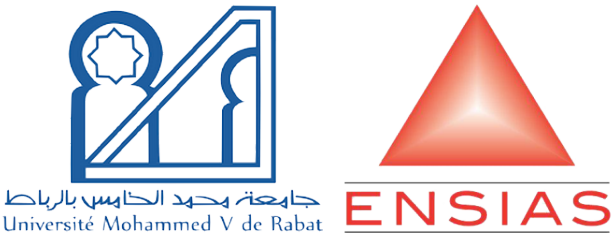


MOHAMMED V UNIVERSITY
ECOLE NATIONALE D'INFORMATIQUE ET D 'ANALYSE DES SYSTÈMES- ENSIAS
INTELLIGENCE ARTIFICIELLE (2IA)

Exploration des Intrusions Réseau

Réalisé par
Salma GUANTOURI

Encadré par
Prof. Mohamed LAZAAR

December 26, 2023

Abstract

In this data engineering project, we aimed to gain a deeper understanding of The exploration of network intrusion data encompasses a comprehensive analysis journey characterized by data visualization, cleansing, feature selection, transformation, and addressing imbalanced data. Leveraging sophisticated visualization tools like Matplotlib and Seaborn, this study unravels intricate trends, anomalies, and patterns within network traffic data, illuminating nuances critical for intrusion detection.

Through meticulous data cleaning, missing values and errors are rectified, ensuring the dataset's integrity and reliability.

Employing dimensionality reduction techniques such as PCA and UMAP aids in reducing feature space while preserving essential information.

Feature selection and transformation techniques further refine the dataset, ensuring its readiness for analysis. Notably, handling imbalanced data by employing oversampling, undersampling, or class weighting techniques is pivotal, impacting model performance. This multidimensional approach aims to enhance our comprehension of network intrusions and fortify intrusion detection systems.

Keywords: intrusion, data cleaning, dimensionality reduction techniques

Résumé

L'exploration des données d'intrusion réseau englobe un parcours d'analyse complet caractérisé par la visualisation des données, leur nettoyage, la sélection des caractéristiques, les transformations et la résolution des problèmes de déséquilibre. En utilisant des outils de visualisation sophistiqués tels que Matplotlib et Seaborn, cette étude dévoile les tendances, anomalies et motifs complexes présents dans les données de trafic réseau, mettant en lumière des nuances cruciales pour la détection des intrusions.

Le nettoyage méticuleux des données, corrigeant les valeurs manquantes et les erreurs, assure l'intégrité et la fiabilité de l'ensemble de données.

L'utilisation de techniques de réduction de dimensionnalité telles que PCA et UMAP contribue à réduire l'espace des caractéristiques tout en préservant les informations essentielles.

La sélection des caractéristiques et les transformations affinent davantage l'ensemble de données, assurant sa préparation pour l'analyse. En particulier, la gestion des données déséquilibrées grâce à des techniques de suréchantillonnage, de sous-échantillonnage ou de pondération des classes est cruciale et impacte les performances des modèles. Cette approche multidimensionnelle vise à améliorer notre compréhension des intrusions réseau et à renforcer les systèmes de détection des intrusions.

Mots-clés: intrusion, nettoyage des données, techniques de réduction de dimensionnalité

Liste des abréviations

AI : Artificial Intelligence
DR : Dimensionality Reduction
SMOTE: Synthetic Minority Over-sampling Technique
ML : Machine Learning
PCA : Principal Component Analysis

Contents

List of Figures

Chapter 1

Introduction générale

Ce chapitre introduit les objectifs de notre exploration de données pour prédire les intrusions. Notre objectif est de comprendre plus en détail le contexte et les facteurs qui influencent les intrusions et comment les prédire. Nous aborderons également la méthodologie et l'approche utilisées pour atteindre nos objectifs. Les résultats de cette exploration fourniront un aperçu des facteurs qui impactent les intrusions et comment les prédire plus efficacement.

1.1 Objectif

L'objectif principal de ce rapport est d'appliquer les techniques apprises dans le cours d'Ingénierie des Données (S3) sur des données réelles :

- Exploration de données
- Visualisation des données
- Nettoyage des données
- Réduction de la dimensionnalité

L'analyse de données est une composante essentielle de l'ingénierie en Intelligence Artificielle (IA). Elle est utilisée pour identifier les schémas et les tendances dans de vastes ensembles de données, fournissant des informations précieuses pour le développement de systèmes d'IA. L'analyse de données permet de découvrir des informations utiles pour concevoir des systèmes d'IA plus intelligents, plus efficaces et précis.

À la fin de ce rapport, nous utiliserons les données que nous avons analysées et nettoyées pour entraîner plusieurs classificateurs avant et après la réduction de leurs dimensions, afin de mettre en évidence l'impact de la réduction dimensionnelle sur l'efficacité du modèle et sa précision.

1.2 Context

Pour notre étude, nous avons opté pour l'analyse d'un ensemble de données brutes liées aux intrusions en sécurité informatique et à leurs relations avec divers facteurs. Ces données contiennent des informations sur des incidents d'intrusion, des attaques, et les

caractéristiques associées. Chaque incident est identifié par un "ID d'incident", et des détails tels que le type d'attaque, la méthode d'intrusion, les paramètres du réseau, les informations sur les paquets, les protocoles utilisés, etc., sont enregistrés.

Il serait également pertinent de définir certains termes clés dans notre contexte :

- **Type d'attaque :** cela désigne le mode d'intrusion spécifique utilisé lors d'une attaque, par exemple, les attaques par déni de service (DDoS), les injections SQL, les attaques par force brute, etc.
- **Méthode d'intrusion :** cela indique comment l'attaque a été menée, comme l'exploitation de vulnérabilités connues, l'utilisation de logiciels malveillants, l'ingénierie sociale, etc.
- **Paramètres du réseau :** ces informations concernent les détails du réseau touché par l'attaque, tels que les ports, les adresses IP source et destination, le nombre de paquets échangés, etc.

L'objectif est de comprendre les différents incidents d'intrusion, d'identifier les modèles et les facteurs prédictifs associés, afin de mieux appréhender la sécurité informatique et de développer des systèmes de détection plus efficaces et précis.

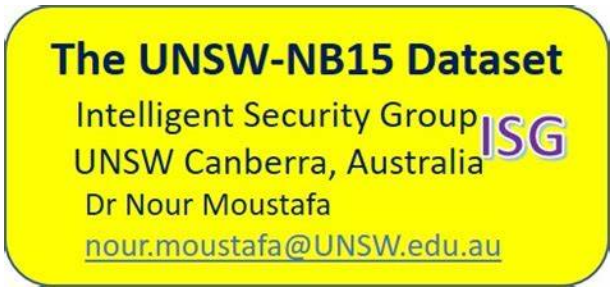


Figure 1.1: Dataset

1.3 Approche et méthodologie

Dans ce projet, nous avons adopté une approche méthodique pour analyser notre jeu de données portant sur les intrusions en sécurité informatique et pour en extraire des informations significatives. Notre méthodologie s'est déroulée selon les étapes suivantes :

- **Chargement et exploration des données :** Nous avons importé le jeu de données et exploré ses caractéristiques de base, telles que le nombre d'enregistrements, le nom et le nombre de fonctionnalités, les données manquantes, ainsi que des statistiques descriptives pour chaque fonctionnalité.
- **Visualisation des données :** Nous avons utilisé une variété de techniques visuelles telles que des histogrammes et des graphiques en boîte pour comprendre la distribution des caractéristiques et détecter d'éventuelles anomalies ou valeurs aberrantes.
- **Nettoyage des données :** Notre processus de nettoyage a impliqué la gestion des valeurs manquantes, l'identification et la suppression des valeurs aberrantes, ainsi que l'encodage des données catégorielles pour préparer le jeu de données à l'analyse.

- **Réduction de la dimensionnalité :** À l'aide de méthodes telles que l'ACP, nous avons réduit la dimensionnalité du jeu de données pour en simplifier la structure tout en conservant au mieux son intégrité.

Cette approche méthodique nous a permis de comprendre les facteurs clés liés aux intrusions en sécurité informatique et de présenter nos découvertes de manière claire et concise. La méthodologie appliquée dans ce projet pourrait servir de référence pour des études similaires dans le domaine de la cybersécurité

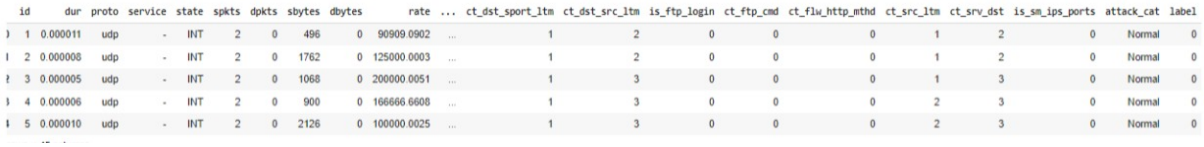
Chapter 2

Dataset

Dans ce chapitre, notre objectif est d'explorer notre jeu de données, de le décrire, de le visualiser pour en acquérir une meilleure compréhension, de le nettoyer des valeurs aberrantes, des données manquantes et des doublons, et enfin de supprimer les colonnes inutiles tout en transformant toutes les valeurs catégorielles en valeurs numériques exploitables pour nos tâches de machine learning ultérieures.

2.1 Data description

Nous commençons par charger notre jeu de données depuis `Sleep_Efficiency.csv`dans un dataframe pandas. Nous pouvons voir les 5 premières lignes de notre jeu de données en utilisant la fonction `head()`



	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd	ct_flw_http_mthd	ct_src_ltm	ct_srv_dst	is_sm_ips_ports	attack_cat	label
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...	1	2	0	0	0	1	2	0	Normal	0
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...	1	2	0	0	0	1	2	0	Normal	0
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...	1	3	0	0	0	1	3	0	Normal	0
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...	1	3	0	0	0	2	3	0	Normal	0
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...	1	3	0	0	0	2	3	0	Normal	0

Figure 2.1: Les 5 premières lignes de l'ensemble de données en utilisant la fonction head()

Nous sommes d'abord curieux de connaître les dimensions de notre ensemble de données. Nous le découvrons en utilisant l'attribut shape du dataframe:

```
(257673, 45)
```

Listing 2.1: shape of the dataset

L'ensemble de données est composé de 257 673 lignes et 45 colonnes que nous développerons quelques un dans la description suivante :

- **"id"**: Identifiant unique pour chaque enregistrement dans l'ensemble de données.
- **"dur"**: Durée du flux de données.
- **"proto"**: Protocole réseau utilisé (TCP, UDP, etc.).

- **"service"**: Type de service ou de port utilisé.
- **"state"**:État de la connexion réseau (établi, en cours de connexion, etc.).
- **"spkts"**: Nombre de paquets envoyés
- **"dpkts"**: Nombre de paquets reçus.
- **"sbytes et dbytes"**: Nombre d'octets envoyés et reçus respectivement.
- **"rate"**:Taux de transmission.
- **"ct-srv-src :"**:Compte du nombre de connexions du même service source pour la même adresse IP source.
- **"ct-dsty-ltm"**: Compte du nombre de connexions vers la même adresse IP de destination.
- **"attack-cat"**: Catégorie d'attaque (si présent) ou étiquette de classification.
- **"label"**: Étiquette indiquant si le flux est normal ou une attaque (1 pour attaque, 0 pour normal).

En utilisant la méthode `info()`, nous pouvons obtenir davantage de détails typiques sur chaque colonne comme suit :

```
    <class 'pandas.core.frame.DataFrame'>
RangeIndex: 257673 entries, 0 to 257672
Data columns (total 45 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               257673 non-null int64
 1   dur              257673 non-null float64
 2   proto            257673 non-null object
 3   service          257673 non-null object
 4   state            257673 non-null object
 5   spkts            257673 non-null int64
 6   dpkts            257673 non-null int64
 7   sbytes           257673 non-null int64
 8   dbytes           257673 non-null int64
 9   rate             257673 non-null float64
 10  sttl             257673 non-null int64
 11  dttl             257673 non-null int64
 12  sload            257673 non-null float64
 13  dload            257673 non-null float64
 14  sloss            257673 non-null int64
 15  dloss            257673 non-null int64
 16  sinpkt           257673 non-null float64
 17  dinpkt           257673 non-null float64
 18  sjit             257673 non-null float64
 19  djit             257673 non-null float64
 20  swin             257673 non-null int64
 21  stcpb            257673 non-null int64
 22  dtcpb            257673 non-null int64
 23  dwin             257673 non-null int64
```

```
 24  tcprtt             257673 non-null float64
 25  synack             257673 non-null float64
 26  ackdat             257673 non-null float64
 27  smean              257673 non-null int64
 28  dmean              257673 non-null int64
 29  trans_depth        257673 non-null int64
 30  response_body_len  257673 non-null int64
 31  ct_srv_src         257673 non-null int64
 32  ct_state_ttl       257673 non-null int64
 33  ct_dst_ltm         257673 non-null int64
 34  ct_src_dport_ltm   257673 non-null int64
 35  ct_dst_sport_ltm   257673 non-null int64
 36  ct_dst_src_ltm     257673 non-null int64
 37  is_ftp_login       257673 non-null int64
 38  ct_ftp_cmd         257673 non-null int64
 39  ct_flw_http_mthd   257673 non-null int64
 40  ct_src_ltm         257673 non-null int64
 41  ct_srv_dst         257673 non-null int64
 42  is_sm_ips_ports    257673 non-null int64
 43  attack_cat         257673 non-null object
 44  label              257673 non-null int64
dtypes: float64(11), int64(30), object(4)
memory usage: 88.5+ MB
```

Listing 2.2: informations sur le datset

Une caractéristique notable de notre ensemble de données est l'absence de valeurs nulles. Cette particularité souligne la qualité des données fournies, offrant une base solide pour une analyse approfondie sans la nécessité de gérer des valeurs manquantes. Pour montrer cela plus clairement, nous utilisons les méthodes `isnull().sum()` :

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	sinpkt	dinpkt	sjit	dji
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 2.2: le nombre des valeurs manquantes par colonne

Maintenant, nous sommes curieux d'en savoir plus sur la distribution statistique de nos données. Dans ce but, nous utilisons la méthode describe() qui nous donne les résultats suivants :

Ces informations peuvent nous donner des perspectives supplémentaires sur les colonnes numériques. Ainsi, nous pouvons identifier les caractéristiques susceptibles de contenir des valeurs aberrantes, ce qui indique quelles caractéristiques doivent être traitées ultérieurement. Enfin, nous pouvons également vérifier s'il y a des valeurs en double dans notre ensemble de données en utilisant la méthode`duplicate()` . Cela montre que notre ensemble de données ne contient aucun doublon.

	id	dur	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	...	ct_sr
count	257673.000000	257673.000000	257673.000000	257673.000000	2.576730e+05	2.576730e+05	2.576730e+05	257673.000000	257673.000000	2.576730e+05	...	25
mean	72811.823858	1.246715	19.777144	18.514703	8.572952e+03	1.438729e+04	9.125391e+04	180.000931	84.754957	7.060869e+07	...	
std	48929.917641	5.974305	135.947152	111.985965	1.737739e+05	1.461993e+05	1.603446e+05	102.488268	112.762131	1.857313e+08	...	
min	1.000000	0.000000	1.000000	0.000000	2.400000e+01	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000e+00	...	
25%	32210.000000	0.000008	2.000000	0.000000	1.140000e+02	0.000000e+00	3.078928e+01	62.000000	0.000000	1.231800e+04	...	
50%	64419.000000	0.004285	4.000000	2.000000	5.280000e+02	1.780000e+02	2.955665e+03	254.000000	29.000000	7.439423e+05	...	
75%	110923.000000	0.685777	12.000000	10.000000	1.362000e+03	1.064000e+03	1.250000e+05	254.000000	252.000000	8.000000e+07	...	
max	175341.000000	59.999989	10646.000000	11018.000000	1.435577e+07	1.465753e+07	1.000000e+06	255.000000	254.000000	5.988000e+09	...	

Figure 2.3: Description statistique des colonnes numériques de l'ensemble de données.

2.2 Visualisation des Données

Dans cette section, nous présenterons nos données à l'aide de différents types de graphiques pour mieux les comprendre. Plus précisément, nous utiliserons :

- **bar plot**: une représentation graphique des données utilisant des barres de hauteurs différentes pour représenter les valeurs. Il est utile pour comparer différentes catégories ou distributions de données.
- **histogram**: un type de diagramme en barres qui montre la fréquence des différentes valeurs dans un ensemble de données. Il est utile pour explorer la distribution des données et identifier les valeurs aberrantes.
- **box plot**:a graphical representation of data that displays the median, quartiles, and extremes of the data. It is useful for identifying outliers and summarizing the distribution of data.
- **heat map**:une représentation graphique bidimensionnelle des données où les valeurs sont représentées par des couleurs. Il est utile pour visualiser les tendances, les motifs et les corrélations dans de grands ensembles de données.

Mais avant cela, nous supprimons d'abord la colonne "ID" car elle ne contient aucune information pertinente pour l'analyse de l'ensemble de données.

2.2.1 Histogramme des données

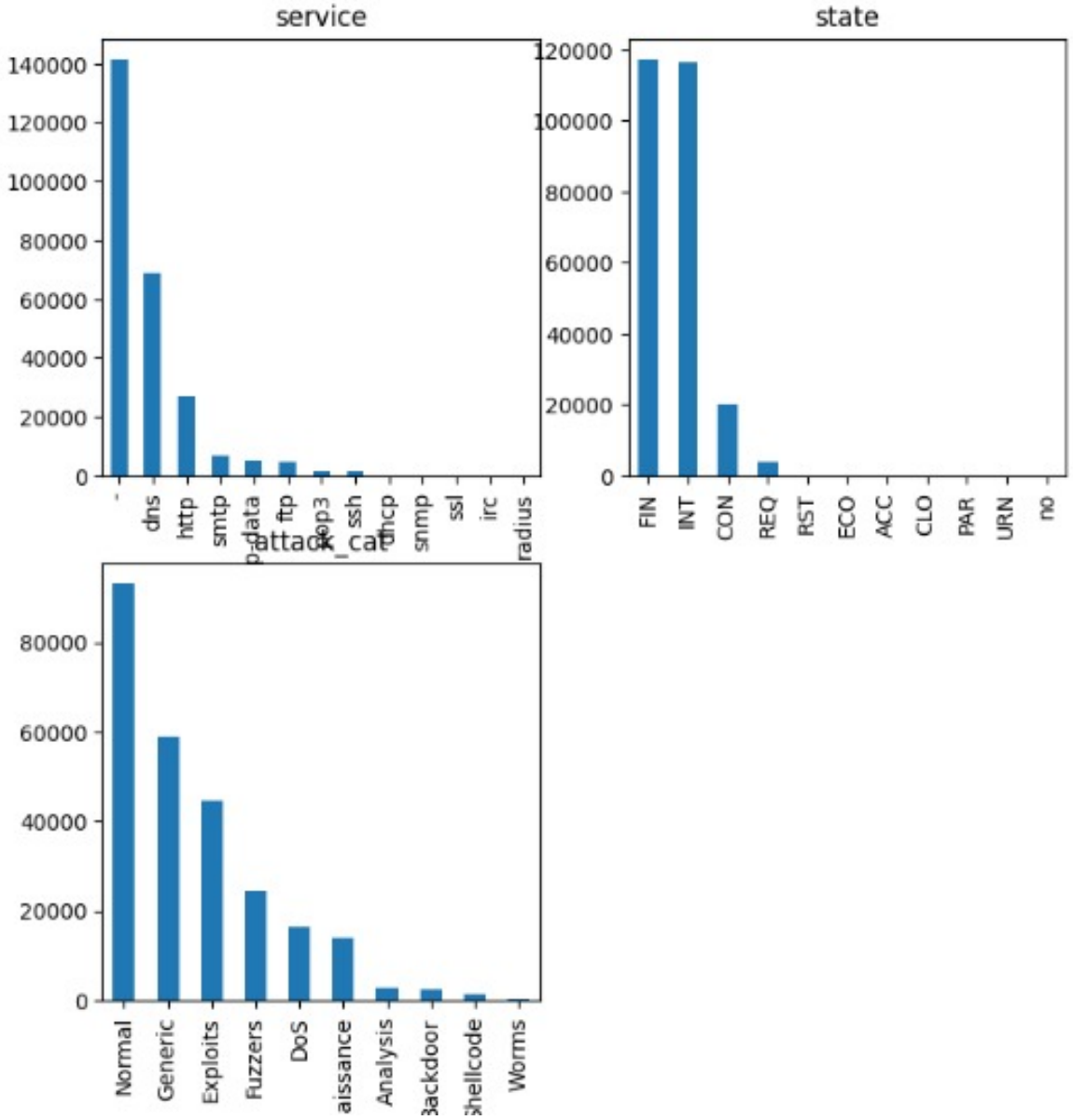


Figure 2.4: histogramme pour les variables catégorielles

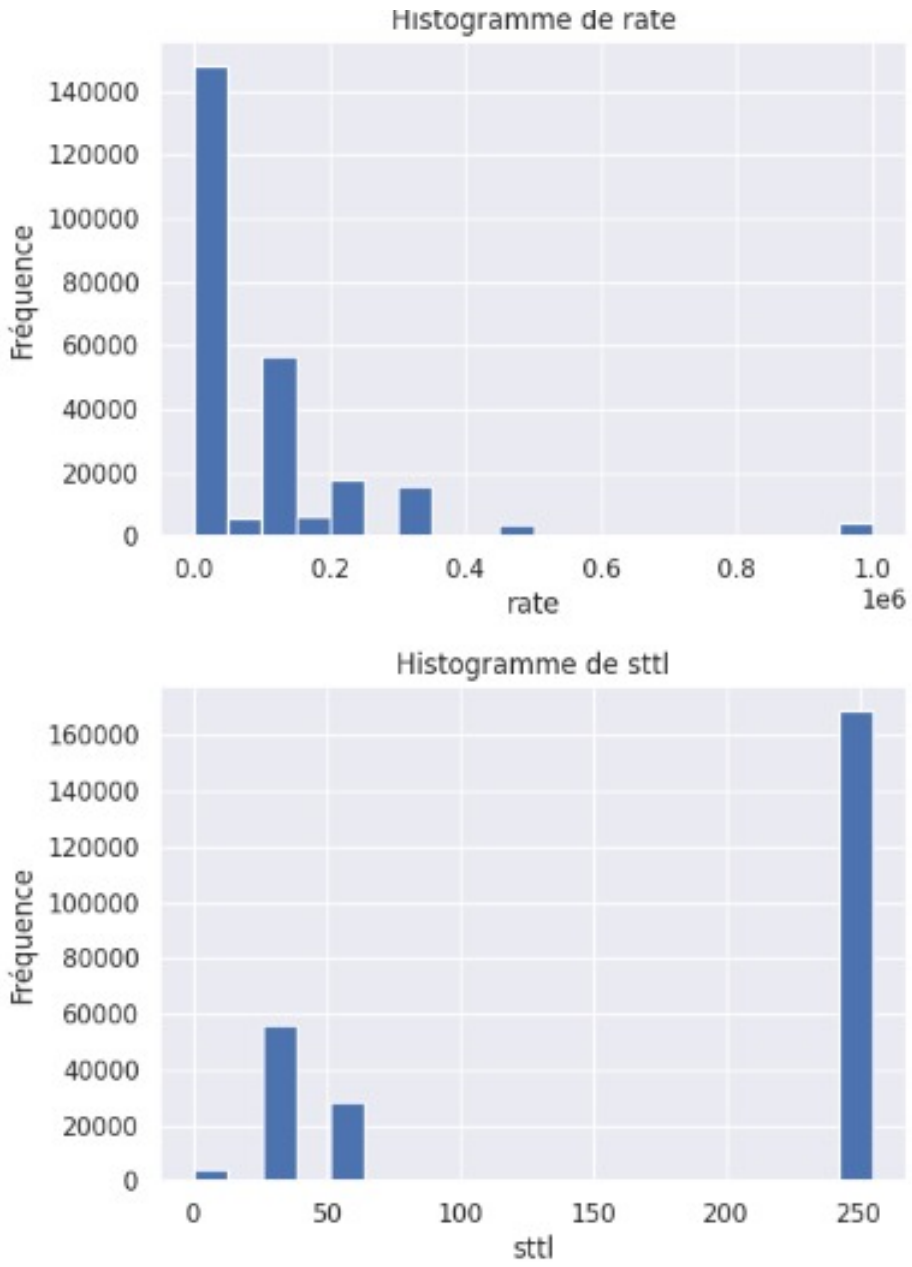


Figure 2.5: histogramme pour les variables numériques

2.2.2 Box plot

Lors de l'analyse des boîtes à moustaches de nos variables, il est apparu clairement que certaines d'entre elles ne présentaient pas une répartition équilibrée de leurs valeurs.

Cette observation souligne des disparités significatives dans la distribution des données pour ces caractéristiques spécifiques. Ces déséquilibres peuvent provenir de diverses causes, telles que la présence d'outliers, des valeurs aberrantes, ou des différences notables de concentration autour de la médiane.

Identifier ces variations nous permet de mieux comprendre l'hétérogénéité des données et peut nécessiter des ajustements supplémentaires dans le processus d'analyse pour garantir des résultats fiables et représentatifs.

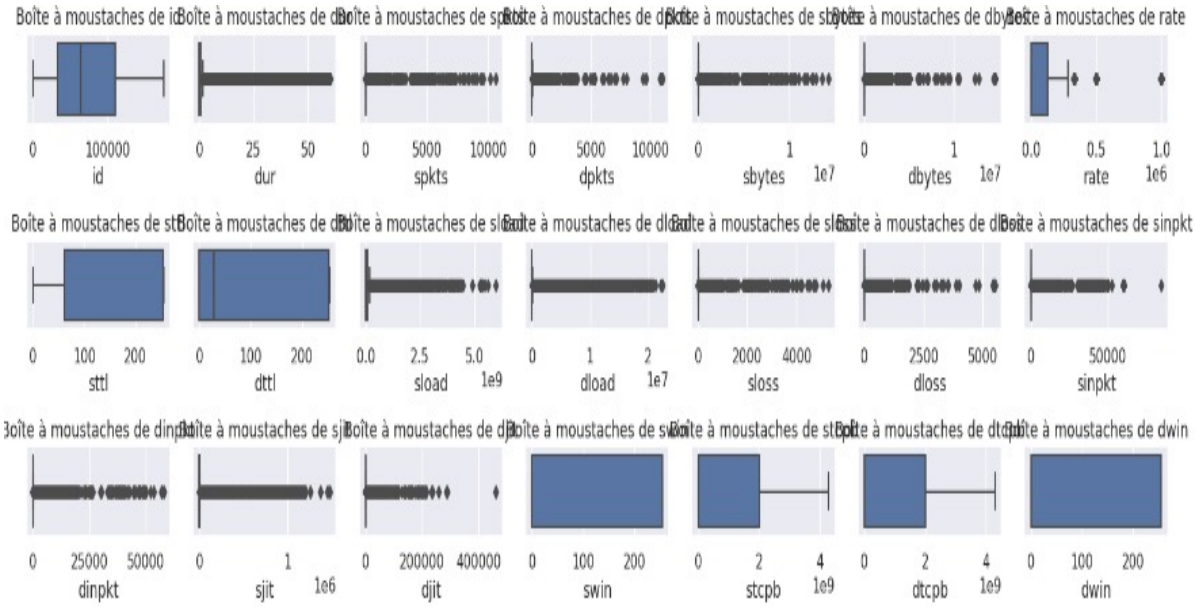


Figure 2.6: Diagramme en boîte des caractéristiques dans le jeu de données

2.2.3 Carte thermique et matrice de corrélation

La matrice de corrélation est une matrice carrée et symétrique qui résume la relation de corrélation entre chaque paire de caractéristiques dans le jeu de données. Formellement, la matrice de corrélation COR est définie comme suit :

$$COR_{i,j} = cor(col_i, col_j)$$

où i et j sont deux indices dans la plage du nombre de caractéristiques.

Pour faciliter la visualisation, nous utilisons une carte thermique pour représenter graphiquement la matrice de corrélation. Nous obtenons le résultat suivant :

Une analyse approfondie de la matrice de corrélation des caractéristiques de notre ensemble de données UNSW-NB-15 a révélé des relations intéressantes.

Plus particulièrement, des corrélations significatives ont été observées entre certaines caractéristiques telles que '**label**', '**sttl**', '**rate**', et '**state**'. La forte corrélation entre ces variables suggère une relation potentielle entre elles.

Cela peut indiquer que ces caractéristiques pourraient avoir une influence importante sur la variable cible, '**label**', utilisée pour identifier les attaques dans notre ensemble de données. Cependant, il est important de noter que la corrélation n'implique pas nécessairement la causalité.

Malgré ces corrélations, il est crucial de mener davantage d'analyses pour comprendre la nature exacte de ces relations et leur impact sur nos tâches de modélisation et de prédiction

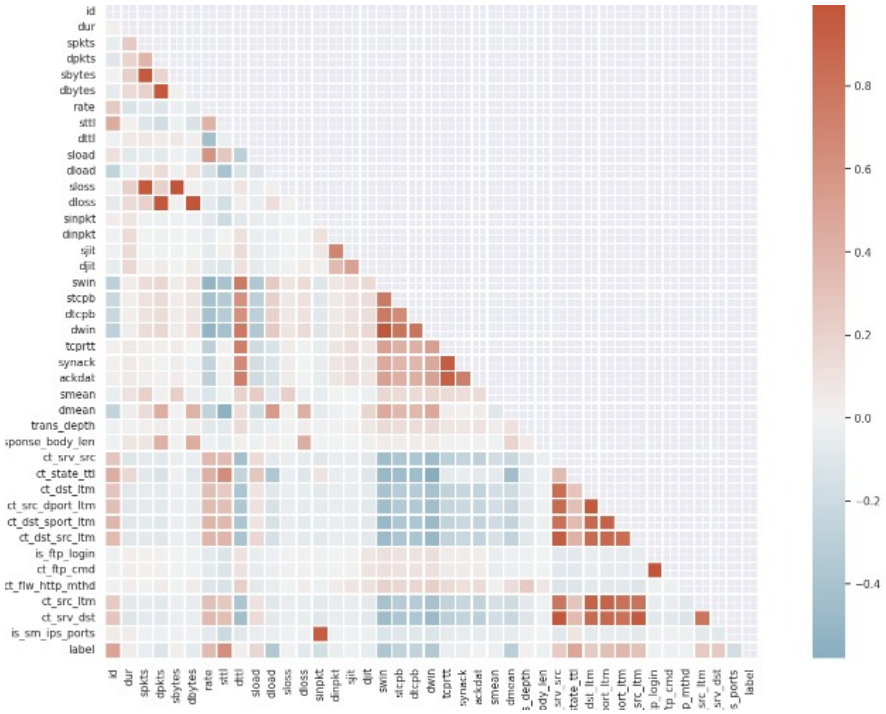


Figure 2.7: matrice de corrélation du dataset

2.3 Nettoyage des données

Dans cette section, nous allons utiliser plusieurs méthodes afin de :

- Éliminer les valeurs manquantes
- Standarisation des données

2.3.1 Traitement des valeurs manquantes

Il existe diverses approches pour traiter les valeurs manquantes dans un ensemble de données, notamment l'imputation, la suppression des lignes ou des colonnes concernées, ou encore l'utilisation de modèles spécifiques.

Cependant, dans notre cas, l'exploration initiale de notre ensemble de données ne révèle aucune valeur manquante. Cela est souvent rare, mais dans ce contexte, notre dataset semble être complet, ce qui élimine le besoin de recourir à des méthodes de gestion de valeurs manquantes.

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	sinpkt	dinpkt	sjit	dji
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 2.8: le nombre des valeurs manquantes par colonne

2.3.2 Stadarisation des données

Après avoir exploré et compris la distribution des données, il est souvent crucial d'appliquer des techniques de standardisation.

Cette étape vise à mettre les différentes caractéristiques sur la même échelle, éliminant ainsi les disparités de variance entre elles. La standardisation permet de rendre les données plus comparables, ce qui est essentiel pour de nombreuses analyses statistiques et algorithmes d'apprentissage automatique.

Cela garantit que chaque caractéristique contribue de manière égale à l'analyse, évitant que certaines ne dominent les autres en raison de leur échelle initiale.

spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	sinpkt	dinpkt	sjit
-0.130765	-0.165331	-0.046480	-0.098409	-0.002151	0.722026	-0.751628	0.590935	-0.272850	-0.074561	-0.125576	-0.131793	-0.090412	-0.110522
-0.130765	-0.165331	-0.039194	-0.098409	0.210460	0.722026	-0.751628	4.363255	-0.272850	-0.074561	-0.125576	-0.131793	-0.090412	-0.110522
-0.130765	-0.165331	-0.043188	-0.098409	0.678204	0.722026	-0.751628	4.220037	-0.272850	-0.074561	-0.125576	-0.131794	-0.090412	-0.110522
-0.130765	-0.165331	-0.044155	-0.098409	0.470318	0.722026	-0.751628	2.850314	-0.272850	-0.074561	-0.125576	-0.131794	-0.090412	-0.110522
-0.130765	-0.165331	-0.037100	-0.098409	0.054546	0.722026	-0.751628	4.198501	-0.272850	-0.074561	-0.125576	-0.131793	-0.090412	-0.110522
...	...	...	...	...	...	...	...	...	...	...	...	...	...
-0.130765	-0.165331	-0.048678	-0.098409	0.123841	0.722026	-0.751628	-0.107371	-0.272850	-0.074561	-0.125576	-0.131793	-0.090412	-0.110522
-0.071919	-0.093893	-0.045766	-0.095988	-0.568903	0.722026	1.483170	-0.380119	-0.270817	-0.044061	-0.106955	-0.123936	-0.029190	-0.034635
-0.130765	-0.165331	-0.048678	-0.098409	0.123841	0.722026	-0.751628	-0.107371	-0.272850	-0.074561	-0.125576	-0.131793	-0.090412	-0.110522
-0.130765	-0.165331	-0.048678	-0.098409	0.123841	0.722026	-0.751628	-0.107371	-0.272850	-0.074561	-0.125576	-0.131793	-0.090412	-0.110522
-0.130765	-0.165331	-0.048678	-0.098409	0.123841	0.722026	-0.751628	-0.107371	-0.272850	-0.074561	-0.125576	-0.131793	-0.090412	-0.110522

Figure 2.9: les données après standarisation

Chapter 3

Réduction de la Dimensionnalité

In this chapter, we will dig into the topic of dimensionality reduction and its role in our analysis of the sleep efficiency dataset. We will focus on one of the most commonly used dimensionality reduction techniques, feature elimination techniques (Correlation and Mutual information) and the Principal Component Analysis (PCA), and their application in our analysis. Additionally, we will discuss the use of regression machine learning models in combination with PCA to evaluate the impact of dimensionality reduction on the performance of the models.

3.1 Élimination de Caractéristiques

Une approche pour réduire la dimensionnalité des bases de données consiste à éliminer les caractéristiques ayant un faible impact sur la variable cible tout en ne conservant que les plus importantes. Cela se fait **sans apporter de modifications supplémentaires aux données.** Dans cette catégorie, nous présenterons 2 méthodes de réduction de la dimensionnalité :

3.1.1 Seuillage de variance

la méthode de seuillage de variance est effectivement considérée comme une technique de sélection de caractéristiques (feature selection). Elle permet de supprimer les caractéristiques qui ont une variance inférieure à un certain seuil spécifié.

En éliminant les caractéristiques avec une faible variance, cette méthode cherche à réduire le bruit ou à éliminer les caractéristiques constantes qui pourraient ne pas contribuer significativement à l'apprentissage du modèle. Cependant, il est important de noter que cette méthode ne prend pas en compte la relation entre les caractéristiques et la variable cible. Elle se concentre uniquement sur la variance de chaque caractéristique individuelle

```
(257673, 22)
```

Listing 3.1: la nouvelle taille du dataset

3.1.2 Analyse de l'importance des caractéristiques avec l'algorithme RandomForest pour la sélection de caractéristiques

L'algorithme RandomForest est souvent utilisé pour estimer l'importance des caractéristiques dans un modèle prédictif.

Cette méthode évalue l'influence de chaque caractéristique sur la précision du modèle. Elle fonctionne en mesurant comment les performances du modèle diminuent lorsque les valeurs d'une caractéristique sont aléatoirement mélangées **(permutation)**. Si la performance du modèle chute considérablement lorsque cette caractéristique est modifiée, elle est considérée comme importante pour la prédiction.

Cette évaluation est répétée pour toutes les caractéristiques, fournissant une mesure relative de leur importance. En visualisant ces importances, on peut identifier les caractéristiques les plus influentes pour la tâche de prédiction, ce qui peut orienter la sélection des caractéristiques pour améliorer les performances du modèle.

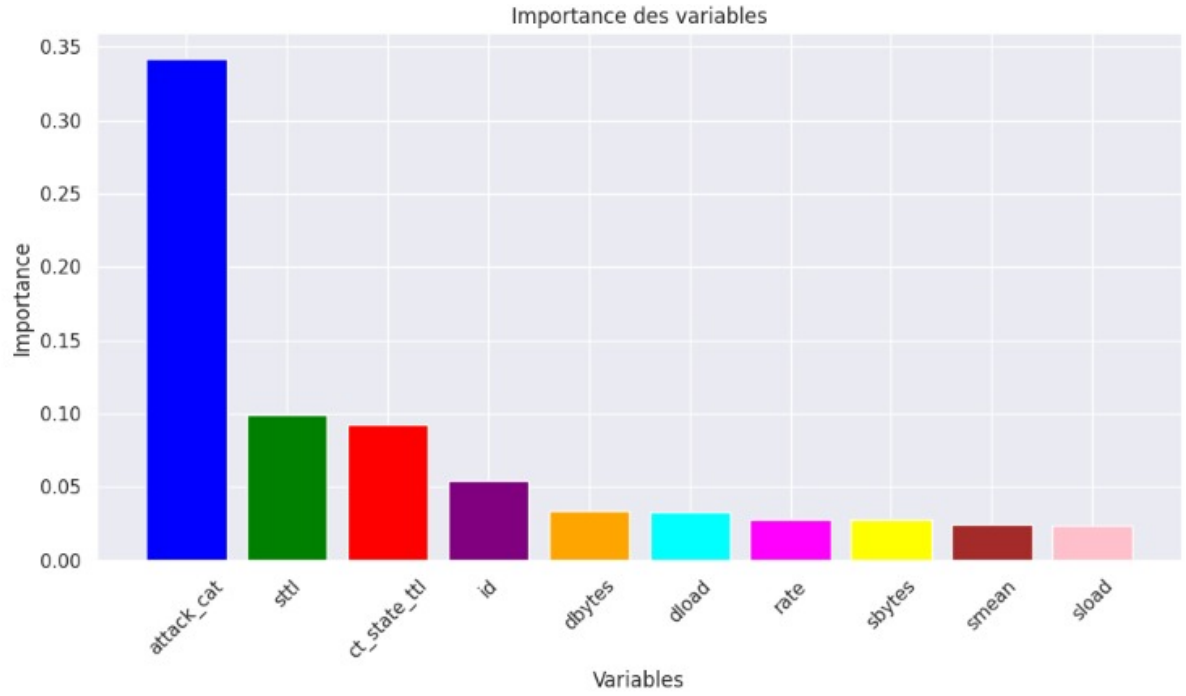


Figure 3.1: l'importance des dix premieres variables

3.2 transformation linéaire des caractéristiques

D'autre part, d'autres méthodes de réduction de dimensionnalité se basent sur **la transformation des données vers de nouveaux espaces vectoriels par projection**. Dans notre rapport, nous ne considérerons que les méthodes utilisant des transformations linéaires, plus spécifiquement : l'Analyse en Composantes Principales (ACP).

3.2.1 Introduction

PCA est une méthode statistique visant à transformer un ensemble de données à haute dimension en un espace de dimension inférieure tout en préservant autant de variance

que possible dans les données. En réduisant le nombre de caractéristiques de l'ensemble de données, l'ACP peut contribuer à simplifier l'analyse, améliorer les performances des modèles d'apprentissage automatique et identifier des schémas et des relations au sein des données.

3.2.2 Principle

Le principe fondamental de l'**ACP** est de trouver de nouvelles variables (appelées composantes principales) qui sont des combinaisons linéaires des variables d'origine. Ces nouvelles composantes captent le maximum de variance possible présente dans les données initiales.

En réduisant la dimension de l'espace des caractéristiques, l'ACP permet de simplifier l'analyse tout en préservant autant que possible les relations et structures importantes entre les données.

3.2.3 Application de PCA

Maintenant, nous appliquons la méthode PCA à notre ensemble de données. Tout d'abord, nous normalisons notre ensemble de données en utilisant la méthode StandardScaler de la bibliothèque scikit-learn.

Après avoir appliqué l'algorithme PCA sur notre ensemble de données en utilisant 70 comme ratio d'information que nous souhaitons conserver

```
1    Variance conserve : 0.7, Accuracy : 0.9931502862132531
2    Variance conserve : 0.9, Accuracy : 0.999010381294266
```

Listing 3.2: Accuracy pour deux valeurs de p

Il semblerait que conserver 0.7 ou 0.9 de l'information à l'aide de la méthode PCA présente des résultats d'accuracy similaires. Ainsi, pour optimiser les ressources et simplifier davantage nos données, nous optons pour conserver uniquement 0.7 de l'information. Cela nous permettra de réduire le nombre de caractéristiques à seulement 6, tout en maintenant des performances satisfaisantes en termes d'accuracy pour notre modèle.

Cette décision est fondée sur la balance entre la complexité des données et les performances du modèle, assurant une efficacité optimale sans compromettre la précision de nos résultats.

Chapter 4

Données Déséquilibrées

Suite à l'exploration visuelle de notre jeu de données, il est devenu évident que celui-ci présente une distribution déséquilibrée de ses classes.

Cette disparité entre les différentes catégories pourrait introduire un biais significatif dans notre modèle. Ainsi, afin de garantir des prédictions précises et équitables, nous allons mettre en œuvre des techniques d'équilibrage des données.

Ces méthodes visent à harmoniser la représentation de chaque classe, permettant ainsi au modèle d'apprendre de manière plus équilibrée et de fournir des résultats plus fiables.

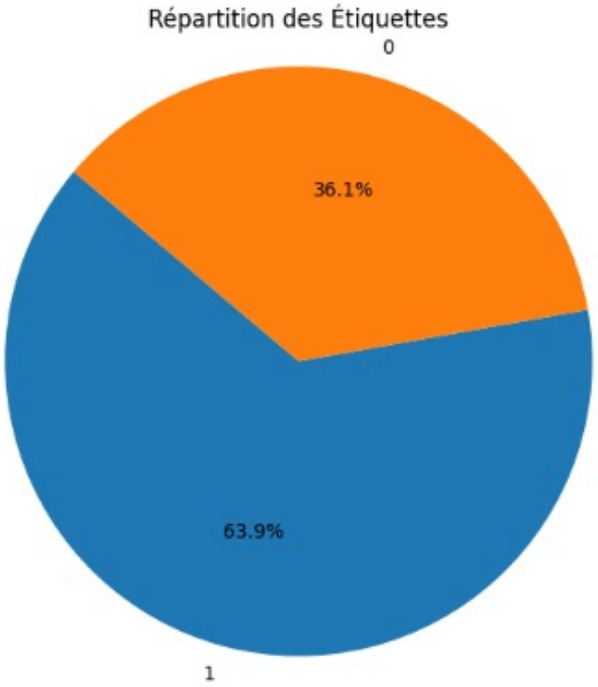


Figure 4.1: Data-set déséquilibrée

4.0.1 Suréchantillonage:SMOTE

SMOTE est une technique de suréchantillonnage utilisée pour équilibrer les classes dans un ensemble de données déséquilibré. Plutôt que de dupliquer simplement des exemples de la classe minoritaire, SMOTE crée de nouveaux exemples synthétiques en prenant des exemples existants et en créant des versions pondérées de ces exemples.

Voici comment SMOTE fonctionne :

1. **Sélection des exemples de la classe minoritaire** : SMOTE sélectionne un exemple de la classe minoritaire.
2. **Calcul des k plus proches voisins** : En utilisant une mesure de distance (souvent la distance euclidienne), SMOTE identifie les k plus proches voisins de cet exemple au sein de la classe minoritaire.
3. **Génération d'exemples synthétiques** : Pour chaque exemple sélectionné, SMOTE crée de nouveaux exemples synthétiques en prenant des combinaisons linéaires des valeurs des caractéristiques des exemples sélectionnés et en les ajoutant à l'ensemble de données.
4. **Répéter le processus** : Ce processus est répété jusqu'à ce que le nombre désiré d'exemples de la classe minoritaire soit généré.

SMOTE aide à atténuer le déséquilibre de classe en créant de nouveaux exemples plutôt qu'en dupliquant simplement ceux existants, ce qui peut aider les modèles à mieux généraliser sans surajuster aux données d'entraînement.

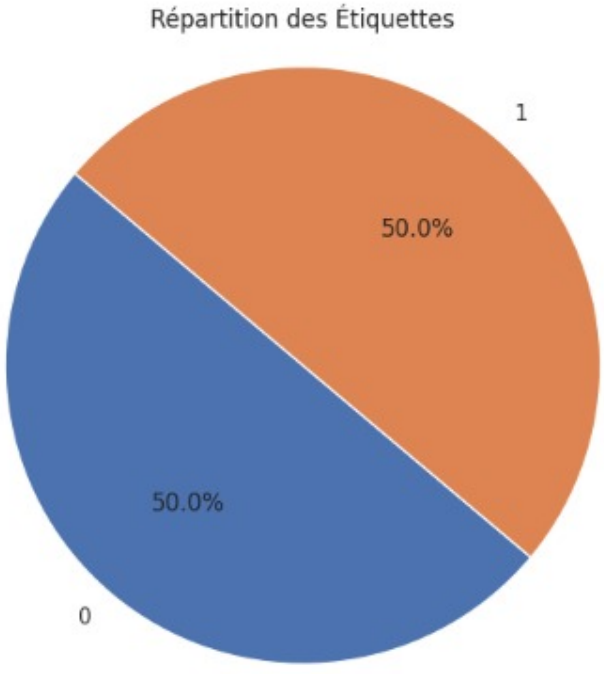


Figure 4.2: Data-set après l'application de la méthode SMOTE

4.0.2 Méthode de pondération

Pondération des Classes: Dans le contexte de l'apprentissage automatique, les poids sont souvent attribués à différentes classes dans un jeu de données. Par exemple, dans un problème de classification, si une classe est sous-représentée, on lui attribue un poids plus élevé.

Fonction de Perte Pondérée: Lors de l'entraînement du modèle, la fonction de perte (loss function) est ajustée pour prendre en compte ces poids. Cela signifie que les erreurs sur les classes minoritaires ont un impact plus important sur la fonction de perte, poussant ainsi le modèle à mieux apprendre ces classes.

Échantillonnage Pondéré: Une autre approche consiste à utiliser un échantillonnage pondéré pour sélectionner des échantillons lors de l'entraînement. Les instances de classes sous-représentées ont une plus grande probabilité d'être sélectionnées. Dans le contexte de

Rapport de classification après la pondération:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	133325
1	0.98	0.98	0.98	74813
accuracy			0.98	208138
macro avg	0.98	0.98	0.98	208138
weighted avg	0.98	0.98	0.98	208138

Figure 4.3: Lapplicatiob de la méthode de pondération au dataset

l'équilibrage des données, la pondération est un outil précieux pour atténuer les biais dus à des déséquilibres de classe. Elle permet de construire des modèles de machine learning plus robustes et équitables, capables de traiter de manière égale des données de différentes classes.

Cependant, comme toute technique, elle doit être appliquée avec soin pour éviter de nouveaux problèmes, comme le surajustement sur des classes spécifiques

Chapter 5

Conclusion générale

En conclusion, cette étude a abordé de manière systématique l'exploration, la visualisation et la préparation des données, ainsi que la réduction de la dimensionnalité pour traiter des données déséquilibrées.

L'analyse approfondie du dataset a permis de comprendre sa structure et d'identifier des méthodes efficaces pour gérer les valeurs manquantes et les outliers. En utilisant des techniques telles que le seuillage de variance et l'analyse de l'importance des caractéristiques avec l'algorithme RandomForest, nous avons pu réduire la dimensionnalité tout en préservant les informations essentielles.

De plus, en utilisant des méthodes de traitement des données déséquilibrées comme SMOTE et la méthode de pondération, nous avons pu améliorer la qualité des modèles prédictifs.

Ces approches ont ouvert la voie à des analyses plus précises et robustes pour notre étude sur les intrusions réseau, fournissant ainsi une base solide pour des travaux futurs dans ce domaine

Bibliography

1: Data Engineering and Dimensionality Reduction course. Fall 2023 - *Prof. Mohamed LAZAAR*

2: Machine Learning Theory. Fall 2023 course, *Prof. Abdellatif EL AFIA*

3: Data Analysis course. Fall 2023 - *Prof. Si Lhoussain AOURAGH*

6: The UNSW-NB15 Dataset , *url: https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo*