République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة أبو بكر بلقايد– تلمسان
Université ABOUBEKR BELKAID – TLEMCEN
Faculté de technologie

Mémoire de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

En : Télécommunications

Spécialité : Réseaux et Télécommunications

Présenté par **:**

* *M* ***MALTI*** *Hamza Ramzy*

* *M*^{elle} ***MERZOUGUI*** *Rania Chahinez*

Thème

Analyse du trafic DNS pour détecter les attaques hybrides en utilisant Deep Learning

Soutenu le 12 Juin 2024 , devant le Jury composé de :

Mr BENDIMERAD Yacine	MCA	Univ. Tlemcen	*Président*
Mme BENAISSA Amel	MAB	Univ- Tlemcen	*Examinatrice*
Mr HADJILA Mourad	MCA	Univ- Tlemcen	*Encadreur*
Mme FERHI Wafaa	Doctorante	Univ-Tlemcen	*Co-encadreur*


بسم الله الرحمن الرحيم

Dédicace

Dédié ce mémoire...

À mes chers parents

À mes parents, dont le soutien indéfectible et les valeurs transmises ont façonné l'individu que je suis aujourd'hui. Aucune dédicace ne saurait pleinement exprimer le respect, la gratitude et l'admiration profonde que je vous porte. Puisse ce travail être un humble témoignage de mon affection et de mon amour éternels.

À mon frère et mes deux sœurs

Dont la présence constante et l'affection sincère ont enrichi ma vie de manière profonde. Leur soutien, leur complicité et leur amour ont été une source inépuisable de force et d'inspiration tout au long de ce parcours.

À mes neveux et nièces

Véritables joyaux de notre famille, dont l'innocence et la vivacité illuminent nos vies. Votre enthousiasme et votre énergie apportent une lumière et une chaleur inestimables à notre quotidien.

À chaque membre de cette famille, je dédie ce modeste travail avec amour et gratitude pour tout ce que vous êtes et tout ce que vous avez apporté dans ma vie.

Ramzy

Dédicace

Dédié ce mémoire...

À mes chers parents

À mon père, dont les valeurs et les leçons continuent de m'inspirer chaque jour. Son absence, bien que silencieuse, demeure une présence constante qui m'incite à toujours donner le meilleur de moi-même.

À ma mère, pour son amour inconditionnel, sa patience infinie et son soutien indéfectible. Elle a été mon pilier, ma source de motivation et un modèle de persévérance sans faille. Nulle dédicace ne puisse exprimer ce que je vous dois.

À mes chers frères

À mes frères NIZAR et RAMZI, qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

À mon amie et co-encadrante Wafaa, merci d'avoir été une source inépuisable de soutien et d'encouragement. Merci pour ton amitié et ton soutien tout au long de cette année.

À mes chères amies , avec qui j'ai partagé les plus beaux et agréables moments de mon parcours universitaire. Merci pour votre soutien inconditionnel, votre amitié et vos sourires

Et en dernier lieu, à moi-même, je souhaite exprimer ma reconnaissance pour les heures prolongées, la dévotion et l'ardeur que j'ai investies dans ce parcours. Cet accomplissement est le reflet de mes compétences, et je suis profondément fière de ce que j'ai accompli. Avec amour et gratitude sincères.

Rania

Remerciements

Nous souhaitons tout d'abord exprimer notre gratitude à Dieu Tout-Puissant et Bienveillant pour nous avoir guidés tout au long de cette aventure académique.

Nous tenons à adresser nos plus sincères remerciements à toutes les personnes ayant contribué à la réalisation de ce mémoire.

Nous exprimons notre profonde reconnaissance à notre directeur de recherche, M. Hadjila, pour son soutien indéfectible et son encadrement précieux durant cette étude.

Nous adressons également nos vifs remerciements à Mlle Ferhi Wafaa, notre co-encadrante, pour ses contributions inestimables. Ses suggestions et critiques constructives ont grandement aidé à affiner et perfectionner notre travail de recherche.

Enfin, nous exprimons notre gratitude aux membres du jury pour l'intérêt qu'ils portent à notre travail et pour leur disponibilité à évaluer notre mémoire.

RÉSUMÉ

Les attaques DNS (Domain Name System) exploitent les vulnérabilités du système DNS, essentiel pour la traduction des noms de domaine en adresses IP. Ces attaques incluent le tunneling DNS, les inondations UDP, l'amplification DNS et l'empoisonnement du cache. Elles peuvent entraîner des vols de données, des interruptions de service et des compromissions de sécurité, soulignant l'importance de stratégies de détection et de prévention robustes. Ce mémoire, intitulé "Lightweight Hybrid Data Exfiltration using DNS based on Deep Learning", examine le développement d'une méthode pour détecter et prévenir les attaques d'exfiltration de données en exploitant les vulnérabilités du DNS. L'objectif principal est de concevoir un modèle efficace utilisant l'apprentissage profond pour analyser et sécuriser le trafic DNS contre les menaces potentielles. Nous avons créer 3 datasets à partir du dataset(CIC-Bell-DNS-EXT 2021) et chaque modèle est entraîné et testé à l'aide de ces 3 datasets, chacun de ses derniers a nécessité plusieurs tâches de prétraitement telles que, la suppression des valeurs manquantes, la conversion des données catégorielles en données numériques et la mise à l'échelle des caractéristiques. Chaque modèle d'apprentissage en profondeur proposé a obtenu des résultats impressionnants et ces derniers démontrent l'efficacité de l'apprentissage en profondeur dans la détection d'attaques DNS et soulignent l'importance d'une analyse et d'un prétraitement de données minutieux.

Mots clés : DNS(Domain Name System), Apprentissage en profondeur(Deep Learning), Dataset(jeu de données), Vulnérabilités.

ABSTRACT

DNS (Domain Name System) attacks exploit vulnerabilities in the DNS system, which is essential for translating domain names into IP addresses. These attacks include DNS tunneling, UDP flooding, DNS amplification and cache poisoning. They can lead to data theft, service interruptions and security compromises, underlining the importance of robust detection and prevention strategies. This dissertation, entitled "Lightweight Hybrid Data Exfiltration using DNS based on Deep Learning", examines the development of a method for detecting and preventing data exfiltration attacks by exploiting DNS vulnerabilities. The main objective is to design an efficient model using deep learning to analyze and secure DNS traffic against potential threats. We have created 3 datasets from the dataset(CIC-Bell-DNS-EXT 2021) and each model is trained and tested using these 3 datasets, each of which required several pre-processing tasks such as, removal of missing values, conversion of categorical data to numerical data and feature scaling. Each proposed deep learning model achieved impressive results, demonstrating the effectiveness of deep learning in detecting DNS attacks and underlining the importance of careful data analysis and pre-processing.

KEY WORDS : DNS (Domain Name System), Deep learning, vulnerabilities, Dataset.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

GLOSSAIRE

A	Address
Adagrad	Adaptive Gradient Algorithm
Adam	Adaptive Moment Estimation
ADASYN	Adaptive Synthetic Sampling Approach for Imbalanced Learning
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
BGD	Batch Gradient Descent
CNAME	Canonical Name
CNN	Convolutif Neural Network
CPU	Central Processing Unit
DDoS	Distributed Denial-Of-Service
DNN	Deep Neural Network
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DoH	DNS Over HTTPS

DoT	DNS Over TLS
ENN	Edited Nearest Neighbors
GPU	Graphic Processing Unit
HDD	Hard Disk Drive
HTTPS	Hypertext Transfer Protocol Secure
IA	Intelligence Artificielle
ICMP	Interner Control Message Protocol
IDS	Intrusion Detection System
IP	Internet Protocol
KPI	Key Performance Indicator
MX	Mail Exchange
NLP	Natural Language Processing
NS	Name Server
PCA	Principal Component Analysis
PTR	Pointer Record
RAM	Random Access Memory
RMSProp	Root Mean Square Propagation
RNP	Réseau Neurones Profond
ROS	Random Over Sampling
RUS	Random Under Sampling
SMOTE	Synthetic Minority Over-sampling Technique
SOA	Start of Authority
SSD	Solid State Drive
TCP	Transmission Control Protocol

TLD	Top Level Domain
TLS	Transport Layer Security
TTL	Time to Live
UDP	User Datagram Protocol

INTRODUCTION GÉNÉRALE

Dans un monde de plus en plus connecté, la sécurité des réseaux de télécommunications est devenue un enjeu primordial. Les attaques informatiques se multiplient et se diversifient, rendant nécessaire le développement de solutions de défense toujours plus sophistiquées. Parmi les nombreuses menaces existantes, l'exfiltration de données via le système de noms de domaine (DNS) est particulièrement préoccupante. Ce type d'attaque, souvent difficile à détecter, permet aux cybercriminels de transférer des données sensibles en utilisant des canaux de communication apparemment légitimes.

Le DNS, composant essentiel de l'infrastructure internet, traduit les noms de domaine en adresses IP, facilitant ainsi la navigation et la communication sur le web. Cependant, cette fonctionnalité peut être exploitée à des fins malveillantes. Les attaquants utilisent des techniques telles que le tunneling DNS pour contourner les pare-feux et autres mesures de sécurité, exfiltrant ainsi des données sans éveiller de soupçons.

Dans ce contexte, le deep learning, une branche avancée de l'intelligence artificielle, offre des perspectives prometteuses. Cette discipline révolutionne l'apprentissage automatique en dotant les systèmes informatiques de capacités exceptionnelles pour comprendre et interagir intelligemment avec leur environnement. Fondé sur les réseaux de neurones profonds, qui imitent les fonctions du cerveau humain pour extraire des représentations hiérarchiques à partir des données, l'apprentissage profond permet aux machines d'apprendre des concepts complexes à partir de vastes ensembles de données et de les appliquer de manière pertinente.

Notre mémoire s'articule autour d'une analyse approfondie des méthodes de Deep Learning et de leur efficacité dans le domaine de la détection d'intrusions. Plus

spécifiquement, nous explorerons les approches qui ont prouvé leur fiabilité et leur pertinence dans la reconnaissance et la prévention des attaques DNS. Le plan de notre mémoire se décline comme suit :

- Dans le premier chapitre, nous avons proposé une introduction détaillée au système de noms de domaine (DNS), en expliquant ses fonctions fondamentales, son fonctionnement interne, ainsi que les protocoles et outils associés. Nous examinerons également les vulnérabilités courantes et les types d'attaques auxquelles le DNS est fréquemment exposé.
- Dans le chapitre suivant, nous explorons l'usage du deep learning pour détecter les attaques DNS. Nous plongerons dans les fondements théoriques du deep learning, les divers modèles neuronaux utilisés, ainsi que les techniques de prétraitement des données qui visent à accroître la précision de cette détection des menaces.
- Et pour le dernier chapitre, nous dévoilons les résultats issus de l'application des modèles de deep learning sur le jeu de données CIC-Bell-DNS-EXT 2021. Nous évaluons les performances de ces modèles en termes de précision, de rappel et de score F1. Ces résultats mettent en lumière l'efficacité remarquable des techniques de deep learning dans la détection des attaques d'exfiltration de données via le protocole DNS.

CHAPITRE 1

DOMAIN NAME SYSTEM

1.1 Introduction

Le système de noms de domaine (DNS) est essentiel à l'Internet, fonctionnant comme un répertoire qui convertit les noms de domaine en adresses IP , permettant aux utilisateurs d'accéder aux sites Web et services en ligne.

Malheureusement, les attaques contre le DNS sont de plus en plus fréquentes et diversifiées, elles peuvent prendre de nombreuses formes, allant des attaques de déni de service distribué (DDoS) visant à saturer les serveurs DNS, aux attaques de détournement DNS où les requêtes DNS sont redirigées vers des serveurs malveillants contrôlés par des pirates informatiques.

Cependant, pour se protéger, les professionnels de la sécurité informatique doivent mettre en œuvre des mesures de protection robustes pour sécuriser leurs infrastructures DNS. Cela peut inclure l'utilisation de pare-feu DNS, de filtrage de contenu, de surveillance du trafic DNS et de solutions de détection des menaces pour détecter et atténuer les attaques DNS avant qu'elles ne causent des dommages.

1.2 Définition

Le système de noms de domaine (DNS), qui est généralement abrégé en DNS (Domain Name System), est un pilier essentiel de l'architecture d'Internet. Son rôle central consiste à fournir une manière pratique de traduire les noms de domaine que les humains utilisent pour accéder aux sites web et aux services en lignes, en adresses IP numériques nécessaires aux ordinateurs et autres appareils connectés pour localiser et communiquer entre eux sur le réseau mondial.

En résumé, le DNS agit comme une sorte de traducteur universel entre les noms de domaine conviviaux pour les humains et les adresses IP numériques nécessaires aux machines pour naviguer sur Internet. Sans DNS, l'accès à des sites web et à d'autres services en ligne serait considérablement plus difficile pour les utilisateurs.

1.3 Fonctionnement du DNS

Lorsque nous recherchons un site Internet à l'aide d'un nom de domaine sur notre navigateur, nous entamons un processus appelé « lookup » (recherche). Le processus de recherche se déroule en 6 étapes (voir Figure 1.1) :

1. Notre navigateur Internet et notre système d'exploitation tentent de rappeler l'adresse IP associée au nom de domaine. Si cette adresse a été visitée précédemment, elle peut être rappelée à partir du stockage interne de l'ordinateur ou de la mémoire cache.
2. Le processus se poursuit si aucun des deux composants ne sait où se trouve l'adresse IP de destination.
3. Le système d'exploitation demande l'adresse IP au serveur de noms de résolution. Cette requête entame la recherche dans la chaîne de serveurs du système de nom de domaine pour trouver l'adresse IP correspondant au domaine.
4. La requête parvient d'abord au serveur de noms racine, qui la dirige vers le serveur de premier niveau (via le résolveur).
5. Le serveur de premier niveau transmet ensuite notre requête, ou la fait pointer, vers le serveur de noms faisant autorité (à nouveau, via le résolveur).
6. Finalement, le résolveur, en communiquant avec le serveur de noms faisant autorité, trouvera l'adresse IP et la fournira au système d'exploitation, qui la transmettra au navigateur Internet, nous présentant ainsi la page ou le site Internet demandé.

Le processus de recherche DNS est le système de base utilisé par l'ensemble du réseau Internet.

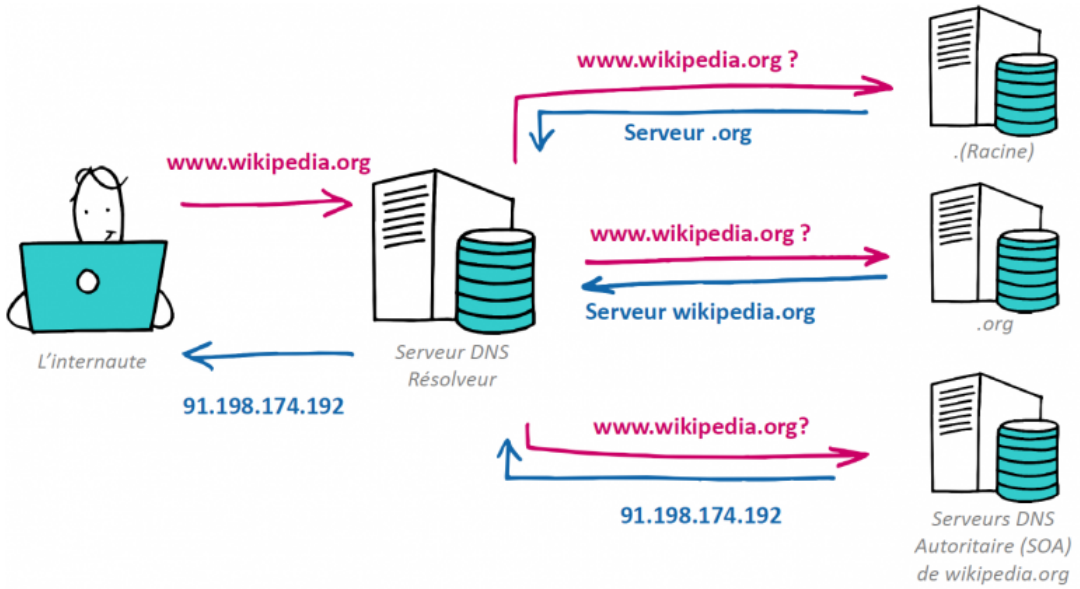


Figure 1.1 – Schématisation du fonctionnement du DNS [1].

1.3.1 Serveur DNS

Un serveur DNS est un composant crucial d'Internet qui traduit les noms de domaine en adresses IP. Il fonctionne en hiérarchie, avec des serveurs racine dirigeant les requêtes vers les serveurs DNS appropriés. Les serveurs DNS conservent souvent en mémoire les résultats de requêtes récentes pour améliorer les performances. Ils sont essentiels pour l'accessibilité et la sécurité d'Internet.

- **Récurseur DNS :** Le récurseur DNS, aussi appelé résolveur DNS, est un composant logiciel qui permet de traduire les noms de domaine en adresses IP et vice versa. Lorsque nous saisissons un nom de domaine dans notre navigateur web, comme "www.example.com", le résolveur DNS est chargé de trouver l'adresse IP correspondante à ce nom de domaine afin que notre navigateur puisse communiquer avec le serveur hébergeant le site web [1].
- **Serveurs de noms racine :** Le serveur de noms racine est chargé de gérer la zone racine DNS d'Internet. Son rôle principal est de répondre aux requêtes pour les enregistrements dans cette zone en fournissant une liste des serveurs de noms faisant autorité associés au TLD approprié [1].
- **Serveurs de noms TLD :** Les serveurs de noms TLD stockent l'adresse IP du domaine de deuxième niveau contenu dans le TLD. Ils résolvent ensuite l'adresse IP du site Web et transfèrent la requête au serveur de noms du domaine [1].

- **Serveurs de noms autorisés :** Un serveur de noms faisant autorité fournit la réponse effective à une requête DNS. Il existe deux types de serveurs de noms faisant autorité : le **serveur maître**, qui détient les copies originales des enregistrements de zone, et le **serveur esclave**, qui est une réplique exacte du serveur maître. Le serveur esclave partage la charge du serveur DNS et agit comme une sauvegarde en cas de panne du serveur maître [1].

1.3.2 Différence entre serveur DNS et résolveur DNS

Un serveur DNS est un composant serveur qui stocke et gère les informations de résolution DNS pour les noms de domaine sous sa responsabilité, tandis qu'un un résolveur DNS est un logiciel client qui envoie des requêtes DNS pour traduire les noms de domaine.

- Serveur DNS :
  - Il répond aux requêtes DNS émises par les résolveurs DNS en fournissant les informations de résolution appropriées.
  - Les serveurs DNS peuvent être configurés pour gérer différents types de zones DNS, telles que les zones racines, les zones de domaine, les zones de recherche inversée, etc.
  - Ils jouent un rôle crucial dans la résolution des noms de domaine et le bon fonctionnement du système DNS.
- Résolveur DNS :
  - Il est généralement installé sur les appareils des utilisateurs, tels que les ordinateurs, les smartphones ou les routeurs.
  - Le rôle principal d'un résolveur DNS est de communiquer avec les serveurs DNS pour obtenir les informations de résolution nécessaires.

1.3.3 Zones DNS

Partie de l'espace de noms DNS gérée par une organisation. Contient les enregistrements autoritaires pour un domaine et ses sous-domaines.

1.3.4 Enregistrements DNS

Entrées de données stockées sur les serveurs DNS autoritaires. Ils contiennent les informations liant noms de domaine et adresses IP.
Les principaux types d'enregistrements DNS comprennent :

- A (Address) : Associe un nom de domaine à une adresse IP.
- CNAME (Canonical Name) : Alias d'un nom de domaine principal (record canonique).
- MX (Mail Exchange) : Indique les serveurs de messagerie associés à un domaine.
- NS (Name Server) : Indique les serveurs DNS autoritaires pour une zone donnée.
- PTR (Pointer) : Associe une adresse IP à un nom de domaine (utilisé dans les recherches inversées).
- SOA (Start of Authority) : Fournit des informations d'autorité sur une zone DNS spécifique.

1.3.5 TTL (Time-to-Live)

Durée de vie des enregistrements DNS dans les caches avant de devoir être actualisés.

1.4 Protocoles et outils

Les protocoles et outils DNS sont essentiels pour la résolution des noms de domaine en adresses IP et pour la gestion des services Internet. Voici quelques-uns des principaux protocoles et outils associés au DNS.

1.4.1 Les protocoles

DNS fonctionne sur deux protocoles :

- **TCP (Protocole de Contrôle de Transmission - Transmission Control Protocol) :** est un des principaux protocoles de la couche transport du modèle TCP/IP, et un protocole de communication réseau orienté connexion. Cela signifie que TCP établit une connexion entre les deux appareils communicants avant de pouvoir transmettre des données. DNS utilise TCP pour le transfert de zone.
  - **Fonctionnement :** Le protocole TCP divise les données en paquets qu'il envoie à la machine de réception. Cette dernière les traite pour les rassembler et reconstituer les données d'origine afin de les utiliser.
- **UDP (Protocole de Datagramme Utilisateur- User Datagram Protocol) :** est un protocole de communication conçu pour les transmissions Internet nécessitant une grande réactivité, comme le streaming vidéo ou les requêtes DNS. Contrairement à d'autres protocoles, il ne nécessite pas l'établissement préalable d'une connexion, ce qui accélère les échanges de données.
  - **Fonctionnement :** Lorsqu'une requête DNS est envoyée à l'aide du protocole UDP, elle est envoyée au serveur DNS sur le port 53, et le serveur répond sur le même port. Si la réponse est trop grande pour tenir dans un seul datagramme UDP, le serveur peut fragmenter la réponse en plusieurs datagrammes. Si le client ne reçoit pas tous les fragments, il peut renvoyer la requête en utilisant TCP, qui fournit une transmission de données fiable [2].

  UDP est approprié pour les échanges de données de taille réduite, tandis que TCP est préférable pour les transmissions impliquant des informations dépassant les 512 octets.

1.4.1.1 Protocoles de sécurité

- **DNS sur HTTPS - DNS over HTTPS (DoH) :** DoH est un mécanisme important pour renforcer la confidentialité et la sécurité des requêtes DNS ,qui fonctionne via le port 443, offrant aux utilisateurs une protection supplémentaire contre la surveillance, la censure et les attaques malveillantes [3].
  - Fonctionnement : Lorsqu'un utilisateur envoie une requête DNS, elle est encapsulée dans un paquet HTTPS et envoyée à un serveur DNS compatible avec DoH via une connexion HTTPS. Le serveur DNS traite ensuite la requête et renvoie la réponse dans un paquet HTTPS, que l'utilisateur décrypte pour obtenir la réponse DNS.

  HTTPS, ou Hypertext Transfer Protocol Secure, est une extension du protocole HTTP utilisé pour sécuriser la communication entre un navigateur web et un serveur.

- **DNS sur TLS - DNS over TLS (DoT) :** DoT est un protocole de sécurité réseau qui consiste à chiffrer et encapsuler les requêtes et réponses du système de noms de domaine (DNS) via le protocole Transport Layer Security (TLS). Il vise à garantir la confidentialité et l'intégrité des requêtes DNS et des réponses en chiffrant les données échangées et en assurant l'authentification de l'identité du serveur DNS [3].
  - Fonctionnement : Il est conçu pour fonctionner sur le port 853, distinct du port DNS traditionnel (53), ce qui permet de le déployer facilement en parallèle avec une infrastructure DNS existante, tout en offrant une couche de sécurité supplémentaire.

  TLS, ou Transport Layer Security, est un protocole de sécurité informatique qui assure la confidentialité et l'intégrité des données échangées sur un réseau informatique, tel qu'Internet.

- **Extensions de sécurité du système de noms de domaine - Domain Name System Security Extensions (DNSSEC) :** DNSSEC est un ensemble de protocoles qui ajoutent une couche de sécurité aux processus de recherche et d'échange des enregistrements DNS. DNSSEC vise à renforcer la confiance des utilisateurs sur Internet, les protéger et les empêcher d'atterrir sur des sites Web frauduleux [4].

- Fonctionnement : Il utilise des signatures cryptographiques pour créer une "chaîne de confiance" afin de valider que les informations que les utilisateurs reçoivent proviennent des serveurs DNS corrects. Si DNSSEC ne peut pas valider les informations, il les rejette [5].

1.4.2 Les outils

Les outils d'analyse du trafic DNS sont des instruments permettant de recueillir et d'examiner des données sur les visiteurs d'un site web, ainsi que leurs interactions avec les pages. Ils analysent des indicateurs clés de performance (Key Performance Indicators ou KPI) révélant les secteurs nécessitant des améliorations sur notre site web. De plus, ils peuvent servir à repérer les interfaces, les applications, les utilisateurs, les ports et les protocoles qui utilisent la bande passante au maximum [3]. Ces KPI comprennent :

⋆ **Volume de trafic :** Le nombre de visiteurs, qu'ils soient nouveaux ou récurrents, sur notre site web, ainsi que leurs caractéristiques démographiques.

⋆ **Sources de trafic :** Les différents moyens par lesquels les utilisateurs accèdent à notre site (par exemple, recherche organique, publicités payantes, e-mails, messages sur les réseaux sociaux).

⋆ **Pages vues :** Le nombre de fois que des pages spécifiques du site sont visitées.

⋆ **Durée de la session :** La quantité totale de temps qu'un visiteur passe sur notre site web lors d'une seule visite.

⋆ **Taux de rebond :** Le pourcentage de visiteurs qui arrivent sur l'une de nos pages web et la quittent sans interagir davantage avec le site.

⋆ **Taux de conversion :** Le pourcentage de visiteurs qui effectuent une action souhaitée sur notre site web, comme remplir un formulaire ou acheter un produit. Cette action est souvent appelée conversion.

⋆ **Visites uniques :** On comptabilise uniquement la première visite d'un individu, ce qui permet de distinguer les nouveaux prospects des utilisateurs réguliers.

⋆ **Pages de sortie :** On identifie les pages à partir desquelles chaque visiteur quitte notre site web.

1.5 Vulnérabilité

Les serveurs DNS (Domain Name System) sont essentiels au bon fonctionnement d'Internet, car ils permettent de traduire les noms de domaine en adresses IP, ce qui facilite la navigation et l'accès aux ressources en ligne. Cependant, étant donné leur rôle central, ils sont également une cible privilégiée pour de nombreuses attaques qui exploitent leurs vulnérabilités pour des raisons malveillantes.

1.6 Les attaques DNS

Nous en trouvons plusieurs :

1.6.1 Tunnel DNS

Le tunnel DNS est un moyen secret souvent utilisé par les cybercriminels ou les logiciels malveillants pour éviter d'être détectés tout en compromettant des appareils, volant des données sensibles ou réalisant des actions frauduleuses à l'insu des victimes ou de tiers. Comme le trafic DNS est habituel, la plupart des pare-feu le laissent passer sans restrictions, et les systèmes de détection d'intrusion ne signalent généralement pas ce type de trafic. Pour échapper à la détection, les données frauduleuses sont encapsulées dans des requêtes DNS [4].

1.6.1.1 Fonctionnement :

Les attaques par tunneling DNS utilisent le protocole DNS pour acheminer des logiciels malveillants et d'autres types de données au sein d'une architecture client-serveur (voir Figure 1.2) :

1. L'attaquant enregistre un domaine malveillant, comme "badsite.com", et configure le serveur de noms pour pointer vers son propre serveur.
2. L'attaquant introduit des logiciels malveillants dans un ordinateur, souvent derrière un pare-feu d'entreprise, permettant d'envoyer des requêtes DNS.
3. Les requêtes DNS étant souvent autorisées par les pare-feux, l'ordinateur infecté les envoie à un résolveur. Ce résolveur transmet ensuite la requête au serveur de commande et de contrôle de l'attaquant.
4. Avec la redirection des requêtes vers le serveur de l'attaquant, un canal de communication est établi entre l'ordinateur infecté et le serveur malveillant.

Cela permet de dissimuler des actions malveillantes sous des requêtes DNS ordinaires.

5. La structure du tunnel DNS rend difficile le suivi et la détection des activités malveillantes, car elle utilise un protocole largement accepté et rarement soumis à des contrôles de sécurité stricts.

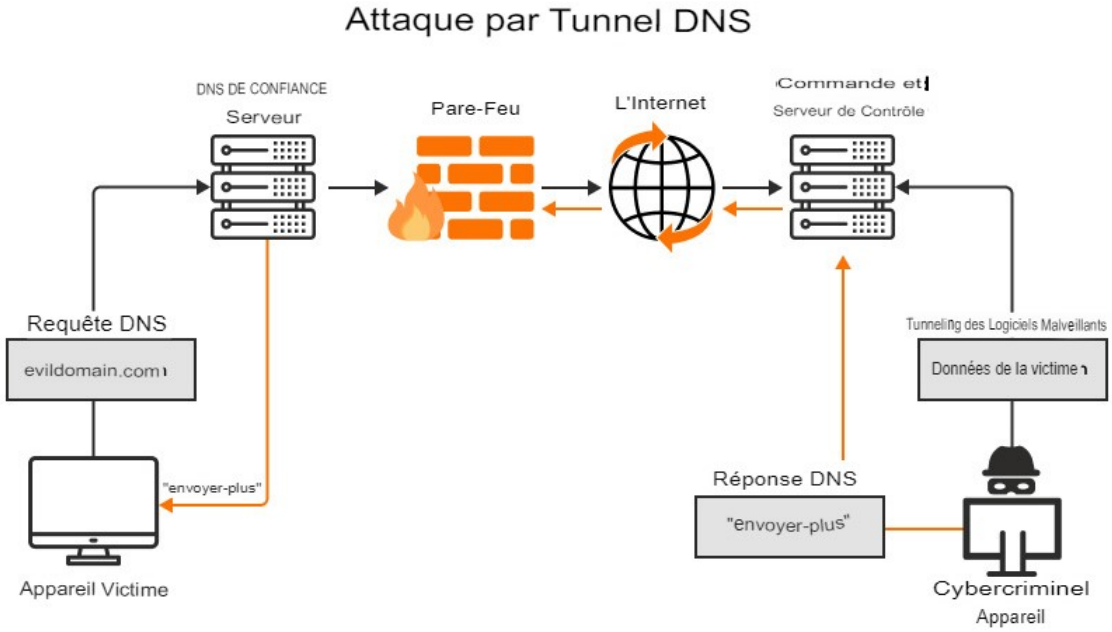


Figure 1.2 – Schématisation d'une attaque par Tunnel [6].

1.6.1.2 Détection des attaques par tunnel DNS

Il existe deux indicateurs majeurs qui suggèrent que vous pourriez être victime d'une attaque par tunneling DNS :

- **Requêtes de domaine inhabituelles :** Un logiciel malveillant utilise le tunneling DNS pour crypter des données dans les noms de domaine demandés. Une organisation pourrait différencier le trafic légitime des tentatives de tunneling DNS en analysant attentivement les noms de domaine sollicités dans les requêtes DNS. Elle doit identifier quels noms de domaine sont normaux et lesquels semblent anormaux ou suspects.
- **Volume de trafic DNS élevé :** Un nom de domaine dans une requête DNS a une limite de taille de 253 caractères. Pour exfiltrer des données ou établir un protocole de commande et de contrôle hautement interactif, un attaquant doit généralement envoyer de nombreuses requêtes DNS malveillantes. Une

augmentation significative du trafic DNS peut donc indiquer l'existence d'un tunneling DNS.

Les techniques à utiliser pour observer ces indicateurs et détecter une attaque par tunneling DNS peuvent être divisées en deux catégories :

- **Analyse de la charge utile :** Nous devons surveiller les éléments suivants : la taille des requêtes et des réponses DNS, les noms d'hôtes inhabituels, les types d'enregistrements DNS atypiques, et les requêtes qui contournent les politiques internes du serveur DNS. Les noms d'hôtes normaux utilisent des mots simples, sans encodage complexe, et contiennent une proportion raisonnable de chiffres.
- **Analyse du trafic :** Nous Surveillons le volume de trafic DNS par adresse IP et par domaine, le nombre de noms d'hôtes associés à chaque domaine, ainsi que l'emplacement géographique des serveurs DNS. Un volume important de trafic DNS vers des régions géographiques avec lesquelles nous n'avons pas de liens directs devrait éveiller des soupçons. Nous pouvons également examiner l'historique d'un domaine, comme la date de création d'un enregistrement A (ou AAAA) ou d'un enregistrement NS pour un nom de domaine donné [5].

1.6.2 Les attaques par inondation UDP

Les attaques par inondation UDP se caractérisent par l'envoi massif de paquets User Datagram Protocol (UDP) vers un serveur cible ou un système réseau. L'objectif principal est de saturer la capacité de traitement du serveur, l'empêchant de répondre aux demandes légitimes des utilisateurs [6].

1.6.2.1 Fonctionnement :

Le principe de fonctionnement des attaques par inondation UDP est illustré par la Figure 1.3.

1. L'attaquant produit un flux important de paquets UDP, souvent à partir de plusieurs sources, afin d'augmenter l'effet de l'attaque. Ces paquets, avec des adresses IP contrefaites, sont dirigés vers le serveur ou la cible choisie, ce qui rend l'attaque plus difficile à tracer.
2. Le serveur cible traite chaque paquet UDP pour déterminer son origine, sa demande ou son port de destination. S'il n'y a pas de service associé à ce port,

le serveur répond par un paquet ICMP indiquant que la destination est injoignable. Ce processus consomme des ressources critiques du serveur, comme la puissance de calcul ou la mémoire, ce qui peut entraîner une dégradation des performances ou un arrêt complet du serveur.

3. Lorsque le serveur ou le réseau ne peut plus gérer le volume de trafic entrant, cela provoque un déni de service, rendant les services inaccessibles pour les utilisateurs légitimes.

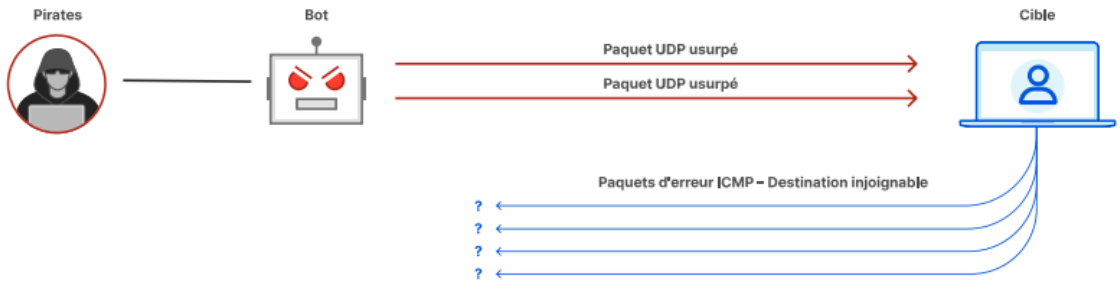


Figure 1.3 – Schématisation d'une attaque par inondation UDP [6]

1.6.3 Les attaques par amplification DNS

Ce Sont des attaques DDoS (déni de service distribué) volumétriques utilisant la réflexion. L'attaquant exploite des résolveurs DNS ouverts pour envoyer une grande quantité de trafic amplifié à un serveur ou un réseau cible, ce qui entraîne une surcharge qui le rend inaccessible, ainsi que son infrastructure associée [6].

Attaques DDos : Les attaques DDoS sont des tentatives malveillantes de rendre un serveur, un service ou une infrastructure indisponible pour les utilisateurs légitimes. Elles visent à épuiser les ressources du système cible en l'inondant de trafic malveillant provenant de multiples sources.

1.6.3.1 Fonctionnement :

Le principe de fonctionnement des attaques par amplification DNS est illustré par la Figure 1.4.

1. Le pirate utilise un point de terminaison compromis pour envoyer des paquets UDP à un récurseur DNS à l'aide d'adresses IP usurpées. L'adresse usurpée contenue dans les paquets renvoie vers l'adresse IP réelle de la victime.

2. Chacun des paquets UDP envoie une requête à un résolveur DNS, en transmettant souvent un argument du type « ANY », afin de recevoir la réponse la plus volumineuse possible.
3. Après avoir reçu ces requêtes, le résolveur DNS (qui essaie d'effectuer son travail en répondant) envoie une réponse volumineuse à l'adresse IP usurpée.
4. L'adresse IP de la cible reçoit la réponse et l'infrastructure de réseau environnante se retrouve submergée sous un flot de trafic, provoquant ainsi un déni de service.

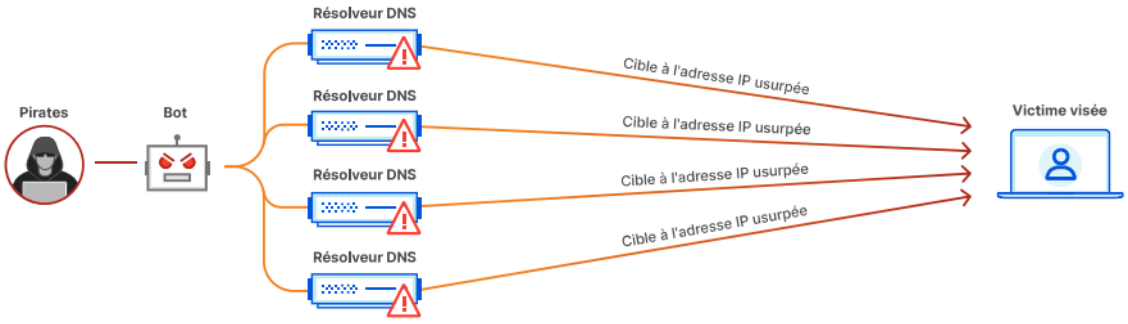


Figure 1.4 – Schématisation d'une attaque par Amplification [6]

1.6.4 Empoisonnement du cache DNS (Cache Poisoning/DNS Spoofing)

L'empoisonnement du cache DNS est une attaque où des informations fausses ou trompeuses sont injectées dans le cache DNS d'un serveur résolveur. Cela entraîne des réponses incorrectes aux requêtes DNS des utilisateurs, les conduisant à des sites web erronés ou malveillants.

1.6.4.1 Fonctionnement :

La figure 1.5 explique le principe de fonctionnement de l'empoisonnement du cache DNS.

1. L'attaquant cherche des informations sur les noms de domaines et les adresses IP stockés dans le cache DNS du résolveur. Cela peut se faire en utilisant des outils de requête DNS ou en exploitant des méthodes de collecte d'informations sur le cache.
2. L'attaquant génère une réponse DNS falsifiée qui contient des données incorrectes, comme une adresse IP différente pour un nom de domaine légitime. Cette réponse falsifiée doit inclure des éléments qui la rendent crédible, comme un identifiant de transaction et une valeur TTL réaliste.
3. L'attaquant envoie un grand nombre de requêtes au résolveur DNS pour un nom de domaine cible, accompagnées de réponses DNS falsifiées qui correspondent à ces requêtes. Le but est de convaincre le résolveur d'enregistrer la réponse erronée dans son cache.
4. Si le résolveur DNS accepte la réponse falsifiée, il la stocke dans son cache selon le TTL spécifié. Cela signifie que toutes les futures requêtes pour ce nom de domaine seront répondues avec la fausse adresse IP [7].

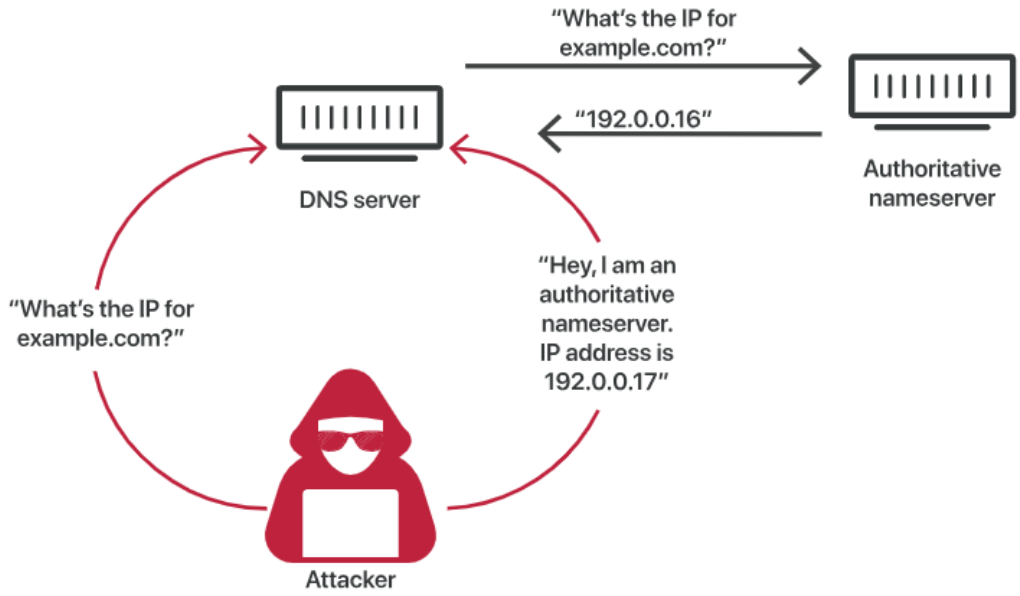


Figure 1.5 – Schématisation d'une attaque par Empoisonnement [7]

Ces vulnérabilités soulignent l'importance de sécuriser le DNS pour protéger l'intégrité et la disponibilité des services en ligne. Les organisations doivent mettre

en place des mesures de sécurité robustes, appliquer rapidement les correctifs et suivre les meilleures pratiques en matière de configuration du DNS.

1.7 Les attaques hybrides

Les attaques hybrides DNS exploitent une combinaison de techniques d'attaques volumétriques et ciblées, comme le DNS Spoofing, l'amplification DNS, et les attaques DDoS.

Ces attaques hybrides sont devenues essentielles pour les cybercriminels car elles leur permettent de contourner plus efficacement les défenses classiques. En effet, en combinant différentes méthodes d'attaque, elles deviennent plus complexes à détecter et à contrer.

1.7.1 DNS Tunneling + Data Exfiltration

Les attaquants utilisent le DNS Tunneling pour exfiltrer des données de manière furtive. Cela peut être combiné avec d'autres attaques pour voler des informations sensibles ou compromettre des systèmes.

1.7.2 DNS Amplification + DDoS Attack

Les attaquants utilisent des techniques de DNS Amplification pour lancer des attaques DDoS massives. Cela peut perturber les services Internet en général, mais lorsqu'il est combiné avec des attaques ciblées, il peut également être utilisé pour désactiver des serveurs DNS ou des sites Web spécifiques.

1.7.3 DDoS + cache poisoning

L'attaquant lance une attaque DDoS pour saturer les serveurs DNS, rendant difficile ou impossible l'accès aux services en ligne. Pendant ce temps, ils exploitent la confusion créée par l'attaque pour injecter des fausses informations dans les caches DNS, redirigeant ainsi le trafic vers des sites malveillants.

1.8 Protection contre les attaques

La sécurité du DNS concerne la protection de l'infrastructure DNS contre les cyberattaques, assurant ainsi son bon fonctionnement et sa fiabilité [8]. Cela implique :

1. Mettre en place des serveurs DNS redondants, utiliser des technologies de sécurité comme DNSSEC, et mettre en œuvre une journalisation rigoureuse des activités DNS.
2. Authentification cryptographique des données DNS, qui utilise une clé symétrique pour fournir un accès aux données DNS.
3. Zones de politique de réponse, qui utilisent des règles concernant ce que les requêtes DNS peuvent faire.
4. L'authentification et l'intégrité des données, qui se concentrent sur l'utilisation de signatures générées par cryptographie, qui sont limitées aux enregistrements de ressources de notre DNS. Cela introduit une protection basée sur la signature cryptographique à toutes les requêtes DNS, car une requête ne peut tout simplement pas être effectuée sans interface avec un enregistrement de ressource DNS.
5. Déni d'existence (DoE) authentifié. Cela permet au résolveur DNS de déterminer si un domaine existe réellement.

En outre, le deep learning peut-être utilisé dans la sécurité pour détecter, prévenir et répondre aux menaces ciblant le système de noms de domaine.

1.9 Conclusion

Le système de noms de domaine (DNS) est fondamental pour le fonctionnement d'Internet, car il convertit les noms de domaine en adresses IP, facilitant la navigation et la communication en ligne. Cependant, cette position centrale en fait également une cible pour de nombreuses attaques. Ces dernières, peuvent causer des perturbations importantes, affectant la confidentialité, l'intégrité, et la disponibilité des services Internet.

La sécurisation du DNS est un défi permanent qui nécessite une approche globale et proactive. Les entreprises doivent allouer des ressources à des technologies et à des pratiques de sécurité efficaces, sensibiliser leurs employés aux risques associés au DNS, et se tenir au courant des nouvelles menaces.

Dans le chapitre suivant, nous allons aborder le concept de deep learning.

CHAPITRE 2

DEEP LEARNING

2.1 Introduction

L'apprentissage profond (Deep Learning) est une branche avancée de l'intelligence artificielle et est devenu une révolution dans le domaine de l'apprentissage automatique. Cette discipline vise à doter les systèmes informatiques de la capacité extraordinaire de comprendre et d'interagir intelligemment avec le monde qui les entoure. Le fondement de l'apprentissage profond repose sur les réseaux de neurones profonds, des structures complexes qui imitent les fonctions du cerveau humain pour extraire des représentations hiérarchiques à partir des données. L'objectif ultime est de permettre aux machines d'apprendre des concepts de haut niveau à partir d'énormes ensembles de données et de les appliquer de manière significative.

Pour comprendre l'apprentissage profond, il faut reconnaître son lien avec l'apprentissage automatique. En fait, le deep learning est une sous-catégorie du machine learning caractérisée par l'utilisation de réseaux de neurones profonds (souvent composés de plusieurs couches) pour effectuer des tâches complexes. Avant d'aborder les mécanismes complexes de l'apprentissage profond, il est nécessaire d'avoir une compréhension de base des concepts de base de l'apprentissage automatique, qui fournit le cadre conceptuel nécessaire à cette discipline en développement.

2.2 Intelligence Artificielle

L'intelligence artificielle (IA) est un processus d'imitation de l'intelligence humaine qui repose sur la création et l'application d'algorithmes exécutés dans un environnement informatique dynamique. Son but est de permettre à des ordinateurs de penser et d'agir comme des êtres humains [9].

2.3 Machine Learning

Machine Learning est une branche de l'intelligence artificielle (IA) qui permet aux ordinateurs de s'auto-apprendre à partir de données d'entraînement et de s'améliorer au fil du temps, sans être explicitement programmés. Les algorithmes d'apprentissage automatique sont capables de détecter des modèles dans les données et d'en tirer des enseignements afin de faire leurs propres prédictions [10].

En bref, les algorithmes et les modèles d'apprentissage automatique apprennent par l'expérience. Il existe plusieurs types d'apprentissage automatique, parmi lesquels on trouve :

2.3.1 L'apprentissage supervisé

En apprentissage supervisé, l'algorithme est guidé avec des connaissances préalables de ce que devraient être les valeurs de sortie du modèle (voir Figure 2.1). Par conséquent, le modèle ajuste ses paramètres de façon à diminuer l'écart entre les résultats obtenus et les résultats attendus. La marge d'erreur se réduit ainsi au fil des entraînements du modèle, afin d'être capable de l'appliquer à de nouveaux cas [11].

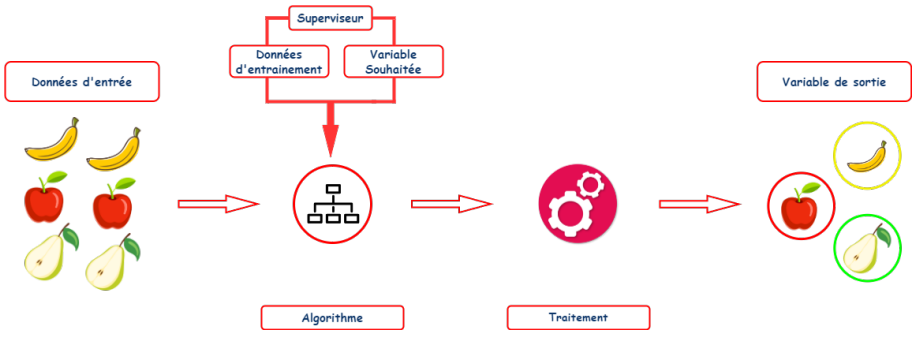


Figure 2.1 – Schéma apprentissage supervisé

2.3.2 L'apprentissage non supervisé

Dans ce type (voir Figure 2.2), on va donner à l'algorithme des données, éventuellement non structurées. Et on le laisse trouver une sorte de structure dans nos données [11].

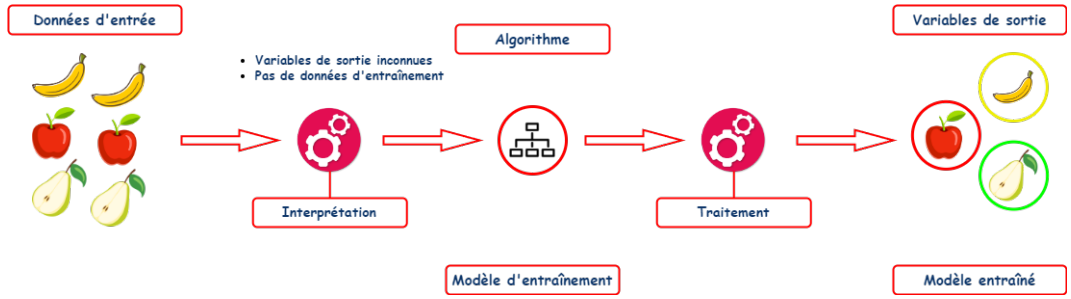


Figure 2.2 – Schéma apprentissage non supervisé

2.3.3 L'apprentissage par renforcement

Dans ce cadre, un agent apprend à prendre des décisions en agissant dans son environnement (voir Figure 2.3). Il reçoit des récompenses ou des pénalités en fonction de ses actions et ajuste son comportement pour maximiser les récompenses au fil du temps. L'apprentissage par renforcement est largement appliqué dans des domaines comme les jeux, la robotique et l'automatisation industrielle.

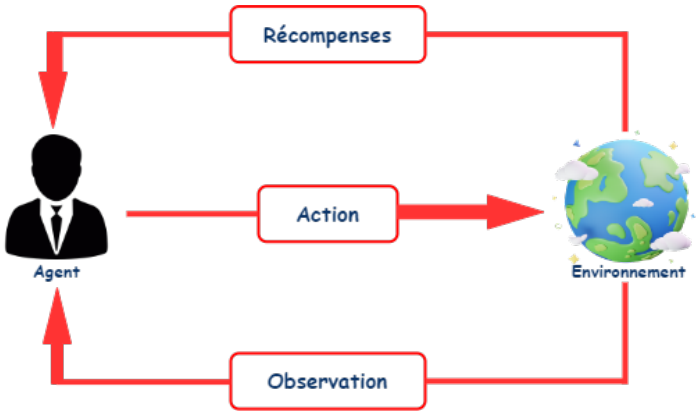


Figure 2.3 – Schéma apprentissage par renforcement

2.4 Deep Learning

L'apprentissage profond est une branche avancée de l'apprentissage automatique, il se repose sur des « réseaux de neurones artificiels », inspirés du cerveau humain, est composés de milliers de couches de « neurones », dont chacune effectue de petites et simples opérations. Les résultats de la première couche de « neurones » servent d'entrée aux calculs de la deuxième couche et ainsi de suite [12].

Les progrès du deep learning ont été rendus possibles grâce notamment à l'augmentation de la puissance des ordinateurs et au développement de grandes bases de données (« big data »).

2.4.1 Les avantages du Deep Learning

Les raisons pour lesquelles le deep learning est devenu la norme de l'industrie [13] :

1. **Gestion des données non structurées :** les modèles formés sur des données structurées peuvent facilement apprendre des données non structurées, ce qui réduit le temps et les ressources nécessaires à la standardisation des ensembles de données.
2. **Gestion de données volumineuses :** grâce à l'introduction d'unités de traitement graphique (GPU) , les modèles d'apprentissage en profondeur peuvent traiter de grandes quantités de données à une vitesse fulgurante.
3. **Haute précision :** les modèles d'apprentissage profond fournissent les résultats les plus précis en matière de vision par ordinateur, de traitement du langage naturel (NLP) et de traitement audio.
4. **Reconnaissance de modèles :** la plupart des modèles nécessitent l'intervention d'un ingénieur en apprentissage automatique, mais les modèles d'apprentissage profond peuvent détecter automatiquement toutes sortes de modèles.

2.4.2 Les réseaux de neurones

Avant d'explorer en détail le fonctionnement spécifique des réseaux de neurones, nous avons envisagé une comparaison avec les neurones biologiques.

Les réseaux neuronaux artificiels imitent le processus d'apprentissage des organismes biologiques. Dans le système nerveux humain, on trouve des cellules nommées neurones.

Le neurone biologique se compose de trois parties principales : le corps cellulaire, les dendrites et l'axone. Le corps cellulaire traite les informations reçues par les dendrites, qui agissent comme des fils conducteurs pour les signaux provenant de l'extérieur. Ensuite, l'axone transmet le signal de sortie du corps cellulaire vers d'autres neurones, quant aux synapses jouent le rôle de connexions et de poids entre les neurones, facilitant ainsi la communication entre eux. Cette organisation permet au neurone de jouer un rôle crucial dans la transmission et le traitement des informations dans le système nerveux biologique [30] (voir Figure 2.4).

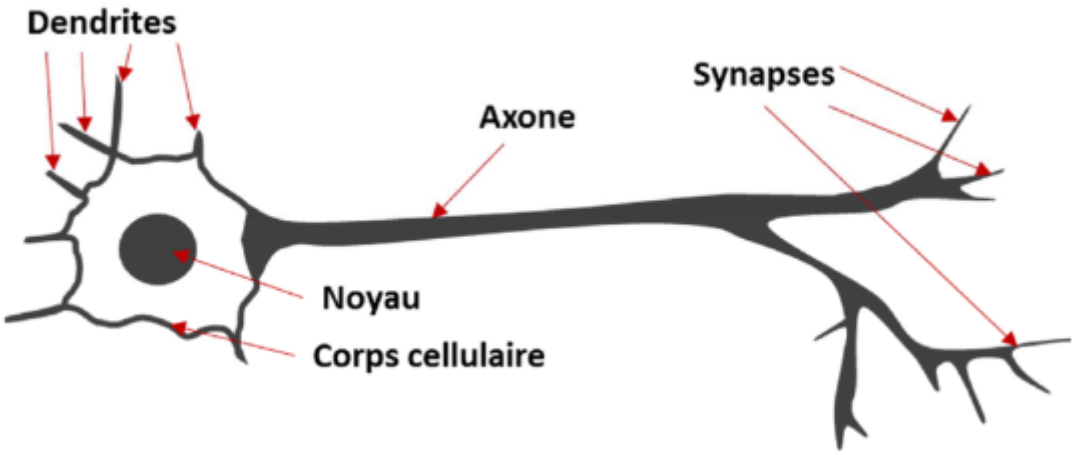


Figure 2.4 – Schématisation d'un neurone biologique[14]

Faisons une analyse simple des principales étapes en les comparant (voir Figure 2.5) :

* **synapses/dendrites :** pondération de chaque élément en entrée x_i, poids synaptiques (w_i).
* **Corps cellulaires :** application d'une fonction d'activation f à la somme des entrées pondérées.
* **Axone :** sortie de notre modèle.

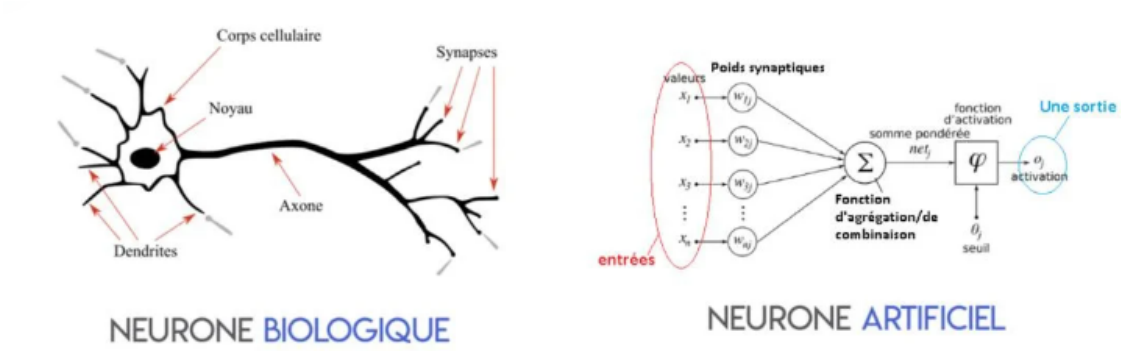


Figure 2.5 – Fonctionnement du neurone artificiel[14]

2.4.2.1 Réseaux Neurones Artificiels (ANNs)

Les réseaux neurones artificiels(ANNs) sont composés de neurones interconnectés, les poids, les fonctions d'activation et les couches, disposés selon une architecture définie, qui se transmettent des informations jusqu'à ce qu'un résultat soit obtenu. Les neurones traitent l'information, les poids représentent la force des connexions entre les neurones, les fonctions d'activation décident de l'activation des neurones en fonction de leurs entrées, et les couches effectuent des calculs intermédiaires. En comprenant ces éléments et leur interaction, nous pouvons mieux comprendre comment les ANNs sont utilisés pour résoudre des problèmes complexes.

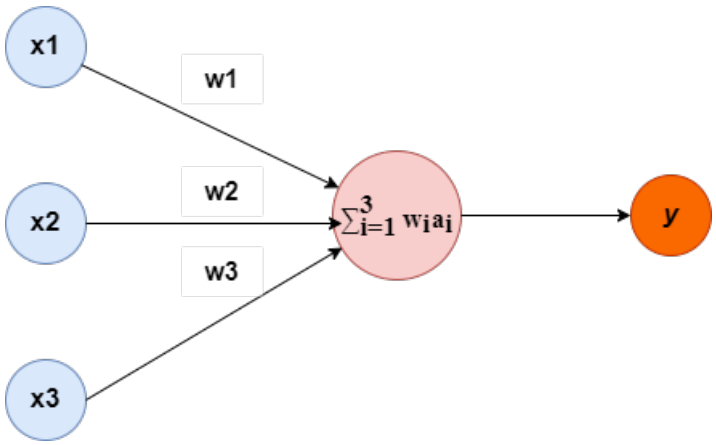


Figure 2.6 – Représentation d'un réseau neurone artificiel- Combinaison linéaire des entrées

Les neurones ici présent vont réaliser une somme pondérée des données d'entrée à disposition. A chaque donnée d'entrée, le neurone va associer un poids (les w1, w2, w3). La somme pondérée de ces poids avec les données d'entrée est donnée par :

$$\sum_{i=1}^{3} w_i a_i = w_1 a_1 + w_2 a_2 + w_3 a_3 \tag{2.1}$$

qui va être utilisée pour estimer un rendement [15] (voir Figure 2.6).

2.4.2.2 Réseaux de neurones profonds (RNP ou DNN)

Les réseaux de neurones profonds sont devenus très populaires en raison de leur capacité exceptionnelle à réussir dans une variété de projets d'apprentissage profond, grâce à leur efficacité remarquable.

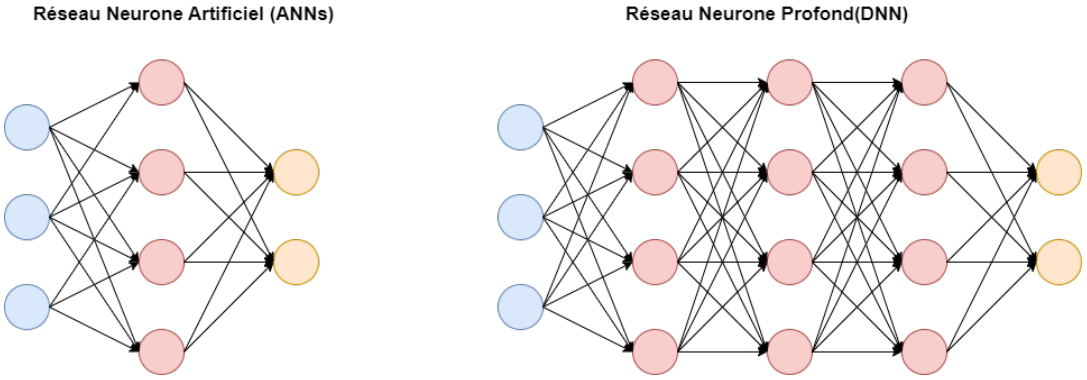


Figure 2.7 – Différence entre ANN et DNN

Les réseaux neuronaux profonds se démarquent des réseaux neuronaux classiques principalement par leur profondeur, qui fait référence au nombre de couches qu'ils intègrent. Contrairement aux réseaux traditionnels, qui ont généralement moins de trois couches, les réseaux neuronaux profonds sont caractérisés par de nombreuses couches cachées beaucoup plus nombreuses. Cette structure complexe leur permet de gérer et de stocker un volume d'informations bien plus important. La profondeur accrue des réseaux neuronaux profonds leur confère la capacité d'analyser de manière autonome des modèles et des caractéristiques complexes, leur offrant ainsi des capacités de traitement de données bien plus avancées que celles des réseaux neuronaux traditionnels (voir Figure 2.7).

2.4.2.3 Architecture

Un réseau de neurones comprend généralement trois types de couches (voir Figure 2.8) :

* **Une couche d'entrée :** Cette couche charge les données sans effectuer de calculs.
* **Des couches cachées :** Ces couches sont interconnectées, et opèrent des calculs mathématiques sur les données pour en extraire des fonctionnalités.
* **Une couche de sortie :** La couche de sortie génère le résultat final en utilisant les informations des couches cachées précédentes.

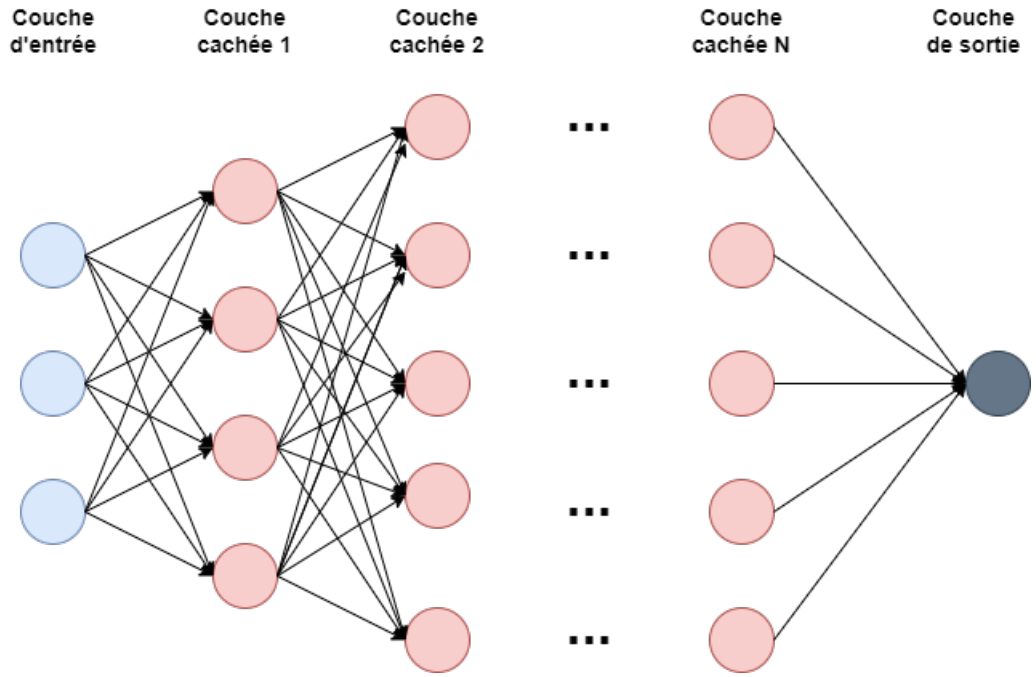


Figure 2.8 – Architecture Réseau Neurone Profond

2.4.2.4 Réseau de neurones convolutifs (CNN)

Le réseau de neurones convolutifs (CNN) est la version étendue des réseaux de neurones artificiels (ANN), spécifiquement conçue pour extraire des caractéristiques à partir de données organisées sous forme de grille, telles que des images ou des vidéos. Les CNN sont particulièrement adaptés à l'analyse de données visuelles où la structure spatiale des données est cruciale. [16]

2.4.2.5 Architecture

Le réseau neuronal convolutif se compose de plusieurs couches telles que la couche d'entrée, la couche convolutive, la couche de pooling , les couches entièrement connectées et la couche de sortie (voir Figure 2.9).

* **Couche d'entrée (input) :** c'est la couche dans laquelle nous donnons une entrée à notre modèle. Dans CNN, généralement, l'entrée sera une image ou une séquence d'images.
* **Couche de pooling :** vise à réduire la dimensionnalité des données d'entrée tout en conservant les informations critiques, améliorant ainsi l'efficacité globale du réseau. Ceci est généralement réalisé grâce au sous-échantillonnage : en diminuant le nombre de points de données dans l'entrée.
* **Couche entièrement connectée(fully connected) :**joue un rôle essentiel dans les étapes finales d'un CNN, où elle est chargée de classer les images en fonction des caractéristiques extraites dans les couches précédentes. Le terme entièrement connecté signifie que chaque neurone d'une couche est connecté à chaque neurone de la couche suivante.
* **Couche de sortie (output) :** La sortie des couches entièrement connectées est ensuite introduite dans une fonction logistique pour les tâches de classification telles que sigmoïde ou softmax qui convertit la sortie de chaque classe en score de probabilité de chaque classe.

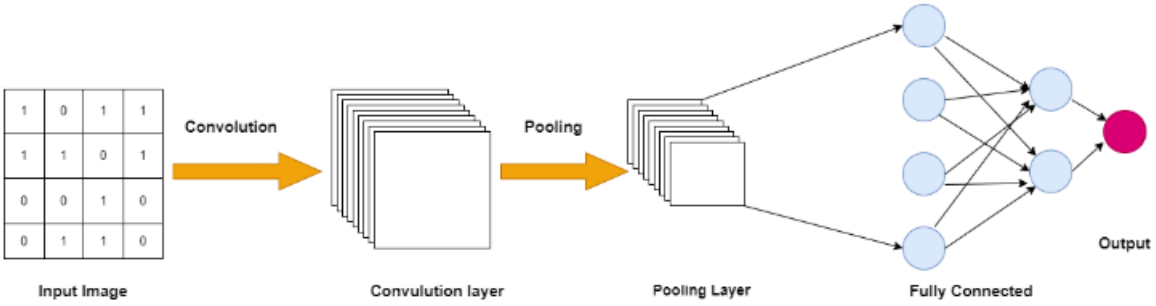


Figure 2.9 – Diagramme CNN

2.5 Dataset (Jeu de Données)

Un jeu de données, ou dataset, regroupe plusieurs données ayant un lien cohérent entre elles. Il se présente sous forme de tableau permettant d'analyser chaque donnée qui le compose. Chaque donnée peut être composée de texte, de chiffres, de coordonnées géographiques ou encore d'éléments multimédia (par exemple une image ou une vidéo).

2.5.1 Imbalanced Dataset (Données déséquilibrées)

Les données déséquilibrées posent de réelles difficultés aux algorithmes de Machine Learning et de Deep Learning.

Un ensemble de données de classification avec des proportions de classe asymétriques est appelé déséquilibré. Les classes qui constituent une grande partie de l'ensemble de données sont appelées classes de majorité. Ceux qui en représentent une plus petite sont des classes de minorité.

2.5.1.1 Solutions pour les données déséquilibrées

Ces solutions consistent à opérer un rééchantillonnage des données utilisées pour l'entraînement des algorithmes de Machine Learning et de Deep Learning. Ceci vise un rééquilibrage des classes, pour faciliter l'apprentissage [17].

1. **Sous-échantillonnage**

   Le sous-échantillonnage permet d'équilibrer l'ensemble des données en réduisant la taille de la ou des classes dominantes pour qu'elles correspondent aux fréquences de la classe la moins répandue (voir figure 2.10). Cette méthode est utilisée lorsque la quantité de données est suffisante.

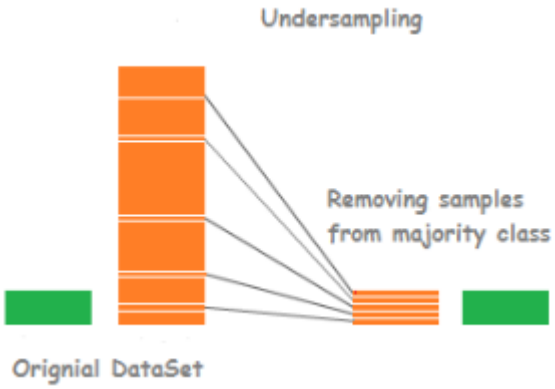


Figure 2.10 – Méthode sous-échantillonnage

Quelques techniques :

* **Sous-échantillonnage aléatoire (Random Undersampling, RUS)**

  Le sous-échantillonnage aléatoire, également connu sous le nom de random undersampling, est une technique simple et couramment utilisée pour équilibrer un ensemble de données déséquilibré en réduisant le nombre d'échantillons de la classe majoritaire jusqu'à ce qu'elle soit équilibrée avec la classe minoritaire.

* **NearMiss :** NearMiss est une technique de sous-échantillonnage qui vise à sélectionner les échantillons de la classe majoritaire en se basant sur leur proximité avec les échantillons de la classe minoritaire.

2. **Sur-échantillonnage**

   Le sur-échantillonnage, est utilisée lorsque la quantité de données est insuffisante. Il tente d'équilibrer l'ensemble des données en augmentant la taille des échantillons plus rares (voir figure 2.11).

Figure 2.11 – Méthode sur-échantillonnage

Quelques techniques :

* **Sur-échantillonnage aléatoire (Random Oversampling, ROS) :**

  Le sur-échantillonnage aléatoire, également appelé random oversampling, est une technique utilisée pour équilibrer un ensemble de données déséquilibré en augmentant le nombre d'échantillons de la classe minoritaire jusqu'à ce qu'elle soit équilibrée avec la classe majoritaire.

* **Sur-échantillonnage synthétique (SMOTE pour Synthetic Minority Oversampling Technique) :**

  Le sur-échantillonnage synthétique, est une méthode plus avancée, qui produit des observations minoritaires ressemblantes mais distinctes de celles déjà existantes.

* **Edited Nearest Neighbors (ENN) :**

  Cette méthode supprime les échantillons de la classe majoritaire qui ne sont pas correctement classés par leurs plus proches voisins de la même classe.

* **SMOTEENN :**

  SMOTEENN est une combinaison de deux techniques : SMOTE (Synthetic Minority Over-sampling Technique) pour le sur-échantillonnage synthétique de la classe minoritaire, et ENN (Edited Nearest Neighbors) pour le sous-échantillonnage de la classe majoritaire. Cette combinaison vise à améliorer la capacité de généralisation du modèle en réduisant le risque de surajustement souvent associé au simple sur-échantillonnage.

* **ADASYN :**

ADASYN est une méthode de suréchantillonnage qui génère des échantillons synthétiques pour les classes minoritaires, ce qui permet d'équilibrer l'ensemble de données et d'améliorer la précision de la classification.

2.5.2 Techniques de prépartion de données

2.5.2.1 Nettoyage des données :

Le nettoyage des données consiste à corriger ou à supprimer les données incorrectes, corrompues, mal formatées, dupliquées ou incomplètes d'un ensemble de données. Lorsque l'on combine plusieurs sources de données, il existe de nombreuses possibilités de duplication ou d'erreur d'étiquetage des données.

Avantages et bénéfices du nettoyage des données :

Le fait d'avoir des données correctes augmentera la productivité globale et permettra d'obtenir des informations de la plus haute qualité pour la prise de décision. Les avantages sont les suivants :

- Moins d'erreurs rendent les clients plus heureux et les employés moins frustrés.
- Possibilité de cartographier les différentes fonctions et ce que vos données sont censées faire.
- Surveillance des erreurs et amélioration des rapports pour voir d'où viennent les erreurs, ce qui facilite la correction des données incorrectes ou corrompues pour les applications futures.

2.5.2.2 Selection des features :

La sélection de features est le processus de réduction du nombre de variables d'entrée lors de l'élaboration d'un modèle prédictif. Il est souhaitable de réduire le nombre de variables d'entrée pour réduire le coût de calcul de la modélisation et, dans certains cas, pour améliorer les performances du modèle.

Méthodes de selection :

- **ANOVA F-test :**

  ANOVA est un acronyme pour analyse de la variance et est un test d'hypothèse statistique paramétrique permettant de déterminer si les moyennes de deux ou plusieurs échantillons de données (souvent trois ou plus) proviennent ou non de la même distribution.

  F-statistic, ou F-test, est une classe de tests statistiques qui calculent le rapport entre les valeurs de variance, telles que la variance de deux échantillons différents ou la variance expliquée et inexpliquée par un test statistique, comme l'ANOVA[17].

- **Mutual Information :**

  L'information mutuelle est calculée entre deux variables et mesure la réduction de l'incertitude pour une variable étant donné une valeur connue de l'autre variable. L'information mutuelle est simple lorsqu'on considère la distribution de deux variables discrètes (catégorielles ou ordinales), telles que des données d'entrée catégorielles et des données de sortie catégorielles[18].

2.5.2.3 Réduction de dimensionnalité

- **Principal component analysis (PCA)**

  L'analyse en composantes principales est une méthode statistique polyvalente qui permet de réduire un tableau de données cas par cas à ses caractéristiques essentielles, appelées composantes principales. Les composantes principales sont quelques combinaisons linéaires des variables originales qui expliquent au maximum la variance de toutes les variables[19].

2.5.2.4 Encodage de données

Il existe plusieurs méthodes pour encoder des variables catégorielles :

- **Encodage ordinal (Ordinal Encoding)**

  L'encodage ordinal consiste à attribuer à chaque catégorie un nombre entier unique en fonction de son ordre dans la variable catégorielle. Cette méthode est utilisée lorsque les catégories ont un ordre naturel.

- **Encodage de label (Label Encoding)**

  Dans cette méthode, chaque catégorie est remplacée par un nombre entier unique. Cependant, contrairement à l'encodage ordinal, l'encodage de label n'impose pas d'ordre aux catégories.

- **Encodage de cible (Target Encoding)**

  Cette méthode consiste à remplacer chaque catégorie par la moyenne de la variable cible pour cette catégorie. Cela peut être utile pour les modèles linéaires ou les modèles basés sur les arbres.

- **Encodage à chaud (One-Hot Encoding)**

  Cette méthode consiste à créer une nouvelle colonne pour chaque catégorie unique dans la variable catégorielle. Chaque colonne représente une catégorie unique, et la valeur est soit 1 (si la catégorie correspondante est présente pour une observation) ou 0 (sinon).

2.6 Optimisation des hyperparamètres d'un modéle du deep learning

Lorsque nous entraînons des modèles, chaque jeu de données ainsi que chaque modèle nécessite un ensemble différent d'hyperparamètres constituant une sorte de variable. La seule façon de les déterminer consiste à effectuer plusieurs expériences en sélectionnant un ensemble d'hyperparamètres pour les exécuter dans notre modèle. C'est ce qu'on appelle le réglage des hyperparamètres.

2.6.1 Le réglage des hyperparamètres

Les hyperparamètres contrôlent directement la structure, la fonction et la performance du modèle. Le réglage des hyperparamètres permet d'ajuster les performances du modèle pour obtenir des résultats optimaux. Ce processus est une partie essentielle, et le choix des valeurs appropriées des hyperparamètres est crucial pour le succès.

2.6.1.1 Les hyperparamètres du DNN

Dans le domaine du deep learning, certains hyperparamètres sont particulièrement cruciaux à optimiser pour garantir des performances optimales des modèles.

Parmi les hyperparamètres les plus importants à optimiser, on retrouve généralement :

1. **Le taux d'apprentissage :** Il contrôle la vitesse à laquelle le modèle apprend et peut influencer la convergence et la qualité des résultats.
2. **La taille du lot (batch size) :** Détermine le nombre d'échantillons utilisés pour calculer le gradient lors de l'entraînement, impactant la stabilité de l'apprentissage.
3. **Le nombre de couches :** Il influence la capacité du modèle à capturer des motifs complexes et à généraliser efficacement.
4. **Le nombre d'époch s :** Correspond au nombre de fois où l'ensemble des données est passé à travers le modèle lors de l'entraînement, affectant la convergence et l'ajustement du modèle.
5. **Le nombre de nœuds :** Dans chaque couche, le nombre de nœuds peut impacter la capacité du modèle à apprendre des représentations significatives.

Certains hyperparamètres sont liés à des fonctions caractérisant le réseau

2.6.1.2 Fonction d'activation

La fonction d'activation prend la combinaison des entrées et la transforme en une sortie non linéaire, ce qui permet au réseau de neurones d'apprendre des relations plus complexes entre les données d'entrée et de produire des résultats plus précis[29].

Le tableau 2.1 montre des exemple de quelques fonctions d'activation.

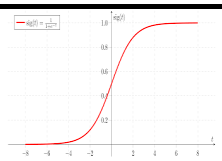
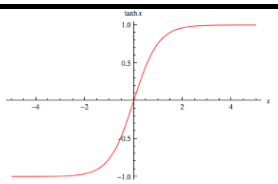
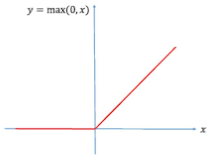
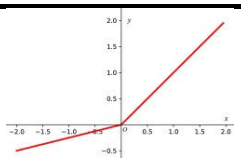
Nom	Graphique	Fonction	Domaine
Sigmoïde		$f(x) = \frac{1}{1+e^{-x}}$	$(-\infty, \infty)$
Tanh		$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$	$(-1, 1)$
ReLU		$f(x) = \begin{cases} 0 & \text{pour } x < 0 \\ x & \text{pour } x \geq 0 \end{cases}$	$(-\infty, \infty)$
LeakyReLU		$f(x) = \begin{cases} 0.01x & \text{pour } x < 0 \\ x & \text{pour } x \geq 0 \end{cases}$	$(-\infty, \infty)$

Table 2.1 – Fonctions d'activation

2.6.1.3 Fonctions d'erreur (loss fonction)

La fonction de perte est une méthode permettant d'évaluer dans quelle mesure notre algorithme d'apprentissage automatique modélise notre ensemble de données présenté.

Elle est cruciale dans le processus d'optimisation du réseau de neurones, car elle guide l'algorithme d'apprentissage pour ajuster les poids et les biais du réseau afin de minimiser cette perte.

$$\varphi = distance(f(xi), y) \tag{2.2}$$

Les fonctions de perte couramment utilisées dans les réseaux de neurones profonds incluent :

- **Perte Entropie Croisée (Cross-Entropy) :** Utilisée souvent dans les tâches de classification, elle mesure la divergence entre la distribution de probabilité prédite par le modèle et la distribution réelle des étiquettes.
- **Perte d'Entropie Croisée Binaire (Binary Cross-Entropy) :** Il s'agit de la fonction de perte la plus couramment utilisée dans les problèmes de classification. La perte d'entropie croisée diminue à mesure que la probabilité prédite converge vers l'étiquette réelle. Elle mesure les performances d'un modèle de classification dont la sortie prédite est une valeur de probabilité comprise entre 0 et 1[20].

$$L = -\frac{1}{m}\sum_{i=1}^{m}(y_i) \times \log(\hat{y}_i) + (1 - y_i) \times log(1 - \hat{y}_i) \tag{2.3}$$

Lorsque le nombre de classes est 2, il s'agit d'une classification binaire.

$$L = -\frac{1}{m}\sum_{i=1}^{m} y_i \times \log(\hat{y}_i) \tag{2.4}$$

- **Perte d'entropie croisée catégorique :** Dans les cas où le nombre de classes est supérieur à deux, nous utilisons l'entropie croisée catégorielle, cela suit un processus très similaire à l'entropie croisée binaire.

$$CELoss = -(\frac{1}{n})\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} \times log(p_{ij}) \tag{2.5}$$

L'entropie croisée binaire est un cas particulier d'entropie croisée catégorielle, où M=2-le nombre de catégories est de 2[21].

2.6.1.4 Optimiseur

Les optimiseurs sont des algorithmes ou des méthodes utilisés pour modifier les attributs de votre réseau neuronal tels que les poids et le taux d'apprentissage afin de réduire les pertes.

Parmi les optimiseurs les plus répandus pour les réseaux neuronaux profonds, on peut citer :

- **Descente de gradient par lots (BGD) :** Cet optimiseur calcule les gradients en utilisant l'ensemble des données de formation pour une mise à jour des paramètres, mais il peut être lent et gourmand en mémoire. La mise à jour des paramètres se fait en utilisant la moyenne des gradients de l'ensemble des données d'entraînement [23] :

$$\theta_t = \theta_{t-1} - \eta(\frac{1}{N})\sum_{i=1}^{N} \nabla J(\theta; x^{(i)}, y^{(i)}) \tag{2.6}$$

où :

* θ représente les paramètres du modèle
* η est le taux d'apprentissage
* N est la taille de l'ensemble de données,
* $J(\theta; x(i), y(i))$ est la fonction de perte pour l'exemple d'entraînement ii avec les entrées $x(i)x(i)$ et les sorties attendues $y(i)y(i)$

- **RMSProp :** RMSProp ajuste les pas d'apprentissage pour chaque composant du modèle en se basant sur l'historique des gradients. Cela aide à entraîner les modèles de manière plus efficace[25].

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{v_t} + \epsilon} \nabla J(\theta) \tag{2.7}$$

où :

* v est une estimation du deuxième moment des gradients
* ϵ est un terme de régularisation pour éviter la division par zéro.

- **Adam (Adaptive Moment Estimation) :** Adam est un optimiseur adaptatif qui calcule des taux d'apprentissage adaptatifs pour chaque paramètre, ce qui permet une convergence plus rapide et une formation efficace[26].

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{v_t} + \epsilon} \hat{m}_t \tag{2.8}$$

où :

* $\hat{m}$ et v sont les estimations corrigées du premier et du deuxième moment des gradients
* ϵ est un terme de régularisation pour éviter la division par zéro.

- **Adagrad :** Adagrad ajuste le taux d'apprentissage pour chaque paramètre sur la base des gradients précédents, ce qui lui permet de traiter efficacement les données éparses[26].

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{G + \epsilon}} \nabla J(\theta) \tag{2.9}$$

où :

* G est l'accumulation des carrés des gradients pour chaque paramètre.

- **Taux D'apprentissage (rate learning) :**

  Le taux d'apprentissage est un hyperparamètre qui joue sur la rapidité de la descente de gradient : un nombre d'itérations plus ou moins important est nécessaire avant que l'algorithme ne converge, c'est-à-dire qu'un apprentissage optimal du réseau soit réalisé.

- **Régularisation :**

  La régularisation est une technique qui apporte de légères modifications à l'algorithme d'apprentissage afin que le modèle se généralise mieux. Cela permet également d'améliorer les performances du modèle sur les données non vues. Comme dans l'apprentissage automatique, la régularisation pénalise les coefficients. Dans l'apprentissage profond, elle pénalise en fait les matrices de poids des nœuds.

  - **L2 et L1 régularisation :** L1 et L2 sont les types de régularisation les plus courants. Ils mettent à jour la fonction de coût générale en ajoutant un autre terme appelé terme de régularisation[22].

$$Cost\ \ function = Loss(say, binary\ \ cross\ \ entropy) + Regularization\ \ term \tag{2.10}$$

2.7 Métriques d'évaluation du modéle DNN

Les métriques d'évaluation d'un modèle de réseau de neurones profonds (DNN) sont essentielles pour mesurer sa performance et son adéquation aux données. Ces métriques permettent de quantifier la qualité du modèle et de comparer différents modèles.

2.7.1 Matrice de confusion (Confusion Matrix)

La matrice de confusion décrit les performances du modèle de classification. En d'autres termes, la matrice de confusion est un moyen de résumer les performances d'un classificateur.La figure 2.12 montre une représentation de base d'une matrice de confusion et représente comment les résultats prédits par le modèle sont comparés aux valeurs réelles [18] :

- **TN (Vrai négatif) :** Il s'agit du nombre de résultats qui étaient initialement négatifs et ont été prédits comme négatifs.

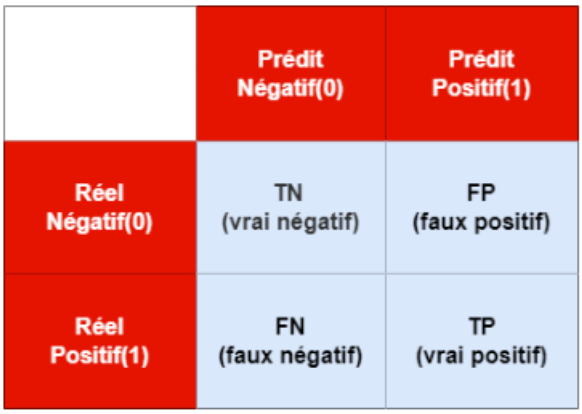


Figure 2.12 – Représentation simple d'une matrice de confusion

- **FP (Faux positif) :** Il s'agit du nombre de résultats qui étaient initialement négatifs mais ont été prédits comme positifs. Cette erreur est également appelée erreur de type 1.
- **FN (Faux négatif) :** Il s'agit du nombre de résultats qui étaient initialement positifs mais ont été prédits comme négatifs. Cette erreur est également appelée erreur de type 2.
- **TP (Vrai positif) :** Il s'agit du nombre de résultats qui étaient initialement positifs et ont été prédits comme positifs.

L'objectif est de maximiser les valeurs dans les cases TN et TP du tableau précédent montré par la Figure 2.12, c'est-à-dire les vrais négatifs et les vrais positifs, et de minimiser les valeurs dans les cases FN et FP, c'est-à-dire les faux négatifs et les faux positifs.

2.7.2 Accuracy :

Accuracy est une mesure de performance couramment utilisée pour évaluer un modèle de machine learning ou deep learning. Elle représente la proportion de prédictions correctes que le modèle a effectuées par rapport au nombre total de prédictions. Par exemple, si un modèle de classification prédit correctement 90% des échantillons de test, son exactitude est de 90%.

La formule pour calculer l'exactitude est la suivante :

$$Accuracy = Exactitude = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

2.7.3 Précision :

C'est la mesure de la proportion d'observations prédites comme positives qui étaient réellement positives. Une précision élevée signifie que le modèle minimise le nombre de fausses alertes.

$$Précision = \frac{TP}{TP + FP} \tag{2.12}$$

2.7.4 Rappel(recall) :

C'est la mesure de la proportion d'observations réellement positives qui ont été correctement prédites par le modèle. Un rappel élevé signifie que le modèle minimise le nombre de cas manqués.

$$Rappel = \frac{TP}{TP + FN} \tag{2.13}$$

2.7.5 Score F1 :

C'est une mesure qui combine la précision et le rappel en une seule valeur, offrant une évaluation globale de la performance du modèle. Un score F1 élevé indique que le modèle a un bon équilibre entre la précision et le rappel.

$$F1 = 2 \times \frac{Précision \times Rappel}{Précision + Rappel} \tag{2.14}$$

2.8 Conclusion

En conclusion, le deep learning représente une avancée majeure dans le domaine de l'intelligence artificielle, permettant aux machines d'apprendre des données complexes et de résoudre une gamme diversifiée de problèmes.

Les datasets jouent un rôle vital dans le deep learning, influençant directement les performances des modèles. Leur évolution constante reflète les changements dans le monde réel et vise à créer des ensembles de données plus diversifiés et équitables, répondant ainsi aux besoins variés des différentes communautés. Le prétraitement des données est crucial pour assurer la qualité des datasets, englobant des processus tels que le nettoyage et la normalisation afin de rendre les données adaptées à l'apprentissage par les modèles de deep learning.

Le chapitre suivant sera consacré à la création d'un modèle deep learning pour la détection des attaques DNS hybrides.

CHAPITRE 3

CRÉATION D'UN MODÈLE DE DEEP LEARNING POUR LA DÉTECTION DES ATTAQUES DNS HYBRIDES

3.1 Introduction

Ce chapitre vise à exposer la mise en œuvre de modèles de réseaux neuronaux profonds (DNN) pour repérer les attaques hybrides, en explorant diverses approches et techniques d'apprentissage en profondeur.

Pour atteindre cet objectif, nous avons exploité le dataset CIC-Bell-DNS-EXT 2021 qui est utilisé pour détecter les attaques d'exfiltration de données via le protocole Domain Name System (DNS). Plus précisément, ce jeu de données, qui consiste en 270,8 Mo de trafic DNS, est utilisé pour identifier les attaques d'exfiltration de données légères et lourdes via DNS.

3.2 Environnement de travail

Le deep learning nécessite d'importantes ressources computationnelles. Même un petit modèle de réseau neuronal contient plus de 100 000 paramètres, tandis que certains en ont des milliards. Par conséquent, l'entraînement de ces réseaux étendus exige une puissance de calcul considérable. les exigences minimales en matière de matériel pour commencer sont les suivantes [33] :

1. **Carte Graphique :** NVIDIA GTX 1060 et au-delà.
2. **CPU :** I5 8ème génération et au-delà.
3. **RAM :** 8GB et au-delà.
4. **HDD/SDD :** 1TB HDD ou bien 256GB SSD.

Même avec toutes les conditions matérielles de base remplies, le traitement de vastes ensembles de données peut s'avérer fastidieux. Heureusement, de nombreuses ressources informatiques en nuage offrent davantage de RAM, de CPU et de GPU, parfois gratuitement ou pour une période limitée. Cette disponibilité accrue peut rendre le traitement des données plus efficace et réduire considérablement les erreurs. C'est pourquoi nous avons opté pour l'utilisation de Kaggle comme alternative matérielle et logicielle dans notre travail.

Kaggle [42] est une plateforme en ligne dédiée aux concours de science des données et d'apprentissage automatique, à la collaboration et à l'apprentissage. Ce qui permet aux utilisateurs de trouver et de publier des jeux de données, d'explorer et de construire des modèles dans un environnement web, de travailler avec d'autres

scientifiques des données et ingénieurs en apprentissage automatique, et de participer à des concours pour résoudre des défis de science des données.

Nous avons choisi Kaggle comme plateforme pour nous assister dans cette étude, car elle présente plusieurs avantages. Tout d'abord, Kaggle offre un accès facile à des ressources de calcul haute performance, telles que des CPU, GPU et TPU, pour des tâches de traitement intensif. De plus, cette plateforme permet l'importation directe de données à partir de sources en ligne, évitant ainsi le téléchargement sur l'ordinateur local, ce qui est particulièrement pratique pour travailler avec de vastes ensembles de données (big data). En outre, Kaggle propose des outils pour la visualisation et l'analyse des données, ainsi que des fonctionnalités pour la collaboration et le partage de projets. En ce qui concerne les logiciels, Kaggle met à disposition une variété étendue d'outils et de packages largement employés dans le domaine de la science des données et de l'apprentissage profond. Parmi ceux-ci, on trouve notamment Jupyter Notebook et le langage de programmation Python, très répandu, ainsi que ses bibliothèques couramment utilisées telles que NumPy, Pandas, Scikit-learn, Tensorflow et Keras.

Jupyter Notebook [43] est une plateforme interactive en ligne proposée par Kaggle, permettant aux utilisateurs de créer et de partager du code, des visualisations et du texte dans un seul document. Cet outil est largement apprécié par les scientifiques des données et les professionnels de l'apprentissage profond, car il prend en charge plusieurs langages de programmation, notamment Python.

Python [44] est un langage de programmation open source multi-plateformes et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données, ou la gestion d'infrastructures.

3.2.1 Bibliothèques utilisées

Le tableau 3.1 liste les bibliothèques couramment utilisées.

Library	Description	Version
Pandas [34]	Pandas est spécifiquement conçue pour la manipulation et l'analyse de données en langage Python. Grâce à Pandas, le langage Python permet enfin de charger, d'aligner, de manipuler ou encore de fusionner des données.	2.2.2
TensorFlow [45]	Il s'agit d'une bibliothèque Open Source de calcul numérique et de Machine Learning compatible avec le langage Python. Elle simplifie le processus d'acquisition de données, d'entraînement des modèles de Machine Learning, de génération de prédictions et de raffinement des résultats futurs.	2.15.0
NumPy [35]	Numerical Python est une bibliothèque Open Source en langage Python. On utilise cet outil pour la programmation scientifique en Python, et notamment pour la programmation en Data Science, pour l'ingénierie, les mathématiques ou la science.	1.26.4
Scikit-learn [46]	Scikit-learn est une bibliothèque Python open source conçue pour faciliter le processus de construction de modèles basés sur des algorithmes de ML et de statistiques intégrés, sans avoir besoin de recourir à des outils de modélisation.	1.2.2
Matplotlib [36]	Matplotlib est une bibliothèque Python open source permettant de créer des visualisations de données. Elle peut générer des graphiques dans divers formats, tels que des graphiques, des histogrammes, des diagrammes de dispersion, et d'autres types de visualisations, afin de répondre à différents besoins de manière efficace.	3.7.5
Seaborn [37]	Seaborn est une bibliothèque permettant de créer des graphiques statistiques en Python. Elle est basée sur Matplotlib, et s'intègre avec les structures Pandas.	0.12.2

Table 3.1 – Liste des bibliothèques Python

3.3 Implémentation

Le système de détection des attaques proposé est illustré à la figure 3.1.

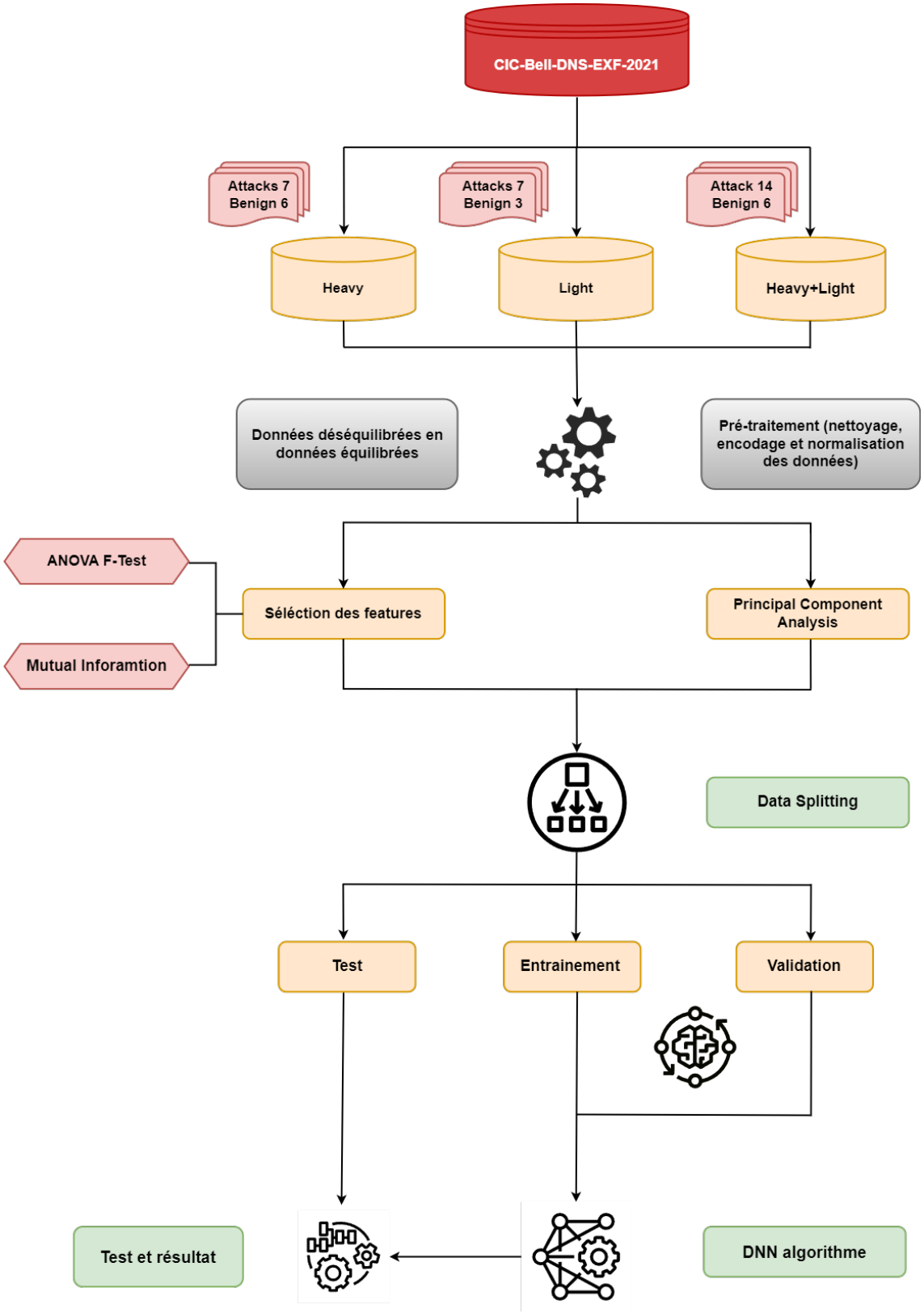


Figure 3.1 – Architecture des composants du travail proposé

3.4 Description de CIC-Bell-DNS-EXF-2021

Le jeu de données CIC-Bell-DNS-EXF-2021 est un ensemble de données de 270,8 Mo de trafic DNS généré par l'exfiltration de différents types de fichiers, allant de petites à grandes tailles. Il a été créé dans le cadre d'un projet collaboratif entre l'Université du Nouveau-Brunswick et Bell Canada, dans le but d'étudier la détection des attaques d'exfiltration de données via le DNS. Pour collecter le trafic de l'attaque d'exfiltration de données DNS, ils ont mené l'attaque dans deux catégories, à savoir l'attaque de fichiers légers et l'attaque de fichiers lourds, pendant cinq jours consécutifs. Chaque catégorie de fichiers lourds et légers comprend six types de fichiers : audio, compressé, .exe, image, texte et vidéo. La taille des fichiers légers est comprise entre 15 et 924 Ko, tandis que la taille des fichiers lourds est comprise entre 4,5 et 26,9 Mo.

Le scénario de l'attaque est le suivant :

$1^{er} jour(Benign)$ **:**

- Vendredi 20 Novembre
- Benign : 9 :59 - 00 :57 (35 636 domains)

$2^{ème} jour(Lightattack)$ **:**

- Samedi 21 Novembre
- Benign : 10 :18 - 14 :00 (9 956 domains)
- Attack
  - ¤ Audio : 15 :13 - 3 :50
  - ¤ Compressed : 18 :09 - 19 :49
  - ¤ Exe : 19 :52 - 20 :46
  - ¤ Image : 20 :48 - 21 :51
  - ¤ Text : 22 :21 - 22 :43
  - ¤ Video : 22 :56 - 23 :37

$3^{ème} jour(Heavyattack)$ **:**

- Dimanche 22 Novembre
- Benign : 6 :53 - 10 :43 (9 956 domains)
- Attack
  - ¤ Audio : 10 :52 - 16 :17

- ¤ Compressed : 16 :46 - 21 :07

$4^{ème} jour(Heavyattack)$ **:**

- Lundi 23 Novembre
- Benign : 11 :06 - 14 :21 (8 403 domains)
- Attack
  - ¤ Image : 14 :27 - 20 :24
  - ¤ Text : 20 :28 - 00 :15

$5^{ème} jour(Heavyattack)$ **:**

- Mardi 24 Novembre
- Benign : 8 :09 - 12 :53 (11 704 domains)
- Attack
  - ¤ Video : 13 :00 - 19 :16
  - ¤ Exe : 19 :18 - 00 :58

Le jeu de données contient 123 698 échantillons d'attaques lourdes, 53 978 échantillons d'attaques légères, et 641 642 échantillons bénins distincts, totalisant 42 caractéristiques et 14 étiquettes de classe différentes. Les tableaux 3.2 et 3.3 offrent une répartition détaillée des caractéristiques du jeu de données combiné. Le tableau 3.4 liste les features du dataset.

Nom du Dataset	CIC-Bell-DNS-EXF-2021
Type Dataset	Multi-classe
L'année de réalisation	2021
Total des colonnes (Nombre de features)	43
Nombre de classes distincte	14

Table 3.2 – Caractéristiques générales du dataset CIC-Bell-DNS-EXF-2021

Catégorie	Stateful	Stateless	DNS Packets
Heavy Attack	72 028	251 670	146.7MB
Heavy-Benign	156 014	402 767	90.3MB
Light Attack	11 295	42 683	20.7MB
Light-Benign	109 766	281 164	62.4MB

Table 3.3 – Statistiques du dataset.

Features	Dtype	Features	Dtype
rr	float64	unique_ttl	object
A_frequency	float64	ttl_mean	float64
NS_frequency	float64	ttl_variance	float64
CNAME_frequency	float64	attacks_types	object
SOA_frequency	float64	timestamp	object
NULL_frequency	float64	FQDN_count	float64
PTR_frequency	float64	subdomain_length	float64
HINFO_frequency	float64	upper	float64
MX_frequency	float64	lower	float64
TXT_frequency	float64	numeric	float64
AAAA_frequency	float64	entropy	float64
SRV_frequency	float64	special	float64
OPT_frequency	float64	labels	float64
rr_type	object	labels_max	float64
rr_count	float64	labels_average	float64
rr_name_entropy	float64	longest_word	object
rr_name_length	float64	sld	object
distinct_ns	float64	len	float64
disntinct_ip	object	subdomain	float64
unique_country	object	unique_asn	object
distinct_domains	object	reverse_dns	object
a_records	float64		

Table 3.4 – Liste des features du dataset

3.4.1 Création d'un dataset étiqueté à partir de CIC-Bell-DNS-EXF-2021

Avant de commencer la phase de prétraitement, il est nécessaire, pour des raisons d'efficacité, de regrouper tous les fichiers en un seul fichier :

- **Heavy :** Nous avons concaténer tous les fichiers concernant la catégorie lourde (heavy) en un seul fichier puis nous avons créer une nouvelle colonne "target" pour classifier les types d'attaque "attacks_types".
- **Light :** En ce qui concerne la catégorie légère (light), nous avons fait la même chose que "Heavy".
- **Heavy + Light :** Dans ce nouveau dataset, nous avons concaténer tous les fichiers des deux catégories à l'exception des trois fichiers heavy-benign et en ajoutant la nouvelle colonne "attacks_types".

3.5 Prétraitement des données

Le prétraitement des données constitue une phase essentielle dans les démarches d'exploration de données et d'apprentissage profond. Cette étape implique une série d'actions cruciales visant à préparer les données initiales en vue de leur analyse ultérieure. Ces actions comprennent notamment le nettoyage des données pour éliminer les valeurs aberrantes et les erreurs, la normalisation pour mettre les données sur une échelle comparable, ainsi que la transformation pour rendre les données plus adaptées aux algorithmes d'apprentissage automatique. En somme, le prétraitement des données joue un rôle fondamental dans la création de jeux de données de qualité et dans l'optimisation des performances des modèles d'apprentissage automatique [38].

3.5.1 Transformation de données (data encoding)

Etant donné que nous avons trouvé deux types de features numériques et catégorielles et que cette dernière nous pose problème dans l'analyse, nous avons décidé de les transformer en numériques en utilisant "**ColumnTransformer**" [47] qui est une classe dans la bibliothèque Python scikit-learn qui permet de sélectionner et d'appliquer des transformations de données à des colonnes spécifiques dans un jeu de données.

Après cela, nous avons utilisé "**OrdinalEncoder**" [48], une classe de prétraitement dans scikit-learn utilisée pour encoder des variables catégorielles en représentations numériques (voir figure 3.2).

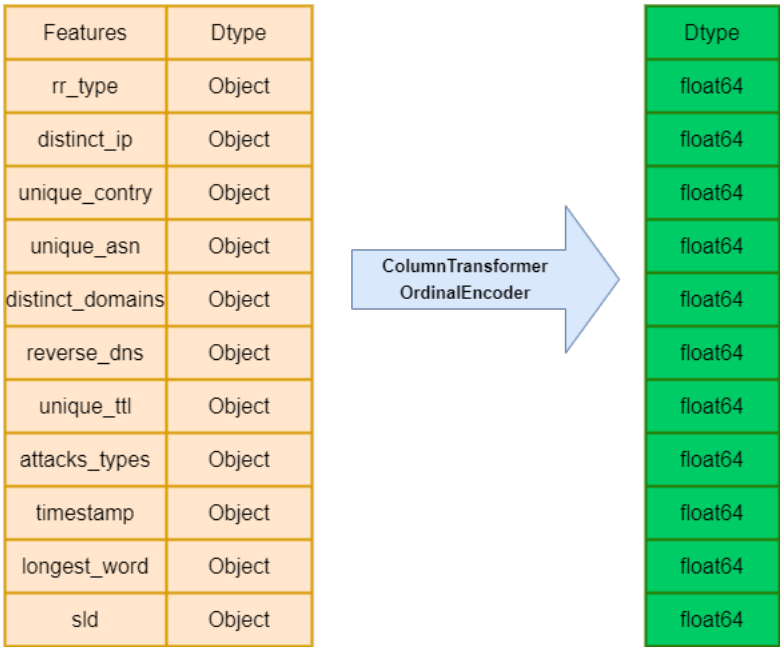


Figure 3.2 – ColumnTransformer/OrdinalEncoder

3.5.2 Nettoyage de données

Après avoir passé en revue notre jeu de données, nous avons éliminé les valeurs manquantes, y compris les valeurs NaN en utilisant la classe de scikit-learn "**SimpleImputer**" [49] qui remplace les valeurs manquantes par une constante, qui peut être spécifiée par l'utilisateur. Dans notre cas, la constante est "mean".

3.5.3 Normalisation

La normalisation d'un ensemble de données est une étape de prétraitement essentielle dans de nombreuses tâches d'apprentissage profond. Elle consiste à mettre à l'échelle les valeurs de toutes les caractéristiques dans un ensemble de données afin qu'elles se situent dans une plage commune.

Pour cela, nous avons utilisé "**MinMaxScaler**" [50], une méthode courante et efficace pour mettre les données à l'échelle d'une plage spécifique. Cette technique transforme les données de manière à ce qu'elles se situent dans une plage spécifiée (généralement entre 0 et 1) en soustrayant la valeur minimale et en la divisant par

la plage des données.

3.5.4 Transformation des données déséquilibrées en données équilibrées

Le processus de transformation de données déséquilibrées en données équilibrées implique diverses techniques pour traiter la distribution déséquilibrée des classes. Nous avons opté pour "SMOTEENN" (voir la page 45), qui a été la plus adaptée à notre dataset (voir figure 3.3).

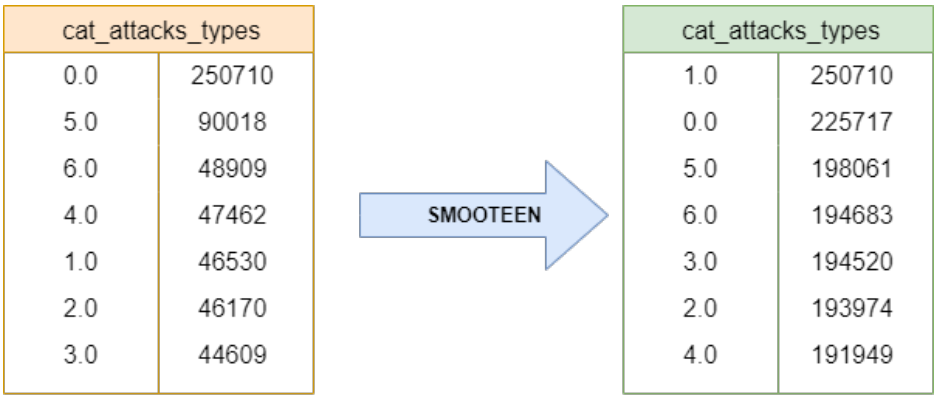


Figure 3.3 – Imbalanced data to balanced data

3.5.5 Sélection des caractéristiques (features)

Après avoir terminer le pré-traitement des données, nous avons entamé la séléction des caractéristiques, un processus qui consiste à trouver un ensemble approprié de caractéristiques sur lesquelles s'entraîner, car un algorithme de DL ne pourra apprendre que si les données d'apprentissage contiennent suffisamment de caractéristiques pertinentes. Concernant cela, vu que notre dataset contient des entrées numeriques et une sortie catégorielle, nous avons opté pour deux approches : **ANOVA F-test** et **Mutual Information**.

Sélection des features pour les attaques heavy : D'après le graphe représenté dans la figure 3.4, le nombre des features sélectionnées est de 31.

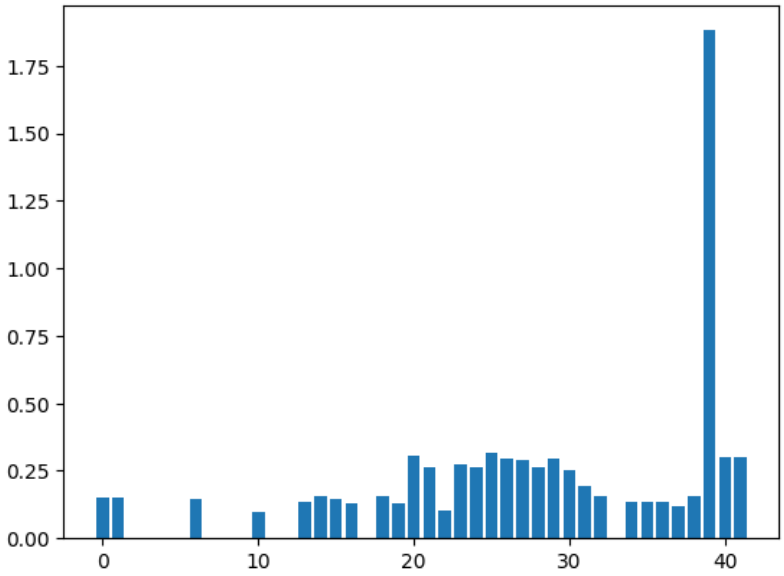


Figure 3.4 – Graphe représentant la sélection des features - Heavy

Sélection des features pour les attaques heavy + light : D'après le graphe représenté dans la figure 3.5, le nombre des features sélectionnées est de 25.

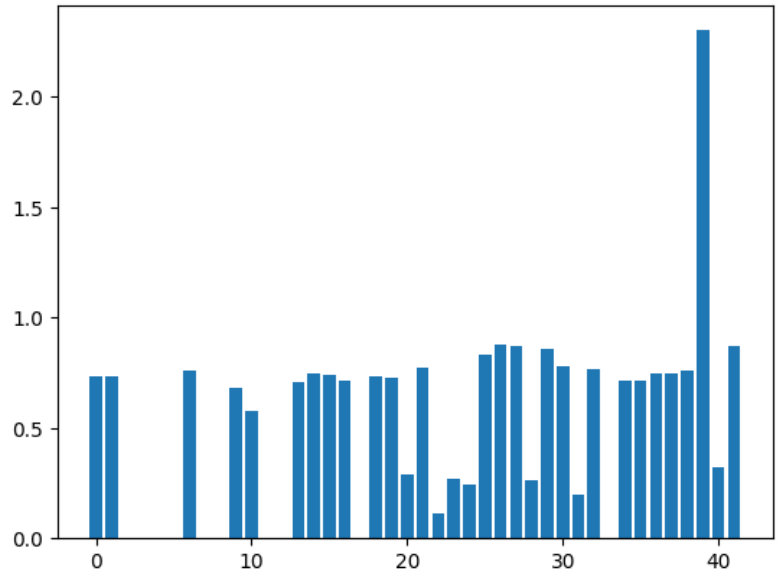


Figure 3.5 – Graphe représentant la sélection des features - Heavy+Light

3.5.6 Réduction de dimensionnalité

La réduction de dimensionnalité en deep learning consiste à réduire le nombre de variables au sein des données d'apprentissage, afin d'obtenir un modèle d'intelligence artificielle plus robuste et un temps de traitement plus rapide [39].

Vu que notre dataset contient plus d'un milion d'enregistrements, nous avons utilisé **PCA (Principal Component Analysis)** qui cherche à simplifier la représentation des données en identifiant les directions dans lesquelles les données varient le plus, permettant ainsi de réduire la complexité des données tout en préservant leur structure essentielle.

3.5.7 Division de données (Data Splitting)

En deep learning, il est essentiel de diviser les données en ensembles d'**entraînement**, de **validation** et de **test** lors du développement des modèles. Cette division permet d'évaluer les performances du modèle sur des données nouvelles et inconnues, tout en évitant le sur-apprentissage (overfitting).

Dans notre cas, l'ensemble d'entraînement comprend 60% des données d'origine, l'ensemble de validation en comprend 20%, et l'ensemble de test en comprend également 20% (voir figure 3.6)

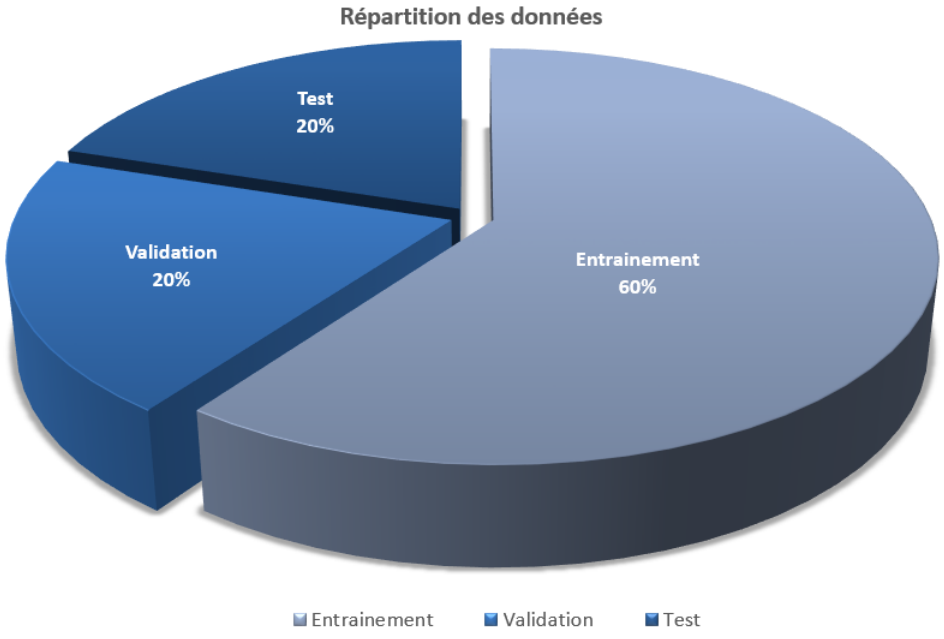


Figure 3.6 – Division des données

3.6 Création du modèle

Nous avons opté pour le modèle séquentiel de la bibliothèque Keras pour notre projet de Deep Learning. Ce choix s'explique par la popularité de ce framework qui offre une grande souplesse et une simplicité d'utilisation très appréciée dans la construction de modèles de Deep Learning. Le modèle séquentiel présente une structure linéaire où les couches sont empilées les unes sur les autres de manière séquentielle. Sa simplicité d'utilisation en fait un choix idéal pour la création de modèles à entrée unique et sortie unique.

La mise en place et l'entraînement d'un réseau neuronal impliquent une série d'étapes cruciales. Tout commence par le choix de l'architecture de réseau neuronal la mieux adaptée au problème à résoudre. Nous avons opté pour un réseau de neurones profond (DNN - Deep Neural Network) pour cette tâche. Une fois l'architecture du modèle de réseau neuronal sélectionnée, vient ensuite l'étape de spécification de ses hyperparamètres.

Nous avons élaboré un modèle en six couches, structuré autour d'une couche d'entrée, de quatre couches cachées, et d'une couche de sortie. Initiée avec un nombre de neurones égal au nombre de features du dataset, la couche d'entrée guide les données dans le réseau. Les couches cachées successives réduisent progressivement la dimensionnalité de l'information, avec 64, 32, 32 et 32 neurones respectivement. Enfin, la couche de sortie se compose de 14 neurones, représentant les classes "Heavy" et "Light", chacune disposant de 7 neurones pour une classification précise selon les deux catégories. La fonction d'activation Rectified Linear Unit (ReLU) [40] est appliquée à toutes les couches cachées, instaurant ainsi une non-linéarité vitale dans le modèle de deep learning, surmontant ainsi l'obstacle des gradients déclinants.

La fonction d'activation Softmax est spécifiée pour la couche de sortie, générant ainsi une distribution de probabilité parmi les différentes classes. Cette fonction est couramment adoptée dans les problèmes de classification multiclasse. Par ailleurs, nous avons opté pour l'optimiseur Adam. La fonction de perte employée est de type categorical_crossentropy, une méthode de choix pour les problèmes de classification multiclasse, servant à évaluer la divergence entre les probabilités prédites et les véritables étiquettes de classe. Pour éviter le sur-apprentissage (overfitting), le modèle a été régularisé en utilisant une pénalisation L2, avec un coefficient de régularisation de 0.0001 appliqué sur le nœud.

Enfin, le modèle est entraîné sur les données d'entraînement, avec un ajustement des poids neuronaux visant à minimiser la fonction de perte. Ce processus itératif se poursuit jusqu'à ce que le modèle atteigne une convergence, aboutissant à un ensemble de poids optimaux pour maximiser la performance sur les données d'apprentissage.

3.7 Résultats et analyses

Nous avons conçu un modèle de réseau neuronal profond (DNN) en explorant trois approches distinctes. Cette section détaille les résultats des tests effectués pour chaque configuration. Afin de déterminer les hyperparamètres optimaux, nous avons réalisé de nombreux essais, ajustant le nombre de couches, de neurones par couche, le nombre d'itérations (epochs), le type d'optimiseur et les fonctions d'activation des couches cachées.

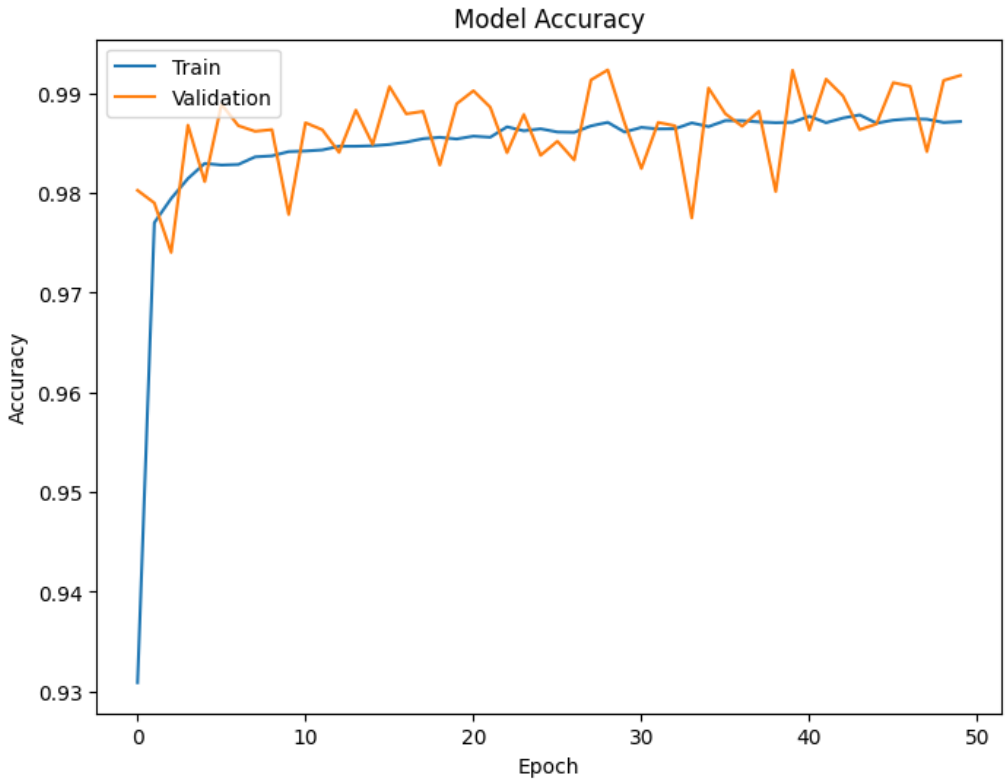
Chaque modèle a été entraîné sur l'ensemble d'entraînement et validé sur un ensemble de validation, répétant ce processus avec diverses configurations jusqu'à atteindre les performances optimales. Les performances finales ont été validées sur l'ensemble de test. Le modèle présentant la meilleure précision ("accuracy") et le taux d'erreur ("loss") le plus faible a été identifié comme meilleur modèle.

3.7.1 Catégorie - Heavy

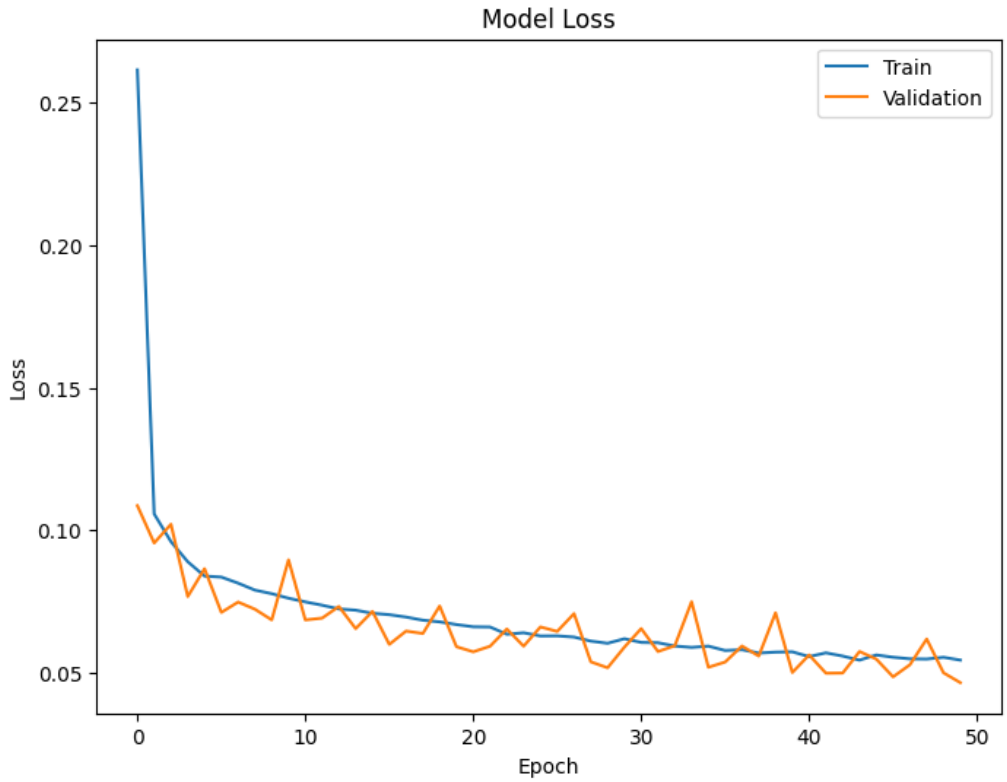
En ce qui concerne la catégorie "heavy", nous avons conçu un modèle DNN en explorant quatre approches distinctes et le modèle présentant la meilleure précision et le taux d'erreur le plus faible était celui qu'on a utilisé : Mutual information + SMOTEENN (voir la figure 3.7 et le tableau 3.5)

Accuracy	**Precision**	**F1 score**	**Loss**
99.15%	99%	99%	4%

Table 3.5 – Métriques d'évaluation



(a) Entraînement et validation Accuracy



(b) Entraînement et validation Loss

Figure 3.7 – Visualitaion des performances du modèle

La figure 3.7-(a) illustre l'évolution de la précision du modèle pour les ensembles d'entraînement (courbe bleue) et de validation (courbe orange) au fil des époques. On constate une augmentation rapide de la précision au début de l'entraînement, suivie d'une stabilisation autour de 98% pour l'ensemble d'entraînement et de 96% pour l'ensemble de validation.

La figure 3.7-(b) montre la progression de la perte (loss) du modèle pour les mêmes ensembles d'entraînement et de validation. On observe une diminution rapide de la perte au début de l'entraînement.

Les métriques d'exactitude, de précision, de rappel et de score F1 ont brillamment éclairé le chemin de l'évaluation de notre modèle (voir tableau 3.6). Leur perspicacité nous a offert une vision approfondie de ses performances, éclairant nos décisions quant à son perfectionnement. Elles ont révélé des pistes d'amélioration, telles que l'ajustement du seuil de décision pour la classification ou l'harmonisation de la distribution des classes dans notre jeu de données.

Classes	**Accuracy**	**Precision**	**Recall**	**F1-score**
Benign heavy	98.44%	100%	98%	99%
Attack heavy audio	98.46%	99%	98%	99%
Attack heavy compressed	99.17%	98%	99%	98%
Attack heavy exe	97.77%	100%	100%	100%
Attack heavy image	99.57%	99%	100%	99%
Attack heavy text	99.61%	99%	100%	99%
Attack heavy video	99.72%	99%	100%	99%

Table 3.6 – Métriques d'évalution de chaque classe

Une matrice de confusion est un outil utilisé en apprentissage profond et en statistique pour évaluer les performances d'un modèle de classification. Chaque cellule de la matrice indique le nombre d'instances assignées à la classe prédite correspondante (colonne), tandis que leur classe réelle est celle de la ligne. Les cellules diagonales, allant du coin supérieur gauche au coin inférieur droit, représentent les instances correctement classées, tandis que les autres cellules mettent en évidence les erreurs de classification (voir figure 3.8)

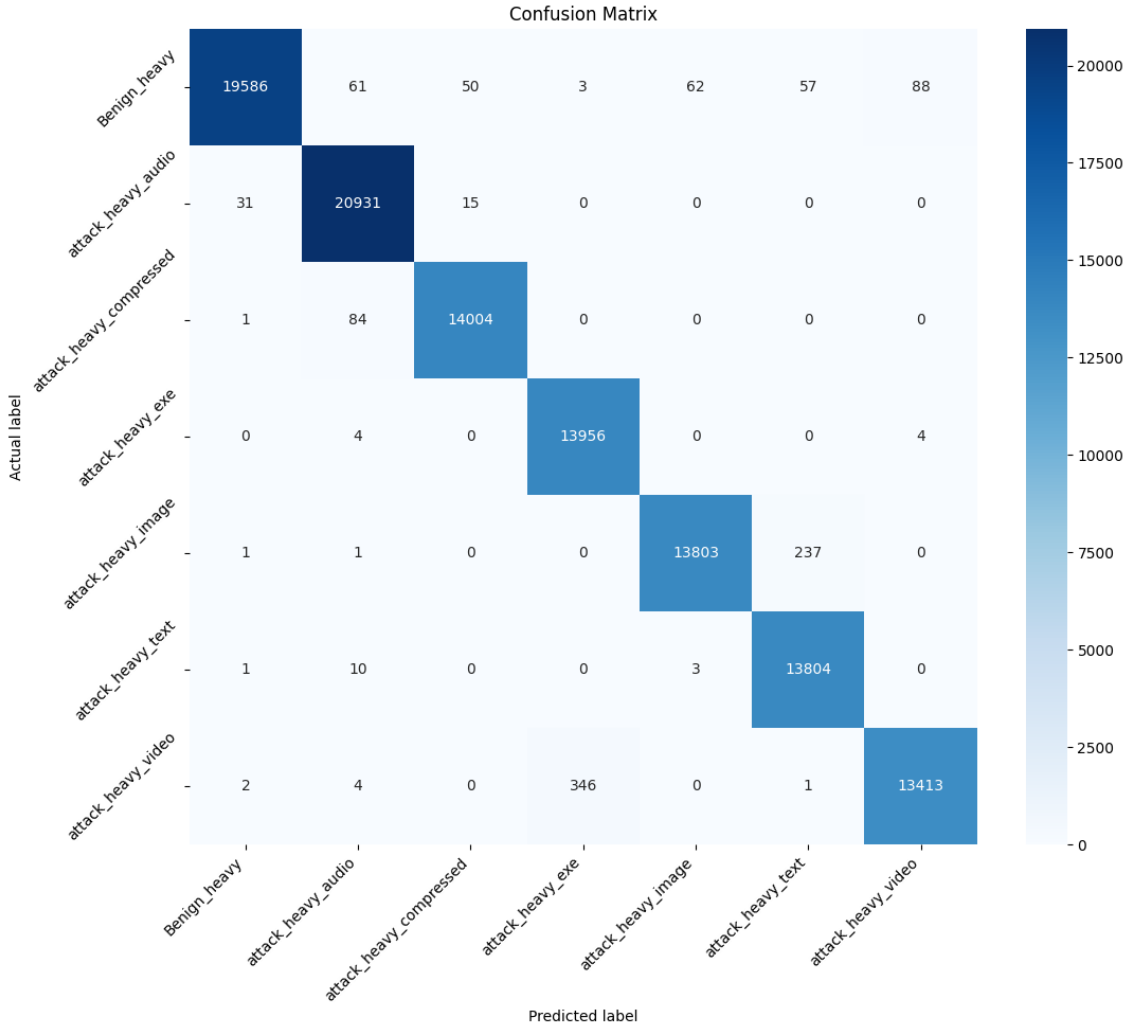


Figure 3.8 – Matrice de confusion multi-classes

Le tableau 3.7 représente les quatre approches que nous avons utilisé.

Méthodes de séléction	**Méthodes d'équili-brage**	**Encoding approches**	**Accuracy**	**Loss**	**Nombre des features**
Anova	SMOTEENN	ColumnTransform (Ordinalencoder)	96.33%	8%	9
Mutual In-formation	SMOTEENN	ColumnTransform (Ordinalencoder)	99.15%	4%	31
Mutual In-formation	NearMiss	ColumnTransform (Ordinalencoder)	76.81%	59%	32
PCA	SMOTEENN	ColumnTransform (Ordinalencoder)	99.01%	34.8%	Toutes

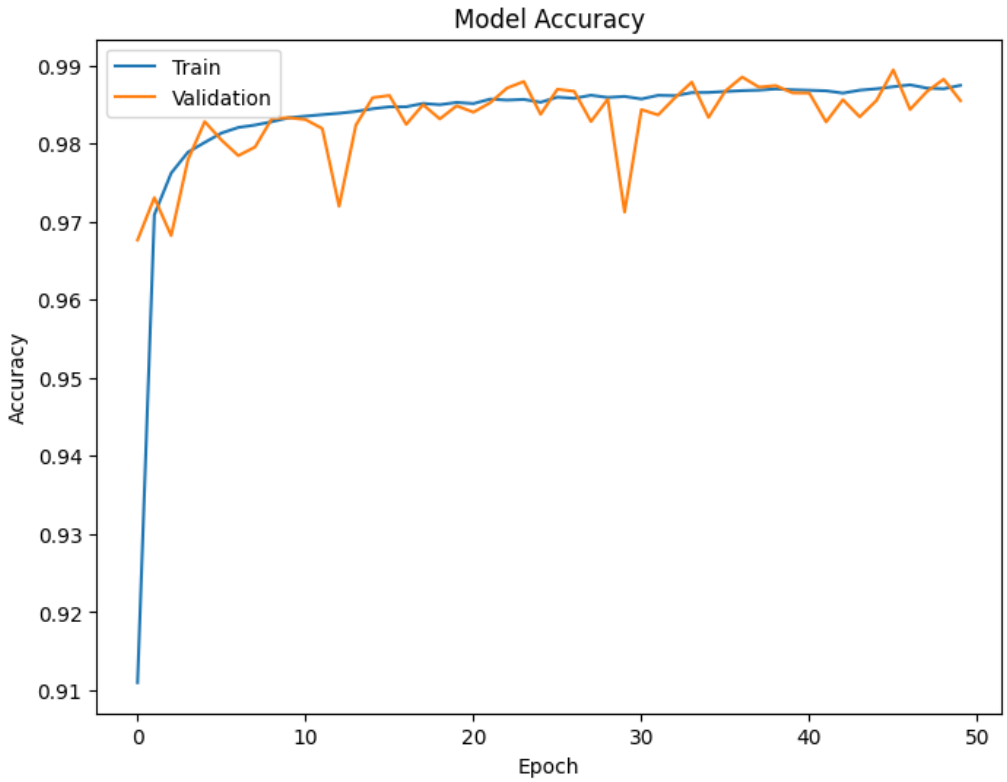
Table 3.7 – Résultats des approches

3.7.2 Catégorie -Light

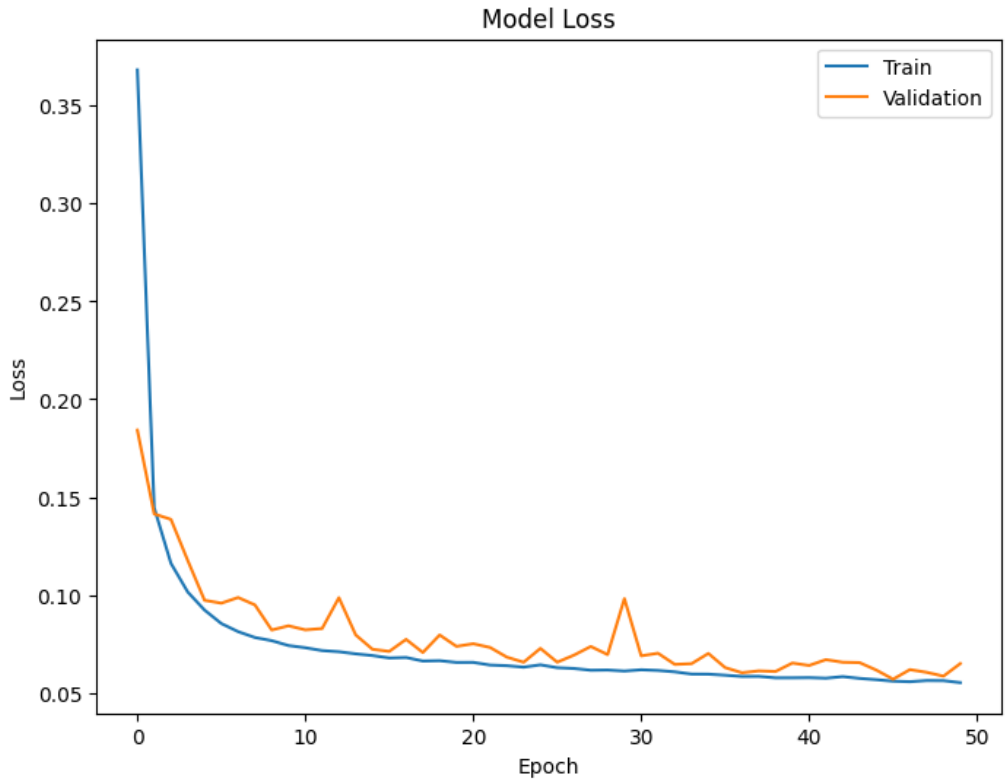
Concernant la catégorie "light", notre exploration a abouti à la conception d'un modèle DNN, résultant de l'examen attentif de six approches distinctes. Parmi celles-ci, celle qui a émergé avec une précision optimale et le plus faible taux d'erreur a été celle que nous avons opté pour PCA + SMOTEENN, comme illustré dans la figure 3.9 et le tableau 3.8.

Accuracy	**Precision**	**F1 score**	**Loss**
98.86%	99%	99%	7%

Table 3.8 – Métriques d'évaluation



(a) Entraînement et validation Accuracy



(b) Entraînement et validation Loss

Figure 3.9 – Visualisation des performances du modèle

La figure 3.9-(a) présente l'évolution de la précision (accuracy) du modèle sur les ensembles d'entraînement et de validation. On observe une amélioration progressive de la précision au cours de l'entraînement, atteignant environ 98% pour l'ensemble d'entraînement et 97% pour l'ensemble de validation. La figure 3.9-(b) illustre la perte (loss) du modèle, une mesure de l'erreur commise par celui-ci. On remarque une réduction continue de la perte au fur et à mesure de l'entraînement, passant d'environ 0.4 à 0.1 pour l'ensemble d'entraînement et à 0.15 pour l'ensemble de validation. Le tableau 3.9 représente les valeurs des métriques d'évaluation pour les différentes classes.

Classes	**Accuracy**	**Precision**	**Recall**	**F1-score**
Benign light	98.88%	98%	99%	98%
Attack light audio	99.43%	99%	99%	99%
Attack light compressed	98.11%	99%	98%	99%
Attack light exe	97.05%	97%	97%	97%
Attack light image	99.25%	99%	99%	99%
Attack light text	99.68%	100%	100%	100%
Attack light video	99.74%	100%	100%	100%

Table 3.9 – Métriques d'évalution de chaque classe

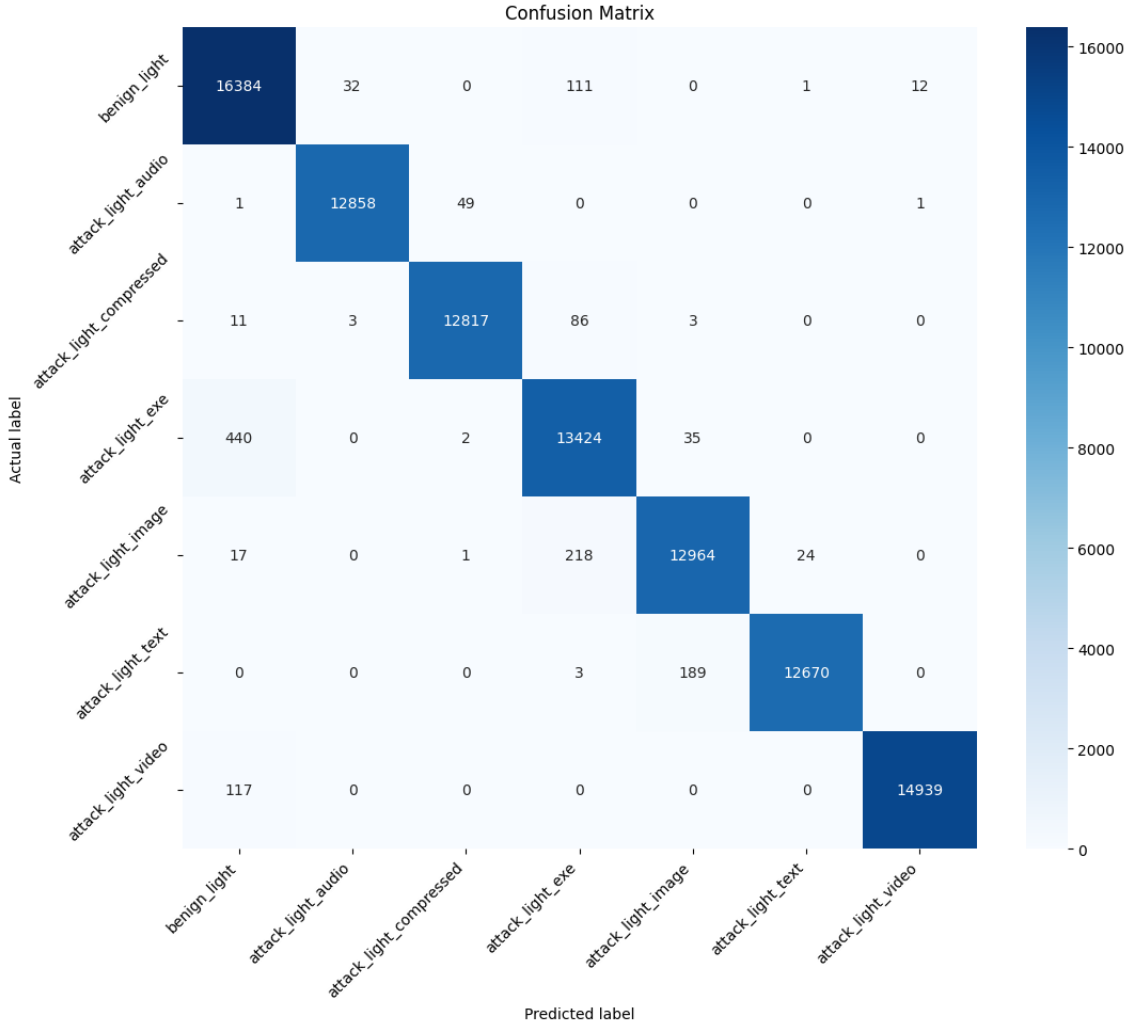


Figure 3.10 – Matrice de confusion multi-classe

Les approches que nous avons employées sont représentées dans le tableau 3.10 ci-dessous.

Méthodes de séléction	**Méthodes d'équili-brage**	**Encoding approches**	**Accuracy**	**Loss**	**Nombre des features**
Anova	SMOTEENN	ColumnTransform (Ordinalencoder)	95.16%	11%	9
Anova	NearMiss	ColumnTransform (Ordinalencoder)	85.4%	4%	15
Anova	RandomOver	ColumnTransform (Ordinalencoder)	81.23%	46%	9
PCA	SMOTEENN	ColumnTransform (Ordinalencoder)	98.88%	6%	Toutes
PCA	SMOTEENN	ColumnTransform (Ordinalencoder)	98.83%	5%	32
Mutual In-formation	SMOTEENN	ColumnTransform (Ordinalencoder)	98.39%	7%	15

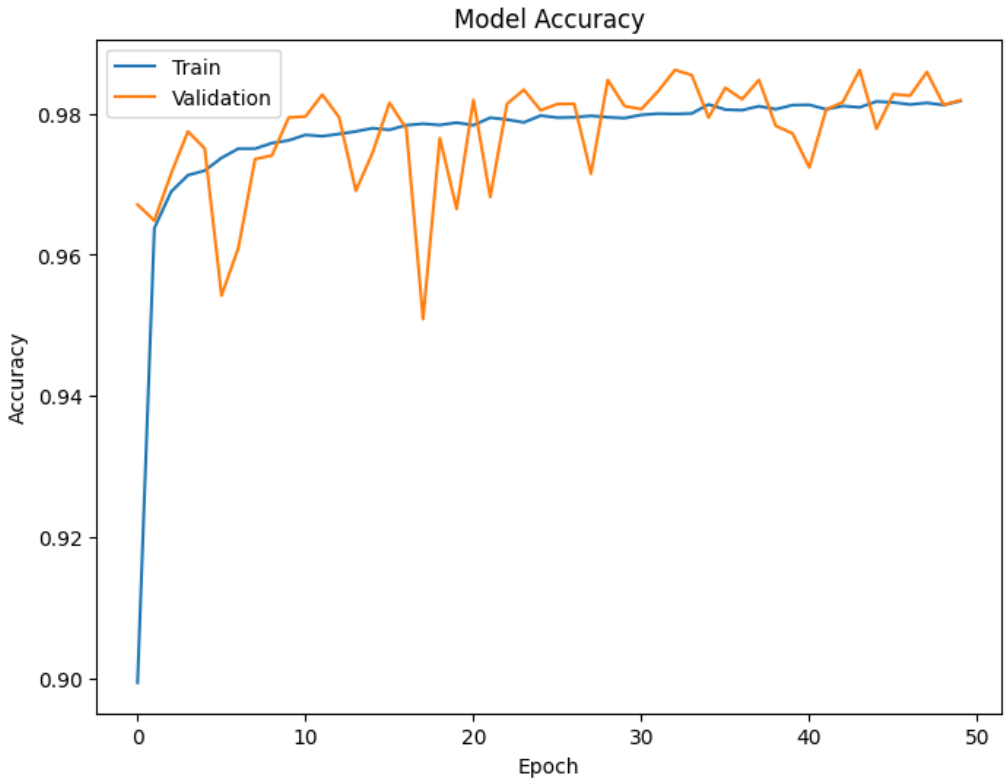
Table 3.10 – Résultats des approches

3.7.3 Attaques "Heavy + light"

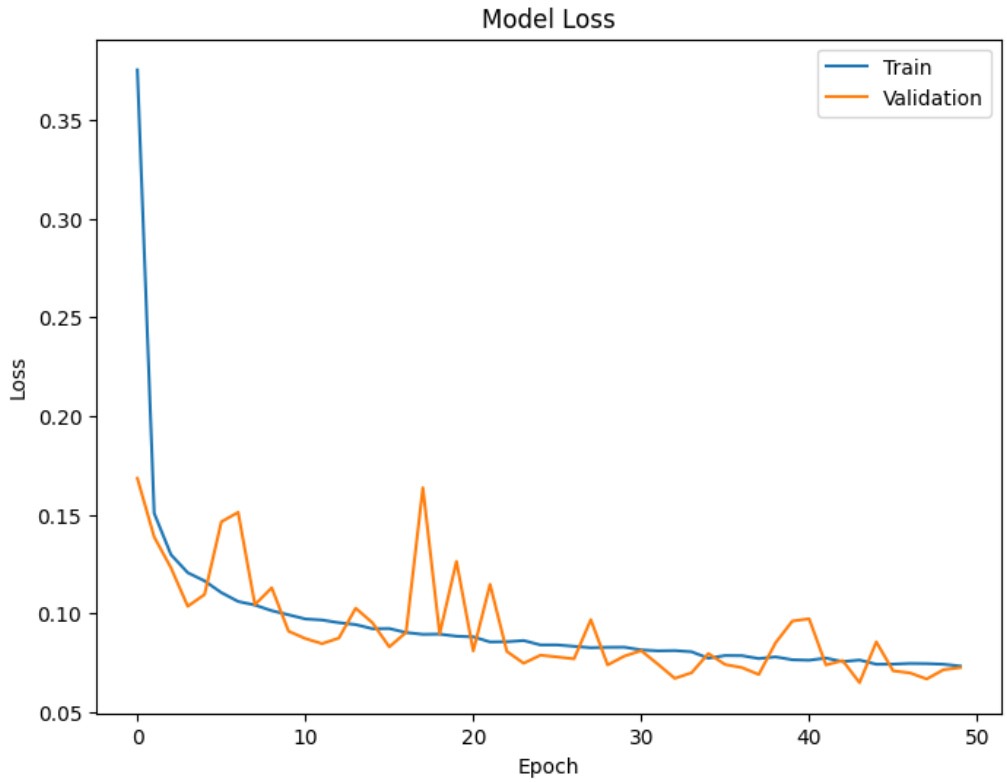
Pour le nouveau dataset comprenant les deux catégories (heavy et light), nous avons élaboré un modèle de réseau de neurones profond (DNN) en intégrant les deux meilleures approches de sélection de caractéristiques : Mutual Information et PCA (Analyse en Composantes Principales). Après une évaluation rigoureuse, il est apparu que le modèle utilisant Mutual Information associé à la technique de rééchantillonnage SMOTEENN offrait les meilleures performances (voir figure 3.11 et tableau 3.11)

Accuracy	**Precision**	**F1 score**	**Loss**
98.2%	98%	98%	7%

Table 3.11 – Métriques d'évaluation



(a) Entraînement et validation Accuracy



(b) Entraînement et validation Loss

Figure 3.11 – Visualisation des performances du modèle

La courbe de perte d'entraînement révèle une tendance générale à la baisse, ponctuée de quelques fluctuations et pics au cours du processus d'apprentissage. Ce comportement est attendu, car le modèle s'adapte progressivement aux données d'entraînement pour optimiser ses performances.

En parallèle, la courbe de perte de validation, qui évalue les performances du modèle sur des données non vues, affiche également une diminution globale. Cependant, elle présente davantage de fluctuations et de pics comparativement à la perte d'entraînement. Cela est typique, car l'ensemble de validation n'est pas utilisé pour l'apprentissage, ce qui entraîne une variabilité plus marquée des performances du modèle sur ces données inédites.

Globalement, ces courbes indiquent que le modèle progresse et améliore ses performances tant sur les données d'entraînement que sur les données de validation au fil des itérations d'apprentissage. Le tableau 3.12 représente les métriques d'évaluation pour les différentes classes alors que la figure 3.12 représente la matrice de confusion.

Classes	**Accuracy**	**Precision**	**Recall**	**F1-score**
Benign light	99.25%	100%	99%	100%
Attack light audio	98.50%	96%	99%	97%
Attack light compressed	98.41%	99%	98%	99%
Attack light exe	93.89%	98%	94%	96%
Attack light image	99.92%	95%	100%	98%
Attack light text	93.35%	98%	93%	96%
Attack light video	94.05%	92%	94%	93%
Benign heavy	98.48%	100%	98%	99%
Attack heavy audio	97.05%	100%	97%	98%
Attack heavy compressed	99.93%	98%	100%	99%
Attack heavy exe	99.36%	100%	99%	100%
Attack heavy image	99.94%	99%	100%	99%
Attack heavy text	99.76%	98%	100%	98%
Attack heavy video	98.57%	99%	99%	99%

Table 3.12 – Métriques d'évalution de chaque classe

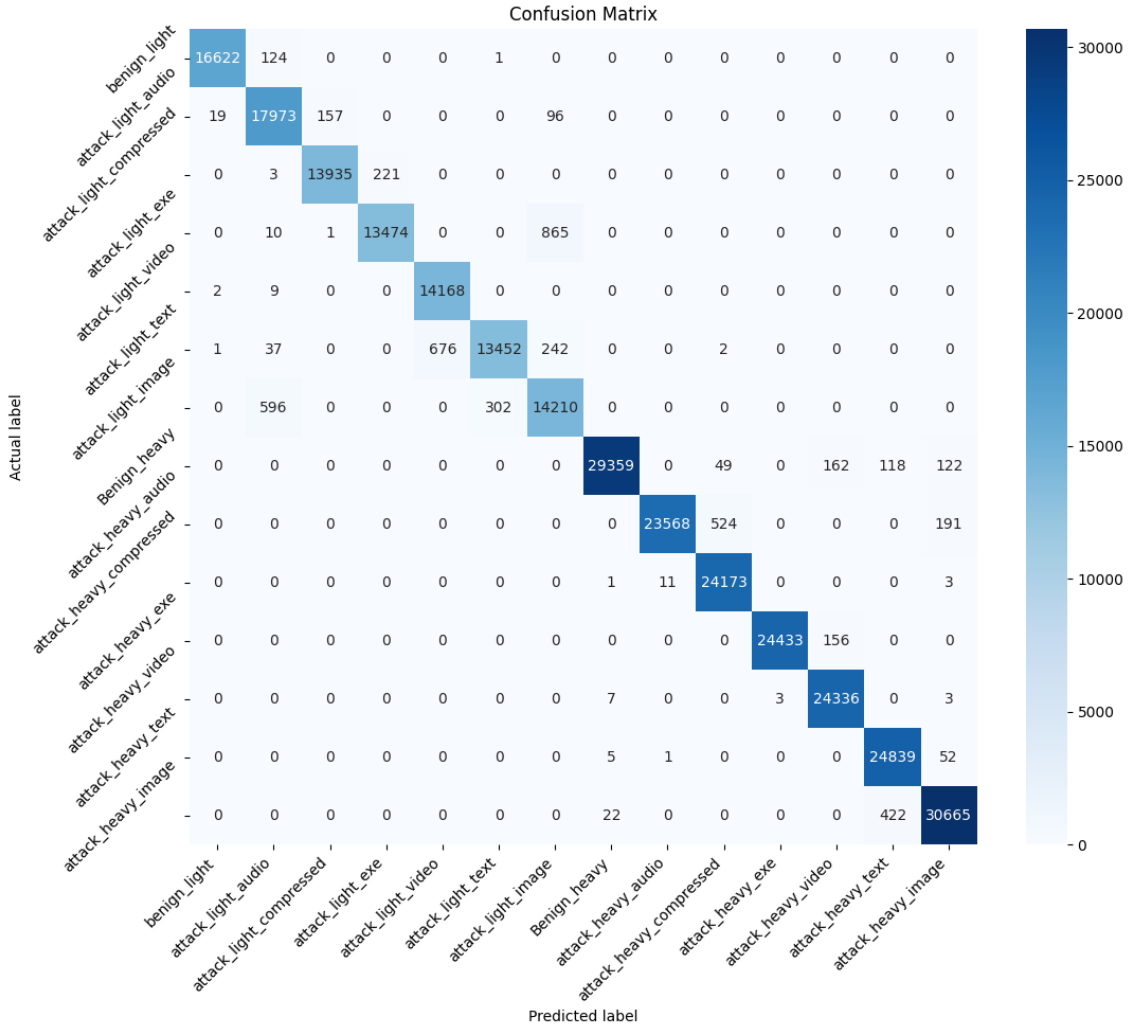


Figure 3.12 – Matrice de confusion multi-classes

En somme, les approches que nous avons employées sont représentées dans le tableau 3.13 ci-dessous.

Méthodes de séléction	**Méthodes d'équili-brage**	**Encoding approches**	**Accuracy**	**Loss**	**Nombre des features**
Mutual Information	SMOTEENN	ColumnTransform (Ordinalencoder)	98.2%	7%	25
PCA	SMOTEENN	ColumnTransform (Ordinalencoder)	98.7%	68%	Toutes

Table 3.13 – Résultats des approches

3.8 Résultats comparitifs

Nous avons comparé nos résultats avec ceux de l'article "Lightweight Hybrid Detection of Data Exfiltration using DNS based on Machine Learning" [41] (voir les deux tableau 3.14 et 3.15).

ML models	Light Attack			Heavy Attack		
	Precision	F1_Score	Accuracy	Precision	F1_Score	Accuracy
GNB	91.41	74.94	69.43	83.46	71.24	71.01
RF	99.94	99.94	99.94	99.97	99.97	99.97
MLP	91.13	93.12	95.32	99.84	99.84	99.84
SVM	78.53	83.25	88.60	65.31	64.90	71.36
LR	92.97	93.39	95.13	92.07	89.59	89.22

Table 3.14 – Résultats obtenus dans [41].

Méthodes	Light Attack			Heavy Attack		
	Precision	F1_Score	Accuracy	Precision	F1_Score	Accuracy
Anova + SOOMTEEN	91.41	74.94	95.16	83.46	71.24	71.01
Mutual Inf + SOOMTEEN	99.94	98.39	99.94	99.97	99.97	99.97
PCA + SOOMTEEN	91.13	93.12	98.88	99.84	99.84	99.84

Table 3.15 – Nos résultats obtenus.

3.9 Conclusion

Ce dernier chapitre visait à présenter une solution novatrice pour surmonter les limitations des systèmes de détection d'attaques. La solution proposée améliore considérablement les performances globales en augmentant la précision de la détection et de la catégorisation d'une vaste gamme d'attaques tout en réduisant les fausses alertes. Pour permettre aux systèmes de détection de reconnaître des motifs et d'identifier des attaques nouvelles et inconnues, il est crucial de disposer d'un large éventail d'échantillons pour chaque type d'attaque. Les expériences menées démontrent l'efficacité de cette approche, produisant des résultats remarquablement satisfaisants.

CONCLUSION GÉNÉRALE

Animés par un intérêt croissant pour l'amélioration de la sécurité du DNS et la prévention des vulnérabilités susceptibles d'entraîner des conséquences désastreuses, les administrateurs de la sécurité des réseaux recherchent constamment des solutions pour assurer un environnement de réseau hautement sécurisé.

Le DNS, en tant que répertoire essentiel qui convertit les noms de domaine en adresses IP, facilite la navigation et la communication sur Internet. Cependant, cette centralité en fait également une cible de choix pour diverses attaques, telles que le DNS Tunneling, l'amplification DNS et le cache poisoning. Ces attaques peuvent gravement compromettre la confidentialité, l'intégrité et la disponibilité des services en ligne.

Pour répondre à ces défis, il est nécessaire d'adopter des mesures de sécurité robustes et diversifiées. L'implémentation de technologies telles que DNSSEC, l'utilisation de serveurs DNS redondants, et la sensibilisation des utilisateurs aux risques associés au DNS sont des étapes fondamentales. Par ailleurs, l'intégration de techniques de deep learning offre des perspectives prometteuses pour améliorer la détection et la réponse aux attaques DNS.

L'application du deep learning, en particulier les réseaux de neurones profonds (DNN) et les techniques de rééchantillonnage comme SMOTEENN, a montré une efficacité notable dans la détection des attaques hybrides sur le DNS. Les modèles développés dans ce mémoire ont démontré une grande précision dans la classification des attaques, réduisant les faux positifs et augmentant ainsi la fiabilité des systèmes de détection.

En conclusion, la sécurisation du DNS nécessite une approche proactive et continue. Les avancées technologiques, combinées à des pratiques de sécurité rigoureuses

et à l'innovation dans le domaine du deep learning, offrent des perspectives prometteuses pour protéger l'infrastructure DNS contre les cybermenaces. La recherche future pourrait se concentrer sur l'amélioration des modèles de deep learning et l'adaptation rapide aux nouvelles formes d'attaques, assurant ainsi une protection durable et efficace des services en ligne.

BIBLIOGRAPHIE

[1] nameshield«DNS (DOMAIN NAME SYSTEM)» https ://www.nameshield.com/ressources/lexique/dns-domain-name-system/ .Consulté le : 30 mars 2024

[2] Fortinet «Qu'est-ce que le DNS ?.» https ://www.fortinet.com/fr/resources/cyberglossary/what is-dns consulté le : 01 avril 2024

[3] Stephane (05 juillet 2011) «Format des messages du protocole DNS»https ://www.altospam.com/actualites/format-des-messages-du-protocole-dns/. Consulté le : 16 avril 2024

[4] heimdal (30 mars 2023). «What Is Encrypted DNS Traffic ?» https ://heimdalsecurity.com/blog/what-is-encrypted-dns-traffic/ Consulté le :20 avril 2024

[5] dnschecker. « DNSSEC - Domain Name System Security Extensions» https ://dnschecker.org/dns-articles/dnssec-domain-name-system-security-extensions.html .Consulté le : 20 avril 2024

[6] Ivan Lee, «Attaque Par Tunnel DNS» https ://www.wallarm.com/what/dns-tunneling-attack .Consulté le : 20 avril 2024

[7] Cyber Security Agency of Singapore. « Domain Name System Security Extensions (DNSSEC)» https ://www.csa.gov.sg/Tips-Resource/internet-hygiene-portal/information-resources/dnssec Consulté le : 20 avril 2024

[8] Ahmed Bahgat, (21 novembre 2022). «Les 15 meilleurs outils pour une analyse efficace du trafic des sites web» https ://kinsta.com/fr/blog/outils-analyse-

trafic-site/les-15-meilleurs-outils-pour-une-analyse-efficace-du-trafic-dun-site-web . Consulté le : 14 avril 2024

[9] J. Liu, S. Li, Y. Zhang, J. Xiao, P. Chang et C. Peng, « Détection du tunnel DNS via une classification binaire basée sur des caractéristiques comportementales », 2017 IEEE Trustcom/BigDataSE/ICESS , Sydney, NSW, Australie, 2017, pp. 339-346, doi : 10.1109/Trustcom/BigDataSE/ICESS.2017.256. Consulté le : 21 avril 2024

[10] heimdal (25 avril 2023). « What Is DNS Tunneling, and How to Detect and Prevent It » https ://heimdalsecurity.com/blog/dns-tunneling/. Consulté le :21 avril 2024

[11] BLANC, LIVRE. "Les DNS et la menaces des attaques DDoS.". Consulté le : 26 avril 2024

[12] cloudflare, (), «Qu'est ce que l'empoisonnement de cache DNS ? | Usurpation de DNS » https ://www.cloudflare.com/fr-fr/learning/dns/dns-cache-poisoning/. Consulté le : 26 avril 2024

[13] Fortinet. (n.d.). «Qu'est-ce que la sécurité DNS ?» https ://www.fortinet.com/fr/resources/cyberglossary/dns-security. Consulté le : 26 Avril 2024.

[14] NetApp. « QU'EST-CE QUE L'INTELLIGENCE ARTIFICIELLE ? » https ://www.netapp.com/fr/artificial-intelligence/what-is-artificial-intelligence/. Consulté le : 11 Mars 2024

[15] Monkeylearn, « An Introduction to Machine Learning » https ://monkeylearn.com/machine-learning/ !. Consulté le : 11 Mars 2024

[16] Younes, Benzaki. (22 novembre 2016). « Introduction au Machine learning : Définitions et Concepts » https ://mrmint.fr/introduction-machine-learning. Consulté le : 11 Mars 2024

[17] Céline, Deluzarche (14 Octobre 2023). «Deep Learning : qu'est-ce que c'est ?» https ://www.futura-sciences.com/tech/definitions/intelligence-artificielle-deep-learning-17262/fonctionnement-du-deep-learning. Consulté le : 14 Mars 2024

[18] Abid Ali, Awan (Octobre 2023). « What is Deep Learning ? A Tutorial for Beginners »https ://www.datacamp.com/tutorial/tutorial-deep-learning-tutorial. Consulté le : 14 Mars 2024

[19] Reche, Jérôme. (2019). Nouvelle méthodologie hybride pour la mesure de rugosités sub-nanométriques.

[20] Aspexit, (08 avril 2019), «Réseau de neurones : on va essayer de démystifier un peu tout ça. » https ://www.aspexit.com/reseau-de-neurones-on-va-essayer-de-demystifier-un-peu-tout-ca-1/. Consulté le : 16 mars 2024

[21] Geeksforgeeks, (14 Mars 2024), « Introduction to Convolution Neural Network » https ://www.geeksforgeeks.org/introduction-convolution-neural-network/ ?ref=lbp. Consulté le : 18 Mars 2024

[22] Kobia. « Imbalanced data et Machine Learning » https ://kobia.fr/imbalanced-data-et-machine-learning/. Consulté le : 14 Mars 2024

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

[24] Michael Greenacre, Patrick J. F. Groenen, Trevor Hastie, Alfonso Iodice D'Enza, Angelos Markos, Elena Tuzhilina. (8 Mars 2023). « Principal component analysis » https ://www.nature.com/articles/s43586-022-00184-w. Consulté le : 18 Mars 2024

[25] BuiltIn, (30 juin 2023). «"Common Loss Functions in Machine Learning".» https ://builtin.com/machine-learning/common-loss-functions. Consulté le : 19 mars 2024

[26] Vishal, Yathish (04 aout 2022). « Loss Functions and Their Use in Neural Networks. » https ://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9. Consulté le : 19 mars 2024

[27] Kaggle. (2019).« "Regularization in Deep Learning".» https ://www.kaggle.com/code/procode/regularization-in-deep-learning. Consulté le : 18 mars 2024

[28] Geeksforgeeks (23 avril 2023).«Difference between Batch Gradient Descent and Stochastic Gradient Descent» https ://www.geeksforgeeks.org/difference-between-batch-gradient-descent-and-stochastic-gradient-descent/.Consulté le : 21 mars 2024

[29] BOUANANE, K., DOKKAR, B., MEDDOUR, B. (2023). FIRST ORDER OPTIMIZATION METHODS FOR DEEP LEARNING (Doctoral dissertation, UNIVERSITY OF KASDI MERBAH OUARGLA). Consulté le : 21 mars 2024

[30] Kingma, Diederik P. et Jimmy Ba. "Adam : Une méthode d'optimisation stochastique." Préimpression arXiv arXiv :1412.6980 (2014). Consulté le 21 mars 2024

[31] CHALIFOUR, V., ASSOCIÉES, P. O. E. S. S. (2019). «COMME EXIGENCE PARTIELLE DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES». Consulté le : 17 mars 2024

[32] SEKHER, Khaled et BOUTAGHANE, Ayoub. Prédiction des paramètres pétrophysiques à l'aide des méthodes de l'intelligence artificielle, cas du réservoir TAGI champs de Chebet El-Nakhla bassin d'Oued M'Ya, sud-est de l'Algérie. Thèse de doctorat. Consulté le : 15 mars 2024

[33] Rukshan Pramoditha(31 Octobre 2022).«Deep Learning Hardware Selection Guide for 2023» https ://medium.com/data-science-365/deep-learning-hardware-selection-guide-for-2023.Consulté le : 20 avril 2024

[34] Jérémy Robert (10 Janvier 2022).«Pandas : la bibliothèque Python dédiée à la Data Science» .https ://datascientest.com/pandas-python-data-science. Consulté le :20 avril 2024.

[35] Raphael Kassel(20 avril 2021).«NumPy : la bibliothèque Python la plus utilisée en Data Science»https ://datascientest.com/numpy. Consulté le : 20avril 2024.

[36] Raphael Kassel(21 février 2021).«Matplotlib : Tout savoir sur la bibliothèque Python de Dataviz»https ://datascientest.com/matplotlib-tout-savoir.Consulté le : 20 avril 2024.

[37] Raphael Kassel(02 avril 2021).«Seaborn : tout savoir sur l'outil de Data Visualization en Python»hhttps ://datascientest.com/seaborn-tout-savoir.Consulté le : 20 avril 2024.

[38] George Lawton (January 2022).«DEFINITION datapreprocessing»https ://www.techtarget.com/searchdatamanagement/definition/datapreprocessing.Consulté le : 21 avril 2024.

[39] Antoine Crochet-Damais(02 juin 2022).«Réduction de dimensionnalité en machine learning : définition».https ://www.journaldunet.fr/intelligence-artificielle/guide-de-l-intelligence-artificielle/1501907-reduction-de-dimensionnalite/. Consulté le : 21 avril 2024

[40] Bharath Krishnamurthy (26 Février 2024)«An Introduction to the ReLU Activation Function» https ://builtin.com/machine-learning/relu-activation-function Consulté le : 22 Avril 2024.

[41] Mahdavifar, S., Hanafy Salem, A., Victor, P., Razavi, A. H., Garzon, M., Hellberg, N., Lashkari, A. H. (2021, December). Lightweight hybrid detection of data exfiltration using dns based on machine learning. In Proceedings of the 2021 11th International Conference on Communication and Network Security (pp. 80-86)

[42] Çağlar, U (Mars 2022) «What is Kaggle ?» https ://www.datacamp.com/blog/what-is-kaggle. Consulté le : 20 Avril2024

[43] Raphael Kassel (30 Mars 2021) «Jupyter Notebook : Un outil indispensable en partage de code» https ://datascientest.com/jupyter-notebook-tout-savoir. Consulté le : 20 Avril 2024

[44] Futura (10 Mai 2021) «Python : qu'est-ce que c'est ?» https ://www.futura-sciences.com/tech/definitions/informatique-python-19349/. Consulté le : 20 Avril 2024

[45] Raphael Kassel (7 Janvier 2021) «TensorFlow : le framework de Machine Learning de Google» https ://datascientest.com/tensorflow. Consulté le : 20 Avril 2024

[46] Margaret Rouse (2 Août 2019) «Scikit-Learn» https ://www.techopedia.com/definition/33860/scikit-learn. Consulté le : 20 Avril 2024

[47] Damien Martin (26 Mai 2019) «Introducing the column transformer» https ://kiwidamien.github.io/introducing-the-column-transformer.html. Consulté le : 21 Avril 2024

[48] Wojtek Fulmyk (25 Juillet 2023) «Ordinal Encoding — A Brief Explanation» https ://medium.com/@WojtekFulmyk/ordinal-encoding-a-brief-explanation-a29cf374dbc1. Consulté le : 21 Avril 2024

[49] Geeksforgeeks (28 Septembre 2021) «ML | Handle Missing Data with Simple Imputer» https ://www.geeksforgeeks.org/ml-handle-missing-data-with-simple-imputer/. Consulté le : 21 Avril 2024

[50] Geeksforgeeks (24 Avril 2023) «StandardScaler, MinMaxScaler and RobustScaler techniques – ML» https ://www.geeksforgeeks.org/standardscaler-minmaxscaler-and-robustscaler-techniques-ml/. Consulté le : 21 Avril 2024