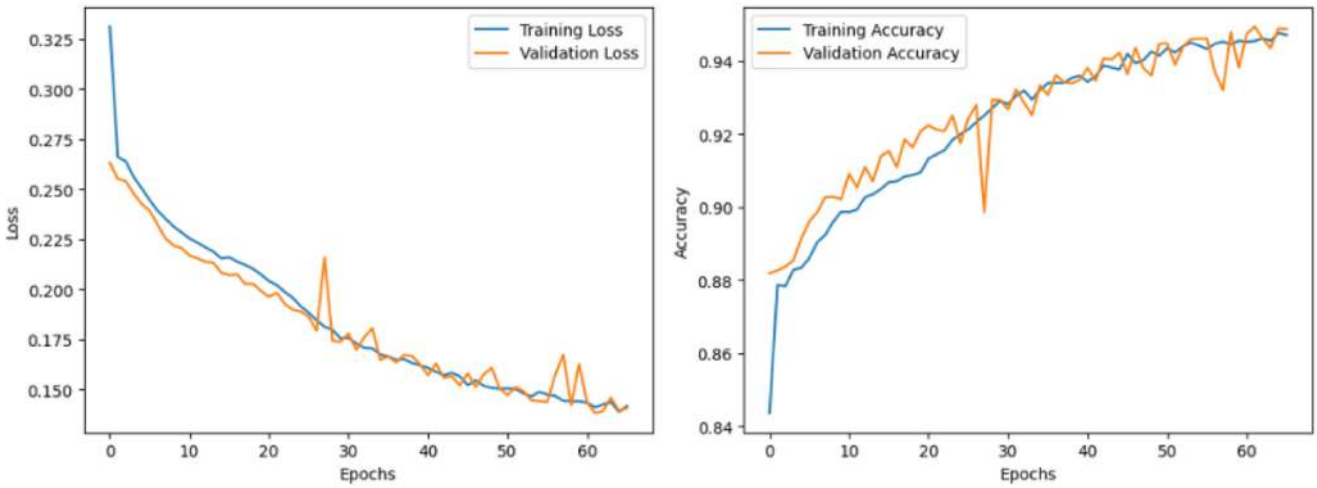

Figure 3.1 – Courbe d'apprentissage du DNN

- **Époque 1** : La perte initiale était de 0.3312 avec une précision de 84,36%, et la perte de validation était de 0.2631 avec une précision de validation de 88,19%.
- **Époque 20** : La perte a diminué à 0.2073 avec une précision de 90,95%, tandis que la perte de validation était de 0.1992 avec une précision de validation de 92,09%.
- **Époque 40** : La perte a continué de diminuer à 0.1619 avec une précision de 93,60%, et la perte de validation était de 0.1625 avec une précision de validation de 93,49%.
- **Époque 66** : La perte finale était de 0.1415 avec une précision de 94,71%, et la perte de validation était de 0.1406 avec une précision de validation de 94,88%.

Les résultats démontrent une amélioration continue de la précision et une réduction de la perte au fil des époques, jusqu'à ce que l'amélioration se stabilise.

3.2.2 Apprentissage de CNN

L'entraînement du modèle de réseau de neurones convolutionnels (CNN) a été réalisé en utilisant une stratégie de "early stopping" pour éviter le sur-apprentissage. L'entraînement s'est arrêté automatiquement après 32 époques. Les résultats montrent que le modèle a atteint une précision (*accuracy*) de 96,6% avec une perte (*loss*) de validation de 0,1006. Voici un aperçu des résultats obtenus au cours des premières et dernières époques de l'entraînement :

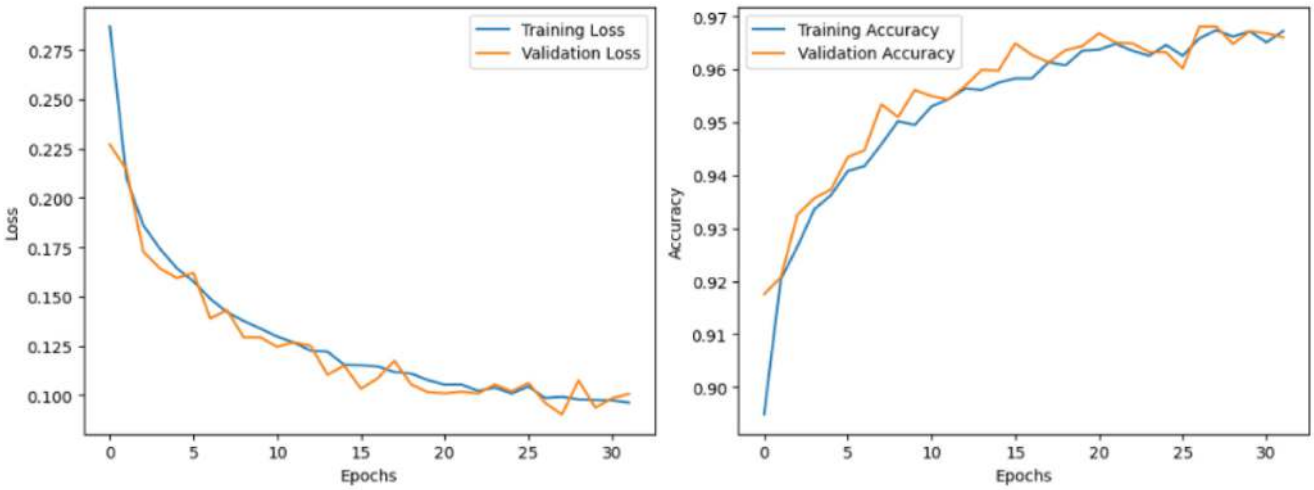

Figure 3.2 – Courbe d'apprentissage du CNN

- **Époque 1** : La perte initiale était de 0.2870 avec une précision de 89,48%, et la perte de validation était de 0.2271 avec une précision de validation de 91,75%.
- **Époque 10** : La perte a diminué à 0.1338 avec une précision de 94,95%, tandis que la perte de validation était de 0.1293 avec une précision de validation de 95,61%.
- **Époque 20** : La perte a continué de diminuer à 0.1076 avec une précision de 96,35%, et la perte de validation était de 0.1015 avec une précision de validation de 96,44%.
- **Époque 32** : La perte finale était de 0.0962 avec une précision de 96,72%, et la perte de validation était de 0.1006 avec une précision de validation de 96,60%.

Les résultats démontrent une amélioration continue de la précision et une réduction de la perte au fil des époques, jusqu'à ce que l'amélioration se stabilise. La stratégie d'*early stopping* a permis d'arrêter l'entraînement à un moment optimal, en évitant le risque de sur-apprentissage.

3.2.3 Apprentissage de Modèle CNN "Validation Croisée"

L'entraînement du modèle de réseau de neurones convolutif (CNN) a été réalisé en utilisant la technique de validation croisée afin d'améliorer la robustesse et la généralisation du modèle. L'ensemble de données a été divisé en 5 sous-ensembles et le modèle a été entraîné pendant 10 époques pour chaque sous-ensemble, atteignant une précision (*accuracy*) de 98,51% et une perte (*loss*) de 0,04.

Voici un aperçu des résultats obtenus au cours des premières et dernières époques de l'entraînement :

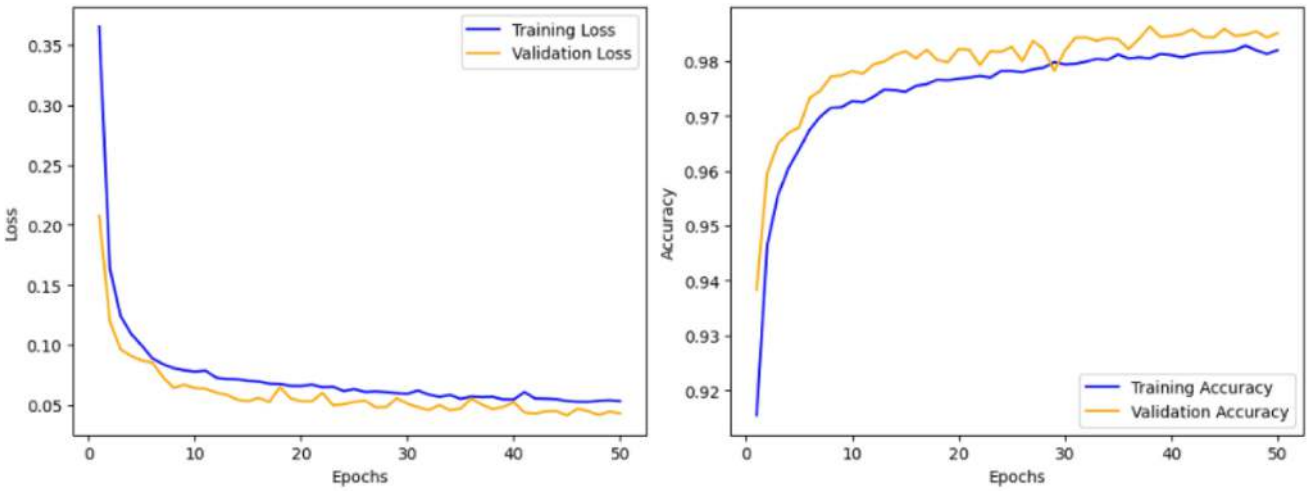

Figure 3.3 – Courbe d'apprentissage CNN avec la validation croisé

— **Époque 1** : La perte initiale était de 0.3654 avec une précision de 91,54%, et la perte de validation était de 0.2076 avec une précision de validation de 93,83%.
— **Époque 10** : La perte a diminué à 0.0777 avec une précision de 97,27%, tandis que la perte de validation était de 0.0639 avec une précision de validation de 97,82%.
— **Époque 20** : La perte a continué de diminuer à 0.0657 avec une précision de 97,68%, et la perte de validation était de 0.0530 avec une précision de validation de 98,22%.
— **Époque 30** : La perte a diminué à 0.0592 avec une précision de 97,99%, tandis que la perte de validation était de 0.0509 avec une précision de validation de 98,19%.
— **Époque 40** : La perte a continué de diminuer à 0.0544 avec une précision de 98,20%, et la perte de validation était de 0.0429 avec une précision de validation de 98,49%.
— **Époque 50** : La perte finale était de 0.0531 avec une précision de 98,27%, et la perte de validation était de 0.0416 avec une précision de validation de 98,51%.

Les résultats montrent une amélioration continue de la précision et une réduction de

la perte au fil des époques pour l'ensemble d'entraînement et de validation. La validation croisée a confirmé la capacité du modèlepour la tâche de classification, démontrant des performances consistantes sur différentes parties des données.

3.3 Analyse des résultats obtenus

Maintenant, passons à la phase d'analyse des résultats obtenus, où nous comparerons les performances des modèles DNN et CNN en utilisant les métriques d'évaluation discutées dans le chapitre 2.

3.3.1 Résultats de DNN

D'après la matrice de confusion et le F1-score, on peut faire les observations suivantes :

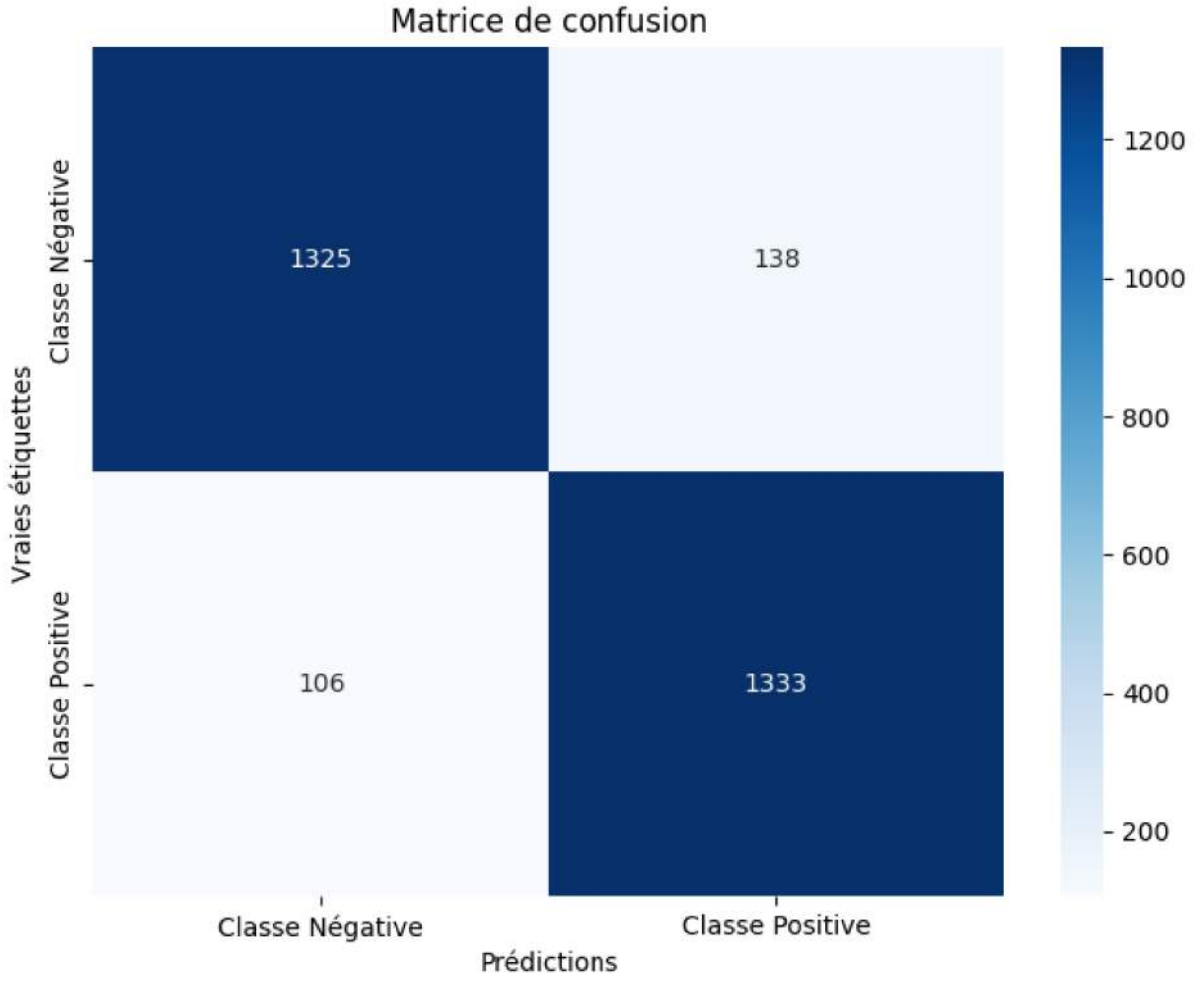

Figure 3.4 – Matrice de confusion du modèle DNN

1. Le modèle semble bien classifier les instances de la classe négative (non-TOR), avec 1325 instances correctement prédites négatives (vrais négatifs) et seulement 106 faux positifs.
2. Pour la classe positive (trafic TOR), le modèle obtient également de bons résultats avec 1333 vrais positifs et seulement 138 faux négatifs.
3. Le F1-score, qui combine à la fois la précision et le rappel, est de 0,9161. Un score F1 supérieur à 0,9 indique de très bonnes performances globales du modèle pour cette tâche de classification.

3.3.2 Résultats de CNN

D'après la matrice de confusion et le F1-score, on peut analyser les performances du modèle CNN comme suit :

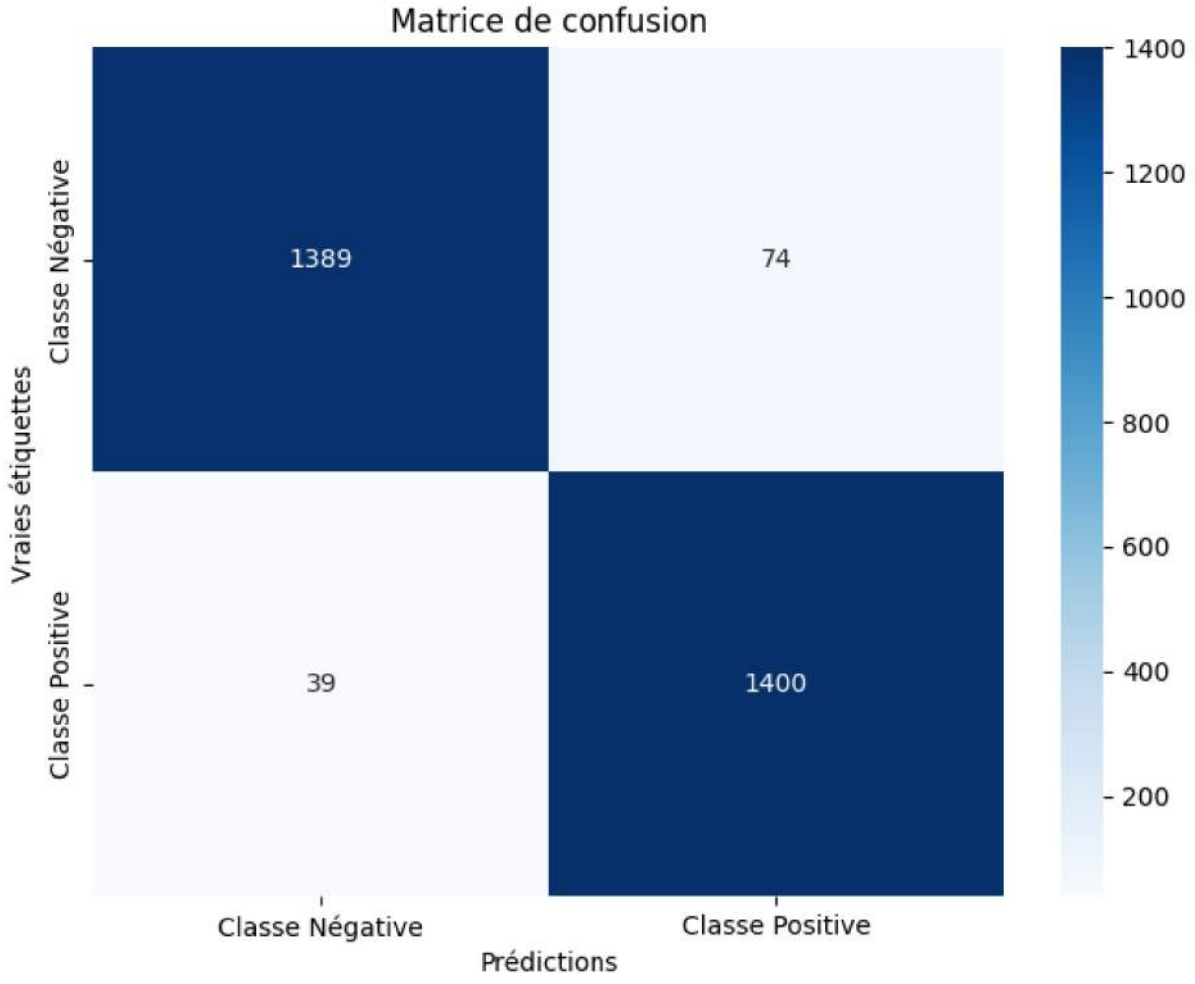

Figure 3.5 – Matrice de confusion du modèle CNN

1. Pour la classe négative (non-TOR), le modèle a correctement prédit 1389 instances négatives (vrais négatifs) et a commis 39 faux positifs.
2. En ce qui concerne la classe positive (trafic TOR), le modèle a identifié avec succès 1400 instances positives (vrais positifs) mais a raté 74 instances positives (faux négatifs).
3. Le F1-score combiné de 0,9612 témoigne d'excellentes performances globales.

3.3.3 Résultats de CNN entraîné avec la méthode de validation croisée

D'après la matrice de confusion et le F1-score, le modèle CNN entraîné avec la méthode de validation croisée semble offrir des performances remarquables :

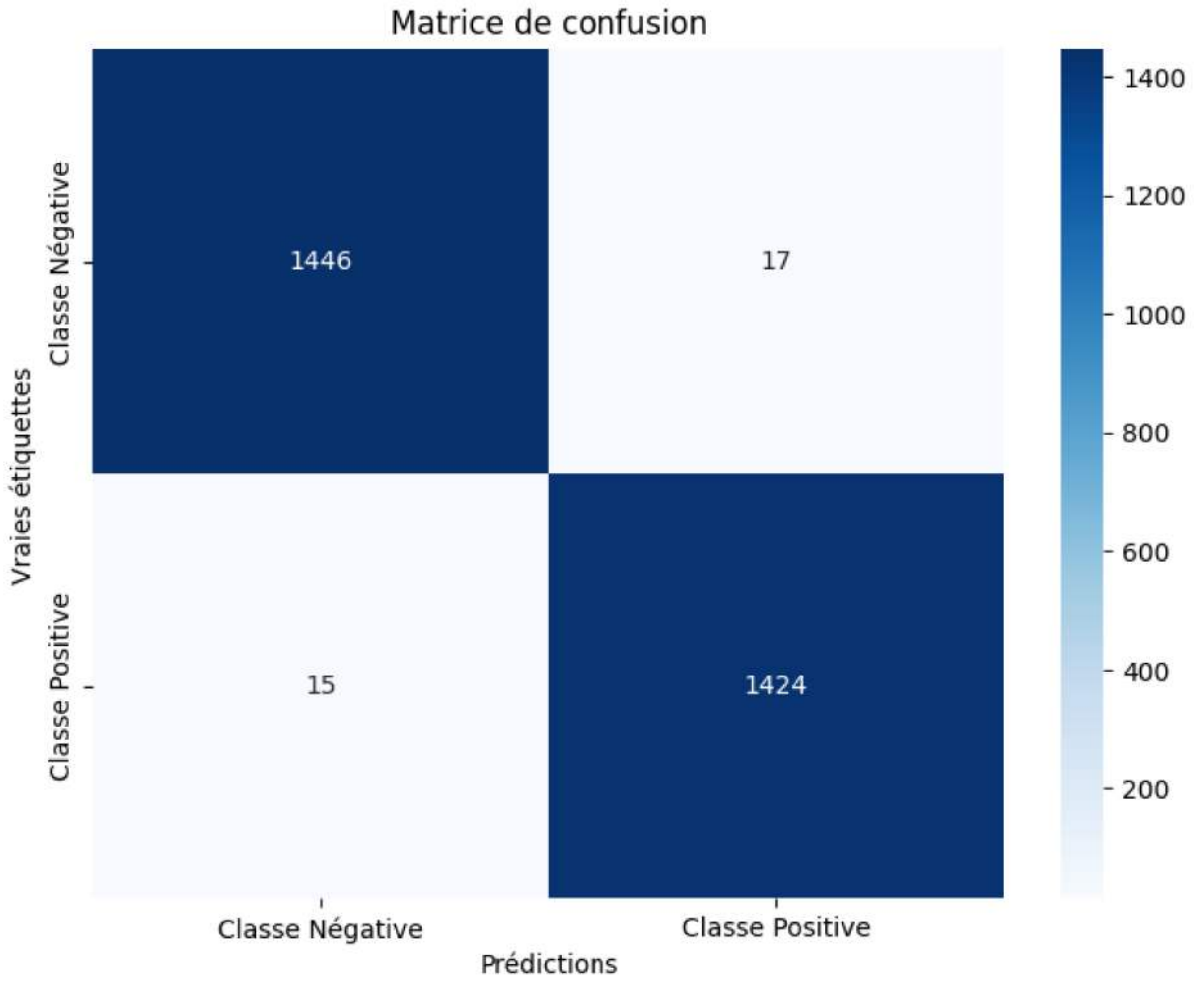

Figure 3.6 – Matrice de confusion du modèle CNN entraîné avec la validation croisée

1. Pour la classe négative (non-TOR), le modèle a correctement prédit 1446 instances sur 1461, avec seulement 15 faux négatifs.
2. Concernant la classe positive (trafic TOR), le modèle a identifié avec une grande précision 1424 instances sur 1441, ne commettant que 17 faux positifs.
3. Le F1-score combiné atteint un niveau exceptionnel de 0,9888.

3.4 Interprétation des résultats

Maintenant, passons à la phase d'interprétation des résultats où nous allons comparer les performances des modèles DNN, CNN, et CNN entraîné avec la validation croisée en utilisant les matrices de confusion et le F1-score.

Modèle	**Vrais négatifs**	**Faux positifs**	**Vrais positifs**	**Faux négatifs**	**F1-score**
DNN	1325	106	1333	138	0,9161
CNN	1389	39	1400	74	0,9612
CNN avec validation croisée	1446	15	1424	17	0,9888

Tableau 3.1 – Comparaison des performances des modèles

En comparant les matrices de confusion et les F1-scores, il est évident que le modèle CNN optimisé par validation croisée offre les performances globales les plus élevées. Il minimise à la fois les fausses positives et les fausses négatives, ce qui se reflète dans son F1-score remarquable de 0,9888.

Pour comparer les performances de trois modèles, nous avons analysé leurs prédictions sur cinq exemples positifs et cinq exemples négatifs. Les résultats obtenus montrent des variations significatives en termes de probabilité de classification correcte.

-pour le modèle DNN :

Exemple	Classe	Probabilité
Exemple positif 1	Positif	89.22%
Exemple positif 2	Positif	95.45%
Exemple positif 3	Positif	52.58%
Exemple positif 4	Positif	89.93%
Exemple positif 5	Positif	78.11%
Exemple négatif 1	Négatif	0.00%
Exemple négatif 2	Négatif	0.00%
Exemple négatif 3	Négatif	0.00%
Exemple négatif 4	Négatif	0.00%
Exemple négatif 5	Négatif	14.67%

Figure 3.7 – Prédictions du modèle DNN

les probabilités de classification des exemples positifs varient entre 52.58% et 95.45%, tandis que les exemples négatifs sont classés avec une probabilité proche de zéro, sauf pour un cas à 14.67%.

-Pour le model CNN :

Exemple	Classe	Probabilité
Exemple positif 1	Positif	99.81%
Exemple positif 2	Positif	94.65%
Exemple positif 3	Positif	77.33%
Exemple positif 4	Positif	85.97%
Exemple positif 5	Positif	99.75%
Exemple négatif 1	Négatif	0.00%
Exemple négatif 2	Négatif	0.00%
Exemple négatif 3	Négatif	0.00%
Exemple négatif 4	Négatif	0.00%
Exemple négatif 5	Négatif	6.03%

Figure 3.8 – Prédictions de modèle CNN

Le modèle CNN, présente une meilleure performance sur les exemples positifs, avec des probabilités allant de 77.33% à 99.81%, et maintient une probabilité proche de zéro pour les exemples négatifs, à l'exception d'un cas à 6.03%.

-Pour le CNN avec la validation croisé :

Exemple	Classe	Probabilité
Exemple positif 1	Positif	98.14%
Exemple positif 2	Positif	99.51%
Exemple positif 3	Positif	95.60%
Exemple positif 4	Positif	99.03%
Exemple positif 5	Positif	99.08%
Exemple négatif 1	Négatif	0.00%
Exemple négatif 2	Négatif	0.00%
Exemple négatif 3	Négatif	0.00%
Exemple négatif 4	Négatif	0.00%
Exemple négatif 5	Négatif	4.70%

Figure 3.9 – Prédictions de modèle CNN avec la validation croisé

Le modèle CNN entraîné avec validation croisée montre les meilleurs résultats parmi les trois, avec des probabilités de classification des exemples positifs allant de 95.60% à 99.51%, et des probabilités pour les exemples négatifs se maintenant à zéro, à l'exception d'un cas à 4.70%. Ces observations indiquent que l'utilisation de la validation croisée améliore la robustesse et la précision du modèle CNN.

Les modèles DNN et CNN présentent également de bonnes performances, mais ils sont dépassés par le CNN optimisé avec la validation croisée.

3.5 Limitations de l'étude

Malgré les performances remarquables obtenues avec le modèle CNN optimisé avec la validation croisée, plusieurs limitations doivent être prises en compte :

1. **Variabilité des données** : Si les caractéristiques du trafic TOR évoluent ou si de nouvelles méthodes de dissimulation du trafic apparaissent, les performances des modèles peuvent diminuer.
2. **Test dans un environnement réel** : Les modèles CNN ont été précis sur les données de test, mais ils n'ont pas été complètement testés dans des environnements réels. Des tests supplémentaires sont nécessaires pour vérifier si les modèles fonctionnent aussi bien dans des conditions réelles.

3. **Équilibre des classes** : Le jeu de données contient beaucoup plus d'exemples de la classe trafic non-TOR que de trafic TOR. Bien que la technique d'undersampling ait été utilisée pour équilibrer les classes, cela peut conduire à une perte d'informations.

3.6 Perspectives d'amélioration

Plusieurs approches peuvent être explorées pour améliorer les performances :

1. **Enrichissement du jeu de données** : Pour que les modèles détectent mieux les nouveaux types de trafic TOR, il faut augmenter la diversité et la quantité des données d'entraînement. En incluant des données de différentes périodes et sources, on capture mieux les variations possibles. Utiliser des modèles comme les réseaux adversariaux génératifs (GAN) peut aussi aider. Les GAN peuvent générer des données synthétiques réalistes qui enrichissent le jeu de données.
2. **Déploiement et tests en conditions réelles** : En testant les modèles dans des conditions réelles, on peut détecter les problèmes pratiques et apporter les modifications nécessaires pour une utilisation optimale en production.

3.7 Conclusion

Ce chapitre a présenté les expérimentations et les résultats obtenus pour l'évaluation de modèles de réseaux de neurones dans la classification du trafic TOR. Nous avons analysé les performances d'un réseau de neurones dense (DNN), d'un réseau de neurones convolutif (CNN) et d'un CNN optimisé par validation croisée.

Les résultats montrent que le modèle CNN, en particulier celui optimisé avec la validation croisée, surpasse les autres modèles en termes de précision et de robustesse. Le F1-score élevé de 0,9888 atteint par ce modèle témoigne de ses performances exceptionnelles dans la tâche de classification, minimisant les faux positifs et les faux négatifs.

Cependant, malgré ces performances, nous avons également identifié certaines limitations de notre étude. La variabilité des données, l'absence de tests en conditions réelles et le déséquilibre des classes restent des défis à surmonter. Nous avons discuté des pistes d'amélioration possibles, telles que l'enrichissement du jeu de données et le déploiement des modèles dans des environnements réels.

En conclusion, bien que les modèles CNN aient montré des performances remarquables dans nos expérimentations, des travaux supplémentaires sont nécessaires pour garantir leur efficacité et leur adaptabilité à des scénarios réels et variés.

Conclusion Générale

En conclusion, ce mémoire démontre l'importance du deep learning dans la détection du réseau TOR pour préserver la confidentialité et renforcer la sécurité.

Après avoir acquis des connaissances approfondies, nous avons développé, entraîné et testé divers modèles de deep learning, notamment des réseaux de neurones profonds (DNN) et des réseaux de neurones convolutifs (CNN), pour identifier le trafic TOR. Nous avons montré que les modèles CNN, surtout ceux optimisés par validation croisée, offrent une grande précision et fiabilité, atteignant un F1-score élevé. Nous avons également souligné l'importance du traitement des données pour de meilleures performances.

Les modèles que nous avons développés peuvent être implémentés sur des pare-feu, des systèmes de détection d'intrusion (IDS) ou de prévention d'intrusion (IPS), ainsi que sur des logiciels antivirus pour détecter les activités associées à TOR.

Nous aimons tester nos modèles dans des conditions réelles, car cela nous permettra de vérifier leur performance en pratique. Pour les travaux futurs, plusieurs approches peuvent être explorées pour améliorer encore nos modèles, notamment l'augmentation du jeu de données en incluant plus d'échantillons. L'utilisation des GANs pourrait être une solution pour produire des échantillons synthétiques, ce qui enrichirait le dataset et améliorerait la capacité du modèle à bien prédire les deux classes. En outre, il est important de traiter le déséquilibre des classes dans les données d'entraînement pour améliorer la détection des classes minoritaires.

Bibliographie

[1] Qu'est-ce qu'un réseau de neurones ? URL : https://www.ibm.com/fr-fr/topics/neural-networks.

[2] The tor relay guide. https://trac.torproject.org/projects/tor/wiki/TorRelayGuide.

[3] Tor, la face chiffrée d'internet : entretien avec lunar. *Vacarme*, 69 :79–98, 2014. doi:10.3917/vaca.069.0079.

[4] Salam Al-E'mari, Yousef Sanjalawe, and Salam Fraihat. Detection of obfuscated tor traffic based on bidirectional generative adversarial networks and vision transform. *Computers & Security*, 135 :103512, 2023. doi:10.1016/j.cose.2023.103512.

[5] ArcGIS Pro. Introduction à l'apprentissage profond, 2024. URL : https://pro.arcgis.com/fr/pro-app/latest/help/analysis/deep-learning/what-is-deep-learning-.htm.

[6] A. Balcerac, B. Tervil, N. Vayatis, and D. Ricard. Principes fondamentaux de l'apprentissage automatique pour les neurologues. *Pratique Neurologique - FMC*, 14(4) :225–236, 2023. doi:10.1016/j.praneu.2023.10.005.

[7] E. T. de O. Chagas. L'apprentissage profond et ses applications aujourd'hui. *Núcleo do Conhecimento*, mai 2019. RC : 29521, 451. URL : https://www.nucleodoconhecimento.com.br/administration-des-affaires/apprentissage-approfondi.

[8] G. Chalons. Le cycle de vie d'un projet de machine learning en 8 étapes. Consulté sur Kaizen Solutions, décembre 2022. URL : https://kaizen-solutions.net/kaizen-insights/articles-et-conseils-de-nos-experts/cycle-de-vie-projet-machine-learning-8-etapes/.

[9] Solemane Coulibaly. *Analyse intelligente des images pour la surveillance dans une agriculture de précision*. PhD thesis, Institut National Polytechnique de Toulouse - INPT ; Université des Sciences Techniques et Technologiques de Bamako (Mali), 2021.

[10] Data Future. [tuto] fabrique & comprends ton premier réseau de neurones en partant de zéro ! Consulté en avril 2024. URL : https://datafuture.fr/post/fabrique-ton-premier-reseau-de-neurones/.

[11] P. De Filippi and F. Huguet. Tor. In C. Méadel and F. Musiani, editors, *Abécédaire des architectures distribuées*. Presses des Mines, Paris, 2015. doi:10.4000/books.pressesmines.2144.

[12] L. Etienne. TOR et « reTORs » : détecter du trafic TOR sur son réseau. Consulté en mars 2024. URL : https://geekeries.org/2016/07/detecter-du-trafic-tor-sur-son-reseau/?doing_wp_cron=1715426911.8267540931701660156250.

[13] Arash Habibi Lashkari, Gerard Draper-Gil, Mohammad Saiful Islam Mamun, and Ali A. Ghorbani. Characterization of tor traffic using time based features. In *Proceedings of the 3rd International Conference on Information System Security and Privacy (ICISSP)*, pages 3–6, Porto, Portugal, 2017. SCITEPRESS.

[14] moncoachdata.com. Comprendre les réseaux de neurones, 8 2022. URL : https://moncoachdata.com/blog/comprendre-les-reseaux-de-neurones/.

[15] primaveradefilippi. Tor : le détournement d'une technologie détournée, 12 2011. URL : https://doi.org/10.58079/alfp.

[16] A. Schmid. Tor browser est désormais optimisé et natif pour les mac apple silicon. *Clubic*, décembre 2022. URL : https://www.clubic.com/navigateur-internet/actualite-450126-tor-browser-fait-son-grand-retour-sur-mac.html.

[17] Label Your Data Team. What is a dataset in machine learning ?, 2023. URL : https://labelyourdata.com/articles/what-is-dataset-in-machine-learning.

[18] Tomorrow Bio. Weight for it : Comment les réseaux neuronaux deviennent plus forts. *Intelligence Artificielle*, juin 2023. URL : https://www.tomorrow.bio/fr/poste/weight-for-It-comment-les-r%C3%A9seaux-neuronaux-deviennent-plus-forts-2023-06-4669545292-ai.

[19] X. Van Caneghem. Surface, deep web, dark web : les 3 couches d'internet. Consulté en mars 2024. URL : https://blog.europ-assistance.be/surface-deep-web-dark-web-3-couches-internet/.

[20] M. Vitali-Rosati and M. E. Sinatra. *Parcours numérique*. Presses de l'Université de Montréal, Montréal, 2014. doi:10.4000/books.pum.306.

[21] Développement Web. Qu'est-ce que le deep web ? Consulté en avril 2024. URL : https://www.ionos.fr/digitalguide/sites-internet/developpement-web/deep-web/.