INSTITUT FRANCOPHONE INTERNATIONAL

TRAVAIL PERSONNEL ENCADRE

Rapport Final

# DEEP LEARNING APPLIQUE A LA CYBERSÉCURITÉ POUR LA DÉTECTION D'INTRUSION

**Promotion 23 RSC**
Année Académique 2018-2019


**Encadreurs :**
Dr. Dr. Nguyen Hong Quang
Dr. HO Tuong Vinh

**Rédigé par :**
TSHIBANGU MUABILA Jean

# Table des matières

# Table des figures

# Liste des tableaux

# INTRODUCTION GÉNÉRALE

Dans le but de développer le sens de l'autonomie, de l'initiative, de l'agencement d'un travail scientifique et du travail individuel aux étudiants, le module du Travail Personnel Encadré, en sigle TPE, a été introduit afin de démontrer notre potentialité en ce qui concerne un travail scientifique, d'approfondir les connaissances dans une perspective bien déterminée et de façon personnelle.
Dans le cadre de notre domaine, nous nous sommes focalisés sur une thématique dont l'intitulée est « Deep Learning appliqué à la cybersécurité pour la détection d'intrusion».
Nous tenterons dans cette partie théorique de montrer l'analyse, la recherche bibliographique et la solution proposée de notre travail.

**Résumé**

Avec la croissance exponentielle de la taille des réseaux informatiques et des applications développées, l'augmentation significative des dommages potentiels pouvant être causée par le lancement d'attaques devient évidente. Concurremment, les systèmes de détection d'intrusion (IDS) et les systèmes de prévention d'intrusion (IPS) sont des outils de détection et défenses les plus importantes contre les attaques réseau sophistiquées, et en croissance constante. Par conséquent, l'objectif de ce travail est d'appliquer une méthode de modélisation de données de pointe d'un système de détection d'intrusion pour prédire une détection d'intrusion dans un environnement réseau à l'aide de l'algorithme de l'apprentissage profond comme outil de prédiction de l'attaque. Pour ce faire, nous avons utilisé le jeu de données ISCX 2017 collecté par l'Institut canadien de cybersécurité. Cet ensemble de données contient sept flux de réseau d'attaques bénignes et communes, qui répond aux critères du monde réel et est publiquement disponible. Vous pouvez le trouver dans le lien https ://www.unb.ca/cic/datasets. L'ensemble de données d'origine comprend 1580215 observations collectées en cinq jours avec une diversité d'attaques, dont 225745 observations le dernier pour l'attaque DDoS, et 85 fonctionnalités. Nous échantillonnons les données au hasard 10 fois avec la validation croisée pour obtenir 10 sous-échantillons de données permettant de créer des modèles de prédiction utilisant l'algorithme d'apprentissage profond, avec une architecture séquentielle. Les modèles obtenus sont comparés à d'autres techniques proposés, pour évaluer les précisions sur la base de la matrice de confusion. La précision trouvée dans notre modèle est 97%. En conséquence, ce document évalue les performances d'un ensemble complet de fonctionnalités de trafic réseau avec l'algorithme d'apprentissage profond, afin de détecter l'attaque dans un réseau informatique.

**Abstract**

With the exponential growth in the size of computer networks and developed applications, the significant increase in potential damage that can be caused by launching attacks becomes evident. Concurrently, intrusion detection systems (IDS) and intrusion prevention systems (IPS) are the most important detection and defense tools against sophisticated and constantly growing network attacks. Therefore, the objective of this work is to apply an advanced data modeling method of an intrusion detection system to predict an intrusion detection in a network environment using the algorithm of deep learning as an attack prediction tool. To do this, we used the ISCX 2017 dataset collected by the Canadian Cybersecurity Institute. This dataset contains seven benign and common attack network flows, which meet real world criteria and are publicly available. You can find it in the link https ://www.unb.ca/cic/datasets. The original dataset includes 1,580,215 observations collected in five days with a variety of attacks, including 225,745 observations the last for the DDoS attack, and 85 features. We randomly sample the data 10 times with cross-validation to obtain 10 subsamples of data to create prediction models using the deep learning algorithm, with a sequential architecture. The models obtained are compared with other proposed techniques, to assess the details based on the confusion matrix. The accuracy found in our model is 97%. As a result, this document assesses the performance of a full set of network traffic features with the deep learning algorithm, in order to detect the attack in a computer network.

# Première partie

# Partie Théorique

# 1) ANALYSE DU SUJET

## 1.1 Contexte général

À l'heure actuelle, les informations à traiter évoluent d'une manière exponentielle, à la mesure où la protection de données est devenue plus complexe, à cause de l'évolution des menaces qui ne cessent d'accroître, la masse de données à traiter par les analystes sécurité devient immense. Avec ce grand nombre d'attaques connues et inconnues, les outils traditionnels tels que le SNORT, SURICATA, Tiger-2.2.4, Logcheck 1.3, Tripwire etc., soulignent des limites, du fait qu'ils sont basés sur les règles pour signaler un comportement inhabituel dans un réseau et/ou système informatique.

Toutes fois, avec les menaces inconnues, l'outil traditionnel sera incapable d'identifier les intrusions, parce qu'il se base sur les règles pour faire l'inférence, où il y a un mappage avec la base des signatures et l'observation. Mais alors, ces limitations peuvent être remédiées avec les solutions analytiques plus avancées, d'où le Deep Learning. Ce dernier peut arriver à traiter et analyser une masse de données dans le système et/ou réseau informatique, ainsi que les activités de réseau. Dans le contexte de notre travail, nous allons montrer l'apport du Deep Learning pour la détection d'intrusion réseau [21].

## 1.2 Problématique

Les systèmes de détections d'intrusions sont un domaine en vogue, et ont fait l'objet de nombreuses recherches, mais la plupart des changements concernent l'ensemble de données collectées, qui contient de nombreux exemples de techniques d'intrusions telles que : la force brute, le déni de service ou même une infiltration depuis un réseau.
À mesure que les comportements et les modèles de réseau changent, et que les intrusions évoluent, on ne peut se dispenser de passer à des ensembles de données statiques et ponctuelles à des ensembles de données générées de manière plus dynamique, reflétant non seulement la composition du trafic et les intrusions de cette époque, mais également modifiables et reproductibles.
Les analystes de sécurité des systèmes d'information envisagent que leurs organisations prennent de l'ampleur des risques nuisibles, et préconisent la détection des menaces en temps réel, afin de les combattre d'une manière efficiente. Les entreprises peuvent diminuer ces risques, en faisant recours à des outils plus avancés, afin de leur aider de parvenir à un niveau de sécurité plus avancé. Pour y arriver, il faut se rendre fortement que ces outils recherchent les menaces inconnues ou virus au bon endroit, vu que les analystes de sécurité veulent des outils qui ont la capacité de leur donner une vue globale, de ce qui se passe sur un réseau et/ou système informatique, et à n'importe quel moment.

Faisant suite du contexte évoqué supra, dans le cadre de notre travail, nous allons nous préoccuper de montrer l'apport du Deep Learning dans la cybersécurité pour la détection d'intrusions, par rapport aux limitations qu'à les méthodes traditionnelles, et comment arriver à la mise en œuvre du Deep Learning pour la détection d'intrusions.

## 1.3 Objectifs concrets

### 1.3.1 théoriques

De tout ce qui précède, nous nous sommes fixé quelques objectifs :

- Montrer l'apport du Deep Learning par rapport aux outils traditionnels ;
- Étudier le modèle de détection d'intrusions existant, avec les outils traditionnels ;
- Étudier comment le Deep Learning est appliqué à la cybersécurité pour résoudre la détection d'intrusion [22].

### 1.3.2 Objectifs Pratiques et Applications possibles

A cet effet, il est espéré au terme de notre TPE le résultat attendu dans ce travail est la conception d'un modèle du Deep Learning, dans le but d'identifier une anomalie à partir d'une masse de donnée bien déterminée ; et faire la prédiction des

attaques futures qui peuvent subvenir dans notre système d'information. Les informations issues des résultats devront être analysées pour faire ressortir les différents mécanismes d'analyse d'intrusion et proposer des politiques système de détection des anomalies. En somme les conclusions de ce TPE pourront servir d'outil d'aide à la détection d'intrusion dans un système d'information, et cela permettra également de vérifier la précision de notre modèle dans la détection des anomalies [10].

## 1.4 Domaine d'application

Notre sujet traite le Deep Learning appliqué à la cybersécurité pour la détection d'intrusion. Un domaine d'étude en vogue à cause des nombreuses recherches qui y sont conduites par divers laboratoires et groupes de recherche, car la détection d'intrusion a un rôle crucial dans la cyberdéfense des systèmes d'information et/ou réseaux informatiques. Dans le contexte de notre TPE, nous aurons principalement se servir des notions dans l'apprentissage en profondeur ; des concepts et techniques dans la cybersécurité ; et le système de détection d'intrusion.

## 1.5 Liste de mots clés

Le Deep Learning : Le Deep Learning est une classe d'algorithmes de Machine Learning, qui utilisent plusieurs couches pour extraire progressivement les entités de niveau supérieur de l'entrée brute [24].
La cybersécurité : la cybersécurité est un ensemble des politiques, outils, dispositifs, concepts et mécanismes de sécurité, méthodes de gestions des risques, actions et technologies qui peuvent être utilisées pour protéger les actifs informatiques matériels et immatériels (connectés directement ou indirectement à un réseau) des entreprises et des organisations (avec un objectif de disponibilité, intégrité  authenticité, confidentialité, preuve  non-répudiation) [23].
L'intrusion : L'intrusion est le fait pour un objet ou une personne d'avoir accès dans un environnement bien déterminé où sa présence n'est pas souhaitée [10].

## 1.6 Difficultés à prévoir

Tout travail scientifique a des difficultés, sous différentes perspectives. Parmi lesquelles, nous pouvons signaler entre autres :

- Prise en main et maîtrise de l'environnement de construction d'un modèle ;
- Établissement dans un environnement avec le réseau ;
- Manque de données pour l'apprentissage
- Les spécifications du modèle etc.

# 2) RECHERCHES BIBLIOGRAPHIQUES

## 2.1 INTRODUCTION

L'intelligence artificielle est l'un de domaines de l'informatique qui amène des innovations très considérable dans le secteur de la nouvelle technologie de l'information et quasiment tous les domaines de la société humaine. Le système de classification et d'apprentissage en profondeur avec des algorithmes rapides en matière de puissance de calcul a donné la réponse à un grand nombre de questions posées dans la société, particulièrement dans la cybersécurité.
Mettre en place une plateforme dans le domaine de la cybersécurité est devenu un sujet d'actualité, un domaine d'étude en vogue à cause des nombreuses recherches qui y sont conduites par divers laboratoires et groupes de recherche, car cette dernière est d'une grande importance afin de connaître les mécanismes utilisés pour protéger les informations circulantes au sein d'une entreprise, organisme, etc.

## 2.2 Notion sur le Machine Learning

### 2.2.1 Définition

L'apprentissage automatique est un type d'intelligence artificielle qui confère aux ordinateurs la capacité d'apprendre sans être explicitement programmés. Il consiste à la mise en place des algorithmes ayant pour objectif d'obtenir une analyse prédictive à partir de données, dans un but précis [11].
Les algorithmes de Machine Learning utilisent donc nécessairement une phase dite apprentissage, qui a pour but de détecter des schémas dans les données et ajuster leur fonctionnement en conséquence.

### 2.2.2 Mode d'apprentissage

La figure 2.1 ci-après tirée illustre quelques types d'apprentissages selon le mode d'apprentissage qu'ils emploient [20].

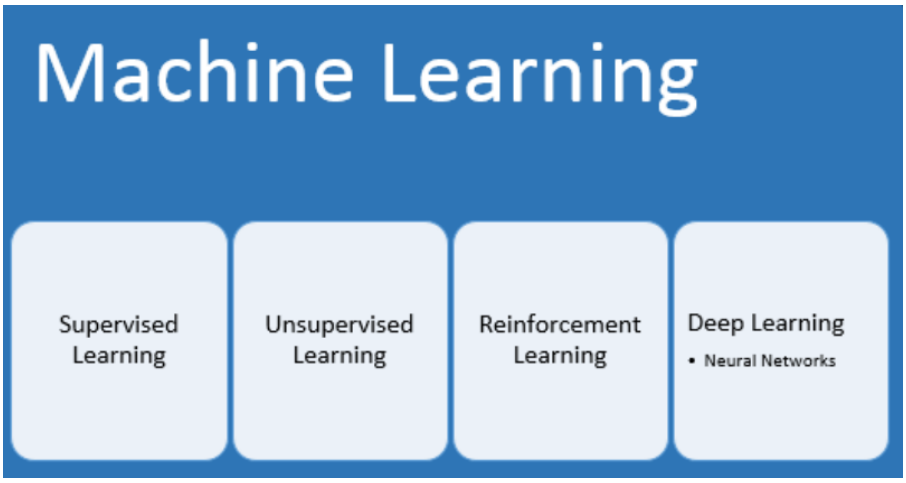


FIGURE 2.1 – Type d'apprentissage automatique

— L'apprentissage supervisé(supervised learning en anglais) est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés [11].

— L'apprentissage non supervisé (Unsupervised Learning en anglais) est un problème d'apprentissage automatique. Il s'agit, pour un logiciel, de trouver des structures sous-jacentes à partir de données non étiquetées [11].

— L'apprentissage par renforcement consiste, pour un agent (robot, etc.), à apprendre les actions à partir d'expériences, de façon à optimiser une récompense quantitative au cours du temps [11].

### 2.2.3 Deep Learning

#### Définition

L'apprentissage profond est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires [14].
Le Deep Learning est un sous-domaine du machine learning, qui repose donc sur ce qu'on appelle des réseaux de neurones artificiels (profonds), c'est-à-dire un ensemble de neurones (ce sont de petites calculatrices qui effectuent une opération mathématique) qui s'envoient des nombres en fonction de leurs liaisons, jusqu'à des neurones de sortie (en général !). Grâce à cette architecture, le Deep learning est capable de reconnaître des visages, de synthétiser des textes ou encore de conduire une voiture autonome ! [11].
Les différentes architectures d'apprentissage profond :

— Réseaux de neurones à convolution (convolutional Neural Networks) ;
— Réseaux de neurones profonds ;
— Réseaux de croyance profonde ;
— Les auto-encoders ;
— Les machines de Boltzmann ;
— Les Self-Organizing Maps (SOM).

#### Principe de fonctionnement

Les réseaux d'apprentissage profond sont entraînés sur la base de structures complexes de données auxquelles ils sont confrontés. Ils élaborent des modèles de calcul composés de plusieurs couches de traitement pour créer plusieurs niveaux d'abstraction afin de représenter les données.
Par exemple, le modèle d'apprentissage profond connu sous le nom de réseau neuronal à convolution (voir figure 2.2) peut être entraîné à l'aide d'un grand nombre (des millions) d'images, des images représentant des chats par exemple.

Ce type de réseau neuronal tire son apprentissage des pixels contenus dans les images reçues. Il peut classer des groupes de pixels en fonction des caractéristiques du chat tel que les griffes, les oreilles, les yeux indiquant la présence de l'animal dans l'image. L'apprentissage profond est très différent de l'apprentissage machine classique. Dans cet exemple, un expert dans ce domaine passerait un temps considérable à mettre au point un système d'apprentissage machine capable de détecter les caractéristiques représentatives du chat. Avec l'apprentissage profond, il suffit de fournir au système un très grand nombre d'images de chats pour qu'il en retienne de façon autonome les caractéristiques [14].

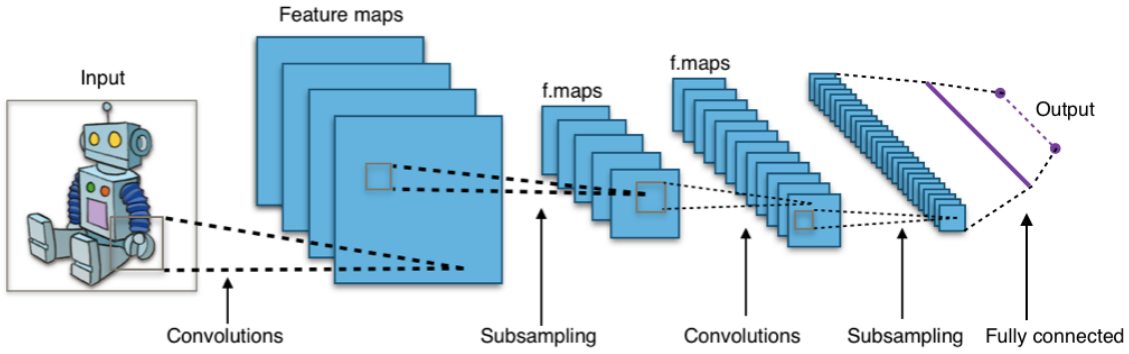


FIGURE 2.2 – Réseau de neurones à convolution avec plusieurs Structure des couches

## 2.3 Quelques concepts sur la cybersécurité

### 2.3.1 La cybersécurité

La cybersécurité consiste en un effort continu pour protéger les systèmes mis en réseau et les données contre l'utilisation ou le méfait non autorisés. À titre personnel, vous devez protéger votre identité, vos données et vos périphériques informatiques, donc protéger votre infrastructure informatique et les informations circulantes.

### 2.3.2 Cyberattaque

C'est une suite ordonnée d'actions menant à une violation de la politique de sécurité. La matérialisation de cette violation se montre par un dysfonctionnement du système d'informations ou du réseau (impossibilité de se connecter, arrêt d'un service, chiffrement des données par un ransomware). Une cyberattaque peut également ne pas être visible mais avoir des conséquences néfastes comme le vol d'informations confidentielles.

### 2.3.3 La Cyberdéfense

C'est un moyen d'attaque et de défense des réseaux et systèmes informatiques.

### 2.3.4 Intrusion

C'est une opération qui consiste à pénétrer, sans autorisation, aux données d'un système informatique, en contournant ou en désamorçant les dispositifs de sécurité mise en place par ce dernier [8]. C'est aussi le fait de se connecter à un système sans autorisation. Une intrusion informatique peut être perpétrée pour diverses raisons, notamment pour modifier ou voler de l'information confidentielle, fausser, contaminer ou détruire les données du système.

### 2.3.5 Cheval de Troie

C'est une porte dérobée installée sur un système à l'insu de ses utilisateurs et administrateurs, permettant à un pirate de s'y connecter régulièrement et facilement sans être vu.

### 2.3.6 Faille(vulnérabilité)

C'est une faille (logicielle) est une erreur (de programmation) faite par le programmeur, qui permet à un attaquant de faire exécuter un programme en dehors de ce qui est prévu.

### 2.3.7 Distributed Denial of Service Attack(DDoS Attack)

C'est une attaque visant à rendre un système indisponible pour ses utilisateurs. Ce type d'attaque sature les ressources (souvent le lien réseau) d'un fournisseur de services, ce qui le rend inaccessible.

### 2.3.8 Malware

C'est un programme dont l'usage contrevient à la politique de sécurité et aux attentes d'un utilisateur. Un malware utilise souvent des vulnérabilités pour s'installer sur un système. Les concepts ci-précédent sont tirés au niveau de référence [2][8].

## 2.4 SYSTÈMES DE DÉTECTION D'INTRUSION(IDS)

### 2.4.1 Généralité

Un système de détection d'intrusion est un mécanisme qui analyse et surveille les activités d'un réseau et/ou système informatique. Les systèmes de détection d'intrusions analysent ses configurations, ses failles, et l'intégrité des fichiers. Ces systèmes peuvent arriver à reconnaître des schémas d'attaque classiques, et tout cela par l'entremise des analyses

de comportements et actions anormaux, et suivent les violations de règles par les utilisateurs, et aussi par la surveillance du trafic réseau [2].

### 2.4.2 Déploiement des IDS

Les figures ci-après tirées de l'ouvrage [10] illustrent la manière de déploiement d'un IDS.

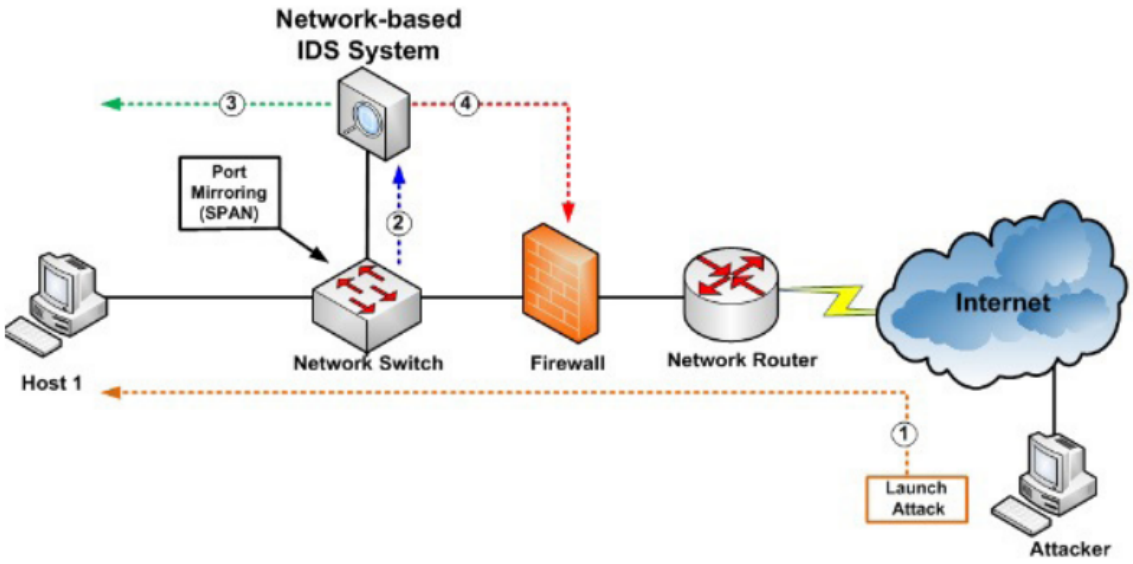


FIGURE 2.3 – NIDS

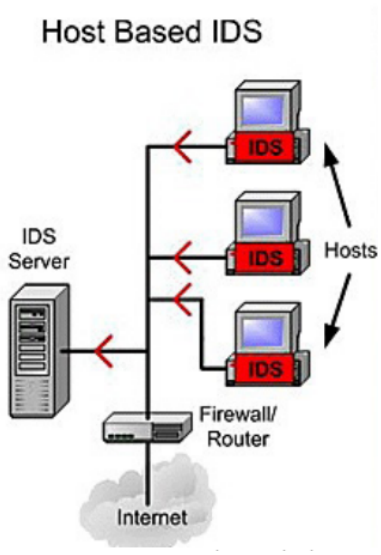


FIGURE 2.4 – HIDS

Un IDS basé sur l'hôte analyse exclusivement l'information concernant l'hôte. Un IDS basé sur le réseau analyse et interprète les paquets circulant dans le réseau.
Il existe aussi une approche hybride, qui consiste à monter l'IDS sur l'hôte et sur le réseau.

### 2.4.3 Techniques des détections d'intrusions

Il existe deux techniques principales de détection :

- Détection par signature (signature-based ou mesure détection) : Cette approche consiste à détecter des attaques, en vérifiant si les observations correspondent à des attaques connues [2].
- Détection par comportement (Détection d'anomalies) : Cette approche consiste à détecter une attaque en vérifiant que les observations ne correspondent pas à des comportements légitimes de référence [2]. Par ailleurs, certains IDS combinent les deux approches afin d'obtenir de meilleurs résultats. Une fois une attaque détectée, un IDS a le choix entre plusieurs types de réponses.

### 2.4.4 Mode de Réponses

Il existe deux types de réponses suivant les IDS utilisés :

- Réponse Passive : la réponse passive d'un IDS consiste à enregistrer les intrusions détectées dans un fichier de log qui sera analysé par le responsable de sécurité. Certains IDS permettent de logger l'ensemble d'une connexion identifiée comme malveillante.
- Réponse active : la réponse active, au contraire a pour but de stopper une attaque au moment de sa détection. Pour cela on dispose de deux techniques : la reconfiguration du firewall et l'interruption d'une connexion TCP.

La réponse passive est disponible pour tous les IDS alors que la réponse active est-elle plus ou moins implémentée.

## 2.5 Quelques travaux connexes

### 2.5.1 Étude sur les outils traditionnels pour la détection d'intrusion

**A. SNORT**

A.1. Généralité

Le SNORT est un Système de détection d'intrusion de réseau (NIDS) Open Source, capable d'analyser en temps réel le trafic sur les réseaux IP [08]. Le SNORT permet la définition précise des signatures, détecte les entêtes et dans le contenu des paquets dans IP, TCP, UDP et ICMP, détection de Nmap (Scan, OS fingerprint), des petits fragments, dénis de service et de débordement de Buffer (script kiddies).

A.2. Architecture de SNORT

Comme le montre la figure 2.5, SNORT comporte cinq parties principales. Le décodeur de paquets est responsable de la réception des paquets provenant de différentes interfaces réseau et de la réalisation de l'analyse initiale des paquets. Le préprocesseur est un plug-in destiné au traitement ultérieur des paquets décodés. Ses fonctions incluent la normalisation HTTP URI, la défragmentation des paquets, le réassemblage du flux TCP, etc.
Le noyau de SNORT est le moteur de détection, qui peut faire correspondre les paquets en fonction des règles configurées. La correspondance des règles est essentielle à la performance globale de SNORT. Une fois que la correspondance est réussie, il en informe le système de journalisation et d'alerte en fonction du comportement défini dans les règles. Ensuite, le système émettra l'alerte ou le journal en conséquence. Les utilisateurs peuvent également définir le module de sortie pour enregistrer les alertes ou les journaux sous une forme spécifique, telle qu'une base de données ou un fichier XML [21].

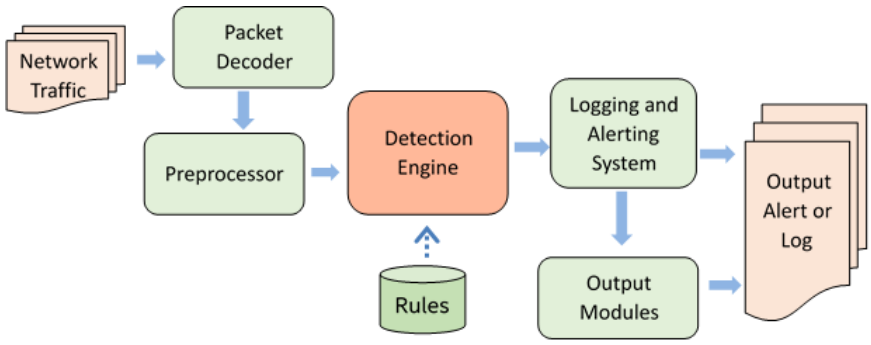


FIGURE 2.5 – Architecture SNORT

A.3. Avantages et Inconvénients de SNORT

L'outil traditionnel SNORT présente quelques avantages :

- Détection en temps réel ;
- Bonne base de signatures (mise à jour  modifiable) ;
- Gratuit (Open Source) ;
- Les alertes peuvent être lues à l'écran, envoyées dans un fichier, envoyées en messages Popup, être stockées dans une base de données MySql,au choix !
- Il vient déjà avec des règles minimales, on peut ainsi avoir des exemples de règles concrètes ;
- Langage de description simple et facile à utiliser ;
- Détecte l'attaque par fragmentation IP ;
- Large communauté d'utilisateurs (Beaucoup de contributions  documentations).

Par ailleurs SNORT présente quelques inconvénients :

- Souvent vulnérable par rapport à des attaques de Denial of Service ;
- Ne peut pas traiter les flux cryptés car la signature de ces attaques dépend évidemment du type de cryptage et aussi évidemment de la clé employée (c'est une des raisons) ;
- Un IDS fonctionne en mode binaire : il traduit toute requête comme une intrusion ou comme une trame "permise". Il n'existe pas d'intermédiaire entre les deux. Tout ce qui est possible, est de donner un degré d'importance à ces alertes ;
- SNORT a plus tendance que d'autres IDS à fournir des fausses alertes (en moyenne, 70 % des alertes remontées sont fausses), par exemple à cause de petites signatures comme phf qui déclenchent une alerte alors que cela peut être une simple requête contenant les mots ‘’dhfudge”ou ”muphf” ;
- Un IDS fonctionne en mode binaire : il traduit toute requête comme une intrusion ou comme une trame "permise". Il n'existe pas d'intermédiaire entre les deux. Tout ce qui est possible, est de donner un degré d'importance à ces alertes ;
- Les vraies attaques menées par exemple sur cgi-bin, ne sont pas toujours détectées.

**B. SURICATA**

B.1. Généralité

Le Suricata est un moteur de détection des menaces réseau, gratuit, open source, mature, rapide et robuste. Le moteur Suricata est capable de détecter les intrusions en temps réel (IDS), de prévention des intrusions en ligne (IPS), de surveiller la sécurité du réseau (NSM) et de traiter les PCAP hors ligne.
Suricata inspecte le trafic réseau à l'aide de règles puissantes et étendues et d'un langage de signature, et dispose d'un puissant support de script Lua pour la détection de menaces complexes. Avec les formats d'entrée et de sortie standard tels que les intégrations YAML et JSON, des outils tels que les SIEM existants, Splunk, Logstash / Elasticsearch, Kibana et d'autres bases de données ne demandent plus aucun effort. Le développement rapide de Suricata axé sur la communauté est axé sur la sécurité, la convivialité et l'efficacité.

B.2. Architecture du SURICATA

La figure ci-après illustre l'architecture de l'outil traditionnel :

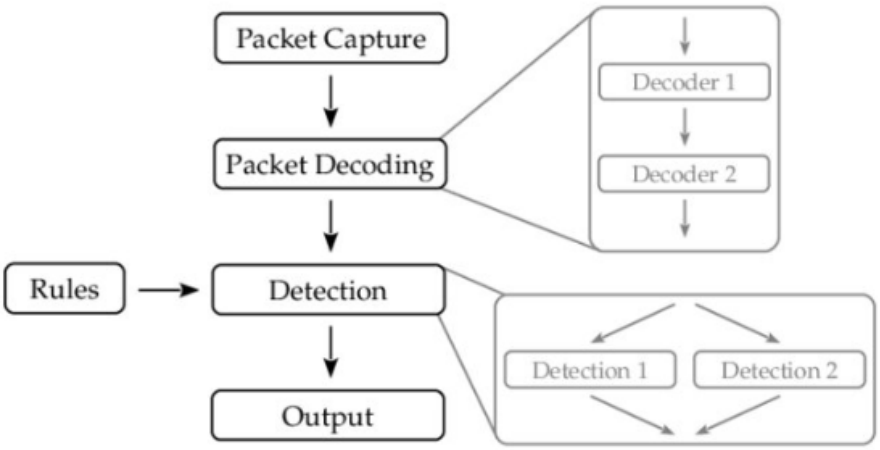


Figure 2.6 – Architecture SURICATA

B.3. Avantages et Inconvénients

L'outil SURICATA présente quelques avantages :
— Un moteur Open Source ;
— Multi-threaded ;
— Prise en charge de la réputation IP (Signaler le trafic provenant de sources malveillantes connues ;
— Détection automatique de protocole (Identification de protocole utilisé dans un flux réseau et applique les règles appropriées ;
— Code moderne et modulaire.
Les inconvénients de SURICATA :
— Difficile de contrôler les intrusions au niveau hôte ;
— Basé sur les règles ;
— Difficile de détecter une menace inconnue.

**C. Tableau comparatif des quelques outils traditionnels pour les IDS**

Le tableau ci-après illustre la comparaison succincte entre quelques outils traditionnels.

| Outil | Description | Avantages | Inconvénients |
|---|---|---|---|
| SNORT | IDS est capable d'analyser le trafic sur le réseau en temps réel et des paquets circulant sur le réseau IP | -Évolutivité<br>-Souplesse et convivialité<br>-Multi processus<br>-Détection à temps réel<br>-Moteur de détection modulaire<br>-Bonne base de signatures (mise à jour & modifiable)<br>-Distribution avec plus 2550 règles et possibilité d'en ajouter<br>-Langage flexible | -Difficultés d'installation et configuration<br>-Pas d'interface graphique<br>-Taux de faux négatifs élevé<br>-Difficulté de contrôler l'intégralité du réseau<br>-pas d'accélération matérielle<br>-Difficile de détecter des nouvelles attaques, si les signatures de ces dernières sont absentes |
| SURICATA | C'est un système de détection d'intrusion basé sur une source ouverte | -Un moteur Open Source<br>-Multi-threaded<br>-Prise en charge de la réputation IP<br>-Détection automatique de protocole<br>-Code moderne et modulaire<br>-Fonctions avancées (flowint, libHTTP, scripts LuaJIT) | -Difficile de contrôler les intrusions au niveau hôte<br>-Basé sur les règles<br>-Difficile de détecter une menace inconnue(indisponible dans la base de signatures) |

Table 2.1 – Comparaison des outils traditionnels (approche classique)

En général, les outils traditionnels sont basés sur les règles, qui leur permettent de faire un mappage entre les observations et la base de signatures. Si la menace est inconnue dans la base de signatures, difficile de détecter une attaque. D'où le plus grand inconvénient des outils traditionnels.

### 2.5.2 Approche analytique avancée

L'approche analytique est une approche qui pourrait remédier aux problèmes, que possèdent les outils traditionnels. L'approche analytique permet d'identifier des attaques ciblées ; montre une corrélation d'identité ainsi qu'une identification et investigation de la menace. L'approche analytique est faite par l'entremise de machine Learning. Ce dernier a la capacité de conférer aux ordinateurs la capacité d'apprendre sans être explicitement programmés [14].
Plusieurs tentatives de détection d'intrusion ont été effectuées avec l'emploi de machine Learning, précisément sur les Réseaux de Neurones et le Deep Learning.

## A. Réseaux de Neurones

### A.1. Bref aperçu

Les réseaux de neurones sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle et reliés par des poids. Ces poids de connexion gouvernent le fonctionnement du réseau. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Les réseaux de neurones ont plusieurs avantages dans la mise en œuvre d'un système de détection d'intrusion. Ils sont très efficaces et rapides dans la tâche de classification. Ils sont capables d'apprendre et d'identifier facilement les nouvelles menaces qui leur sont soumises [9].

### A.2. Travaux Antérieurs

Plusieurs travaux antérieurs ont été réalisés, James Canady [7] est le premier chercheur à proposer un système de détection d'intrusion basé sur l'analyse des protocoles réseau à base de réseaux de neurones. Il utilise les perceptrons multicouches (MLP) pour l'apprentissage. Le tableau ci-dessous (Tableau II.2) illustre quelques résultats des chercheurs ayant utilisé le modèle de réseaux de neurones pour concevoir un IDS réseau. Le processus d'apprentissage utilisé pour modifier les paramètres du réseau utilisé par ces chercheurs sont[17] :

— Vérifier si la sortie générée corresponds à la sortie désirée
— Présenter au réseau de neurones les données sous forme d'un vecteur
— Changer les paramètres du réseau afin de tendre vers la sortie désirée

Le tableau ci-après illustre les résultats de quelques travaux, utilisant le jeu de données KDD pour la validation leur modèle.

| Auteur Critères | Taux de reconnaissance | Faux Positif | Faux négatif |
|---|---|---|---|
| Vaitsekhovich 2008[16] | 93,21% | 12,90% | - |
| Khattab 2009 [15] | 97% | 2,4% | 0,8% |
| Iftikar 2009 [12] | 98% | 1,5% | - |
| Muna 2010 [18] | 78% | - | - |
| Aslihan 2012 [4] | 93,42% | 2,95% | - |
| Yousef 2012 [26] | 95,4% | 2,6% | - |
| Berlin et Gilbert 2012 [6] | 93,55% | 1,9% | - |

Table 2.2 – Résultats de quelques travaux

## B. Deep Learning

### B.1. Bref aperçu

Le Deep Learning fait partie des techniques de machine Learning, rendu possible par un accroissement de la performance de calcul des machines (GPU) et d'une base de données d'apprentissage de plus en plus gigantesque [24].
Les caractéristiques de l'image sont extraites automatiquement par le réseau de neurones. On parle de l'apprentissage hiérarchique (Hierarchical Feature Learning).

Pour ce faire, on utilise généralement des réseaux de Neurones à convolution (CNN) [11] :

— La couche de convolution(CONV) qui traite les données d'un champ récepteur : Permet d'extraire des caractéristiques déjà observées
— La couche de pooling(POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par le sous-échantillonnage). Ce qui réduit la sensibilité aux bruits.
— La couche correction (ReLU), en référence à la fonction d'activation (Unité de rectification linéaire). Ce qui entraine comment le signal circule d'un neurone à l'autre
— La couche entièrement connectée(FC), qui est une couche du type multilayer perceptron, ce qu'entraine que toutes les connexions entre neurones sont étudiées
— La couche de perte (LOSS) : Elle spécifie comment l'entrainement du réseau pénalise l'écart entre le signal prévu et réel

## B.2. Étude de faisabilité du Deep Learning à la cybersécurité pour la détection d'intrusion

Le Deep Learning fait partie des solutions analytiques avancées pour la détection d'intrusion. Grâce au machine Learning, particulièrement Deep Learning et aux algorithmes de détection d'anomalies à la cybersécurité offre une identification d'attaque ciblée.
Le Deep Learning arrive à détecter une intrusion grâce à sa phase d'apprentissage, sur laquelle plusieurs données sont appliquées, et arrivent à apprendre par lui-même un ensemble de données, et détecte les caractéristiques importantes dans cet ensemble de données pour construire un modèle, qui permettra de faire des prédictions avec des attaques inconnues qui peuvent subvenir dans le futur. Une fois obtenue un modèle, le Deep Learning arrive à faire une prédiction précise avec les futures données.
Le Deep Learning peut traiter et analyser d'énormes quantités de données et d'activités de réseau, ce que les outils traditionnels ne permettent pas [22].

## B.3. Travail antérieur

Tamin Mirza a proposé dans son travail [22], une approche pour Construire un système de détection d'intrusion en utilisant le Deep Learning.

### B.3.1. Bref Aperçu

Dans cet article, l'auteur nous présente les méthodologies de l'application de l'apprentissage approfondie pour identifier une anomalie à partir d'une masse de données circulant dans le réseau. Pour arriver à cette analyse, l'auteur à utiliser le jeu de données collecté par l'Institut canadien pour la cybersécurité ISCX2012.
Après prétraitement de données, deux données de chaque flux sont retenues :
— Données utiles de destination au format UTF : il s'agit du paquet de données entrant reçu par l'utilisateur ;
— Tag : Ceci qualifie la charge de données de normale ou anomale.

La conversion de données du Dataset avec l'extension pcap en un format de fichier xml lisible s'est effectuée au moyen d'un outil appelé ISCX Flowmeter. Après l'obtention de plusieurs fichiers XML, on extrait deux valeurs de données principales de chaque arbre par fichier, et concaténé ces données utiles d'une façon d'avoir une longueur 7500.
Et alors prendre ces données et le mettre dans un tableau NumPy de dimensions 50X50X30.
Ensuite, ces tableaux sont empilés et en ajoutant une colonne qui contient l'étiquette de chaque charge utile, donc si la charge est normale ou anormale. Une fois le traitement de données terminé, les données sont stockées dans des fichiers NPY afin qu'elles soient utilisées comme entrées par le modèle d'apprentissage en profondeur.
L'auteur a utilisé la technique d'apprentissage par transfert sur le modèle préformé pour VGG-19 Keras, parce que c'est celle qui a donné la précision satisfaisante.
En ce qui concerne l'entraînement du modèle, l'auteur a utilisé la technique conventionnelle de détection des anomalies qui comprend deux phases :
— Phase de formation
— Phase de test

### B.3.2. Résultats

Pour la phase du test, l'auteur a fait recours à la fois aux données anormales et normales lors de l'entraînement. Donc il s'agissait d'un problème pour détecter les valeurs aberrantes. Au cours de la phase de test, le modèle a pu identifier les données normales avec une précision de 100% et les données d'anomalies avec une précision de 85%. D'où en faisant suite du Dataset considéré, le résultat obtenu sont satisfaisants. Les résultats obtenus lors de la phase de test sont les suivants :

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| normal | 1.00 | 0.99 | 1.00 | 191266 |
| anomaly | 0.85 | 0.98 | 0.91 | 9433 |
| avg / total | 0.99 | 0.99 | 0.99 | 200699 |

FIGURE 2.7 – Évaluation scientifique du modèle à l'aide de la bibliothèque Scikit-learn

Le tableau ci-après présente succinctement les avantages de cet article, et leurs limitations.

| Description | Avantages | Inconvénients |
|---|---|---|
| Présentation de l'approche analytique avancée de l'application du Deep Learning à la cybersécurité pour identifier une anomalie dans le réseau | -Vue globale de l'ensemble du réseau<br>-Utilisation de données de l'Institut Canadien pour la Cybersécurité (ISCX2012) dans son ensemble<br>-Identification des données normales avec une précision de 100%<br>-Identification des données anormales avec une précision de 85% | -Utilisation de jeu de données pas récentes, datant de 2012<br>-Plus de temps consacré à la digestion directe du format de fichier pcap dans le modèle<br>-Plus de temps pour la conversion de format pcap en XML, et XML en image<br>- |

TABLE 2.3 – Avantages et Inconvénients du travail [22]

L'étude sur le travail [22] arrive à bien résoudre les problèmes que les outils traditionnels possédés. Cette étude montre le processus analytique avancé pour la détection d'anomalies dans un réseau et/ou système. Mais ce travail ne tient pas compte d'un ensemble de données mis à jour par l'Institut Canadien pour la Cybersécurité de 2017.
Est-ce possible d'utiliser l'ensemble de données récentes de l'Institut Canadien pour la Cybersécurité afin de trouver un modèle plus précis ? Au vu de ces analyses, nous nous proposons dans le cadre de notre travail personnel encadré, de faire une étude approfondie sur la limitation des outils traditionnels face au Deep Learning ; et arriver à apporter un édifice dans le travail [22], en tenant compte des données mises à jour par l'Institut National pour la Cybersécurité, dans la perspective de construire un modèle d'apprentissage plus précis que le précédent. Notre cas d'étude se portera dès la survenue d'une intrusion(anomalie) dans le réseau.

## Conclusion

En ce qui concerne ce travail d'étude bibliographique, nous avons présenté dans un premier temps les concepts de base du système de détection d'intrusion réseau et hôte, qui montrent les approches de détection d'intrusion ou d'anomalie dans un réseau et/ou système.
Par la suite, nous avons porté une étude comparative entre les outils traditionnels et le Deep Learning appliqué à la cybersécurité afin de détecter les intrusions et/ou anomalies dans le réseau et/ou système.
A la fin, nous avons présenté et donné des motivations de notre travail personnel en expliquant succinctement l'approche optée pour ce dernier.

# 3) SOLUTION PROPOSEE

## 3.1 Solution proposée

Dans cette section, nous allons exposer les différents composants de ce que nous allons faire, et comment le faire avec une définition plus précise.
Dans le cadre de notre travail, nous nous sommes décidé à reprendre le travail [22], avec un nouveau jeu de données de l'Institut Canadien pour la Cybersécurité 2017 (ISC 2017), afin de maximiser la précision de la détection d'intrusion, car les données ISC2017 renferme plus de caractéristiques que ISC2012

Nous allons prendre les données utilisées dans le travail [22] de l'Institut Canadien pour la cybersécurité de 2012, ensuite le réappliquer dans le modèle et appliquer les données ISC 2017, pour qu'on puisse faire une comparaison entre ces deux résultats obtenus.

## 3.2 Outils à utiliser et Types de données à utiliser

Dans notre étude, nous ferons usage de quelques outils utilisés dans le travail [22].

### 3.2.1 Logiciels, plateforme et algorithmes

— Keras avec version GPU TensorFlow de Google comme moteur principal
— IDE Python 3.5.6
— Plateforme ANACONDA, gestionnaire de packages Python à code source ouvert pour la science de données
— Editeurs : Spyder, Jupyter Notebook.
— Langage de Programmation : Python.
— Apprentissage profonds (Architecture séquentielle)

Le choix porté sur le langage de programmation Python se justifie par son environnement entièrement organisé autour de graphes computationnels ; puissants ; facile et flexible à utiliser.

## 3.3 Architectonique Fonctionnel

La figure ci-après répresente le modèle fonctionnel de notre système.

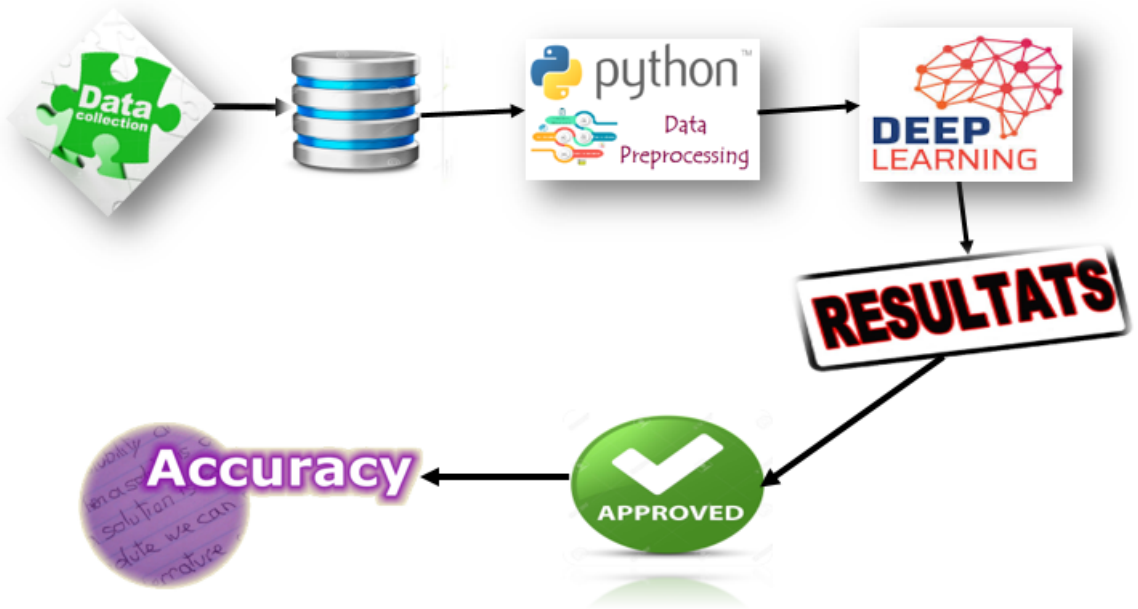


Figure 3.1 – Architectonique Fonctionnelle de la solution proposée

## 3.4 Recherche méthodologique de jeu de données

### 3.4.1 Collecte de données : Infrastructure

Pour créer un banc d'essai complet dans le travail [21], il a fallu concevoir et mettre en œuvre deux réseaux, à savoir Attack-Network et Victim-Network.Le Victim-Network est une infrastructure hautement sécurisée comprenant un pare-feu, un routeur, des commutateurs et la plupart des systèmes d'exploitation courants, ainsi qu'un agent fournissant les comportements inoffensifs sur chaque PC. Le réseau d'attaque est une infrastructure complètement séparée conçue par un routeur, un commutateur et un ensemble de PC avec des adresses IP publiques et différents systèmes d'exploitation nécessaires à l'exécution des scénarios d'attaque. La figure 3.2 illustre cette infrastructure.

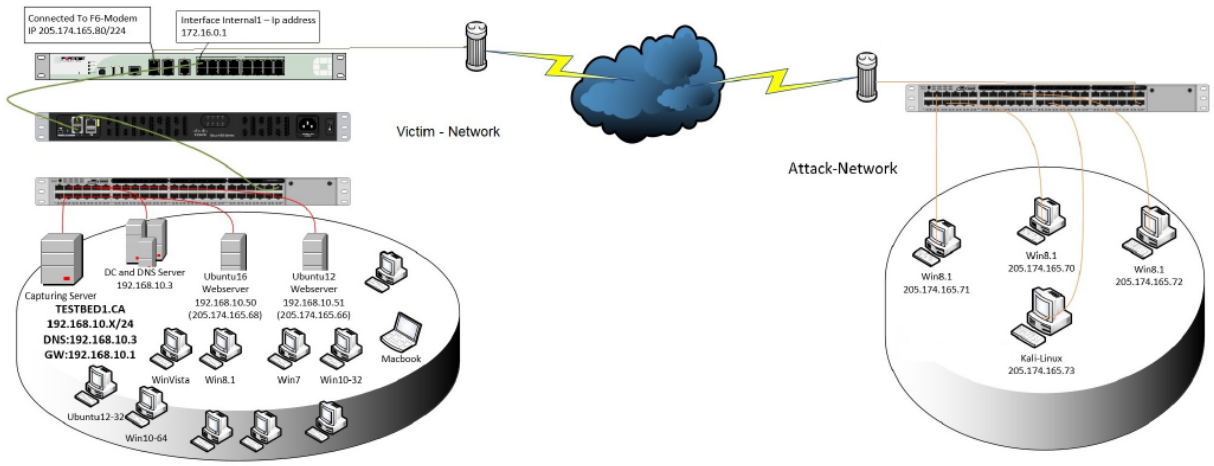


FIGURE 3.2 – Banc d'essai complet

La période de capture a débuté le lundi 3 juillet à 9h00, et a duré de manière continue pendant 5 jours, pour se terminer le vendredi 7 juillet à 17h00. Les attaques ont ensuite été exécutées pendant cette période. Comme le montre la figure 3.3, le lundi est le jour normal et ne comprend que le trafic bénin. Les attaques implémentées incluent FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet et DDoS, exécutées respectivement le matin et l'après-midi des mardi, mercredi, jeudi et vendredi. Sur la base des scénarios d'attaque, pour exécuter chaque attaque, nous avons utilisé l'un des meilleurs outils, le plus accessible au public, ou développé le code par Python (Jeu de données disponible publiquement à l'adresse http ://www.unb.ca/cic/datasets/IDS2017.html). Attaque par force brute (mardi matin après-midi) : il existe de nombreux outils pour mener des attaques par force brute contre la fissuration des mots de passe, tels que Hydra, Medusa. , Ncrack, modules Metasploit et scripts Nmap NSE. En outre, il existe des outils tels que hashcat et hash-pump pour la fissuration par mot de passe.

| Days | Labels |
|---|---|
| Monday | Benign |
| Tuesday | BForce,SFTP and SSH |
| Wednes. | DoS and Hearbleed Attacks slowloris, Slowhttptest, Hulk and GoldenEye |
| Thurs. | Web and Infiltration Attacks Web BForce, XSS and Sql Inject. Infiltration Dropbox Download and Cool disk |
| Friday | DDoS LOIT, Botnet ARES, PortScans (sS,sT,sF,sX,sN,sP,sV,sU, sO,sA,sW,sR,sL and B) |

FIGURE 3.3 – Attaques détectées dans l'infrastructure

### 3.4.2 Déscription du DataSet

Pour la description de donnée, nous avons utilisés l'analyseur de flux de trafic Réseau CICFlowMeter. NetFlowMeter est un générateur de flux de trafic réseau écrit en Java. Il offre plus de flexibilité en termes de choix des fonctionnalités que vous souhaitez calculer, d'ajout de nouvelles fonctionnalités et de contrôle accru de la durée de temporisation du flux. CICFlowMeter génère des flux bidirectionnels (Biflow), où le premier paquet détermine les directions vers l'avant (source à destination) et arrière (destination vers la source), d'où les 83 fonctions statistiques telles que Durée, Nombre de paquets, Nombre d'octets, Longueur des paquets, etc. Sont également calculés séparément dans les sens aller et retour. La sortie de l'application est le format de fichier CSV avec six colonnes étiquetées pour chaque flux, à savoir :

— IP source
— IP Destination
— Port source
— Port destination
— Protocole avec plus de 80 fonctionnalités de trafic réseau.

Notez que les flux TCP sont généralement terminés lors du démantèlement de la connexion (par paquet FIN), tandis que les flux UDP se terminent par un délai d'expiration du flux. La valeur de temporisation du flux peut être attribuée de manière arbitraire par le schéma individuel, par exemple 600 secondes pour TCP et UDP.

Le tableau ci-dessous décris l'extrait des quelques 80 fonctionnalités de jeu de données.

| Nom de la fonctionnalité | La description |
|---|---|
| Feduration | Durée du flux en microseconde |
| total_fpackets | Nombre total de paquets dans le sens aller |
| total_bpackets | Nombre total de paquets dans la direction arrière |
| total_fpktl | Taille totale du paquet dans le sens aller |
| total_bpktl | Taille totale du paquet en arrière |
| min_fpktl | Taille minimale du paquet dans le sens aller |
| min_bpktl | Taille minimale du paquet en arrière |
| max_fpktl | Taille maximale du paquet dans le sens aller |
| max_bpktl | Taille maximale du paquet en arrière |
| mean_fpktl | Taille moyenne du paquet dans le sens aller |

Figure 3.4 – Extrait de 80 fonctionnalités du Dataset

Signalons que le Dataset de 2017 comprends plus de 1580215 exemples, pour servir d'apprentissage.

### 3.4.3 Différentes caractérisques des Jeux de données 2012 vs 2017

Le tableau ci-après illustre la comparaison entre le DataSet de 2012 et de 2017 du point de vue caractéristiques.

| Dataset 2012 | Dataset 2017 |
|---|---|
| RESEAU ET TRAFIC REALISTES | CONFIGIRATION RESEAU COMPLETE |
| JEU DE DONNEES ETIQUETTE | TRAFIC COMPLET |
| CAPTURE D'INTERACTION TOTALE | JEU DE DONNEES ETIQUTTE |
| CAPTURE COMPLETE | INTERACTION COMPLETE |
| DIVERS SCENARIOS | CAPTURE COMPLETE |
| | PROTOCOLES DISPONIBLES |
| | DIVERSITE DES ATTAQUES |
| | HETEROGENEITE |
| | FEATURE SET |
| | METADATA |

FIGURE 3.5 – Comparaison entre Dataset 2012 et 2017

### 3.4.4 Difficultés de la réalisation

— Maîtrise de la plateforme, adaptation de l'environnement
— Au vu de la masse de donnée à traiter, l'ordinateur à utiliser doit avoir une performance énorme dans la puissance de calcul

## 3.5 Planification des taches

| TACHE | Description | Temps estimé(semaine) |
|---|---|---|
| Recherche sur l'API Keras et TensorFlow | o Étude de l'API Keras<br>o Étude du TensorFlow<br>o Étude de l'IDE Python | 3 |
| IDE DE L'IMPLEMENTATION | o Plateforme ANACONDA;<br>o Jupyter NotBook;<br>o Pyhcarm;<br>Recherche sur leLangage Python. | 4 |
| Implémentation et test résultats | o Récupération Jeu de données 2012;<br>o Test résultats;<br>o Récupération Jeu de données 2017;<br>o Prétraitement de données<br>o Test résultats;<br>o Présentation de résultats aux encadreurs et amélioration; | 5 |
| Constitution du rapport final | o Rassemblement des sous rapports élémentaires;<br>o Présentation et correction du premier rapport final<br>o Présentation finale. | 4 |

FIGURE 3.6 – Planification des taches

## 3.6 Difficultés à prévoir

Les précautions à prendre pour réaliser ce travail sont :
— Un ordinateur avec une capacité élevée (ou soit utilisé google collab)
— Traitement de données de 2017 de l'Institut Canadien pour la cybersécurité
— Adaptation avec l'environnement google collab

## 3.7 Conclusion

Au égards de tout ce qui précède, l'actuel rapport de la solution proposée nous a aidé à exposer le mode opératoire que nous allons observer pour la concrétisation de l'étape pratique de notre travail. L'élaboration de la solution proposée en passant par les outils et algorithmes optés ; les sources et volumes de données ; les différents scénarios de tests ; ainsi que la planification des taches d'une manière sommaire, indique la mise à termes de la partie théorique du TPE. Rappelons que cette partie nous a permis de cerner notre sujet, l'état de l'art et quelques objectifs à atteindre.
La partie pratique est consacrée à l'implémentation de notre solution, nonobstant, nous allons développer nos potentialités techniques enfin que nous puissions atteindre le but voulu.

## Deuxième partie

# PARTIE PRATIQUE

# 4) IMPLEMENTATION

## 4.1 PREPARATION DE DONNEES

Le jeu de données de 2017 possèdent plusieurs attaques, et explorer toutes ces dernières nécessites l'analyses minitieuses pour chaque fonctionnalités, dans le cadre de notre travail, nous nous sommes focalisés sur un seul type d'attaque « DDoS » détecté le Vendredi après-midi. Ce jeu de données possède 225745 exemples.

### 4.1.1 Visualisation de données

Le graphique ci-après illustre la proportion des exemples anormales et normales du Dataset.

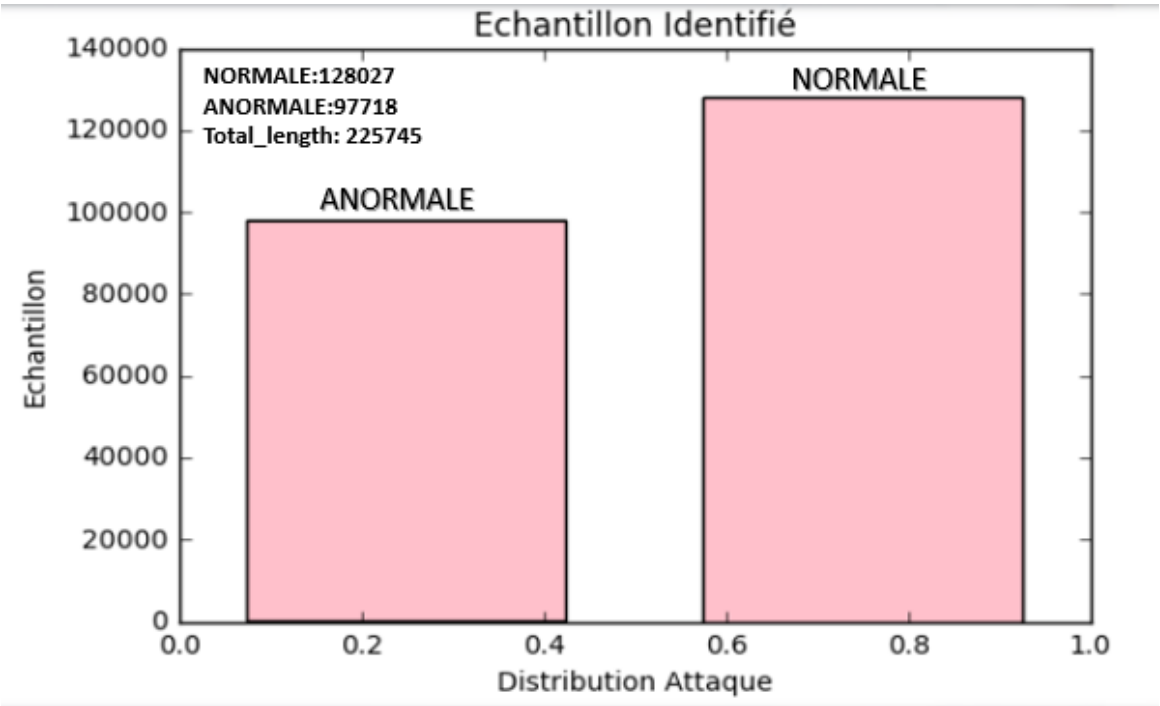


FIGURE 4.1 – Proportion des exemples normales et Anormales

### 4.1.2 Pretraitement de données

Le prétraitement des données est une technique utilisée pour convertir les données brutes en un ensemble de données vierge. En d'autres termes, chaque fois que les données sont collectées à partir de différentes sources, elles sont collectées au format brut, ce qui n'est pas réalisable pour l'analyse.

Pour pouvoir traiter nos données, nous nous sommes servis de quelques librairies Python.

```
In [8]: import statsmodels as stat
        #import seaborn as sbrn
        import pandas as pds
        import matplotlib.pyplot as mplt
        import numpy as np
        from sklearn.preprocessing import Imputer
        from sklearn.preprocessing import LabelEncoder, OneHotEncoder
        #from sklearn.cross_validation import train_test_split
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import StandardScaler
        from sklearn.preprocessing import normalize
```

FIGURE 4.2 – Importation Librairies

**QUELQUES EXTRAITS DU PRETRAITEMENT DES DONNEES**

```
In [4]: print("Shape:", data_set_DDoS.shape)

        Shape: (225745, 85)
```

FIGURE 4.3 – Nombre d'attributs

En statistique, les données manquantes, ou valeurs manquantes, apparaissent lorsqu'une observation d'une variable n'a pas de valeur. Les données manquantes sont fréquentes et peuvent avoir un effet significatif sur les conclusions qui peuvent être tirées à partir des données. Dans notre cas, nous avons utilisé un code python pour tester les valeurs manquantes, et il n'y a pas eu des données manquantes. La figure 4.4 illustre l'extrait du résultat de test des données manquantes.

```
In [39]: print(dtst.isnull().sum)
<bound method DataFrame.sum of        Flow ID  Source IP  Source Port  Destination IP  I
0        False      False        False           False             False
1        False      False        False           False             False
2        False      False        False           False             False
3        False      False        False           False             False
4        False      False        False           False             False
5        False      False        False           False             False
6        False      False        False           False             False
7        False      False        False           False             False
8        False      False        False           False             False
9        False      False        False           False             False
10       False      False        False           False             False
11       False      False        False           False             False
12       False      False        False           False             False
13       False      False        False           False             False
14       False      False        False           False             False
15       False      False        False           False             False
16       False      False        False           False             False
17       False      False        False           False             False
18       False      False        False           False             False
19       False      False        False           False             False
```

FIGURE 4.4 – Données manquantes

### CHOIX DES ATTRIBUTS PERTINENTS

La sélection des fonctionnalités est un processus dans lequel vous sélectionnez automatiquement les fonctionnalités de vos données qui contribuent le plus à la variable de prédiction ou à la sortie qui vous intéressent. Le fait de disposer de trop nombreuses fonctionnalités non pertinentes dans vos données peut réduire la précision des modèles.

Quatres façons de sélectionner les fonctionnalités pour l'apprentissage automatique en Python.

— Univariate Selection : Les tests statistiques peuvent être utilisés pour sélectionner les entités ayant la relation la plus forte avec la variable de sortie.
— Recursive Feature Elimination(RFE) : L'élimination des caractéristiques récursives (ou RFE) fonctionne en supprimant récursivement les attributs et en construisant un modèle sur les attributs restants. Il utilise la précision du modèle pour identifier quels attributs (et combinaison d'attributs) contribuent le plus à la prédiction de l'attribut cible. La RFE est l'objet de notre travail.
— Principal Component Analysis(ACP) : L'analyse en composantes principales (ou ACP) utilise l'algèbre linéaire pour transformer l'ensemble de données à ! un formulaire compressé. Généralement, cela s'appelle une technique de réduction des données. Une propriété de PCA est que vous pouvez choisir le nombre de dimensions ou de composant principal dans le résultat transformé.
— Feature Importance : Des arbres de décision en sac comme Random Forest et Extra Trees peuvent être utilisés pour estimer l'importance des caractéristiques.

L'extrait du code RFE pour la selection de meilleures fonctionnalités

```
21 from sklearn.feature_selection import SelectKBest
22 from sklearn.feature_selection import RFE
72 model = LogisticRegression()
73 rfe = RFE(model, 3)
74
75 fit = rfe.fit(X,Y)
76 fit
77
78 print("Num Features: %d" % fit.n_features_)
79 print("Selected Features: %s"% fit.support_)
80 print("Feature Ranking: %s"% fit.ranking_)
81
```

FIGURE 4.5 – Code RFE

```
In [375]: print("Num Features: %d" % fit.n_features_)
     ...: print("Selected Features: %s"% fit.support_)
     ...: print("Feature Ranking: %s"% fit.ranking_)
Num Features: 3
Selected Features: [False False False False False False False False False False False False False
 False False False False False False False False False  True False
 False False False False False False False False False  True False
 False False False False False False False False False False False
  True False False False False False False False False False]
Feature Ranking: [18 22 21 20 30 17 11 16 19 12  6 42 54 57 56 27 31  7  8 29  3 10  1 32 45
 43 46 41 44 40 55 47 39  4  1  2 28 52 53 51 48 49 50 34  5 36 15  9  1 37
 33 14 35 24 13 38 23 26 25]
```

FIGURE 4.6 – Résultat attributs pertinents

Après analyse et plusieurs entrainements de la méthode RFE, les meilleurs fonctionnalités trouvées sont :

— Flow-Duration :Durée du flux en microseconde
— Flow iat Std :Ecart-type entre deux paquets envoyés dans le flux
— Average Packet size :Taille moyenne du paquet
— Bwd Packet Length std :Écart type de la taille du paquet en arrière

Les attributs pertinents trouvés, nous donnent le jeton de passer à la phase d'apprentissage. Signalons que les attributs pertinents sont trouvés par rapport au target "Label", qui qualifie un exemple d'anormal ou normal.

### CHOIX DU FRAMEWORK

Dans le cadre de ce travail, nous utiliserons Keras qui est une librairie Python de haut niveau pour, entre autre, pilote Tensorflow. Ce dernier est le framework de machine learning de Google. Keras propose principalement deux types de modèles : le modèle séquentiel et la classe de modèle utilisée avec l'API fonctionnelle. Le modèle séquentiel est l'objet de notre travail.

### ARCHITECTURE DU MODELE

L'architecture séquentielle de notre modèle est représentée à la fgure 4.7.

```
def classifier(optimizer):
    model = Sequential()
    model.add(Dense(units=16, kernel_initializer='uniform', activation='relu', input_dim=4))
    model.add(Dense(units=8, kernel_initializer='uniform', activation='relu'))
    model.add(Dense(units=6, kernel_initializer='uniform', activation='relu'))
    model.add(Dense(units=1, kernel_initializer='uniform', activation='sigmoid'))
    model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])
    return model
```

FIGURE 4.7 – Architecture du modèle

### CHOIX DE L'OPTIMISEUR ET SES HYPERPARAMETRES

Le travail de l'optimiseur consiste à mettre à jour les pondérations du réseau de neurones de manière à minimiser l'erreur des pertes d'entrainement.Dans le cadre de ce travail, nous avons utilisé la famille d'optimiseur à la descente de gradient.

— Optimizer : adam
— Epochs :120
— Batch-size :1
— Loss : binary-crossentropy
— Metrics : Accuracy

# 5) EXPERIMENTATIONS

Dans cette section, nous allons présenter quelques résultats obtenus dans notre entrainement.

## 5.1 VALIDATION CROISEE

La validation croisée désigne le processus qui permet de tester la précision prédictive d'un modèle dans un échantillon test (parfois aussi appelé échantillon de validation croisée) par rapport à la précision prédictive de l'échantillon d'apprentissage à partir duquel le modèle a été développé. Dans l'idéal, avec un échantillon suffisamment grand, une certaine proportion d'observations (disons la moitié ou les deux-tiers) peut être affectée à l'échantillon d'apprentissage et les observations restantes sont affectées à l'échantillon test. Le modèle peut être construit en utilisant les observations de l'échantillon d'apprentissage, et la puissance prédictive peut être testée en utilisant les observations de l'échantillon test. Si le modèle s'exécute aussi bien sur l'échantillon test que sur l'échantillon d'apprentissage, nous pouvons dire que la validation croisée est bonne ou plus simplement, qu'il y a validation croisée.

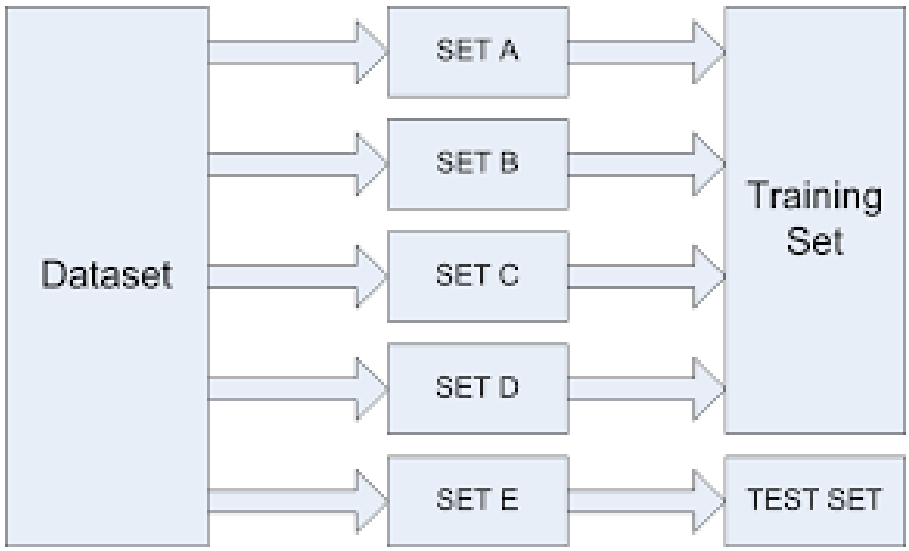


FIGURE 5.1 – validation croisée

## 5.2 TRAINING SET ET TEST SET

Faisant suite de la validation croisée, nous avons disséqué notre dataset en sous échantillon. Nous entrenons par exemple 20% de test et 80% de training, ensuite nous changéons les pourcentages après chaque phase d'apprentissage.

```
In [ ]: X_DDoS_train,X_DDoS_test,Y_train,Y_test = train_test_split(X_DDoS,Y_tag, test_size = 0.2, random_state = 0)

In [ ]: X_DDoS_train,X_DDoS_test,Y_train,Y_test = train_test_split(X_DDoS,Y_tag, test_size = 0.25, random_state = 0)

In [ ]: X_DDoS_train,X_DDoS_test,Y_train,Y_test = train_test_split(X_DDoS,Y_tag, test_size = 0.1, random_state = 0)
```

FIGURE 5.2 – Données Entrainement et Test

## 5.3 MATRICE DE CONFUSION

La matrice de confusion est un indicateur de performance en classification. Elle peut avoir deux classes ou plus. Dans le cadre de ce travail nous avons deux classes, la classe des observations normales et anormales. Pour mesurer les performances de ce classifieur, il est d'usage de distinguer 4 types d'éléments classés pour la classe voulue :

— Vrai positif VP : Elément de la classe 1 correctement prédit
— Vrai négatif VN : Elément de la classe 0 correctement prédit
— Faux positif FP : Elément de la classe 1 mal prédit
— Faux négatif FN : Elément de la classe 0 mal prédit

Ces informations peuvent être rassemblés et visualisés sous forme de tableau dans une matrice de confusion. Dans le cas d'un classifieur binaire, on obtient :

| | | Classe prédite | |
|---|---|---|---|
| | | Classe 0 | Classe 1 |
| Classe réelle | Classe 0 | VN | FN |
| | Classe 1 | FP | VP |

FIGURE 5.3 – Matrice de confusion

En particulier, si la matrice de confusion est diagonale, le classifieur est parfait. Partant de la matrice de confusion, Il est possible de calculer plusieurs indicateurs résumant la matrice de confusion. Par exemple si nous souhaitons rendre compte de la qualité de la prédiction sur la classe 1, on définit :

— Précision : Proportion d'éléments bien classés pour une classe donnée :

$$Precision_{\text{de la classe 1}} = \frac{VP}{VP + FP}$$

— Rappel : Proportion d'éléments bien classés par rapport au nombre d'éléments de la classe à prédire :

$$Rappel_{\text{de la classe 1}} = \frac{VP}{VP + FN}$$

— F-mesure. Mesure de compromis entre précision et rappel :

$$F - mesure_{\text{de la classe 1}} = \frac{2 * (Précision * Rappel)}{Précision + Rappel}$$

Il est possible de calculer tous ces indicateurs pour chaque classe. La moyenne sur chaque classe de ces indicateurs donne des indicateurs globaux sur la qualité du classifieur.

$$Precision = \frac{1}{k} \sum_{i=1}^{k} \frac{VP_i}{VP_i + FP_i}$$

$$Rappel = \frac{1}{k} \sum_{i=1}^{k} \frac{VP_i}{VP_i + FN_i}$$

$$F - mesure = \frac{2 * (Précision * Rappel)}{Précision + Rappel}$$

La matrice de confusion trouvée à la prémière phase d'apprentissage est :

| | ANORMAL | NORMAL |
|---|---|---|
| ANORMAL | 2589 | 462 |
| NORMAL | 938 | 42560 |

FIGURE 5.4 – Matrice de confusion de notre modèle

En considérant la matrice de confusion trouvée ci-précédente, nous avons appliqué les indicateurs de performances afin de trouver la précision, qui est décrit dans le tableau ci-après :

| | PRECISION | RECALL | F-MESURE |
|---|---|---|---|
| ANORMAL | 0.84 | 0.73 | 0.78 |
| NORMAL | 0.97 | 0.98 | 0.98 |

FIGURE 5.5 – Précision du modèle

Nous avons calculer aussi le score de précision( Accuracy-Score) du modèle par la relation :

$$\text{Accuracy-score=} \frac{Matrice\_de\_Confusion[i][i] \; + \; Matrice\_de\_Confusion[j][j]}{\sum_{i=0}^{2} Matrice\_de\_Confusion[i][i]}$$

Le score de précision du modèle à l'entrainement i est :

$$\text{Accuracy-score=} \frac{2589 \; + \; 42560}{2589+462+938+42560} = 0,9699$$

Le score de précision après plusieurs entrainements est décrit au tableau ci-après avec la moyenne(AVG) comme la précision finale :

| N° | ACCURACY |
|---|---|
| Entrainement 1 | 0,9699 |
| Entrainement 2 | 0,9692 |
| Entrainement 3 | 0,9710 |
| Entrainement 4 | 0,9612 |
| Entrainement 5 | 0,9777 |
| Entrainement 6 | 0,9770 |
| Entrainement 7 | 0,9792 |
| Entrainement 8 | 0,9711 |
| Entrainement 9 | 0,9633 |
| Entrainement 10 | 0,9777 |
| AVG | **0,9717** |

FIGURE 5.6 – Validation croisée

Ce tableau répertorie les différentes précisions obtenues lors de dix itérations de notre modèle avec le nombre d'époques bien déterminés. La précision moyenne trouvée est évaluée à 97%. La section analyse des résultats explicite quelques discussions sur les résultats trouvés et l'etat de l'art.

# 6) ANALYSE DES RESULTATS

## 6.1 RESULTATS ET DISCUSSIONS

Les résultats de la détection des valeurs abérrantes à partir des dix itérations de l'algorithme d'apprentissage profond sont illustrés dans la figure 5.6. En effet, nous avons formé notre modèle sur les valeurs de données dont les labels ont été étiquetés comme normales. Juste à la prémière époque, le modèle a atteint une précision de 97% . Par ailleurs, lors de l'introduction des valeurs des données d'anomalies, le modèle a atteint une précision incorrecte de 84%, ce qui a conduit de trouver un score de précision de 97% dans l'ensemble. En faisant suite de ces résultats, on conclu que le modèle a correctement identifié les données normales, mais en cas d'anomalies, il les a simplement contournées sans les classer.

### 6.1.1 Performances comparatives de deux travaux sur le Deep Learning

Faisant suite de l'état de l'art de notre travail, l'illustration des précisions entre les deux travaux est décrite dans la figure ci-après :

| Méthode de modélisation | Accuracy_score |
|---|---|
| Notre Deep Learning | 97% |
| Deep Learning par Tamim Mirza(2018) | 99% |

FIGURE 6.1 – Comparaison des précisions

Les résultats de deux travaux divergent, ceci explique qu'il y a plusieurs facteurs qui ont impacté la précision de notre modèle. Nous pouvons cités les facteurs tels que :

— Une architecture séquentielle et l'autre travail[22] à utilisé l'architecture convolutionnelle
— Le travail[22] utilise le modèle pré-formé VGG-19 Keras.
— Le travail[22] utilise le jeu de données de 2012, nous nous avons utilisé le jeu de données de 2017
— Le jeu de données 2017 renferme plus de caractéristiques que celui de 2012.

D'une manière succincte, dans l'ensemble de notre travail, nous avons donc porté notre regard à l'utilisation des données anormales et normales dans nos données d'entraînement. En considérant la nature de données, il est évident de savoir qu'il s'agit d'un problème de détection de valeurs aberrantes.
Dans la phase de test, nous avons obtenu des résultats satisfaisants dont le modèle a pu identifier les données normales avec une précision de 97% et les données d'anomalie avec une précision de 84%. Vu notre ensemble de données, les résultats obtenus étaient satisfaisants.

# 7) CONCLUSION GENERALE

## 7.1 Conclusion

Dans ce travail, nous avons étudié les performances de l'algorithme de l'apprentissage profonds lorsqu'il a été appliqué au domaine de la Cybersécurité pour prédire la détection d'intrusion dans un système informatique. Dans la partie théorique, nous avons montré les limitations des outils traditionnels par rapports à l'apprentissage automatique et l'apport du Deep Learning à la détection d'intrusion. Le Deep Learning et d'autres modèles informatiques tels que ID3, KNN et autres algorithmes ont récemment été appliqués à la modélisation de la détection d'intrusion. L'avantage d'une prévision précise avec des modèles informatiques est de contrôler efficacement les détections d'intrusions dans un réseau informatique à temps réel. Dans la partie pratique, nous avons construit un modèle de Deep Learning, qui exprime au mieux la détection d'intrusion réseau. Compte tenu du double objets et résultats obtenus, nous sommes persuadés avec des identifications réussies sur notre modèle, et nous pouvons affirmer en toute confiance que notre modèle peut être utilisé comme moteur d'arrière-plan d'une application de système de détection d'intrusion, qui peut être monté à la frontière de n'importe quel réseau informatique.

## 7.2 Aperçu succinct et Limitations

Nous avons passé en revue les systèmes des détections d'intrusions sous l'optique outil traditionnel, et analyses avancées(Deep Learning). Faisant suite aux analyses supra, nous avons remarqué que les outils traditionnels possèdent des limitations quant à leurs façons d'être basé sur les règles appliquées à une base des signatures, donc qui nécessite un effort de la programmation, quoi qu'il en soit, cette manière leur rend vulnérables face à l'augmentation exponentielle des attaques. Nonobstant, les systèmes des détections d'intrusions basés sur l'analyse avancée utilisent les techniques de machine learning(Deep Learning), qui fait l'apprentissage machine des différentes caractérisques d'une attaque, selon que le modèle a été bien entrainer. Signalons que l'emploi du Deep learning renferme quelques limitations telles que :

— L'impossibilité de faire une traçabilité de l'attaque ;
— Il se limite à la détection d'intrusion sans protéger le système informatique ;
— La complexité pour sa mise en oeuvre dans le système informatique ;

Au vu de ces quelques limitations, nous pensons que les systèmes des détections d'intrusions avec la machine learning(Deep learning) ne peuvent remplacer en integralité les outils traditionnels, mais apportés un complement à ces derniers. La perfection en terme de sécurité n'existant pas, alors nous avons imaginé que cette manière de faire les choses permettra aux analystes de la sécurité, d'avoir une vue globale du système informatique, toujours sous l'optique «tolérance zéro en termes de sécurité». Et l'intégration du Deep Learning comme module aux outils traditionnels serait un atout pour les systèmes des détections d'intrusions.

## 7.3 Perspectives

Ce travail a été réalisé avec plusieurs contraintes, bien que nos objectifs ont été atteints, nonobstant, il est sage de tester davantage, et si possible de reformer le modèle avec une autre architecture, afin d'obtenir une précision optimale. Etant donné que nos résultats sont satisfaisants, nous pensons davantage à un deploiement de ce module dans n'importe quel niveau d'un réseau informatique, et aussi s'il y a lieu de reformer avec le modèle du nouvel ensemble de données mise à jour publié par l'Institut Canadien pour la Cybersécurité connu sous le nom d'ensemble de données IDS.

# Bibliographie


[1] Aissaoui Sihem, Apprentissage automatique et sécurité des systèmes d'information : Application : un système de détection d'intrusion basé sur les (SVM), Université d'oran,2008

[2] Angelo Rossi, Systèmes de détection d'intrusion pour les réseaux mobiles ad hoc, Éditions universitaires européennes,8 Juillet 2010

[3] Aurélien Géron, Deep Learning avec TensorFlow - Mise en oeuvre et cas concrets, Novembre 2017

[4] Aslihan Ozkaya  Bekir Karlik "Protocole Type Based Intrusion Detection Using RBF Neural Network" International Journal of Artificial Intelligence and Expert Systems(IJAE), volume (3): Issue (4):2012

[5] Bourouh Mouloud, KANOUN ZAKARIA Détection d'intrusions à base des réseaux de neurones et algorithmes génétiques, 2017

[6] Berlin Hervé Djionang Lekagning  Gilbert Tindo, "Vers une nouvelle architecture de détection d'intrusion réseaux à base de réseaux neuronaux", Avril 2016

[7] Cannady,J. , "Artificial Neural Network s for Misuse Detection," Proc eedings, National Information Systems Security Conference (NISSC '98 ), October, Arlington , VA, pp . 443 -456. 1998

[8] Claude Touzet, LES RESEAUX DE NEURONES ARTIFICIELS, INTRODUCTION AU CONNEXIONNISME, 2016

[9] G. DREYFUS "les réseaux de neurones" Mécanique Industriel et Matériaux, n51, septembre 1998

[10] M.DOMINGUEZ Hugo,LE SYSTÈME DE DÉTECTION DES INTRUSIONS ET LE SYSTÈME D'EMPÊCHEMENT DES INTRUSIONS (ZERO DAY), Montréal, Février 2015.

[11] Emmanuel Grolleau, Introduction à l'apprentissage automatique (Machine Learning), Observatoire de Paris-LESIA-Service d'Informatique Scientifique Aout 2017

[12] Iftikhar Ahmad  all "Application of Artificial Neural Network in Detection of Probing Attacks" 2009 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), October 4-6, 2009, Kuala Lumpur, Malaysia

[13] Jean-Pierre Briot, Deep learning, Laboratoire d'Informatique de Paris 6 (LIP6) Université Pierre et Marie Curie (Paris 6) - CNRS,Mars 2017

[14] Judith Hurwitz and Daniel Kirsch, Machine Learning for durmmies, Septembre 2018

[15] M. Khattab Ali  all "The Affect of Fuzzification on Neural NetworksnIntrusion Detection System", IEEE computer society: 2009

[16] Leanid VAITSEKHOVICH, Vladimir GOLOVKO « Employment of neural network baser classifier for intrusion detection » acta mechanica et automatica, vol2, n°4 ,2008.

[17] Mehdi MORADI and Mohammad ZULKERNINE, "A Neural Network based System for intrusion detection and Classification of Attacks" In 2004 IEEE International on Advances in Intelligent Systems

[18] Muna Mhammad  Monica Mehrotra "Design Network Intrusion Detection System using Hybrid Fuzzy-Neural Network" International Journal of Computer Science and Security, volume (4): Issue (3): 2010

[19] Olivier Markowitch  Gianluca Bontempi, Systèmes de détection d'intrusion basés sur du machine Learning, Université Libre de Bruxelles, Faculté des Sciences Département d'Informatique, Septembre 2015

[20] Rémy Sun, « Apprentissage profond et acquisition de représentations latentes de séquences peptidiques », Campus de Beaulieu, Rennes, France 28 février 2017

[21] Sharafaldin;Habibi Lashkari, A. et Ghorbani, A. (2018). « Vers la génération d'un nouveau jeu de données de détection d'intrusion et la caractérisation du trafic d'intrusion ».

[22] Tamin Mirza, « Construire un système de détection d'intrusion en utilisant le Deep Learning »,https://towardsdatascience.com/building-an-intrusion-detection-system-using-deep-learning-b9488332b321, 08 Septembre 2018

[23] Wikipédia, Définition cybersécurité.

[24] Yann LeCun, Yoshua Bengio et Geoffrey Hinton. Deep Learning. Nature, vol. 521, pages 436–444, 2015.http://dx.doi.org/10.1038/nature14539

[25] Yoshua Bengio, La révolution de l'apprentissage, La révolution de l'apprentissage profond - Interstices.html, Janvier 2019

[26] ousef Farhaoui, « Evaluation des systèmes de détection et de prévention des intrusions et la conception d'un BIDS », thèse de doctorat, Université Ibn Zohr, 2012