الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et de Technologie
Département de Mathématiques et de l'Informatique

Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en : Informatique
Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Réseaux et ingénierie des données (RID)

Thème

Détection d'intrusions informatiques via le deep learning.

Présenté par :

1) Mr. Madouni Riad
2) Mr. Hadj Said Houcine Zakaria

Devant le jury composé de :

Dr Bouafia Zoheir	MAA	UAT.B.B (AinTemouchent)	Président
Dr Benomar Mohammed Lamine	MCA	UAT.B.B (AinTemouchent)	Examinateur
Dr Belgrana Fatima Zohra	MCA	UAT.B.B (AinTemouchent)	Encadrant

Année Universitaire 2022/2023

Remerciement

Nous commençons par remercier Dieu du fond du cœur de nous avoir donné la force et l'endurance nécessaires pour terminer ce travail. Nous remercions sincèrement docteur Belgrana Fatima Zohra pour sa supervision, sa direction et son soutien continu exceptionnel au cours de ce projet de recherche. Leurs connaissances et son engagement ont été cruciaux dans la formation de notre travail. Nous tenons aussi à remercier les membre de jurys monsieur Bouafia Zoheir et docteur Benomar Mohammed Lamine d'avoir accepter de juger notre travail.

Nous apprécions énormément l'amour, le soutien et la foi de nos parents, aussi nous tenons également à exprimer notre gratitude à nos amis, collègues et étudiants pour leur soutien et leurs conseils avisés tout au long du processus. Leurs efforts ont approfondi notre compréhension et nous ont aidés à atteindre nos objectifs.

Nous remercions sincèrement l'administration de l' Université Belhadj Bouchaib, les employés du département et tous ceux qui nous ont aidés à progresser sur le plan académique. Nous apprécions grandement le travail qu'ils ont fait pour créer un environnement d'apprentissage positif.

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Acronymes

Acronyme	Description
SSI	Sécurité des Systèmes d'Information
SI	Systèmes d'Information
SDI	Système de détection d'intrusion
IA	intelligence artificielle
AA	Apprentissage automatique
AP	Apprentissage profond
RN	réseaux de neurones
RNA	réseaux de neurones artificiels
PMC	Perceptron Multicouches
RPV	Réseau Privé Virtuel
SDIH	SDI basé sur l'Hôte
SDIR	SDI basé sur le Réseau
RNR	Les réseaux de neurones récurrents
RNC	Réseaux de Neurones Convolutifs
TVi	Transformateur de Vision
RNA	Les Réseaux de Neurones Artificiels
MCLT	Mémoire à Court et Long Terme
TLN	Traitement de Langage Naturel
SDIR	Système détection d'intrusion réseau basé sur réseau
SDIH	Système détection d'intrusion basé sur l'hôte
RVB	Rouge vert bleu
VP	vrai positif
VN	vrai négatif
FP	faux positif
FN	faux négatif

Tab. 1 : Acronymes et Descriptions.

Acronyms

Acronyms	Description
UNet	U Network
IDS	Intrusion detection system
BDA	Before, During and After
ML	Machine Learning
DP	Deep learning
MLP	Multi Layer Perceptron
DoS	Denial of service
DDOS	Distributed denial of service
Resnet	Residual Network
SVM	Support vector machines
LSTM	Long short term memory
CNN	Convolutional Neural Network
KNN	k-nearest neighbors
DPC	Dropout Probabilistic Confidence
ViT	Vision Transformer
U2R	Remote to user
R2L	Remote-to-Local
RGB	Red Green Blue
API	Application Programming Interface

TAB. 2 : Acronymes et Descriptions en anglais.

Introduction Générale

Internet, un vaste réseau de périphériques tous connectés les uns aux autres, a révolutionné notre façon de vivre, travailler, de communiquer et d'accéder à l'information. Ce qui a permis un grand nombre d'avantages. Internet contient en contrepartie des inconvénients, notamment en matière de sécurité, comme toute technologie.

La sécurité sur Internet constitue l'un des principaux enjeux, la protection des réseaux et des terminaux sont devenus des défis cruciaux en raison de la croissance explosive du monde en ligne et de la dépendance croissante aux plateformes numériques. Internet met à disposition aux pirates de nouvelles méthodes pour menacer les systèmes, de lancer des attaques et d'exploiter les faiblesses. Les répercussions de ces menaces peuvent être chaotiques.

Plusieurs stratégies ont été proposées pour contrer ces risques de sécurité. Le pare-feu, un mur qui filtre le trafic réseau et régule l'accès en fonction de règles de sécurité établies, est l'un des types de défense les plus utilisés. Bien qu'ils puissent enlever l'accès illégal à un réseau, les pare-feu seuls ne sont pas efficaces. Ils peuvent être dépassés par quelques attaques, ce qui permet l' infiltration facile des pirates dans le system et causer le chaos, les proxys et les réseaux privés virtuels sont vulnérables et ne sont pas efficaces contre de nombreuses attaques à leur tour.

Le domaine de la cyber sécurité a connu une autre technologie de sécurité, qui est venu renforcé celles qui existent déjà , il s' agit des systèmes de détection d'intrusion (SDI).

Les approches de l'intelligence artificielle (IA) sont souvent utilisées par les SDIs pour améliorer leurs performances et leur efficacité. Les SDIs sont capables de reconnaître et de répondre plus efficacement aux menaces potentielles en apprenant des modèles et des comportements à partir de grands ensembles de données qui incluent à la fois des scénarios normaux et d'attaque. Les algorithmes d'apprentissage automatique des SDI peuvent évoluer et progresser au fil du temps, ce qui les rend plus aptes à identifier de nouvelles voies d'attaque.

Pourtant, la création d'un SDI réussi ne peut se faire sans obstacles. Il faut beaucoup d'investigation, d'analyse et de collecte de données pour en développer un qui soit fiable. Pour ce faire, de nombreux ensembles de données représentant différentes activités réseau et scénarios d'attaque doivent être rassemblés. Pour que les modèles d'IA soient formés avec précision, ces données doivent être méticuleusement nettoyées, traitées et étiquetées. La procédure peut prendre beaucoup de temps et de ressources et nécessite souvent beaucoup de connaissances et de puissance de calcul.

Notre approche proposée est basée sur l' intelligence artificielle, et plus particulièrement l' apprentissage profond, où nous avons opté pour l' algorithme transformateur de vision (Vision transformer en anglais), nous avons utilisé deux bases de données qui sont NSL KDD et CIC 2017. La base CIC 2017 est beaucoup plus importante, nous avons dû faire recoure au sous-échantillonnage afin de réduire la base d' apprentissage optimiser en temps d' exécution et pour pallier les problèmes de capacité mémoire.

Notre mémoire est constitué de 3 chapitres, dans le premier chapitre nous présentons quelques notions de base sur la sécurité informatique. Dans le deuxième chapitre, nous discutons des réseaux de

neurones, quelques approches de l' apprentissage profond, nous présenterons également un état de l' art sur les différentes techniques utilisées en vue d' une réalisation d' un SDI en utilisant l' IA.

Dans le dernier chapitre, nous introduisons notre approche proposée où nous présentons en détail les différentes étapes nécessaires pour réaliser notre SDI dans une démarche en quête d' une amélioration des performances de détection. Nous discutons à la fin de ce chapitre des résultats obtenus et une comparaison afin de démontrer l' efficacité de notre SDI. Nous clôturons ce mémoire avec une conclusion générale.

Chapitre 1

Sécurité Informatique

1.1 Introduction

Les réseaux informatiques sont des systèmes qui connectent des ordinateurs et des dispositifs ensemble pour permettre la communication et le partage de données, d'informations et de ressources. Bien qu'ils soient très utiles et largement utilisés, les réseaux informatiques sont également vulnérables à de nombreuses menaces de sécurité telles que les virus informatiques, les attaques par déni de service (DoS), les vols de données et les piratages.

La vulnérabilité des réseaux informatiques peut être attribuée à plusieurs facteurs, tels que la mise en œuvre incorrecte de la sécurité, la configuration défectueuse des systèmes, la défaillance des logiciels et la perte ou le vol de matériel. Les attaquants peuvent utiliser ces vulnérabilités pour accéder aux données confidentielles et sensibles, altérer ou détruire les systèmes informatiques, ou utiliser les réseaux pour diffuser des virus et des logiciels malveillants.

Il est donc crucial de prendre des mesures de sécurité pour protéger les réseaux informatiques contre les menaces potentielles. Cela peut inclure plusieurs méthodes qui peuvent minimiser les risques pour les systèmes informatiques et les données sensibles, et garantir que les réseaux continuent à fonctionner de manière efficace et sécurisée.

Dans ce premier chapitre nous allons voir quelques notions de base sur la sécurisé informatique ou nous allons explorer les différents type d' attaque ainsi que quelques moyens de protection et plus particulièrement les systèmes de détection d' intrusions.

1.2 La sécurité

La Sécurité des Systèmes d'Information (SSI) fait référence à la protection des systèmes d'information contre l'accès non autorisé, l'utilisation, la divulgation, la perturbation, la modification ou la destruction, afin d'assurer la confidentialité, l'intégrité et la disponibilité des informations, C' est une discipline de première importance car le Système d' Information (SI) est pour toute entreprise un élément absolument vital, la sécurité de ces derniers est devenue une préoccupation majeure. [Rivet et Alain, 2007].

La cybersécurité consiste alors à protéger les systèmes informatiques, les réseaux et les actifs numériques contre les accès non autorisés, le vol ou les dommages. Elle vise à garantir la confidentialité, l'authenticité, l'intégrité, la disponibilité et la non-répudiation des données grâce à une gamme de pratiques et de technologies. Une cyber sécurité efficace nécessite une approche globale qui prend en compte tous les aspects de l'écosystème numérique d'une organisation.

- Confidentialité : seules les personnes autorisées peuvent avoir accès aux ressources et bloquer l' accès aux autres.
- Authenticité : cet objectif garantis à l' expéditeur que le destinataire est bien celui qui veut lui envoyer des données.
- L' intégrité : cet objectif assure que le message n' a pas été modifié.
- La disponibilité : l'accès aux ressources du système d'information doit être permanent et sans faille.
- La non-répudiation : consiste à assurer qu' aucun des utilisateurs peuvent contester les opérations qu'ils ont réalisées.

Deux axes fondamentaux fon partie de la sécurité à savoir la cryptographie et le BDA (Before, During and After).

1.2.1 La cryptographie

La cryptographie est une pratique ancienne qui remonte à plusieurs siècles. C'est une technique utilisée pour sécuriser les informations en les rendant illisibles pour quiconque n'a pas la clé de déchiffrement appropriée. Elle était initialement employée pour protéger les secrets militaires, la correspondance diplomatique et les lettres privées. Cependant, l'utilisation de la cryptographie est de plus en plus courante dans les systèmes informatiques modernes pour la sécurité des données, y compris dans les systèmes de détection d'intrusion (SDI), où il est crucial de protéger les données sensibles collectées et stockées par ces systèmes.

BDA :

pour début, pendant, et après permet de minimiser les conséquences d'une attaque informatique et de reprendre rapidement les opérations normales. En élaborant un plan de réponse aux incidents bien conçu est crucial pour les organisations afin de protéger leurs systèmes et données sensible

- Début : Cette phase consiste à se préparer en cas d'attaque en établissant des politiques et des procédures de sécurité. Les actifs critiques sont identifiés et classés par ordre de priorité, et des mesures de sécurité sont mises en place pour prévenir et détecter les cyber-attaques. En somme, cette phase est cruciale pour anticiper les incidents de sécurité et agir rapidement en cas d'urgence.
- Pendant : Cela implique l'activation de l'équipe de réponse aux incidents, la collecte et l'analyse d'informations sur l'attaque, et la limitation et l'éradication de la menace pour empêcher d'autres dommages.

- Après : Cette phase est celle où l'on analyse l'incident, récupère les données et produit un rapport. On évalue la réponse à l'incident, on identifie les points d'amélioration et on met en place des changements pour éviter que des incidents similaires ne se reproduisent.

1.3 Catégories d'attaques

Les attaques informatiques peuvent causer des dommages considérables, tels que la perte de données confidentielles, la perturbation des activités commerciales et l'obtention d'un accès au système. Il est donc important de comprendre les différentes catégories d'attaques qui peuvent se produire (voir la figure 1.1 ci-dessous), à savoir :

1.3.1 L'interruption des données

L'interruption des données fait référence à un type d'attaques qui se produit lorsqu'il y a une demande d'entrée ou de sortie de données provenant d'un contrôleur d'entrée/sortie, Les attaques d'interruption dans les réseaux sont des attaques qui visent à perturber le fonctionnement (la disponibilité) normal des réseaux en utilisant divers moyens, tels que les dénis de service distribués (DDOS).

1.3.2 L'interception des données

L'interception de données est l'un des vecteurs d'attaque possibles au niveau des canaux de communication, ce qui nécessite que l'attaquant obtienne l'accès au canal de transmission afin de capturer et potentiellement modifier les messages échangés, Cette attaque peut être réalisée en utilisant des techniques telles que la mise en place de points d'accès malveillants, l'interception de paquets de données, ou l'utilisation de logiciels malveillants pour espionner les activités des utilisateurs sur le réseau.

1.3.3 Modification des données

La modification des données fait référence à l'acte de modifier ou de changer le contenu des informations dans un système informatique, Il s'agit d'une attaque portée à l'intégrité, les données ne doivent pas être altérées de façon fortuite, illicite ou malveillante pour ne pas endommager les données ou changer leurs significations.

1.3.4 Fabrication des données

La fabrication de données peut être réalisée par l'insertion de messages dans un canal de communication, permettant à un attaquant de créer et d'envoyer de fausses informations, Cette type d'attaque vise à compromettre la sécurité des réseaux en injectant des composants malveillants dans les équipements réseau. Cette forme d'attaque peut entraîner une interruption du service, une perte de confidentialité et d'intégrité des données. [Sang et Fernand, 2012].

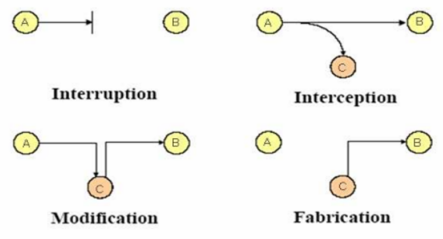

Fig. 1.1 : Les principales catégories d' attaques [Dumont, 2009].

1.4 Outils de Sécurité

Pour se protéger contre ces attaques, il est important d'utiliser des outils de sécurité qui ont pour objectif de minimiser les risques d' intrusions que ça soit dans les machines hôtes ou bien le réseau où se trouve ces machines. Il existe plusieurs outils de sécurité, parmi ces outils on mentionnera :

- Anti-virus
- Pare-feu
- Serveur proxy
- VPN

1.4.1 Antivirus

Un Antivirus est un programme capable de détecter les virus, les vers, les troyens et parfois les spywares qui peuvent infecter un ordinateur et nuire à ses performances, voire le rendre inutilisable. Les antivirus fonctionnent en utilisant une base de données de signatures de virus, en examinant les fichiers pour détecter toute correspondance avec les virus connus. [Salma et Tidjani, 2015].

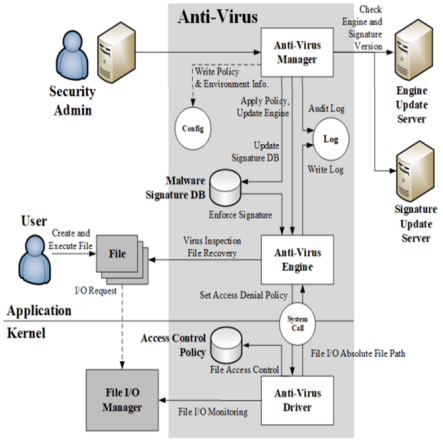

Fig. 1.2 : L' architecture d' un antivirus [Han et al., 2020].

1.4.1.1 Fonctionnement d' un antivirus

Les antivirus fonctionnent de différentes manières, mais la plupart utilisent une combinaison de méthodes pour protéger leurs appareils contre les logiciels malveillants. Les méthodes courantes comprennent :

- Analyse de la mémoire : l'antivirus examine la mémoire de votre ordinateur pour détecter les logiciels malveillants actifs qui pourraient causer des problèmes.

- Analyse du disque dur : l'antivirus analyse les fichiers stockés sur votre disque dur pour détecter les virus connus.

- Analyse du disque dur : l'antivirus analyse les fichiers stockés sur votre disque dur pour détecter les virus connus.

- Analyse comportementale : l'antivirus analyse le comportement des programmes sur votre ordinateur pour détecter les activités suspectes.

1.4.1.2 Ces limitations

Les antivirus sont des outils très utiles pour protéger nos ordinateurs contre les virus, Il ne faut pas croire qu'ils sont infaillibles, malgré tout. Les antivirus ne peuvent détecter que les virus qu'ils connaissent déjà, ce qui signifie que les pirates peuvent facilement créer de nouveaux virus que les antivirus ne peuvent pas encore détecter. De plus, il y a beaucoup de formes de logiciels malveillants différents que les antivirus n' arrivent pas à détecter, tels que les logiciels espions ou les ransomwares. Il est essentiel de se souvenir que les antivirus ne suffisent pas à garantir la sécurité de nos ordinateurs et de nos données. Il est important de mettre en place d'autres mesures de protection pour assurer leur sécurité.

1.4.2 Pare-feu

Les pare-feux sont des systèmes de sécurité réseau largement utilisés pour filtrer le trafic de données entrant et sortant afin de protéger un hôte ou un système de réseau contre les anomalies de réseau et les attaques malveillantes (voir figure 1.3) [SUNGHWAN et al., 2020].

Cela est réalisé en employant des règles de filtrage du trafic entrant et sortant du réseau, qui sont spécifiées par l'administrateur du réseau pour déterminer quelles communications sont autorisées et quelles communications sont bloquées.

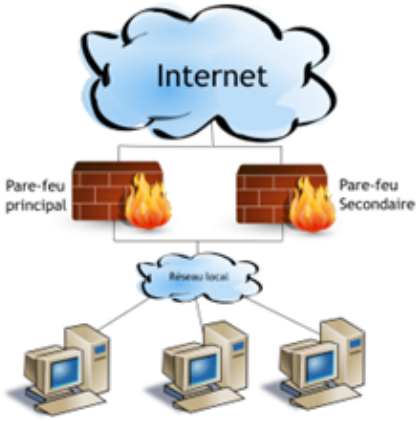

FIG. 1.3 : L' architecture d' un pare feu [Herrb, 2022].

1.4.2.1 Fonctionnement du Pare-feu

Les principales taches d' un pare-feu sont comme suit :

- Analyse de chaque paquet de données entrant ou sortant du réseau.
  - Origine ou la destination des paquets (adresse IP, port TCP).
  - Origine ou la destination des paquets (adresse IP, port TCP).
  - Les Utilisateurs eux-mêmes. [Prieur et Benoît, 2020].
- Décision de bloquer ou d' accepter le paquet en fonction des règles définies.
- Possibilité de réaliser des analyses plus avancées, telles que l'inspection du contenu pour détecter les virus et les logiciels malveillants.
- Vérification de l'identité de l'utilisateur pour s'assurer qu'il a l'autorisation d'accéder aux ressources du réseau.

1.4.2.2 Ces limitations

Les pare-feux sont des gardiens indispensables pour protéger les réseaux informatiques, mais ils ont leurs limites, les pare-feu ne peuvent protéger que contre les attaques qui passent par les connexions entrantes et sortantes. Ils peuvent également être contournés si un attaquant parvient à pénétrer dans le réseau.

1.4.3 Le serveur proxy

Un serveur proxy est essentiellement un ordinateur connecté à Internet qui est équipé de sa propre adresse IP. Au lieu que l'utilisateur se connecte directement au site Web prévu, il envoie sa demande à un serveur proxy qui évalue la demande puis la complète(voir figure 1.4) [Kocaleva et al., 2022].
Un serveur proxy agit comme un intermédiaire entre l' ordinateur client et les sites Web visités, il permet de protéger la vie privée des internautes en cachant leur adresse IP et leur emplacement réels.

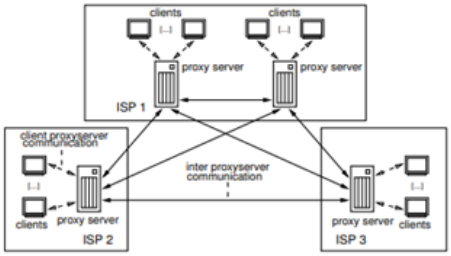

FIG. 1.4 : L' architecture d' un serveur proxy [Muller et al., 2004].

1.4.3.1 Fonctionnement d'un serveur proxy

La procédure de fonctionnement d' un serveur proxy est comme suit :

- L'utilisateur envoie une demande à un serveur cible.
- La demande est redirigée vers un serveur proxy.
- Le serveur proxy envoie une nouvelle demande au serveur cible, en utilisant sa propre adresse IP.
- Le serveur cible renvoie la réponse au serveur proxy, qui la transmet ensuite à l'utilisateur.

Ces étapes permettent à l'utilisateur de rester anonyme et de protéger sa confidentialité en ligne, car le serveur proxy masque l'adresse IP réelle de l'utilisateur et remplace l'adresse IP d'origine par la sienne. De plus, le serveur proxy peut effectuer des tâches de filtrage et de contrôle d'accès, ce qui améliore la sécurité et les performances du réseau.

1.4.3.2 Ces Limitations

Ces étapes permettent à l'utilisateur de rester anonyme et de protéger sa confidentialité en ligne, car le serveur proxy masque l'adresse IP réelle de l'utilisateur et remplace l'adresse IP d'origine par la sienne. De plus, le serveur proxy peut effectuer des tâches de filtrage et de contrôle d'accès, ce qui améliore la sécurité et les performances du réseau.

1.4.4 Réseau Privé Virtuel (RPV)

Avec la montée en puissance des cyberattaques et de la surveillance en ligne, l'utilisation d'un RPV ou encore VPN pour virtual private Network, est devenue de plus en plus courante pour protéger la vie privée et la sécurité des utilisateurs sur Internet. En effet, un RPV est un outil qui permet de crypter toutes les communications entre un ordinateur et Internet, rendant ainsi impossible pour quiconque d'intercepter ou d'espionner ces communications. C' est un réseau privé au sein d'une infrastructure de réseau public qui permet un accès contrôlé aux connexions pairs au sein d'une communauté d'intérêt définie, De cette façon, les utilisateurs peuvent surfer sur Internet en toute sécurité et confidentialité, sans craindre que leurs informations personnelles ne soient exposées. [Ferguson et al., 1998].

1.4.4.1 Fonctionnement du RPV

Les RPVs créent une connexion sécurisée entre le client utilisateur et le point distant. Le trafic est crypté pour le protéger des autres. Tout cela se passe via les serveurs RPV, vers lesquels le trafic Internet de l'utilisateur est envoyé avant d'atteindre sa destination (voir figure 1.5). Cela permet d'utiliser les connexions Wi-Fi publiques de manière plus sûre [Antonio et al., 2022].

Le fonctionnement du RPV suit ces étapes :

- Lorsqu'un utilisateur se connecte à un RPV, son trafic Internet est acheminé vers un serveur RPV situé dans un autre pays.
- Le serveur RPV crypte le trafic de l'utilisateur avant de le renvoyer à Internet.
- L'adresse IP visible par les sites web est celle du serveur RPV, plutôt que celle de l'utilisateur.
- Cela permet à l'utilisateur de surfer sur Internet de manière plus anonyme et de contourner les restrictions géographiques.

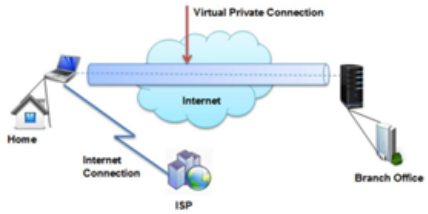

Fig. 1.5 : Architecture du RPV [Sistani et Patel, 2016].

1.4.4.2 Ces limitations

Les RPVs sont de plus en plus utilisés pour protéger la vie privée et assurer la sécurité des données en ligne. Cependant, il est important de reconnaître leurs limites. Tout d'abord, bien que les RPVs soient très sûrs, Aucune protection des données entre les stations et les pare-feu dans le cas d' un VPN site à site car le tunnel entre eux n'est installé qu'entre deux pare-feu. De plus, les RPVs peuvent ralentir la vitesse de navigation, car les données doivent être acheminées via un serveur distant et les fournisseurs de RPVs ont parfois accès aux données des utilisateurs. [MAHREZ et al., 2020].

1.5 Les Systèmes de Détection d' Intrusion (SDIs)

Les systèmes de détection d'intrusion sont des outils essentiels dans le domaine de la cyber-sécurité, Conçus pour surveiller les réseaux ou les systèmes informatiques à la recherche de signes d'activité malveillante ou de violations des politiques de sécurité. Avec la fréquence et la complexité croissantes des menaces informatiques, les SDI sont devenus un élément essentiel de toute stratégie de sécurité informatique complète. Ces outils puissants sont conçus pour détecter des comportements suspects ou des violations de politique et alerte le personnel de sécurité ou prend des mesures automatisées lorsqu'une telle activité est détectée, Lorsqu'un SDI identifie une intrusion potentielle, il signale immédiatement

l'incident à l'équipe de sécurité informatique, ce qui permet une réponse rapide et efficace.
Il est donc crucial de comprendre le fonctionnement des SDIs, leur architecture et leurs capacités de détection pour les utiliser efficacement dans un environnement de sécurité informatique. [Liao et al., 2013].

1.5.1 Types des SDIs

Nous allons approfondir notre compréhension des systèmes de détection d'intrusion (SDI) et nous concentrer sur l'exploration des différents types d'SDI. Nous allons examiner en détail les forces et les faiblesses de chaque type d'SDI afin de comprendre comment chaque méthode de détection peut être adaptée pour répondre aux besoins spécifiques d'une organisation.

Il existe plusieurs critères permettant la classification des SDIs, à savoir la méthode de détection, les types de données analysées, l'emplacement du SDI dans l'architecture du réseau, le niveau de granularité de la surveillance et le type d'analyse effectuée par le SDI Nous allons voir dans ce qui suit quelques types D' SDI en considérant les critère suivants : La méthode de détection utilisée, l'emplacement d'installation du SDI (hôte ou réseau), le type de trafic surveillé et la manière dont le SDI gère les événements détectés.[Liao et al. 2013].

A. SDI basé sur les signatures : ces SDIs reposent sur des modèles préconfigurés d'attaques connues. L'SDI compare le trafic réseau ou l'activité du système à la base de données de signatures pour identifier toute correspondance. Cette approche est utile pour détecter les attaques connues, mais elle peut ne pas être efficace contre les menaces nouvelles ou émergentes.

B. SDI basé sur les anomalies : ces SDIs sont également appelés SDI basés sur le comportement, ils reposent sur des techniques statistiques ou d'apprentissage automatique pour établir une ligne de base de l'activité normale du réseau ou du système. Toute déviation par rapport à la ligne de base est signalée comme une intrusion potentielle. Cette approche peut détecter des attaques précédemment inconnues (des attaques zero-day). Cependant, elle peut générer des faux positifs ou manquer les attaques sophistiquées.

C. SDI basé sur l'Hôte(SDIH) : les SDIHs est une technologie IDS qui surveille et collecte les caractéristiques pour les hôtes contenant des informations sensibles. L'SDI peut détecter une activité suspecte en utilisant un agent pour surveiller et analyser les activités, telle que des modifications de fichiers non autorisées, des modifications de la configuration du système ou de nouveaux processus en cours d'exécution.

D. SDI basé sur le Réseau(SDIR) : ces capteurs SDIRs sont installés sur un dispositif réseau à des points stratégiques pour capturer et analyser le trafic à mesure qu'il passe, L'SDIR peut détecter une large gamme d'attaques, y compris les scans de port, les attaques par déni de service (DoS) et les infections de logiciels malveillants qui sont transmises sur le réseau, ils sont utiles pour surveiller de grands réseaux, mais peuvent générer des faux positifs ou manquer des attaques qui utilisent le chiffrement, SDIRs peut également être utilisé pour surveiller le trafic sortant afin de détecter les tentatives d'exfiltration de données.

E. SDI hybride : c' est un type de système de détection d'intrusion qui combine deux ou plusieurs des techniques SDI pour fournir une capacité de détection complète. L'SDI hybride peut exploiter les forces de chaque type d'SDI et atténuer leurs faiblesses. Par exemple, un SDI hybride peut utiliser un SDI basé sur des signatures pour détecter les attaques connues et un SDI basé sur les anomalies pour détecter les attaques inconnues. De cette façon, l'SDI hybride peut être plus efficace que tout type d'SDI individuel, car il peut détecter un plus grand nombre de menaces de sécurité. En utilisant un mélange de techniques, l'SDI hybride peut également minimiser les fausses alarmes et fournir une détection plus précise des activités malveillantes.

Les différents types des SDI offrent différentes capacités pour faire face à différentes menaces et environnements, Chaque type d' SDI offre des capacités uniques pour détecter et prévenir les intrusions selon les besoins de l'entreprise et les menaces actuelles, Tous ces types d' SDI sont des composants clés d'une stratégie globale de défense de la cyber-sécurité.

Le SDI peut également être classé en tant que SDI passif ou actif. Sans intervenir activement dans le réseau ou le système, les SDI passifs se contentent d'observer le trafic et de signaler tout événement ou anomalie suspecte. Ils sont généralement utilisés pour la collecte de données et la surveillance à des fins d'audit ou de conformité réglementaire. En revanche, les SDI actifs peuvent agir pour bloquer ou neutraliser les menaces identifiées. Par exemple, ils peuvent bloquer l'accès à une adresse IP douteuse ou mettre fin à une connexion malveillante. Pour protéger un réseau ou un système, les SDI actifs peuvent être plus efficaces.

1.5.2 Types d'intrusions

Un SDI est capable de détecter différents types d'intrusions, tels que des scans de ports, des attaques par déni de service (DoS) et des infections par des logiciels malveillants.

A. Les scans des ports : ce sont une technique utilisée pour repérer les ports ouverts sur un système informatique, ils sont souvent utilisés par les pirates informatiques pour repérer des vulnérabilités exploitables sur un système, les SDI peuvent détecter ces scans en surveillant le trafic réseau et en identifiant les demandes de communication sur des ports inhabituels ou inconnus, Ces scans de ports peuvent causer des problèmes de disponibilité du réseau en surchargeant le trafic, ce qui peut ralentir ou arrêter le fonctionnement normal du réseau. Mais grâce à une surveillance et à une détection précoce efficaces des SDIs, les scans de ports peuvent être repérés avant qu'ils ne causent des dommages réels. [De Vivo et al., 1999].

B. Les attaques par déni de service (DoS) : sont des attaques contre la disponibilité, qui cherchent à empêcher les utilisateurs légitimes d'accéder au réseau. Les attaques DoS sont différentes des comportements égoïstes motivés par un résultat bénéfique possible. En raison de la nature de diffusion des réseaux sans fil, les attaques DoS sont faciles à mener, en particulier dans le domaine sans fil. De plus, il existe de nombreuses vulnérabilités DoS spécifiques à la norme 802.11 démontrées expérimentalement dans la littérature ces dernières années. Un scénario générique d'attaque DoS dans un réseau 802.11 représentatif est présenté dans la figure 1.6 [Kemal et Bulent, 2009].

FIG. 1.6 : Un scénario générique d'attaque DoS sur un réseau 802.11. [Kemal et Bulent, 2009].

C. Les logiciels malveillants : Un logiciel malveillant est un programme informatique quiréalise volontairement une action allant à l' encontre de l' intéré et de l' utilisateur. tels que les virus, les chevaux de Troie et les ransomwares sont des infections qui se produisent souvent, lorsque des programmes malveillants sont installés sur des ordinateurs à l'insu des utilisateurs, ou lorsqu'ils sont utilisés pour accéder à des informations sensibles. C'est là que les SDIs interviennent, ils peuvent détecter ces infections en surveillant les activités suspectes sur un système, telles que la modification de fichiers système ou l'installation de logiciels malveillants. [Calvet et Joan, 2013].

Pour prévenir des d'intrusions, les SDIs peuvent être configurés pour bloquer le trafic malveillant, envoyer des alertes ou déclencher des mesures de sécurité supplémentaires pour empêcher les intrusions de causer des dommages. En détectant les intrusions avant qu'elles ne puissent causer des dégâts.

1.5.3 Architecture et composants d'un SDI

L'architecture d'un SDI est généralement composée de trois éléments clés : les capteurs, les analyseurs et les interfaces utilisateur.

- Capteur : un logiciel générant des événements en filtrant et en formatant les données brutes provenant d'une source de données. Les capteurs sont conçus pour capturer et collecter des informations sur le trafic réseau à partir de différents points du réseau, tels que les commutateurs, les routeurs et les pare-feu.

- Analyseur : c'est un outil matériel ou logiciel qui met en œuvre l'approche choisie pour la détection d'intrusion en utilisent des algorithmes et des règles prédéfinies pour analyser les données collectées par les capteurs et identifier les comportements suspects. Lorsqu'un comportement suspect est identifié, il génère des alertes lorsqu'il détecte une intrusion.

- Les interfaces utilisateur : sont des outils logiciels utilisés par les équipes de sécurité pour gérer les alertes, C'est l'unité élémentaire utilisée pour représenter un élément d'un scénario d'attaque. ça permet aux équipes de sécurité de surveiller et d'analyser les données collectées par les capteurs.

En somme, l'architecture et les composants d'un SDI collaborent pour surveiller le trafic réseau, repérer les comportements inhabituels et alerter rapidement les équipes de sécurité informatique. Grâce à cette réactivité, ces dernières sont en mesure de réagir rapidement et de protéger le réseau et les systèmes contre les potentielles menaces de sécurité.[Bouzayani et Hatem, 2012].

1.5.4 L'intégration des SDIs avec d'autres systèmes de sécurité

Un SDI est un composant important de l'infrastructure de sécurité globale d'une entreprise. Il fonctionne en surveillant le trafic réseau et en l'analysant pour identifier toute activité malveillante ou suspecte. Cependant, un SDI seul peut ne pas être suffisant pour protéger entièrement un réseau contre les différentes menaces cybernétiques. Pour améliorer son efficacité, un SDI doit être intégré à d'autres systèmes de sécurité tels que les antivirus, les pare-feu, les serveurs proxy, etc, L'intégration des SDIs avec d'autres systèmes de sécurité, d' autres SDIs ou même une hybridation d' approches et de méthodes complémentaires permet plusieurs avantages tels que :

- L' amélioration de la détection des menaces : en combinant les capacités de plusieurs systèmes de sécurité, les organisations peuvent améliorer leur capacité à détecter les menaces et à y répondre efficacement.

- Réduction des faux positifs : en utilisant plusieurs systèmes de sécurité, le nombre de faux positifs peut être réduit grâce à la combinaison de différentes méthodes de détection de menaces.

- Réponse aux incidents plus rapide : les systèmes intégrés peuvent travailler ensemble pour isoler automatiquement les systèmes affectés, bloquer les attaquants et informer rapidement les équipes de sécurité.

- Analyse globale des menaces : l'intégration des SDIs avec d'autres systèmes de sécurité permet aux organisations de réaliser une analyse complète des menaces, ce qui leur permet d'identifier et de résoudre les éventuelles faiblesses de sécurité dans l'ensemble du réseau.

L'intégration des SDIs avec d'autres systèmes de sécurité peut améliorer la sécurité globale d'un réseau et améliorer les temps de réponse aux incidents. La maintenance et les mises à jour régulières sont également importantes pour s'assurer que le SDI est à jour et efficace pour détecter les nouveaux types d'intrusions.

1.6 Conclusion

Dans ce chapitre, nous avons examiné les notions de base de la sécurité informatique, les catégories d'attaques, les moyens de se protéger contre ces attaques, ainsi que les risques liés aux réseaux et leurs vulnérabilités. Un aspect clé de la sécurité informatique est la détection et la prévention des attaques, il est crucial d'avoir une stratégie de sécurité robuste qui inclut divers outils et techniques, tels que les pare-feu, les logiciels antivirus, les VPN et les SDI pour se défendre contre différents types d'attaques.

Les SDIs sont des outils qui aident à détecter les activités suspectes sur un réseau en surveillant le trafic et en signalant les événements potentiellement dangereux. Ils fournissent une couche de défense supplémentaire contre les attaques. Les SDIs peuvent être configurés pour générer des alertes, des journaux ou même prendre des mesures automatiques lorsqu'une activité suspecte est détectée. Il est essentiel de configurer les SDIs avec soin pour éviter les faux positifs, qui peuvent conduire à des alertes inutiles et à une utilisation inefficace des ressources.

La protection des systèmes informatiques contre les attaques nécessite une combinaison de mesures et d'efforts continus pour rester informé des dernières tendances en matière de sécurité. En mettant en place des mesures de sécurité appropriées et en restant informé, les organisations et les individus peuvent réduire les risques et assurer la sécurité de leurs systèmes et de leurs données.

Dans le prochain chapitre, nous approfondirons l'intersection entre les SDI et l'apprentissage profond est l'état de l'art. Nous explorerons les différentes techniques utilisées dans l'apprentissage profond et leur application dans l'analyse des données en utilisant les SDI.

Chapitre 2

Les SDIs et l' apprentissage profond

2.1 Introduction

L' apprentissage automatique (AA, ou ML pour Machine Learning) est issu de l' intelligence artificielle, l' apparition des grandes masses de données a donnée à son tour naissance à l' Apprentissage Profond (AP, ou DP pour Deep Learning) qui est largement utilisé ces derniers temps dans divers domaines, tel que la reconnaissance vocale, le traitement du langage naturel, la vision par ordinateur, la robotique, etc.

L' AP consiste à apprendre aux Réseaux de Neurones (RN) à effectuer des tâches difficiles en utilisant une énorme quantité de données. Le terme "profond" décrit comment ces réseaux de neurones comprennent souvent de nombreuses couches, ce qui leur permet d'apprendre des représentations de données complexes.

Dans ce chapitre nous allons d' abord explorer l' ingrédient clé de l'apprentissage profond qui est les réseaux de neurones, puis nous allons présenter quelques modèles de l' AP. Ce dernier a été notamment utilisé pour améliorer la détection et la prévention des intrusions informatiques, raison pour laquelle nous allons présenter une section état de l' art, où nous verrons quelques travaux récents de la littérature, qui démontre l' intérêt de son usage. Ceci nous a aussi permis d' apprendre et de connaitre davantage les architectures d'apprentissage en profondeur existantes, les approches déjà proposées en vue d' une détection d' intrusion surtout, tout en examinant comment les différentes méthodes sont utilisées, afin de les exploiter pour éventuellement pouvoir apporter des améliorations en matière de résultats et de performances.

2.2 Les Réseaux de Neurones Artificiels(RNA)

Un RNA est un ensemble interconnecté d'éléments de traitement simples, d'unités ou de nœuds (neurone), dont la fonctionnalité est basée sur le neurone biologique. La capacité de traitement du réseau est stockée dans les forces de connexions inter unitaires ou poids qui sont obtenus par un processus d'apprentissage. [Gurney et Kevin, 1997].

Le premier modèle de RNA qui a vu le jour est le Perceptron Multi Couches (PMC ou MLP pour Multi Layer Perceptron), constitué principalement d' une couche d' entrée, une couche de sortie et plusieurs

couches cachées (voir figure 2.1). Mais il existe plusieurs types de RNA, il y a ceux qui appliquent un apprentissage supervisé tel que le réseau ADALINE, qui ressemble au MLP mais avec plutôt une fonction objectif linéaire, et qui est basée principalement sur la méthode des moindres carrées. D' autres avec un apprentissage non supervisé tel que la carte auto-adaptative de Kohonen ou le réseau de Hopfield. Il existe d' autres classifications possibles des RNAs, à savoir, les réseaux linéaires ou non par rapport à la fonction objectif, ou encore des RNAs avec ou sans rétropropagation par rapport à l' ajustement des poids.

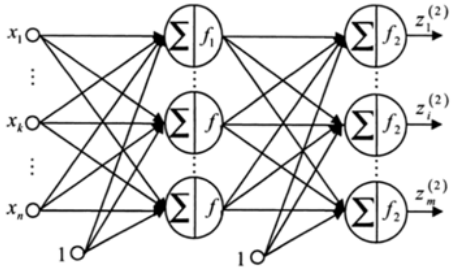

FIG. 2.1 : Réseau multicouche **[Borne et al. 2007]**.

2.3 Apprentissage Profond

L'apprentissage profond est une forme d'apprentissage automatique qui permet aux ordinateurs d'apprendre à partir de l'expérience et de comprendre en termes d'une hiérarchie de concepts, chaque concept étant défini en fonction de sa relation avec des concepts plus simples. En recueillant des connaissances à partir d'expérience, cette approche évite le besoin pour les opérateurs humains de spécifier formellement toutes les connaissances nécessaires à l'ordinateur. [Goodfellow et al., 2016].

Les réseaux de neurones artificiels sont utilisés dans l'apprentissage profond pour analyser et extraire des informations à partir de grandes quantités de données complexes. Ces RNs peuvent identifier des motifs et des relations dans de vastes volumes de données car ils sont conçus pour imiter la structure et le fonctionnement du cerveau humain. Ils sont souvent configurés en couches, chacune traitant d'un composant particulier des données et transmettant les résultats à la couche suivante. Cela permet au réseau de capter progressivement des liens et des propriétés plus profonds dans les données, produisant des prévisions et des classifications plus précises.

2.3.1 Les Réseaux de Neurones Récurrents (RNR)

Les réseaux de neurones artificiels avec des connexions récurrentes sont appelés réseaux de neurones récurrents (voir figure 2.2). Ils sont capables de modéliser des données séquentielles pour la reconnaissance et la prédiction de séquences. Ils sont constitués d'états cachés de haute dimension avec une dynamique non linéaire qui fonctionne comme la mémoire du réseau. Les RNRs peuvent stocker, se rappeler et traiter les signaux complexes passés sur de longues périodes de temps, et ils peuvent mapper une séquence d'entrée vers la séquence de sortie au pas de temps courant et prédire la séquence au prochain pas de temps. [Salehinejad et al., 2017].

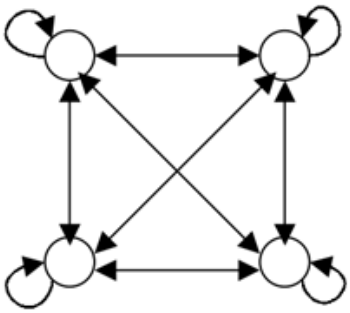

FIG. 2.2 : Un réseau de neurones récurrent entièrement connecté. **[Medsker et al., 2001]**.

- Les réseaux de neurones récurrents à Mémoire à Court et Long Terme (MCLT) :

Les réseaux de neurones récurrents (RNR) peuvent parfois rencontrer des problèmes lorsqu'il s'agit de modéliser des données séquentielles à long terme, en raison de la difficulté à stocker et à traiter des informations sur de longues périodes de temps. C'est là qu'interviennent les MCLTs (ou LSTM pour Long Short Term Memory), il s' agit d' une forme avancée de RNR qui utilisent des mécanismes de porte pour contrôler la quantité d'informations qui est stockée ou oubliée dans l'état caché du réseau. Ces portes permettent au MCLT de gérer efficacement la mémoire à court et à long terme, ce qui le rend capable de modéliser des données séquentielles complexes sur de très longues périodes de temps. Les MCLTs sont particulièrement utiles pour des tâches telles que la traduction automatique, la reconnaissance de la parole et la prédiction de séries temporelles. [Hochreiter et al., 1997].

2.3.2 Les Réseaux de Neurones Convolutifs (RNC)

Le RNC (ou CNN pour Convolutional neural network) a réalisé des avancées remarquables, il est devenu l'un des réseaux de neurones les plus représentatifs dans le domaine de l'apprentissage profond. La vision par ordinateur basée sur les RNCs a permis aux gens de réaliser ce qui avait été considéré comme impossible au cours des derniers siècles, tels que la reconnaissance faciale, segmentation des images, les véhicules autonomes, etc. Le RNC a été aussi utilisé dans d' autres types d' application et plus particulièrement dans la cyber sécurité. Le RNC est un type de réseau de neurones à propagation avant qui est capable d'extraire des caractéristiques à partir de données à l'aide de structures de convolution différentes des méthodes d'extraction de caractéristiques traditionnelles. [Li et al., 2021].

Les RNC apprend automatiquement les représentations hiérarchiques des données visuelles en utilisant plusieurs couches de neurones interconnectés. Explorons les différentes couches que l'on trouve généralement dans un CNN :

- Couche d'entrée : reçoit des données d'entrée brutes, généralement des images ou un ensemble d'images.
- Couche convolutive : applique des filtres pour détecter des modèles et des caractéristiques dans l'entrée.
- Couche d'activation : applique une fonction d'activation pour introduire la non-linéarité.

- Couche de regroupement : sous-échantillonne les cartes d'entités pour extraire les entités dominantes.
- Couche entièrement connectée : apprend les représentations de haut niveau en connectant les neurones de la couche précédente.
- Couche de sortie : fournit les prédictions ou les décisions finales du CNN. (voir figure 2.3).

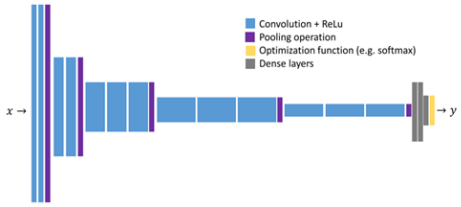

Fig. 2.3 : Diagramme schématique de l' architecture générale d' un modèle de RNC : le VGG-16 [Kattenborn et al., 2021].

2.3.2.1 Les Réseaux Résiduels (Resnet : Residual Network)

L'apprentissage résiduel fait référence à un cadre d'apprentissage où chaque couche doit ajuster une transformation résiduelle, c'est-à-dire la différence entre l'entrée et la sortie de la couche. Les connexions résiduelles permettent de sauter certaines couches de neurones, ce qui permet à l'information d'être transférée directement aux couches suivantes et facilite ainsi l'apprentissage de fonctions complexes (voir figure 2.4). Les réseaux résiduels ont été largement utilisés dans des tâches de reconnaissance d'images et de classification d'objets, et ont montré des performances remarquables dans ces domaines. [He et Kaiming, 2016].

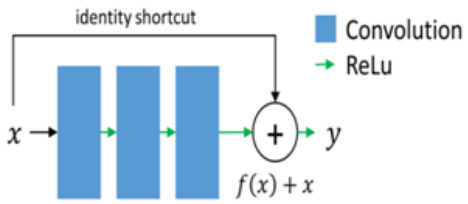

Fig. 2.4 : Diagramme schématique de l' architecture générale du modèle ResNet. [Kattenborn et al., 2021].

2.3.2.2 Réseau U (U-Net : U-Network)

Ce nom vient de la forme en U de l'architecture du réseau de neurones (voir figure 2.5), où l'image est réduite de taille à travers une série de couches d'encodage, U-Net est un type d'architecture de réseau de neurones convolutifs couramment utilisé pour les tâches de segmentation d'images.

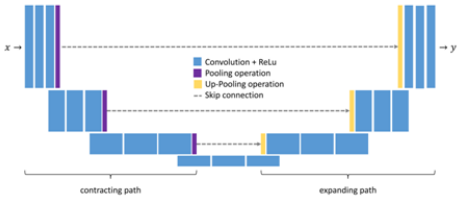

FIG. 2.5 : L'architecture U-Net. [Kattenborn et al., 2021].

La caractéristique de cette architecture est son utilisation de connexions de saut entre les couches correspondantes dans les chemins de contraction et d'expansion. Ces connexions de saut permettent au réseau de combiner des caractéristiques de bas niveau avec des caractéristiques de haut niveau, ce qui peut améliorer la précision de la segmentation.

2.3.3 Transformateur de Vision (TVi)

ViT (Vision Transformer en anglais) est un type de modèle d'apprentissage en profondeur qui utilise une architecture transformer pour les tâches de reconnaissance d'images, il est devenu bien connu dans le domaine de la vision par ordinateur pour sa capacité à atteindre le plus haut niveau sur des référentiels de classification d'images en utilisant seulement un petit échantillon d'exemples étiquetés. Ceci est réalisé en découpant une image en patchs et en fournissant la séquence d'incorporations linéaires de ces patchs en entrée dans l' encodeur de transformateur qui reçoit les incorporations de patch et les encodages de position. Il existe plusieurs couches dans le Transformer Encoder, et elles sont souvent constituées de réseaux de neurones à propagation avant et de processus d'auto-attention. Chaque patch peut prêter attention à tous les autres patchs grâce au mécanisme d'auto-attention, qui capture efficacement le contexte global de l'image. Les données de cette couche sont traitées par les réseaux de neurones à propagation avant, ce qui donne des représentations plus abstraites.
En règle générale, une tête de classification composée d'une ou plusieurs couches entièrement connectées est utilisée pour traiter la sortie du Transformer Encoder. La tête de classification permet de classifier des images en mappant la représentation de grande dimension produite à partir du Encodeur Transformeur au nombre souhaité de classes de sortie (voir figure 2.6). Les patchs d'images sont traités de la même manière que les jetons (mots) dans une application de traitement du langage naturel (NLP), et le modèle est entraîné sur la classification d'images de manière supervisée. [Dosovitskiy et al., 2020].

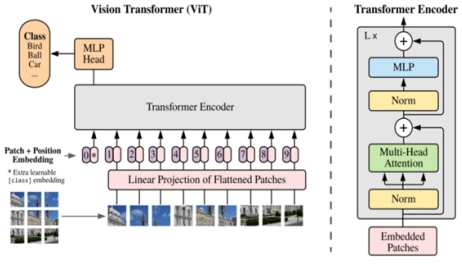

FIG. 2.6 : Vue d'ensemble du modèle ViT. [Dosovitskiy et al., 2020].

Voici quelques avantages de ViT par rapport à d'autres modèles :

- ViT est plus facile à entraîner et plus rapide à exécuter que RNC et ResNet car il nécessite moins de paramètres.

- Les réseaux de neurones convolutifs traditionnels tels que RNC et ResNet ont des difficultés à apprendre des dépendances à longue portée et à collecter le contexte global dans les images, tandis que ViT est assez efficace pour le faire.

En plus de ces avantages, ViT utilise une technique d'attention multi-têtes qui permet au modèle de se concentrer sur plusieurs aspects de l'image en même temps. Grâce à ce processus, ViT est capable de collecter les informations multi- aspects et multi-niveaux de l'image d'entrée, ce qui est essentiel pour effectuer des tâches de reconnaissance d'images avec une grande précision.

2.4 Etat de l' art

L'état actuel de la progression ou de l'avancement dans un domaine ou une industrie particulière est désigné par l'expression "état de l'art" ou bien "the state of art" en anglais. Cela indique généralement les méthodes, technologies ou processus les plus avancés et modernes qui sont utilisés ou développés à ce moment-là. En d'autres termes, cela signifie le summum de la réussite et de l'avancement dans un domaine de connaissance particulier. Au fur et à mesure que de nouvelles découvertes et des développements technologiques sont produits, l'état de l'art est continuellement en train de changer et d'être mis à jour.

Puisque nous nous situons tous au long de notre étude dans une approche de détection d' intrusion via l' apprentissage profond, alors nous nous focalisons dans cette section à présenter un état de l' art dans ce contexte-là.

2.4.1 Les Réseaux MCLT

Les réseaux de neurones MCLT sont une méthode populaire de l' AP qui utilise des unités de mémoire pour traiter efficacement des séquences de données telles que des textes, des signaux audio ou des images.

Les auteurs Imrana et al. (2021) ont proposé une méthode de détection d'intrusion en utilisant une architecture d'apprentissage profond bidirectionnelle à mémoire à long court terme (MCLT) à l'aide de l'ensemble de données NSL-KDD, qui est un ensemble de données de référence largement utilisé pour évaluer les systèmes de détection d'intrusion. Cette étude a fourni une solution prometteuse pour détecter les intrusions sur le réseau (voir figure 2.7).
Altunay et Albayrak (2023) ont utilisé un RNC, un MCLT et un modèle hybride RNC + MCLT pour détecter les intrusions dans les réseaux IoT industriels, tandis que Mushtaq et son équipe [Mushtaq et al., 2022] ont combiné un auto-encodeur avec un MCLT pour extraire et classer les entités. Les deux études ont évalué leurs modèles proposés sur différents ensmbles de données.

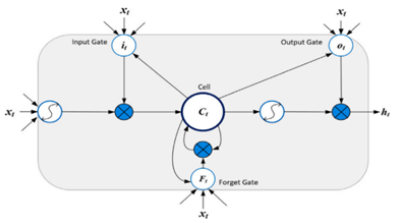

FIG. 2.7 : Architecture d'une cellule de mémoire à court et long terme (MCLT). [Imrana et al. 2021].

2.4.2 Réseaux de type RNC

Les auteurs ElSayed et al. (2021) proposent une approche pour améliorer la sécurité des réseaux définis par le logiciel SDN (software-defined networking), ils introduisent un modèle hybride qui combine RNC et une nouvelle technique de régularisation probabiliste (DPC : Dropout with Probabilistic Confidence), comme indiqué dans le diagramme de flux de l' approche proposée par les auteurs (voir Figure 2.8). Le RNC a été utilisé pour extraire les caractéristiques des données du trafic réseau, tandis que le DPC a été utilisé pour améliorer la capacité de généralisation du modèle et réduire le sur-apprentissage en effectuant également une analyse de sensibilité pour évaluer l'impact de différents paramètres sur les performances du modèle. Les résultats indiquent que le modèle présente une bonne précision et un faible taux de faux positifs.
Le diagramme illustre l'objectif d'exploiter des algorithmes d'apprentissage profond pour la détection d'anomalies et la classification des attaques dans le contexte des réseaux définis par logiciel (SDN). Il commence par une étape de prétraitement, où les données d'entrée sont préparées pour le modèle d'apprentissage profond (AP). Les données sont transformées en un format d'image en les convertissant en une matrice bidimensionnelle. Une fois que les données d'entrée sont prêtes, le modèle d'apprentissage CNN est initialisé et entraîné en utilisant le flux d'entrée préparé.

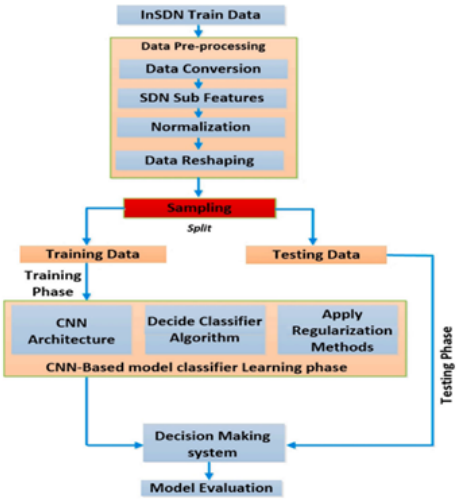

FIG. 2.8 : Diagramme de flux de la méthode proposée dans [ElSayed et al., 2021].

2.4.3 Transformateur de Vision (TVi)

Ho et al. (2022) ont introduit le TVi dans le domaine de la cyber sécurité, où ils ont proposé de combiner la conversion du flux du réseau (paquets) en image et la classification via les transformateurs de vision. Les auteurs ont utilisé les arbres de décisions dans une étape de prétraitement de données pour mettre en œuvre leur approche de détection d'intrusion réseau. Ils ont utilisé un transformateur de vision pour identifier le type de paquet comme normales ou anormales en exploitant les images obtenues suite à la conversion préalable du flux réseau. (voir Figure 2.9).

L' approche a été entrainée, testée et validée sur les ensembles de données CIC IDS2017 et UNSW-NB15, ces enssemble de données sont des collections de données d'intrusions réseau, elle comprend des enregistrements de trafic réseau capturés, reproduisant des scénarios d'attaques réelles. Les données sont organisées en différentes catégories, telles que les attaques par déni de service, les attaques de type infiltration, les attaques Web, etc. Chaque enregistrement de données contient des informations telles que les caractéristiques du trafic, les types d'attaques, les adresses IP source et destination, les protocoles utilisés, etc.

Les résultats obtenus surpassent un certain nombre de techniques de détection d'intrusion en termes de précision, de rappel, de F1-score et d'exactitude. La stratégie suggérée offre une nouvelle façon d'utiliser l'architecture du transformateur de vision pour identifier les intrusions réseau et a le potentiel de surpasser les techniques les plus avancées pour le faire.

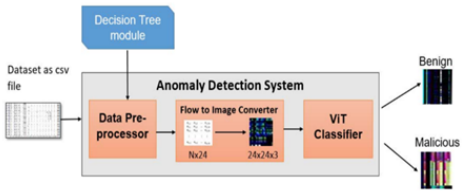

FIG. 2.9 : La structure générale de l' approche TVI proposé dans [Ho et al. 2022].

2.4.4 Autres Travaux

Les techniques d'apprentissage automatique se sont avérées être l'une des méthodes les plus efficaces pour lutter contre les intrusions dans les systèmes informatiques, application, hôtes ou même réseau. Nous avons hélas exploré d' autres travaux dans une quête d' une approche efficace, nous allons dans cette section en présenter quelques-uns.

Les techniques d'apprentissage automatique standard, y compris MVS (machines à vecteur de support), KPPV(K Plus Proches Voisins) et la méthode NB(Naive Bayesienne), ont été utilisées par les auteurs Ikram et Cherukuri (2016), Ingre et Yadav (2015), Nie et al. (2017), Parwez et al. (2017) et Reddy et al. (2016) en vue d' une détection d'intrusions. Bien que ces techniques aient obtenu des résultats encourageants dans le passé, elles présentent des problèmes inhérents, qui ont conduit d' autres auteurs à faire appel aux réseaux de neurones profonds, parmi les limitations de ces techniques :

- La limitation de la complexité des modèles : Limitation dans leurs capacité à représenter des modèles complexes.

- Base de données de grandes dimensionnalité : Les techniques d'apprentissage automatique standard peuvent être moins efficaces dans des base de données de grande dimentionalité.

Tang et al. (2016) sont un exemple, effectivement les auteurs ont proposé une méthode de l'apprentissage profond pour la détection d'intrusions, leur modèle a été entraîné et testé sur l'ensemble de données NSL-KDD, une version réduite de KDD Cup 99, et a abouti à de bonnes performances.

Kim et al. (2016) ont proposé un modèle de détection d'intrusion dans les réseaux basé sur un réseau MCLT-RNR. Leur modèle a atteint une grande précision après avoir été entraîné sur l'ensemble de données KDD Cup 1999.

Ces approches ayant recours à l' apprentissage profond, surpassent généralement les méthodes classiques, ce qui prouve encore une fois l' efficacité de cette approche.

2.5 Conclusion

Dans ce chapitre, nous avons présenté d' abord quelques méthodes issues de l' apprentissage profond, ce dernier qui est le fruit d' une combinaison entre les réseaux de neurones et les grandes masses de données (Big data). Par conséquent, nous avons dû parcourir plusieurs types de réseaux de neurones utilisés en AP tels que les réseaux de neurones récurrents (RNN), les Mémoires à Court et Long Terme (MCLT), les Réseaux de Neurones Convolutionnels (RNC), les réseaux résiduels (ResNets), le réseau U(U-Net) et le Transformer de Vision (TVi).

En outre, il est important de rester à jour avec les dernières recherches dans le domaine de l'apprentissage profond, sur ce nous avons illustré plusieurs travaux ayant fait recours à cette approche afin d' améliorer les performances des SDIs. Nous avons examiné de prés, différentes approches et développements réalisés dans ce domaine ou nous avons illustré la façon dont l'apprentissage profond a permis aux SDIs de devenir plus performants. Effectivement les approches qui ont utilisé les algorithmes de l' AP, ont put surpasser les résultats obtenus via les méthodes issues de l' apprentissage automatique classique, notant toutefois que pas mal de combinaison ont été réalisées entre ces méthodes dites classiques et celles issues de l' apprentissage profond.

Notre objectif principal est de construire un SDIs efficace et performant, en augmentant les taux de bonne détection et en réduisant les fausses alarmes afin d'identifier et d'arrêter les cybers attaques, rendant ainsi le réseau un environnement plus sûr. Pour ce faire nous avons choisi une méthode issue de l' AP qui a montré son efficacité, le chapitre suivant est consacré a notre approche proposée.

Chapitre 3

Approche proposée, Résultats et Présentation de l'application

3.1 Introduction

La sécurité informatique est devenue un élément essentiel à l'ère numérique d'aujourd'hui, car les informations sont facilement partagées et la technologie affecte tous les domaines de notre vie. Il est impératif que nos réseaux et systèmes informatiques soient protégés contre les intrusions hostiles. Les systèmes de détection d'intrusions (SDIs), représentent solution logicielle et parfois aussi matérielle créée pour surveiller et identifier les accès non autorisés ou les actions nuisibles au sein d'un réseau. Maximiser l'efficacité d'un SDI, cependant, présente un ensemble unique de problèmes.

La sécurité informatique est devenue un élément essentiel à l'ère numérique d'aujourd'hui, car les informations sont facilement partagées et la technologie affecte tous les domaines de notre vie. Il est impératif que nos réseaux et systèmes informatiques soient protégés contre les intrusions hostiles. Les systèmes de détection d'intrusions (SDIs), représente solution logicielle et parfois aussimatérielle créée pour surveiller et identifier les accès non autorisés ou les actions nuisibles au sein d'un réseau. Maximiser l'efficacité d'un SDI, cependant, présente un ensemble unique de problèmes.

L' objectif de notre étude est de réaliser un SDI permettant d' obtenir de bonnes performances de détection, nous avons alors opté pour une méthode issue de l' apprentissage profond, ce dernier qui a largement montré son efficacité et surtout avec les bases de données importantes. Il est cependant aussi question de trouver une méthode efficace pour traiter les problèmes posés par des ensembles de données déséquilibrés et bruités.

Dans ce chapitre, nous allons présenter notre approche proposée pour réaliser notre SDIR, résultat d' une démarche en quête d' amélioration des performances de détection d' intrusions, durant laquelle nous avons adopté alors l' algorithme de transformateurs de vision. Les étapes d' apprentissage et de test ont été réalisées sur deux bases de données très utilisées qui sont NSL KDD et CIC 2017, pour pallier les problèmes de déséquilibre des classes, nous avons choisi des algorithmes d' échantillonnage, quant aux attributs non significatifs représentants du bruit, nous avons dans un premier temps utilisé l' algorithme basé sur la variation [Belgrana et al., 2020] qui s' est avéré efficace avec la classification binaire, puis dans un deuxième temps, nous avons appliqué l' algorithme des forêts aléatoires « Random Forest » qui plus performant avec la classification multi classes.

Pour évaluer les performances de notre SDIR, nous avons utilisé trois mesures de performances, à savoir, le taux de bon apprentissage, la précision, les fausses alarmes. Pour démontrer l' efficacité de notre approche proposée, nous avons réalisé une comparaison avec un travail de la littérature [Ho et al., 2022].

3.2 Base de données utilisées

Tout au long de notre travail, nous avons manipulé deux bases de données largement utilisées dans la recherche en sécurité informatique : NSL-KDD [NSL KDD, 2009] et CIC-IDS-2017 [CIC, 2017].Les deux bases sont différentes en matière de contenu des paquets, nous allons voir dans ce qui suit la structure de chacune. Nous examinerons également les différents types d'attaques qu'elles contiennent.

3.2.1 NSL KDD

La base de données NSL-KDD fait référence à un ensemble de données utilisé pour la recherche en systèmes de détection d'intrusion, elle contient divers types d'attaques réseau. Cet ensemble de données est une version améliorée du KDD Cup 1999 d'origine, qui a été critiquée pour être trop simple et irréaliste. NSL-KDD a été développée pour résoudre ces problèmes-là, sa base d' entrainement contient 125 973 instances et sa base de test contient 22 544 instances [Dhanabal et Shantharajah, 2015]. Le jeu de données est largement utilisé par les chercheurs et les praticiens dans le domaine de la sécurité des réseaux pour évaluer les performances des différents SDIs [Dhanabal et Shantharajah, 2015].
La colonne "Label" dans l'ensemble de données NSL-KDD est une variable qui indique si une connexion réseau est normale ou malveillante. En d'autres termes, elle spécifie le type d'attaque ou d'intrusion présente dans les données de trafic réseau. Elle prend deux valeurs possibles : "Normal" et "Attaque". La catégorie "attaque" comprend différents types d'attaques (Déni de Service (DoS), les attaques de sondage (Probing attacks), les attaques U2R et R2L). Cette colonne est un composant crucial de l'ensemble de données NSL-KDD, car elle est utilisée lors de l' apprentissage supervisé et pour évaluer la performance des systèmes de détection d'intrusion.

TAB. 3.1 : Attributs de la base NSL KDD

Attribut	**Description**
Duration	La durée de la connexion
protocol_type	Protocole utilisé dans la connexion
Service	Service réseau de destination utilisé
Flag	Statut de la connexion - Normal ou Erreur
src_bytes	Nombre d'octets de données transférés de la source à la destination en une seule connexion
dst_bytes	Nombre d'octets de données transférés de la destination à la source dans une connexion unique
Land	Si les adresses IP source et de destination et les numéros de port sont égaux, cette variable prend la valeur 1 sinon 0
wrong_fragment	Nombre total de fragments incorrects dans cette connexion
Urgent	Nombre de paquets urgents dans cette connexion. Les paquets urgents sont des paquets avec le bit urgent activé
Hot	Nombre d'indicateurs "hot" dans le contenu tels que : entrer dans un répertoire système, créer des programmes et exécuter des programmes
num_failed_logins	Nombre de tentatives de connexion échouées
logged_in	État de connexion : 1 si la connexion a réussi ; 0 sinon
num_compromised	Nombre de conditions compromises
root_shell	1 si la coque racinaire (root shell) est obtenue ; 0 sinon
su_attempted	1 si la commande ńsu rootż a été tentée ou utilisée ; 0 sinon
num_root	Nombre d'accès "root" ou nombre d'opérations effectuées en tant que root dans la connexion
num_file_creations	Nombre d'opérations de création de fichiers dans la connexion
num_shells	Nombre d'invités du shell
num_access_files	Nombre d'opérations sur les fichiers de contrôle d'accès

is_host_login	1 si la connexion appartient à la liste "hot", c'est-à-dire root ou admin ; sinon 0
is_guest_login	1 si la connexion est une connexion "invite" ; 0 sinon
Count	Nombre de connexions au même hôte de destination que la connexion actuelle au cours des deux dernières secondes
srv_count	Nombre de connexions au même service (numéro de port) que la connexion actuelle au cours des deux dernières secondes
serror_rate	Pourcentage de connexions qui ont activé l'indicateur "flag" (4) s0, s1, s2 ou s3, parmi les connexions regroupées dans count (23)
srv_serror_rate	Pourcentage de connexions qui ont activé l'indicateur "flag" (4) s0, s1, s2 ou s3, parmi les connexions regroupées dans srv count (24)
rerror_rate	Pourcentage de connexions qui ont activé l'indicateur "flag" (4) REJ, parmi les connexions regroupées dans count (23)
srv_rerror_rate	Pourcentage de connexions qui ont activé l'indicateur "flag" (4) REJ, parmi les connexions regroupées dans srv count (24)
same_srv_rate	Le pourcentage de connexions qui étaient au même service, parmi les connexions regroupées en count (23)
diff_srv_rate	Le pourcentage de connexions qui étaient à différents services, parmi les connexions regroupées en count (23)
srv_diff_host_rate	Le pourcentage de connexions qui étaient vers différentes machines de destination parmi l'agrégat des connexions dans srv count (24)
dst_host_count	Nombre de connexions ayant la même adresse IP d'hôte de destination
dst_host_srv_count	Nombre de connexions ayant le même numéro de port
dst_host_same_srv_rate	Le pourcentage de connexions qui étaient au même service, parmi les connexions regroupées dans dst host count (32)
dst_host_diff_srv_rate	Le pourcentage de connexions qui étaient à différents services, parmi les connexions regroupées dans dst host count (32)
dst_host_same_src_port_rate	Pourcentage de connexions qui se trouvaient sur le même port source, parmi les connexions regroupées dans dst host srv count (33)
dst_host_srv_diff_host_rate	Pourcentage de connexions qui étaient vers différentes machines de destination, parmi les connexions regroupées dans dst host srv count (33)
dst_host_serror_rate	Pourcentage de connexions qui ont activé l'indicateur (4) s0, s1, s2 ou s3, parmi les connexions regroupées dans dst host count (32)
dst_host_srv_serror_rate	Pourcentage de connexions qui ont activé l'indicateur (4) s0, s1, s2 ou s3, parmi les connexions regroupées dans dst host srv count (33)
dst_host_rerror_rate	Pourcentage de connexions qui ont activé l'indicateur (4) REJ, parmi les connexions agrégées dans dst host count (32)
dst_host_srv_rerror_rate	Pourcentage de connexions qui ont activé l'indicateur (4) REJ, parmi les connexions agrégées dans dst host srv count (33)
Label	Classification de l'entrée de trafic

3.2.2 CIC 2017

CIC-IDS-2017 est un ensemble de données qui contient différents types de trafic réseau, elle est conçue pour entrainer et évaluer les systèmes de détection d'intrusion à son tour. Il s'agit d'un ensemble de données disponible publiquement utilisé par les chercheurs et les praticiens pour analyser les caractéristiques du trafic réseau, y compris le trafic normal et divers types d'attaques.

L' ensemble de données est composé du trafic réseau capturé dans un environnement contrôlé, conçu pour imiter des scénarios du monde réel. L'ensemble de données a été publié par l'Institut canadien de la cyber sécurité (CIC) en 2017 et est composé à la fois de trafic réseau bénin et malveillant. CIC-IDS-2017 contient un total de 85 caractéristiques extraites à partir des flux de trafic réseau à l'aide de l'outil "CIC Flow Meter" développé par l' institut Canadien de la cyber sécurité. L'ensemble de données contient différents types d'attaques réseau, telles que le déni de service (DoS), le déni de service distribué (DDoS), les attaques par force brute, les scans de réseau et les attaques de botnet, entre autres. Les attaques ont été générées à l'aide d'une combinaison de méthodes automatisées et manuelles pour créer un ensemble de données réaliste qui peut être utilisé pour évaluer les SDIs. [Panigrahi et al., 2018].

CIC-IDS-2017 contient 84 attributs, comme c' est illustré dans la table 3.2 présentée ci-dessous.

Tab. 3.2 : Attributs de la base CIC 2017

Attribut	**Description**
Flow ID	Identifiant de flux unique
Source IP	Adresse IP source de la connexion
Source Port	Port source de la connexion
Destination IP	Adresse IP de destination de la connexion
Destination Port	Port de destination de la connexion
Protocol	Protocole utilisé (TCP, UDP, ICMP)
Timestamp	Horodatage de la connexion
Flow Duration	Durée totale du flux
Total Fwd Packets	Nombre total de paquets en direction de la destination
Total Backward Packets	Nombre total de paquets en provenance de la destination
Total Length of Fwd Packets	Longueur totale des paquets en direction de la destination
Total Length of Bwd Packets	Longueur totale des paquets en provenance de la destination
Fwd Packet Length Max	Longueur maximale des paquets en direction de la destination
Fwd Packet Length Min	Longueur minimale des paquets en direction de la destination
Fwd Packet Length Mean	Longueur moyenne des paquets en direction de la destination
Fwd Packet Length Std	Écart type des longueurs des paquets en direction de la destination
Bwd Packet Length Max	Longueur maximale des paquets en provenance de la destination
Bwd Packet Length Min	Longueur minimale des paquets en provenance de la destination
Bwd Packet Length Mean	Longueur moyenne des paquets en provenance de la destination

Bwd Packet Length Std	Écart type des longueurs des paquets en provenance de la destination
Flow Bytes/s	Débit en octets par seconde du flux
Flow Packets/s	Débit en paquets par seconde du flux
Flow IAT Mean	Moyenne des intervalles inter-arrivées du flux
Flow IAT Std	Écart type des intervalles inter-arrivées du flux
Flow IAT Max	Intervalle inter-arrivées maximum du flux
Flow IAT Min	Intervalle inter-arrivées minimum du flux
Fwd IAT Total	Durée totale des intervalles inter-arrivées en direction de la destination
Fwd IAT Mean	Moyenne des intervalles inter-arrivées en direction de la destination
Fwd IAT Std	Écart type des intervalles inter-arrivées en direction de la destination
Fwd IAT Max	Intervalle inter-arrivées maximum en direction de la destination
Fwd IAT Min	Intervalle inter-arrivées minimum en direction de la destination
Bwd IAT Total	Durée totale des intervalles inter-arrivées en provenance de la destination
Bwd IAT Mean	Moyenne des intervalles inter-arrivées en provenance de la destination
Bwd IAT Std	Écart type des intervalles inter-arrivées en provenance de la destination
Bwd IAT Max	Intervalle inter-arrivées maximum en provenance de la destination
Bwd IAT Min	Intervalle inter-arrivées minimum en provenance de la destination
Fwd PSH Flags	Nombre de paquets TCP avec le drapeau PSH en direction de la destination
Bwd PSH Flags	Nombre de paquets TCP avec le drapeau PSH en provenance de la destination
Fwd URG Flags	Nombre de paquets TCP avec le drapeau URG en direction de la destination
Bwd URG Flags	Nombre de paquets TCP avec le drapeau URG en provenance de la destination
Fwd Header Length	Longueur de l'en-tête TCP en direction de la destination
Bwd Header Length	Longueur de l'en-tête TCP en provenance de la destination
Fwd Packets/s	Débit en paquets par seconde en direction de la destination
Bwd Packets/s	Débit en paquets par seconde en provenance de la destination
Min Packet Length	Longueur minimale des paquets
Max Packet Length	Longueur maximale des paquets
Packet Length Mean	Longueur moyenne des paquets
Packet Length Std	Écart type des longueurs des paquets
Packet Length Variance	Variance des longueurs des paquets

FIN Flag Count	Nombre de paquets TCP avec le drapeau FIN
SYN Flag Count	Nombre de paquets TCP avec le drapeau SYN
RST Flag Count	Nombre de paquets TCP avec le drapeau RST
PSH Flag Count	Nombre de paquets TCP avec le drapeau PSH
ACK Flag Count	Nombre de paquets TCP avec le drapeau ACK
URG Flag Count	Nombre de paquets TCP avec le drapeau URG
CWE Flag Count	Nombre de paquets TCP avec le drapeau CWE
ECE Flag Count	Nombre de paquets TCP avec le drapeau ECE
Down/Up Ratio	Ratio de trafic descendant/ascendant
Avg Fwd Segment Size	Taille moyenne des segments en direction de la destination
Avg Bwd Segment Size	Taille moyenne des segments en provenance de la destination
Fwd Header Length.1	Longueur de l'en-tête TCP en direction de la destination (2ème occurrence)
Fwd Avg Bytes/Bulk	Taille moyenne des octets par lot en direction de la destination
Fwd Avg Packets/Bulk	Nombre moyen de paquets par lot en direction de la destination
Fwd Avg Bulk Rate	Taux moyen des lots en direction de la destination
Bwd Avg Bytes/Bulk	Taille moyenne des octets par lot en provenance de la destination
Bwd Avg Packets/Bulk	Nombre moyen de paquets par lot en provenance de la destination
Bwd Avg Bulk Rate	Taux moyen des lots en provenance de la destination
Subflow Fwd Packets	Nombre de paquets de sous-flux en direction de la destination
Subflow Fwd Bytes	Nombre d'octets de sous-flux en direction de la destination
Subflow Bwd Packets	Nombre de paquets de sous-flux en provenance de la destination
Subflow Bwd Bytes	Nombre d'octets de sous-flux en provenance de la destination
$\text{Init}_W in_b ytes_f orward$	Taille de la fenêtre TCP initiale en direction de la destination
$\text{Init}_W in_b ytes_b ackward$	Taille de la fenêtre TCP initiale en provenance de la destination
$\text{act}_d ata_p kt_f wd$	Nombre de paquets TCP avec des données réelles en direction de la destination
$\text{min}_s eg_s ize_f orward$	Taille minimale du segment TCP en direction de la destination
Active Mean	Durée moyenne de la période d'activité du flux
Active Std	Écart type de la période d'activité du flux
Active Max	Période d'activité maximale du flux
Active Min	Période d'activité minimale du flux
Idle Mean	Durée moyenne de la période d'inactivité du flux
Idle Std	Écart type de la période d'inactivité du flux
Idle Max	Période d'inactivité maximale du flux
Idle Min	Période d'inactivité minimale du flux
Label	Étiquette de la classe

3.3 Approche proposée

3.3.1 Principe général

Afin de réaliser un SDIR avec de bonnes performances de détection d'intrusion, nous avons dû dans un premier tempschoisir un algorithme issu de l' apprentissage profond, un sous-domaine de l' apprentissage automatique et qui a déjà montré son efficacité, surtout avec la disponibilité de grands volumes de données. Nous avons opté pour l' algorithme TVi en raison de ses avantages évidents par rapport aux modèles concurrents. TVi utilise la force des processus d'auto-attention pour capturer les dépendances globales dans les données, contrairement aux réseaux de neurones conventionnels (CNN) qui reposent sur des interactions locales. Pour cette raison, le modèle peut découvrir avec succès des corrélations et des modèles complexes dans d'énormes ensembles de données.

Nous avons travaillé initialement avec la base NSL KDD, mais pour pouvoir comparer avec le travail de Ho et son équipe [Ho et al., 2022] qui ont aussi fait recours au TVi, nous avons utilisé notamment la base CIC 2017. Les deux bases sont différentes en matière de structure interne des paquets (attributs), les deux base données nécessitaient un néttoyage, les lignes qui n' étaient pas importantes, incompléte ou null ont été éliminées. Nous avons d' abord commencé par une classification binaire en deux classes, normale et intrusion, puis nous avons réalisé une classification sur plusieurs classes.

Sur ceux nous avons adopté deux stratégies en utilisant le même algorithme qui est TVi, la première était appliquée sur l'ensemble de données NSL KDD. Les lignes qui n'étaient pas importantes, incompléte ou null ont été éliminées et nous avons appliqué la variance pour la sélection des attributs pertinents, aprés plusieur test nous avons opté pour un seuil de «0.01». Pour équilibrer les données entre les classes nous avons opté pour l' algorithme d' échantillonnage aléatoire « Random Oversampling » qui duplique aléatoirement les lignes de la base de données. Après avoir équilibré l'ensemble de données, nous avons utilisé le modèle TVi pour réaliser la classification.

La deuxième stratégie que nous avons adoptée sur l'ensemble de données CIC 2017 consistait à réaliser un sous-échantillonnage en utilisant l' algorithme « Random Undersampling ». L' objectif était de réaliser une réduction de dimensionnalités ; bien que les algorithmes de l' AP nécessitent une base importante, notre choix était dû suite à des limitations en matière de matériel (capacité de la RAM), car la base CIC est très volumineuse, la réduire nous a permis de réaliser l' étape d' apprentissage. Grâce à cet algorithme de sous-échantillonnage, nous avons abouti à un ensemble réduit généralisable. Malheureusement, l'application précoce de la sélection des caractéristiques avec variance lors de la classification binaire a produit des résultats acceptables, mais pas assez intéressants sur cette base-là lors de la classification en plusieurs classes ; normal et intrusion avec spécification du type de l' attaque. Afin d' améliorerces résultats, nous avons altérer vers l' algorithme des forêts aléatoires qui est capable de gérer des ensembles de données de grande dimension, il offre une métrique pour l'importance des caractéristiques, afin d' identifier celle les plus pertinentes. Hélas l' amélioration n' était pas satisfaisante, alors nous avons procéder à un équilibrage de la base CIC 2017 cette fois-ci en utilisant un sur-échantillonage, car la différence en nombre de paquets entreles classes normales, et ceux des différentes attaques est importante.

Ces méthodes d' échantillonnage ont démontré la valeur d'une méthodologie personnalisée pour divers ensembles de données, traitant des lignes et les attributs non pertinentes, équilibrant les données et améliorant les modèles pour une meilleurs détection des intrusions.

Une des particularités de l' algorithme TVi, est qu' il faut convertir les données à traiter dans un format compatible. Effectivement, lorsque l'étape de prétraitement des données (nettoyage,codage et normalisation) est terminée, nous avons transformé les données brutes en images, afin de capturer les informations nécessaires à l'intérieur du format d'image. Cette transformation impliquait d'exprimer les points de données sous forme de pixels, permettant au TVi d'apprendre avec succès des représentations structurées des données modifiées. L'intégration des données traitées dans le pipeline de détection d'intrusion a été rendue possible par cette transformation, qui a comblé le vide entre le format de données d'origine et les spécifications du modèle TVi.

L' organigramme décrivant notre approche proposée et la méthodologie adoptée est illustré dans la figure 3.1 présentée ci-dessous.

Fig. 3.1 : Organigramme de l' approche proposée.

3.3.2 Prétraitement des données

Pour s'assurer que les données sont appropriées pour l'analyse, certaines préparations doivent être effectuées. Le codage, la normalisation et la transformation des données sous forme d' images, que nous allons voir dans cette section, sont des étapes essentielles qui font partie du prétraitement des données.

3.3.2.1 Codage

Pour convertir les données catégorielles en une représentation numérique que les algorithmes d'apprentissage automatique peuvent traiter efficacement, les valeurs de chaîne sont converties en valeurs entières. Cette méthode de codage nous permet de transformer des données qualitatives, telles que des étiquettes ou des catégories, en une structure quantitative que les modèles informatiques peuvent rapidement examiner et comprendre. Le codage a été utilisé dans les deux ensembles de données NSL KDD et CIC. Il est crucial de garder à l'esprit que les qualités précises qui nécessitent un codage dans ces deux ensembles de données peuvent ne pas être les mêmes. Chaque ensemble de données à son propre ensemble de caractéristiques et de qualités.

Les attributs qui ont été codés dans la base de données NSL KDD sont les suivants :

A. L' attribut « Protocol type »

Tab. 3.3 : Codage de l' attribut "Protocol type"

Protocol type	**Codage**
ICMP	0
TCP	1
UDP	2

B. L' attribut « Flag »

Tab. 3.4 : Codage de l' attribut "Flag"

Flag	**Codage**
OTH	0
REJ	1
RSTO	2
RSTOS0	3
RSTR	4
S0	5
S1	6
S2	7
S3	8
SF	9
SH	10

C. L' attribut « Service »

Tab. 3.5 : Codage de l' attribut "Service"

Service	**Codage**	**Service**	**Codage**
IRC	0	X11	1
Z39_50	2	Aol	3
	3	Auth	4
bgp	5	courier	6
Csnet_ns	7	Ctf	8
daytime	9	discard	10
domain	11	domain_u	12
echo	13	Eco_i	14
Ecr_i	15	Efs	16
exec	17	Finger	18
ftp	19	ftp_data	20
gopher	21	harvest	22
hostnames	23	http	24
http_2784	25	http_443	26
http_8001	27	Imap4	28
Iso_tsap	29	Klogin	30
kshell	31	Ldap	32
link	33	Login	34
mtp	35	Name	36
Netbios_dgm	37	Netbios_ns	38
Netbios_ssn	39	netstat	40
nnsp	41	nntp	42
Ntp_u	43	Other	44
Pm_dump	45	Pop_2	46
Pop_3	47	printer	48
private	49	Red_i	50
Remote_job	51	Rje	52
shell	53	Smtp	54
Sql_net	55	Ssh	56
sunrpc	57	supdup	58
systat	59	telnet	60
Tftp_u	61	Tim_i	62
time	63	Urh_i	64
Urp_i	65	Uucp	66
Uucp_path	67	Vmnet	68
whois	69	/	/

Puisque les deux bases NSL KDD et CIC 2017 sont différentes au niveau des attributs des paquets, l' étape de codage et alors à son tour différente. Nous allons voir dans ce qui suit les quatre attributs à qui nécessite un codage dans la deuxième base de données CIC 2017, « Flow ID », « Source IP », « Destination IP » et « Timestamp » qui contiennent respectivement 1085071, 17005, 19112 et 27965 catégories.

Voici quelques exemples de chaque attribut :

D. Flow ID

Tab. 3.6 : Codage de l' attribut "Flow ID"

Flow ID	**Codage**
192.168.10.9	906481
192.168.10.3	696110
192.168.10.1	313099

E. Source IP

Tab. 3.7 : Codage de l' attribut "Source IP"

Source IP	**Codage**
35.185.54.22	8355
172.16.0.1	3310
104.88.40.28	999

F. Timestamp

Tab. 3.8 : Codage de l' attribut "Timestamp"

Timestamp	**Codage**
6/7/2017 4:24	26436
7/7/2017 2:52	26830
7/7/2017 3:02	26840

3.3.2.2 Nettoyage

Une autre étape essentielle pour préparer les données en vue de l'analyse consiste à la nettoyer en éliminant tous les enregistrements nuls ou les enregistrements qui ont des valeurs manquantes qui peuvent se produire pour plusieurs raisons, telles que des erreurs de saisie de données ou des données insuffisantes. Ces enregistrements peuvent affecter l'exactitude des résultats, l'équité des jugements ou la capacité prédictive.
Après nettoyage des bases de données, nous avons éliminé un total de «204 202» paquets (lignes contenant des champs d'attributs complétement vide) et supprimé 8 colonnes (complétement vide) dans l'ensemble de données CIC 2017. En revanche, l'ensemble de données NSL KDD n'a nécessité aucune suppression car les données étaient complétes.

3.3.2.3 Normalisation

La méthode de normalisation consiste à réduire les valeurs des caractéristiques à une plage comprise entre 0 et 1. Cette phase est très importante lorsqu'on travaille avec des données numériques qui ont une grande plage de valeurs ou lorsqu'on utilise des algorithmes sensibles à l'échelle des données d'entrée. La méthode "MinMaxScaler" est un choix populaire pour la normalisation car elle met à l'échelle les données de manière linéaire dans la plage comprise entre 0 et 1. Elle est définit par l' équation suivante :

$$X_{\text{nouv}} = \frac{X_{\text{anc}} - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{3.1}$$

Où :

- Xnouv : la nouvelle valeur obtenue.
- Xanc : la valeur a normalisée.
- Xmax : la valeur maximale du champ.
- Xmin : la valeur minimale du champ.[Belgrana et al., 2020].

3.3.2.4 Sélection des caractéristiques

Une étape critique de l'apprentissage automatique est la sélection des caractéristiques, qui tente d'isoler les caractéristiques les plus pertinentes et les plus instructives dans un ensemble de données. Différentes méthodes de sélection des caractéristiques dans les bases de données NSL KDD et CIC 2017 ont été utilisées dans notre étude. Initialement, nous avons reproduit l'approche utilisée par les étudiants précédents [Belgrana et al., 2020] qui avaient obtenu des résultats positifs en utilisant la sélection de caractéristiques basée sur la variance dans NSL KDD. Nous avons appliqué la même approche et vu des résultats encourageants, nous avons donc décidé de continuer à l'utiliser comme méthode de sélection des attributs pour NSL KDD.

Cependant, les résultats de sélection des caractéristiques basée sur la variance dans l'ensemble de données CIC 2017 n'étaient pas satisfaisants. Dès que nous avons pris conscience des lacunes de cette technique, nous sommes rapidement passés à l' algorithme des forêts aléatoires (en anglais : Random Forest). Cet algorithme construit de nombreux arbres de décision et agrège les résultats pour produire des prédictions. Cela fonctionne en créant des arbres de décision qui utilisent des sous-ensembles d'attributs choisis au hasard. Cette technique est réputée pour sa capacité à capturer des corrélations complexes entre les variables et est efficace pour gérer des ensembles de données de grandes dimensions.

Les performances du modèle de détection d' intrusion sur l'ensemble de données CIC 2017 se sont considérablement améliorées après l'utilisation des forêts aléatoires pour sélectionner les attributs pertinents ce qui illustre son efficacité et justifie notre choix.

3.3.2.5 Équilibrage des données

Pour résoudre le problème du déséquilibre des classes dans les ensembles de données, nous avons utilisé à la fois des algorithmes de sur et de sous-échantillonnage dans notre approche proposée. Pour augmenter la proportion des occurrences des classes minoritaires, nous avons sur échantillonné les ensembles de données NSL KDD et CIC 2017. Cette méthode nous a aidé à améliorer les performances globales de notre modèle. D'autre part, comme l'ensemble de données CIC 2017 était nettement plus volumineux que la base de données NSL KDD, nous avons utilisé le sous-échantillonnage de la base de données CIC 2017 pour surpasser les limitations en matière de matériel, en procédant à une réduction de dimensionnalité de la base.

Pour NSL KDD, nous avons équilibré les classes en fonction de la classe la plus élevée, qui est "normale" avec «67 343» lignes. Cette approche égalise la distribution des instances entre les classes. De même, dans le cas de CIC 2017, nous avons équilibré les données en ajustant uniformément toutes les classes pour avoir «50 000» lignes chacune.

3.3.2.6 Transformation de données numérique en image

Lors de l'utilisation de l'approche du transformateur de vision (TVI), il est important de convertir les données brutes en images RGB (Red Green Blue en anglais) car cette technique ne fonctionne qu'avec des données d'image et non avec des valeurs numériques brutes. Chaque pixel de l'image RGB finale de cette transformation indique la valeur d'une certaine caractéristique à partir des données brutes, chaque ligne représente un paquet. Les valeurs des attributs sont souvent mises à l'échelle ou normalisées pour s'adapter à l'intérieur de la bande de 0 à 255, qui représente toute la gamme des valeurs d'intensité pour chaque pixel, afin de garantir la compatibilité avec l'espace colorimétrique appelé RVB.

Il faudrait toute fois savoir, que l'analyse peut ne pas nécessiter les informations de couleur exactes. Dans ces circonstances, donner aux images un thème en niveaux de gris peut être avantageux. Les images RGB peuvent être transformées en images en niveaux de gris où chaque pixel indique l'intensité d'une valeur caractéristique particulière, ce qui simplifie la concentration sur les données importantes pour la recherche. La précision et l'efficacité de l'algorithme du transformateur de vision peuvent toutes les deux bénéficier de cette traduction en niveaux de gris.

Lorsqu' on travaille avec des ensembles de données de grandes dimensions, l'image finale peut parfois être trop grande pour être bien gérée. L'image est fréquemment divisée en parties plus petites ou sous-images pour résoudre ce problème. La taille de ces patchs peut changer en fonction des techniques de sélection de caractéristiques particulières employées ou des exigences de l'analyse. Les patchs ont une dimension de 224 pixels en hauteur et une quantité particulière de caractéristiques en largeur. La figure 3.2 présentée ci-dessous montre une image qui est une partie d' une autre image DoS plus grande.

Les informations sont moins accablantes pour la méthode du transformateur de vision à analyser en patchant l'image en minuscules sous-images. Chaque patch contient un sous-ensemble de fonctionnalités, permettant d'extraire et d'analyser les fonctionnalités à un niveau local plus précis. Étant donné que différentes méthodes de sélection d'entités peuvent utiliser une quantité différente d'attributs, cette technique de correction est utile pour les adapter.

FIG. 3.2 : Une partie de la grande image DoS avec une dimension (224x 31).

3.3.3 Méthodes et mesures adoptées

3.3.3.1 Transformateur de vision

Les avantages extraordinaires des transformateurs de vision (TVi), en particulier le mécanisme d'attention multi-têtes, ont influencé notre décision de l'utiliser dans notre approche. Les TVis utilisent plusieurs chefs d'attention pour enregistrer les interactions complexes entre les différents patchs d'image, permettant une meilleure compréhension du matériel visuel. Avec l'aide de ce mécanisme d'attention, le modèle peut se concentrer sur les correctifs pertinents et combiner des données provenant de divers domaines, améliorant ainsi la représentation des caractéristiques et la compréhension contextuelle. Ils ont également démontré des performances exceptionnelles dans des applications telles que la segmentation, la détection d'objets et la classification d'images, correspondant ou même surpassant les réseaux de neurones convolutionnels conventionnels.

Notre TVi est construit par une fonction appelée « create vit classifier() ». On commence par définir la couche d'entrée du modèle. L'augmentation de données est appliquée aux entrées, cette étape permet d'améliorer la généralisation et la robustesse du modèle en générant des variations des données d'entrée. Les images d'entrée sont divisées en patchs avec une taille précise. Les patchs sont aprés encodés. Ensuite, on crée les couches du bloc Transformer qui se compose de plusieurs sous-couches, y compris la normalisation, l'attention multi-tête, les connexions de saut et les couches PMC (Perceptron Multi Couche). Ces couches aident le modèle à capturer les dépendances à longue portée et à apprendre les relations complexes au sein de l'image. Après les blocs Transformer, une normalisation de couche est appliquée aux patchs encodés. Cette étape normalise les valeurs des caractéristiques et améliore la stabilité du modèle pendant l'entraînement.les valeurs sont ensuite véhiculées dans une couche MLP (Multi-Layer Perceptron) pour traiter les données et extraire les caractéristiques de niveau supérieur, Enfin, une couche entièrement connectée est ajoutée pour produire les sorties du modèle, qui représentent les classes (ou les sorties). Dans la classification binaire nos classes sont normale est anormale, et pour la classification multi classes nos class sont normale et les différentes catégories d'intrusion.

Algorithme 1 : Transformateur de vision

```
function create_vit_classifier() :
    inputs ← layers.Input(shape = input_shape) ;
    augmented ← data_augmentation(inputs) ;
    patches ← Patches(patch_size)(augmented) ;
    encoded_patches ← PatchEncoder(num_patches, projection_dim)(patches) ;
    for ß ← 1 to transformer_layers do
        x1 ← layers.LayerNormalization(epsilon = 1e − 6)(encoded_patches) ;
        attention_output ← layers.MultiHeadAttention(num_heads = num_heads, key_dim =
          projection_dim, dropout = 0.1)(x1, x1) ;
        x2 ← layers.Add()([attention_output, encoded_patches]) ;
        x3 ← layers.LayerNormalization(epsilon = 1e − 6)(x2) ;
        x3 ← mlp(x3, hidden_units = transformer_units, dropout_rate = 0.1) ;
        encoded_patches ← layers.Add()([x3, x2]) ;
    representation ← layers.LayerNormalization(epsilon = 1e − 6)(encoded_patches) ;
    representation ← layers.Flatten()(representation) ;
    representation ← layers.Dropout(0.5)(representation) ;
    features ← mlp(representation, hidden_units = mlp_head_units, dropout_rate = 0.5) ;
    logits ← layers.Dense(num_classes)(features) ;
    model ← keras.Model(inputs = inputs, outputs = logits) ;
```

3.3.3.2 Algorithme d' échantillonnage

- Algorithme de sur échantillonnage : il permet de traiter les déséquilibres dans les classes dans les ensembles de données d'apprentissage automatique. Nous avons opté pour l' algorithme de sur échantillonnage aléatoire (Random Over Sampler), où les échantillons de la classe minoritaire sont dupliqués dans le but d'augmenter la représentativité des données. Cet algorithme a l'avantage d'être simple et il conserve les données d'origine de l'ensemble de données tout en éliminant l'utilisation d'échantillons artificiels.

Algorithme 2 : Sur échantillonnage aléatoire

Data : Ensemble de données d'entrée D avec déséquilibre de classes
Result : Ensemble de données sur échantillonné D' avec des classes équilibrées

```
D' ← D ;
N ← Nombre d'instances dans la classe minoritaire ;
M ← Nombre d'instances dans la classe majoritaire ;
while N < M do
    Sélectionner une instance aléatoire x de la classe minoritaire ;
    Dupliquer x et l'ajouter à D' ;
    N ← N + 1 ;
end
```

- Algorithme de sous-échantillonnage : c' est une deuxième méthode pour traiter le déséquilibre des classes dans les ensembles de données d'apprentissage automatique. Nous avons aussi opté pour l' algorithme de sous-échantillonnage aléatoire, qui implique d'éliminer aléatoirement les instances de la classe majoritaire. Cette stratégie à l' avantage d' être simple et rapide.

Algorithme 3 : Sous-échantillonnage aléatoire

Data : Ensemble de données d'entrée D avec déséquilibre de classes
Result : Ensemble de données sous-échantillonné D' avec classes équilibrées

```
D' ← D ;
N ← Nombre d'instances dans la classe majoritaire ;
M ← Nombre d'instances dans la classe minoritaire ;
while N > M do
    Sélectionner une instance aléatoire x de la classe majoritaire ;
    Supprimer x de D' ;
    N ← N − 1 ;
end
```

3.3.3.3 Méthodes utilisées pour la sélection des attributs pertinents

- Variation : Un concept statistique appelé variance quantifie la variation d'une caractéristique dans un ensemble de données. L'utilisation de la variance dans la sélection des caractéristiques présente divers avantages. Premièrement, cela aide à identifier les caractéristiques à faible variance, signe d'informations insuffisantes ou de peu de variété entre les points de données. Deuxièmement, choisir des caractéristiques basées sur la variance permet de découvrir des caractéristiques hautement informatives avec des variances notables, ce qui peut améliorer la capacité de prédiction du modèle, La table 3.9 et la table 3.10 montrent les attributs selectionnés (les plus significatifs) selon la varience dans les deux base de données NSL KDD et CIC 2017.

Algorithme 4 : Sélection de caractéristiques basée sur la variance

Data : Ensemble de données d'entrée X avec des caractéristiques
Result : Ensemble de données d'entrée X' avec des caractéristiques sélectionnées basées sur la variance

```
Calculer la variance de chaque caractéristique dans X ;
Initialiser l'ensemble de caractéristiques sélectionnées X' ← ∅ ;
for chaque caractéristique x_i dans X do
    if la variance de x_i dépasse un seuil prédéfini then
        Ajouter x_i à l'ensemble X' des caractéristiques sélectionnées ;
    end
end
```

TAB. 3.9 : Taux de variance des valeurs des attributs gardés de la base NSL-KDD

Attribut	**Variance**
flag_enc	0.07
diff_srv_rate	0.03
same_srv_rate	0.19
dst_host_srv_serror_rate	0.19
dst_host_diff_srv_rate	0.03
count	0.05
srv_serror_rate	0.19
difficulty_level	0.01
dst_host_same_src_port_rate	0.09
protocol_type_enc	0.04
dst_host_serror_rate	0.19
dst_host_same_srv_rate	0.20
dst_host_srv_diff_host_rate	0.01
logged_in	0.23
dst_host_count	0.15
serror_rate	0.19
srv_count	0.02
dst_host_srv_count	0.18
service_enc	0.05
dst_host_rerror_rate	0.09
rerror_rate	0.10
dst_host_srv_rerror_rate	0.19
srv_rerror_rate	0.10
srv_diff_host_rate	0.06
is_host_login	0.01

Tab. 3.10 : Taux de variance des valeurs des attributs gardés de la base CIC-2017

Attribut	**Variance**
Source Port	0.11
Destination Port	0.07
Protocol	0.09
Flow Duration	0.07
Bwd Packet Length Max	0.01
Bwd Packet Length Mean	0.01
Bwd Packet Length Std	0.01
Flow IAT Max	0.04
Fwd IAT Total	0.07
Fwd IAT Std	0.01
Fwd IAT Max	0.04
Bwd IAT Total	0.05
Bwd IAT Max	0.02
Fwd PSH Flags	0.04
Fwd Header Length	0.01
Bwd Header Length	0.01
Packet Length Mean	0.01
Packet Length Std	0.01
FIN Flag Count	0.03
SYN Flag Count	0.04
PSH Flag Count	0.20
ACK Flag Count	0.21
URG Flag Count	0.08
Avg Bwd Segment Size	0.01
Fwd Header Length	0.01
Init_Win_bytes_forward	0.04
Init_Win_bytes_backward	0.01
min_seg_size_forward	0.01
Idle Mean	0.03
Idle Std	0.01
Idle Max	0.04
Idle Min	0.03
Flow ID_enc	0.07
Source IP_enc	0.01
Destination IP_enc	0.02
Timestamp_enc	0.04

- Algorithme des forets aléatoires : Random Forest est une technique d'apprentissage d'ensemble puissante qui est fréquemment utilisée dans l'apprentissage automatique pour la sélection de fonctionnalités. Plusieurs arbres de décision sont construits à l'aide de sous-ensembles aléatoires de données, et leurs prédictions sont combinées par vote ou par moyenne. La sélection d'entités à l'aide de Random Forest présente divers avantages. Parmi ces avantages, il est capable de gérer des ensembles de données de grande dimension et de capturer efficacement des relations d'entités complexes. Aussi, il offre une métrique pour l'importance des fonctionnalités, permettant d'identifier les caractéristiques les plus cruciales. La table 3.11 montre les attributs selectionnés selon les forets aléatoires dans la base de données CIC 2017.

Algorithme 5 : Sélection de caractéristiques basée sur la Random Forest

Data : Ensemble de données d'entrée X avec des caractéristiques

Result : Ensemble de caractéristiques sélectionnées X' basées sur le Random Forest

1 Construire un ensemble d'arbres de décision sur des sous-ensembles aléatoires de X ;

2 Calculer l'importance des caractéristiques basée sur les arbres de décision ;

3 Trier les caractéristiques par ordre décroissant d'importance ;

4 Sélectionner les k premières caractéristiques les plus importantes pour former X' ;

Tab. 3.11 : les attributs choisis selon les forets aléatoires

Attribut	**Importance**
Source IP_enc	13%
Flow ID_enc	9%
Timestamp_enc	6%
Bwd Packet Length Std	4%
Destination IP_enc	4%
Avg Bwd Segment Size	3%
Destination Port	3%
Bwd Packet Length Mean	3%
Average Packet Size	2%
Packet Length Mean	2%
Packet Length Std	2%
Bwd Packet Length Max	2%
Bwd Packet Length Min	2%
Subflow Fwd Bytes	2%
Max Packet Length	2%
Packet Length Variance	2%
Init_Win_bytes_backward	1%
Total Length of Fwd Packets	1%
Avg Fwd Segment Size	1%
Fwd Packet Length Max	1%
Init_Win_bytes_forward	1%
Total Fwd Packets	1%
Bwd Packets/s	1%
Fwd Packet Length Mean	1%
Source Port	1%
Fwd Packet Length Std	1%
PSH Flag Count	1%
Flow IAT Mean	1%
Fwd IAT Std	1%
Flow IAT Max	1%

3.4 Mesures d' évaluation

La matrice de confusion est un outil utile pour évaluer le bon fonctionnement d'un modèle de classification. En évaluant les prédictions du modèle avec les véritables étiquettes des informations, il fournit un bref aperçu de ces prédictions. Quatre catégories (vrais positifs, vrais négatifs, faux positifs et faux négatifs) sont utilisées par la matrice pour regrouper les résultats [Kulkami et al. 2020]. Nous pouvons en savoir plus sur l'efficacité, la précision, le rappel et les fausses alertes du modèle en examinant ces valeurs. La matrice de confusion nous permet également d'identifier les domaines spécifiques qui nécessitent des travaux.

De plus, la matrice nous permet de reconnaître les différents types d'erreurs commises par le modèle, ce qui peut nous aider à décider s'il faut changer le critère de classification ou se concentrer sur des classes particulières. En général, la matrice de confusion est un outil puissant pour évaluer, affiner et affiner les modèles de classification.

Tab. 3.12 : Matrice de confusion

Classe prédite	**Classe actuelle**	
	Normal	**Anormal**
Normal	VP	FP
Anormal	FN	VN

A l' aide de cette matrice de confusion, nous allons calculer Trois paramètres pertinents qui sont le taux de bon apprentissage, la précision et le taux de fausses alertes. Vous trouverez ci-dessous les formules de ces paramètres.

- Taux de bon apprentissage (Accuracy)
  Généralement utilisé pour évaluer l'efficacité d'un modèle de classification. Le taux de bon apprentissage affiche la proportion d'instances classées avec succès dans tous les exemples d'un jeu de données.

$$\text{Taux de bon apprentissage} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{VN} + \text{FP} + \text{FN}} \times 100 \tag{3.2}$$

- Précision
  Le succès d'un modèle de classification est mesuré par une statistique appelée précision, qui compte le nombre de cas positifs correctement prédits parmi toutes les occurrences positives.

$$\text{Précision} = \frac{\text{VP}}{\text{VP} + \text{FP}} \times 100 \tag{3.3}$$

- Taux de fausses alertes (fausse alarmes)
  La classification des paquets réguliers (qui n'incluent aucune agression) en tant qu'attaques est représentée par les fausses alertes (FP). C'est ainsi que leur taux est calculé.

$$\text{Fausse alarmes} = \frac{\text{FP}}{\text{FP} + \text{VN}} \times 100 \tag{3.4}$$

3.5 Résultats et discussions

Nous avons évalué l'efficacité de notre modèle de classification sur les ensembles de données NSL KDD et CIC 2017, en utilisant la précision, le taux de bon apprentissage et les fausses alarmes.

Nous avons cherché à déterminer le sous-ensemble de caractéristiques qui reflétaient le plus efficacement les relations sous-jacentes dans les données en utilisantdeuxméthodes de sélection de caractéristiques.

Pour résoudre le problème du déséquilibre entre les classes en matière de nombre de paquets, nous avons appliqué un sur échantillonnage sur les deux bases de données.

Pour surpasser le problème de la saturation de la RAM dû à la taille dès la base de données CIC 2017 nous avons appliqué un sous-échantillonnage permettant une réduction de dimensionnalité avec un maintien de généralisation de la base de données. Nous pouvons évaluer le succès de ces stratégies dans l'amélioration des performances de notre modèle de classification en comparant les résultats obtenus avant et après leur utilisation.

A. Classification binaire
   La classification binaire consiste à déterminer si le paquet testé et normal ou il représente une intrusion, sans spécification du type de ce dernier.

   - NSL-KDD
     Les différents résultats de la simulation en utilisant la base de données NSL KDD sont illustrés dans la table 3.13 présentée ci-dessous. Nous avons opté dans un premier temps pour la variance lors du choix des attributs avec un seuil de 0.01, nous avons pu sélectionner 25 attributs et c' était le meilleur choix vu qu' on a eu des très bons résultats, ou les taux de bonne détection et la précision on atteint tous deux les 100%, quant aux fausses alarmes sont complètement inexistantes.

Tab. 3.13 : Performances de la classification binaire via TVi sur NSLKDD avant et âpres sélection des attributs parla méthode de "variation"

Avant et après réduction de dimensionnalité	Classe	Taux de bon apprentissage	Précision	Fausses alarmes
Avant : 42 attributs	Normal	93%	100%	0%
	Anormal	93%	87%	13%
Après : 25 attributs	Normal	100%	100%	0%
	Anormal	100%	100%	0%

   - CIC 2017
     Vu que la base CIC 2017 est très volumineuse par rapport à NSL-KDD, où elle compte en totale 2 827 876 (voir table 3.14), il n' était malheureusement pas possible de mener a bien l' étape d' apprentissage, suite a des problèmes de capacité de la RAM et même avec usage de Google Colab, par conséquent nous avons réalisé une réduction de dimensionnalité en utilisant un sous-échantillonnage (voir section 3.3.3.2 (2)). Le nombre de paquets retenus suite à l' application de cet algorithme est de « 200000 » paquets pour chacune des deux classes « Normal » et « Anormal » pour résoudre le problème des limitations matérielles.

Tab. 3.14 : Nombre de paquets dans la base CIC 2017

Nombre de paquets avant réduction	Classe : normal	Classe : anormal
Total	2271320	556556
Base d'apprentissage	200000	200000
Base de test	20000	20000

Nous avons appliqué par la suite une sélection des attributs pertinents via la méthode des variations, nous avons obtenu 38 attributs, puis nous avons exécuté l' algorithme TVi, les résultats obtenus sont illustrés dans la table 3.15 présentée ci- dessous, et montrent une amélioration en matière de performances de classification suite à la sélection des attributs via la variance.

Tab. 3.15 : Résultat de classification via TVi sur CIC avant et âpres sélection des attributs par "variation"

Nombre d'attributs	Classe	Taux de bon apprentissage	Précision	Fausses alarmes
Avant (85 attributs)	Normal	87%	87%	12%
	Anormal	87%	87%	12%
Après (38 attributs)	Normal	98%	98%	1%
	Anormal	98%	98%	1%

Afin d' améliorer les performances de notre modèle de détection encore une fois, nous avons dû réviser les étapes en amant de l' apprentissage et l' élaboration du modèle, or nous nous sommes résolu au changement de l' algorithme de sélection des caractéristiques, où nous avons testé cette fois-ci l' algorithme des forêts aléatoires « Random forest » qui à déterminé 31 attributs significatives. L' algorithme a cependant prouvé son efficacité, ceci est illustré dans la table 3.16 présentée ci-dessous. Nous remarquons clairement une maximisation des taux de bon apprentissage et de précision à 100%.

Tab. 3.16 : Performance de classification via TVi sur la base CIC 2017 réduite par échantillonnage et sélection des attributs avec "Forets aléatoires"

Performance	Normal	Anormal
Taux de bon apprentissage	100%	100%
Précision	100%	100%
Fausses alarmes	0%	0%

B. Classification multi classes

- Sélection des caractéristiques
  Dans un premier temps, nous avons testé notre modèle en utilisant la base de données NSL KDD, la classe anormale de cette dernière contient à son tour 4 classes qui sont les suivantes :« DoS », « Probe », « R2L » et « U2R » en plus de sa classe normal. Dans la première simulation nous avons utilisé le même algorithme de sélection des caractéristiques qui basé sur les variations, vu qu' il a donné d' excellent résultats dans la classification binaire et, nous avons notamment gardé le même nombre de paquets initial dans chaque class. Les résultats obtenus présentés dans la table 3.17 ne sont pas satisfaisants. Nous pouvons voir que la précision de la classe DoS est égale à 0% car tous simplement les vrais positifs de cette classe sont nuls.

Tab. 3.17 : Résultats de classification via TVi sur NSL-KDD avec sélection des attributs par la méthode de "variation"

Performance	Normal	DoS	Probe	R2L	U2R	Total
Nombre de Paquets	67343	12703	46974	57635	58578	243233
Taux de bon apprentissage	89%	94%	97%	97%	84%	88%
Précision	76%	0%	94%	95%	68%	66%
Fausses alarmes	10%	0%	1%	1%	10%	4%

Nous avons testé dans un deuxième temps notre modèle sur la base de données CIC, sa classe anormale contient à son tour 7 classes à savoir :« DoS », « DDoS », « Web Attack », « Botnet », « Scan », « Heartbleed » et « Patator » en plus de la classe normale. Cette base de données contient plus de classes que la base NSL KDD. Dans cette simulation nous avons choisi de sélectionner nos attributs avec le même algorithme qui est la « Variance », nous avons également procéder à la réduction de la base via un sous-échantillonnage pour les mêmes raisons déjà citées dans la classification binaire. Selon les résultats illustrés dans la table 3.18, nous remarquons qu' il existe 4 classes avec 0% au niveau de la précision, il s' agit des classes Web Attack, Botnet, Heartbleed et Patator. Nous remarquons également que le taux de bon apprentissage n' est pas satisfaisant.

Tab. 3.18 : Performance de classification via TVi sur la base CIC 2017 réduite via un sous-échantillonnage, et attributs sélectionnés avec la "Variance"

Performance	Normal	DoS	DDoS	Web Attack	Botnet	Scan	Heartbleed	Patator	Total
Taux de bon apprentissage	63%	76%	86%	99%	99%	90%	99%	96%	88.5%
Précision	39%	0%	81%	0%	0%	80%	0%	0%	25%
Fausses alarmes	47%	0%	3%	0%	0%	6%	0%	0%	7%

Afin d' améliorer les performances de notre SDI, nous avons changé l' algorithme de sélection des attributs, ou nous avons opté cette fois -ci pour l' algorithme des forêts aléatoires (voir section 3.3.3.3 (2)). La table 3.19 ci-dessous présente lesnouveaux résultatsobtenus suite à ce changement de l' algorithme de réduction de caractéristiques.

Comme nous pouvons le voir, certaines classes, en particulier les classes « Normal » et « DoS », ont connu une nette amélioration, tandis que d' autres telles que « DDoS » et « Scan » ont vu leurs performances baisser mais très légèrement.

Tab. 3.19 : Performances de classification via TVi sur la base CIC 2017 âpres sélection des attributs via "Random Forest" et réduction via un sous-échantillonnage aléatoire

Performance	Normal	DoS	DDoS	Web Attack	Botnet	Scan	Heartbleed	Patator	Total
Taux de bon apprentissage	84%	84%	85%	99%	99%	86%	99%	96%	91%
Précision	66%	67%	68%	0%	0%	69%	0%	0%	33.75%
Fausses alarmes	10%	10%	10%	0%	0%	11%	0%	0%	5%

- Équilibrage des données

Notre objectif était d' apporter une amélioration en matière de performances de détection, les résultats obtenus suite au changement effectué concernant l' algorithme de sélection des attributs les plus significatifs étaient meilleurs, mais pas assez. Nous avons envisagé d'équilibrer les données afin de réduire l'impact d'une distribution non équilibrée des classes, plus particulièrement la classe « DoS », qui contient dans NSL KDD « 12703 » paquets par rapport à la classe « Normal » qui contient « 67343 » et qui est considéré par conséquent comme une classe majoritaire. La base de données CIC 2017 est caractérisée également par un déséquilibre au niveau des classes.
Dans la simulation illustrée dans la table 3.20, nous exposant les résultats obtenus suite à un équilibrage des données entre les 5 classes de la base NSL-KDD, en considérant le nombre de paquets de la plus grande classe qui est « Normal ». Par conséquent les cinq classes contiennent après cette étape, toutes le même nombre de paquets « 67343 ». L' algorithme de sur échantillonnage aléatoire adopté a permis d'améliorer la précision des prédictions, où nous remarquons une augmentation au niveau des valeurs totales du taux de bon apprentissage, précision et les fausses alarmes qui ont atteint respectivement « 91% », « 91% » et « 2% ».

Tab. 3.20 : Performances de classification via TVi sur la base NSL KDD équilibrée viaun sur échantillonnage et une sélection des attributs avec la "Variance"

Performance	Normal	DoS	Probe	R2L	U2R	Total
Taux de bon apprentissage	97%	98%	95%	94%	96%	91%
Précision	92%	97%	89%	87%	90%	91%
Fausses alarmes	2%	1%	2%	3%	2%	2%

Concernant la base de données CIC 2017, l' équilibrage suivant la classe majoritaire a permis l' obtention de « 50000 » paquets par classe. Les résultats illustrés dans la table 3.21 démontrent une augmentation des valeurs de précision et du taux d' apprentissage et une réduction des fausses alarmes, pour les deux méthodes de sélection d' attributs, ce qui démontre l' efficacité de l' équilibrage de la base via le sur échantillonnage.
Notant toutefois que l' algorithme des forêts aléatoires est meilleur en matière de sélection des caractéristiques sure la base CIC 2017.
Suite à toutes ses simulations, la meilleure combinaison qui représente notre approche proposée pour la base CIC 2017 est TVi, forêts aléatoire et sur échantillonnage.

Tab. 3.21 : Performances de classification de la base CIC 2017 via TVi après équilibrages des données avec un sur échantillonnage aléatoire et deux méthodes de sélection d' attributs

Sélection des attributs	Performance	Normal	DoS	DDoS	Web Attack	Botnet	Scan	Heartbleed	Patator	Total
Via Variance	Taux de bon apprentissage	97%	88%	87%	98%	85%	91%	98%	98%	92.75%
	Précision	88%	53%	0%	95%	43%	59%	92%	94%	65.5%
	Fausses alarmes	1%	9%	0%	1%	8%	9%	1%	1%	3.75%
Via Random Forests	Taux de bon apprentissage	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Précision	100%	100%	100%	100%	100%	100%	100%	100%	100%
	Fausses alarmes	0%	0%	0%	0%	0%	0%	0%	0%	0%

3.6 Étude comparative

Afin de prouver l' efficacité de notre modèle nous avons réalisé une comparaison avec le travail de Ho et al. (2022), les auteurs ont proposé une approche basée sur un TVi en effectuant une sélection des caractéristiques via un arbre de décision sur la base de données CIC 2017. Nous avons apporté une amélioration en matière de performances de classification de notre SDI, grâce à l' équilibrage des données en appliquant sur échantillonnage aléatoire. Les histogrammes présentés dans les figures 3.3 et 3.4 ainsi que la table 3.22 démontrent une comparaison entre notre SDI proposé et celui de Ho et son équipe. Nous remarquons clairement que les valeurs des taux de bon apprentissage et de la précision ont atteint les 100% pour les deux types de classification, binaire et multi-classes via notre SDI proposé, contre respectivement les valeurs de 98,5 et 98.23 dans la classification binaire et 96,40 et 88.75 dans la classification multi-classes.

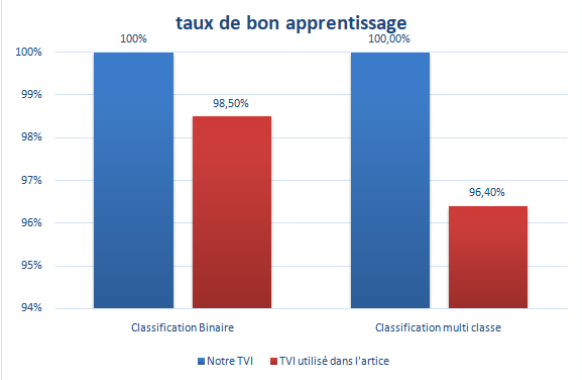

Fig. 3.3 : Comparaison des taux de bon apprentissage entre notre approche proposée et celle de Ho et al. (2022) en utilisant la base CIC 2017.

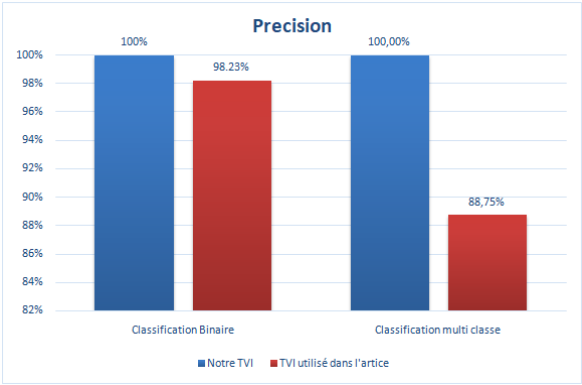

Fig. 3.4 : Comparaison en matière de précision entre notre approche et celle de Ho et al. (2022) en utilisant la base CIC 2017.

Tab. 3.22 : Comparaison des performances de classification avec l' estimateur "Précision" entre notre SDI proposé et celui de Ho et al. (2022) de la base CIC 2017.

	Normal	**DoS**	**DDoS**	**Web Attack**	**Botnet**	**Scan**	**Heartbleed**	**Patator**	**Total**
TVI utilisé dans l'article	99%	95%	100%	93%	84%	50%	89%	100%	88.75%
Notre TVI	100%	100%	100%	100%	100%	100%	100%	100%	100%

3.7 Simulation de notre SDI

3.7.1 Matériel utilisé

- HP probook 450 G8.
- Processeur : I5-1135G7.
- Mémoire : 8 Go DDR4.
- Système d' exploitation : Windows 10 64bits.

3.7.2 Environnement de développement

Le puissant langage de programmation Python est la base de l'environnement de codage que nous avons utilisé pour créer notre système de détection d'intrusion. La grande variété de bibliothèques de Python contribue à sa popularité. Python présente de nombreux avantages en matière d'exploration de données. Ses bibliothèques, telles que Pandas et NumPy, offrent des outils efficaces d'analyse et de manipulation des données, nous permettant de tirer des conclusions précieuses à partir d'ensembles de données complexes. La lisibilité et la simplicité de Python permettent un développement rapide, ce qui nous permet de tester rapidement différentes stratégies d'exploration de données.

Python excelle également dans le domaine du deep learning, élément clé des systèmes de détection d'intrusion contemporains. TensorFlow, un cadre d'apprentissage en profondeur populaire, offre une API de haut niveau pour créer et former rapidement des réseaux de neurones. La mise en œuvre de systèmes complexes d'apprentissage en profondeur est simplifiée par la syntaxe de Python et l'adaptabilité de TensorFlow. Python est une excellente option pour notre cas, afin de pouvoir exploiter python, nous avons opté à l' utilisation de « Visual Studio Code » comme environnement de développement.

3.7.3 Présentation de notre application

- Fenêtre principale : dès le lancement de l' exécution de notre application,la fenêtre illustrée par la figure ci-dessous s' ouvre.

  A gauche de la fenêtre nous avons un panneau contenant les étapes de prétraitement, afin nettoyer, coder et normaliser les données pour les rendre exploitable par notre modèle TVI.

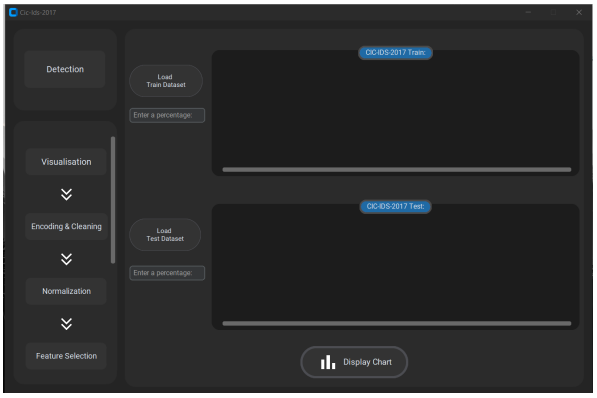

FIG. 3.5 : Fenêtre de principale.

- Affichage de la base d' origine : c' est la première étape de visualisation, on peut choisir un pourcentage de la base de données à afficher que ça soit celle de l' apprentissage ou test, ceci est illustré par la figure 3.5.

- Information sur la base : la figure 3.7 présente les différentes classes existantes dans la base de données accompagnée de leur distribution.

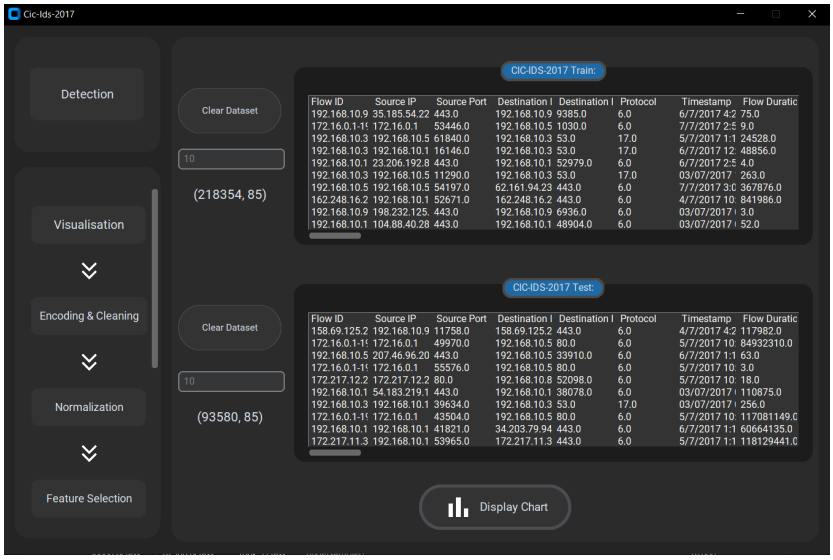

Fig. 3.6 : Affichage de la base de donnée âpres lecture.

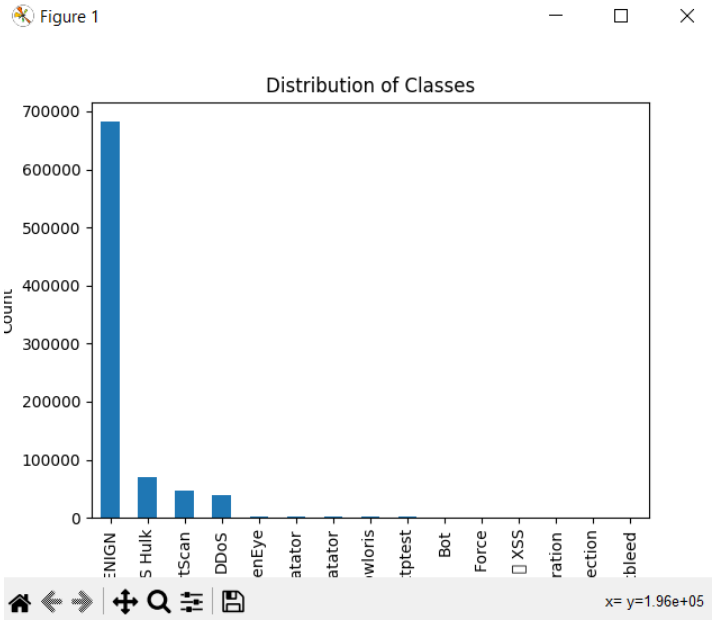

Fig. 3.7 : Différentes classes qui existe dans l' ensemble de données.

- Nettoyage et codage : cette option permet d' afficher les données avant et âpres codage et nettoyage tout en affichant le nombre de paquets traités (voir figure 3.8).

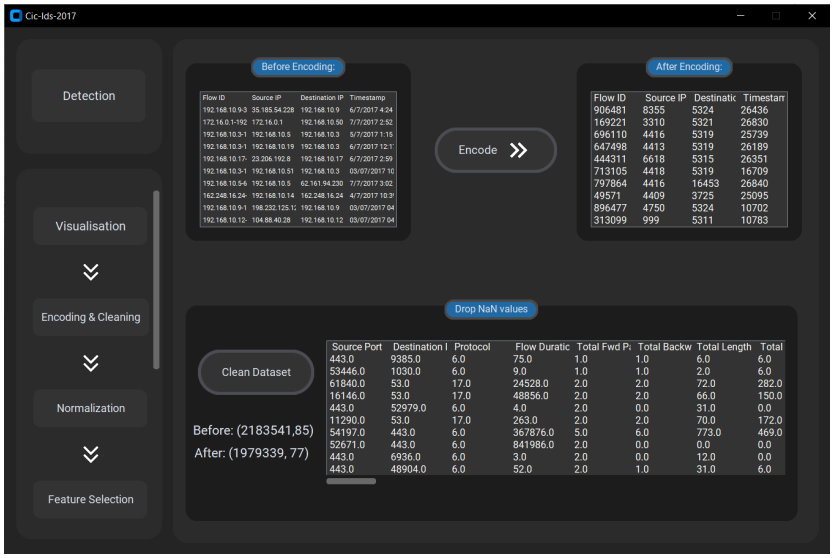

Fig. 3.8 : Fenêtre de codage et nettoyage

- Normalisation : une autre étape de prétraitement qui peut être aussi visualisée (voir figure 3.9).

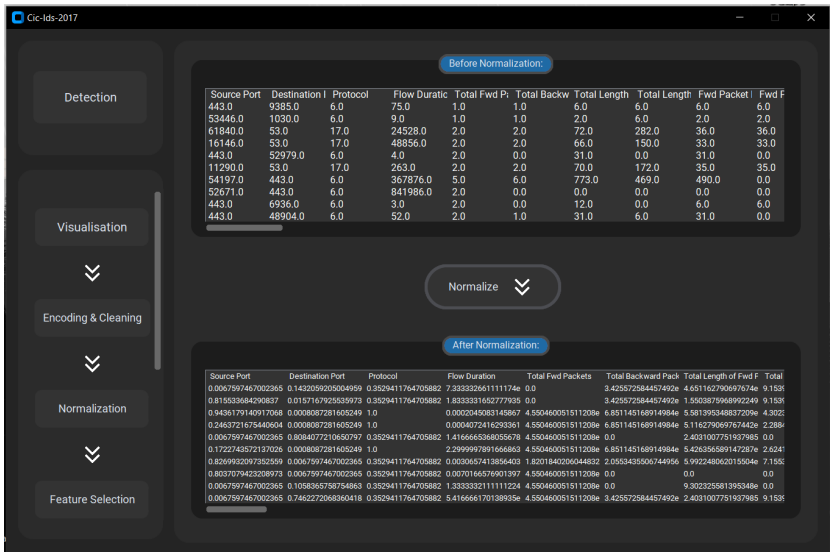

Fig. 3.9 : Fenêtre de normalisation

- Choix des paramètres : la fenêtre illustrée par la figure 3.10 permet la sélection de l' algorithme à utiliser et les attributs à enlever ou à garder pour chaque algorithme, elle permet également d' afficher la taille après suppression des attributs.

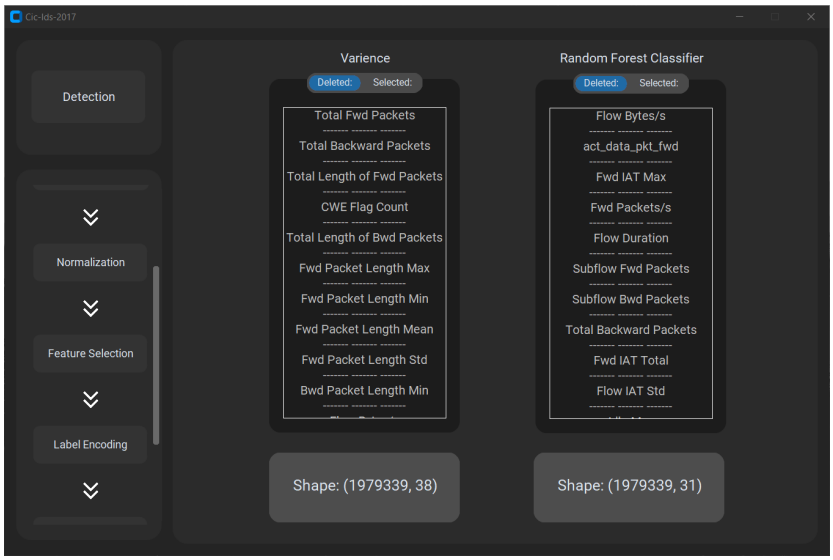

Fig. 3.10 : Fenêtre de sélection d' attributs.

- Codage de la colonne « Label » : Rassemblement des sous classes en classe mère et codage.

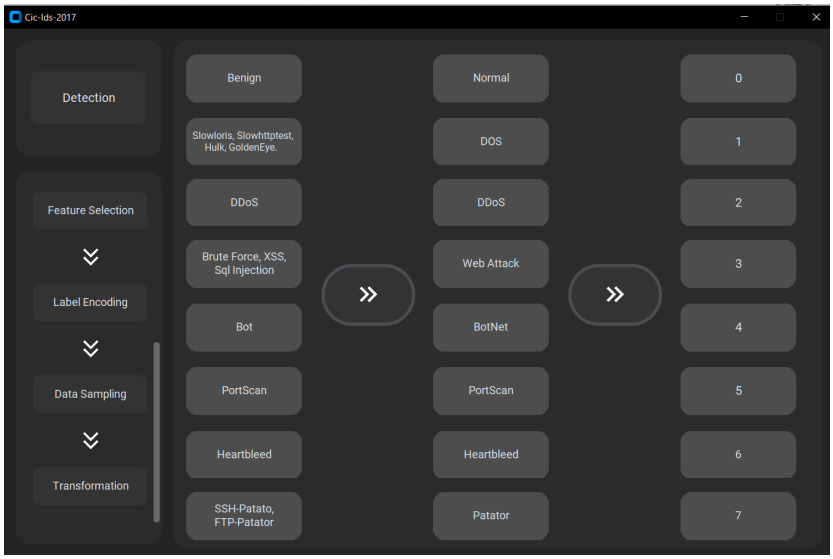

Fig. 3.11 : Fenêtre du codage de la colonne "Label"

- Sur et sous échantillonnage : après l' encodage des classes, il est possible maintenant de procéder au sur échantillonnage ou au sous échantillonnage comme illustré dans la figure suivante.

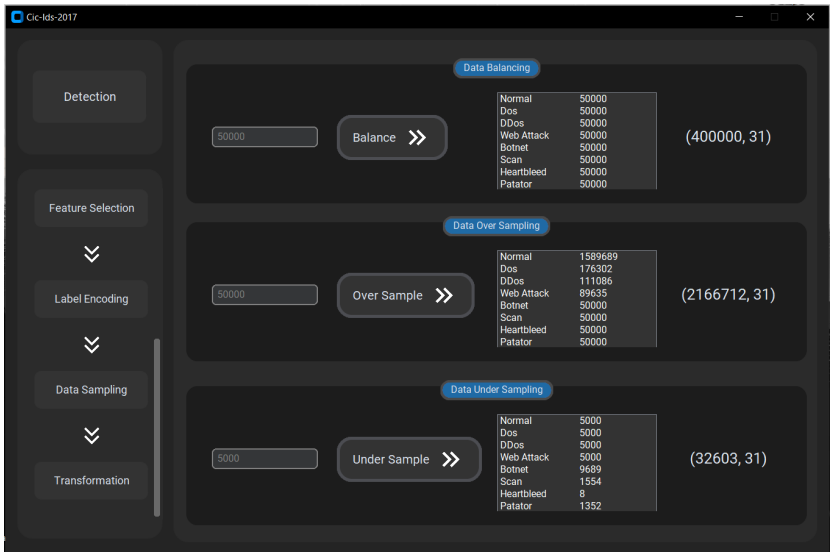

Fig. 3.12 : Fenêtre de sur et sous échantillonnage

- Transformation des données en image : suite à cette opération, chaque classe sera représentée par une image, qui sera découpée au format « 224x31 », le « 31 » représente le nombre d' attributs obtenus dans cette exemple via laméthode « Random Forest » pour le TVi s' encharge de rendre les images sous forme de matrice carrée « 224x224 ».

  On peut voir dans la figure 3.13 un exemple des patchs de chaque classe qui vont servir comme entrée X et leurs labels des classes qui vont servir comme entrée Y.

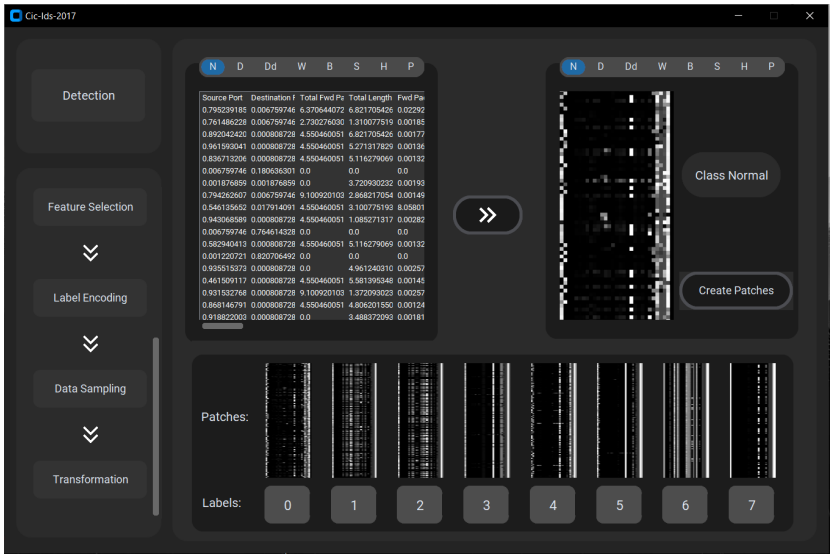

Fig. 3.13 : Fenêtre de transformation en image.

- Détection d' intrusion : maintenant après avoir préparé les entrées de notre modèle TVi, on peut procéder à l' étape « Détection » qui est la dernière étape, et qui est illustrée dans les figures 3.14, 3.15.On peut lancer la détection en cliquant sur le bouton « Start Scanning » , nous également une console en bas pour montrer la prédiction de notre modèle, qui spécifie la classe actuelle et la classe prédite. A droite de la console, on a les deux paramètres « Accuracy » et « Loss » qui sont affichés.

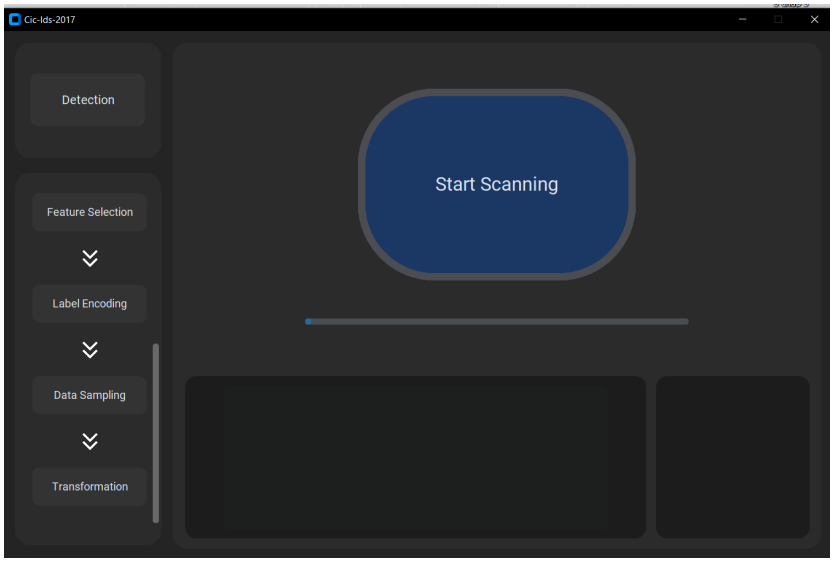

Fig. 3.14 : Fenêtre de détection partie 1

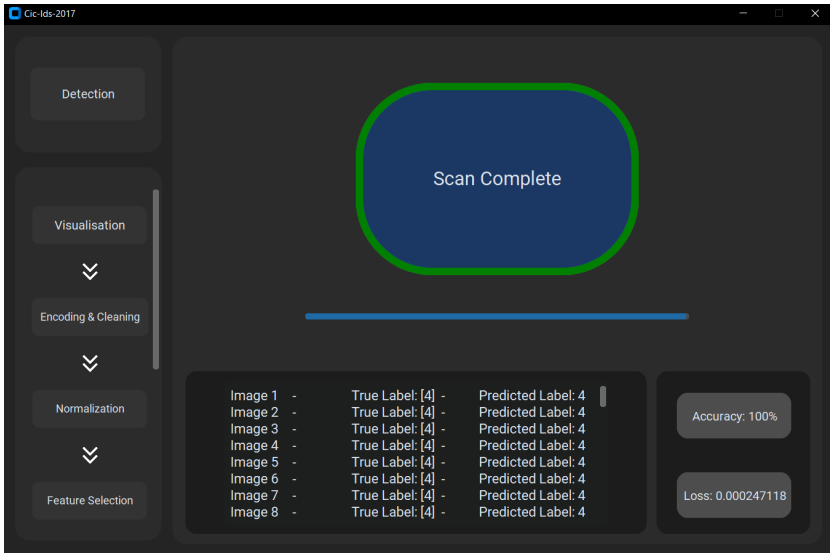

Fig. 3.15 : Fenêtre de détection partie 2

3.8 Conclusion

Dans ce chapitre, nous avons présenté notre système de détection d' intrusion proposée, où nous avons commencé par décrire le principe général de notre approche en fournissant une représentation algorithmique. Nous avons illustré les méthodes utilisées et décrit chaque étape, en insistant sur sa fonction et son importance pour l'ensemble de la procédure. De plus, nous avons souligné l' importance des stratégies d'équilibrage des données, qui ont réduit les problèmes de déséquilibre de classe et améliorer l'efficacité de notre approche dans son ensemble.

Lors de l' élaboration de notre SDI, nous avions comme objectif d' aboutir à de bonnes performances de détection, nous étions dans une quête d' identification du meilleur algorithme de sélection des caractéristiques qui s' est avérée déterminante, effectivement selon les résultats illustrés le choix des attributs influence vraiment la prise de décision lors de la classification.

L'utilisation du modèle Vision Transformer (TVI), qui est bien connue pour ses performances exceptionnelles dans les tâches de classification d'images, est l'un des éléments essentiels qui ont aidé à avoir des résultats satisfaisants. La couche distinctive connue sous le nom « attention multi-têtes » dans le modèle TVi le distingue. Cette couche donne au modèle la capacité d'apprendre et de classer efficacement diverses caractéristiques en lui permettant de capturer des dépendances et des relations complexes au sein de l'image, ce qui a permis une précision puissante lors de la classification.

Cette analyse complète a démontré l'efficacité de notre approche proposée, justifiant son potentiel à surpasser les solutions existantes sur le terrain, ceci a été illustré avec une comparaison avec un travail de la littérature.

Conclusion Générale

Le monde a connu plusieurs attaques informatiques et il est toujours en train de les subir, différentes méthodes de détection d'intrusion pour assurer la sécurité informatique ont été réalisés, tous avec le même objectif, arrêter ces attaques et protéger les systèmes informatiques et les réseaux.

Afin de seriner ce domaine qui est assez vaste, nous avons commencé par examiner différentes techniques issues de la cyber sécurité traditionnelle qui existent telles que les pare-feu, les logiciels antivirus et les serveurs proxy. Cependant, tous au long de notre mémoire, nous nous sommes concentré sur les SDIs , sujet de notre projet de fin d' études. Notant toute fois que ces solutions efficaces viennent compléter celle qui existe déjà.

L' apprentissage Profond a fait ces preuves dans divers domaines, et a connu une large utilisation ces dernières années, surtout lorsqu' on dispose d' une base de données assez importante et étiqueté. Les méthodes issues de l' AP utilisées dans la cyber sécurité ont été présentées dans le deuxième chapitre, tel que le RNN, TVi, UNet, ResNet et RNC. Nous avons par conséquent pu comprendre les avantages de ses méthodes-là et leur potentiel dans notre domaine qui est la détection d' intrusions réseau, or nous avons opté pour l' algorithme TVi qui est doté d' un mécanisme d'auto-attention, qui permet au réseau de se concentrer sur différentes parties de l'image pour capturer les dépendances à longue portée.

Notre approche proposée en vue d' une détection d' intrusion est présentée dans le troisième chapitre. Pour entraîner notre modèle SDI, nous avons utilisé deux ensembles de données CIC 2017 et NSL-KDD. Notre objectif principal était d'utiliser efficacement l'algorithme on Transformateur de Vision) pour détecter les intrusions de manière précise et rapide. Nous nous sommes assurés que nos données d'entraînement étaient fiables en les nettoyant et en les préparant soigneusement. De plus, nous avons procéder à la sélection des attributs les plus pertinents afin d'améliorer l' efficacité du TVi dans l'identification et la catégorisation des intrusions.

Les résultats de nos tests étaient très encourageants pour NSL KDD et trés satisfaisante pour CIC 2017, et notre SDI a atteint un taux de précision remarquable de 100% pour la base NSL KDD dans la classification binaire et la base CIC 2017 dans la classification binaire et celle en multi classes. Une comparaison avec un travail de la littérature a montré notamment l' efficacité de notre approche basée sur l' algorithme TVi et l' échantillonnage.

Cependant comme perspective, nous proposons de revoir la classification multi classes de la base NSL-KDD afin d' optimiser les résultats obtenus en cherchant un autre algorithme plus adéquat que ça soit dans la classification ou au niveau de la sélection des attributs, pour cela il faudrait étudier la structure de la base de plus prés.

Annexe 1 Bibliothèques utilisées

Bibliothèque	Description
Pandas	Bibliothèque puissante pour la manipulation et l'analyse de données structurées.
Numpy	Bibliothèque fondamentale pour le calcul numérique en Python, offrant des structures de données efficaces.
Matplotlib	Bibliothèque de visualisation de données en 2D, permettant de créer des graphiques et des tracés.
Sklearn.preprocessing	Module de prétraitement des données de Scikit-learn, proposant des outils pour la préparation des données.
RandomForestClassifier	Classe d'algorithme d'apprentissage automatique basé sur les forêts d'arbres décisionnels utilisé lors des sélection de caractéristiques
PIL	Bibliothèque Python pour le traitement d'images, offrant des fonctionnalités pour manipuler des images.
Patchify	Bibliothèque utilisée pour découper une grande image en petites patchs, souvent utilisée en imagerie.
OS	Module Python fournissant des fonctionnalités liées au système d'exploitation, comme la gestion des fichiers et des répertoires.
Csv	Module intégré de Python pour la lecture et l'écriture de fichiers CSV (Comma-Separated Values).
imblearn.over_sampling	Module de sur échantillonnage des données déséquilibrées dans la bibliothèque imbalanced-learn.
imblearn.under_sampling	Module de sous-échantillonnage des données déséquilibrées dans la bibliothèque imbalanced-learn.

keras	Bibliothèque d'apprentissage profond (deep learning) haut niveau, basée sur TensorFlow.
sklearn.metrics	Module de Scikit-learn qui contient des métriques d'évaluation pour évaluer les performances des modèles.
layers	Module de TensorFlow/Keras qui fournit des couches pour construire des modèles d'apprentissage profond.
tensorflow_addons	Extension officielle de TensorFlow fournissant des fonctionnalités supplémentaires pour TensorFlow.

Annexe 2 Fonctions utilisées

Fonctions	Description
read_csv()	Fonction de la bibliothèque pandas pour lire un fichier CSV et le charger dans un DataFrame.
dropna()	Fonction de pandas pour supprimer les lignes contenant des valeurs manquantes (NaN) dans un DataFrame.
LabelEncoder()	Fonction de la bibliothèque scikit-learn qui convertit les étiquettes de classe en nombres entiers pour une utilisation avec les algorithmes d'apprentissage automatique.
MinMaxScaler()	Fonction de scikit-learn pour mettre à l'échelle les caractéristiques d'un ensemble de données dans une plage spécifiée, généralement entre 0 et 1.
var()	Méthode de pandas qui calcule la variance des données dans un DataFrame.
drop()	Méthode de pandas pour supprimer des colonnes ou des lignes spécifiées d'un DataFrame.
save()	Méthode pour enregistrer un objet Python (comme un modèle d'apprentissage automatique) dans un fichier pour une utilisation ultérieure.
to_csv()	Méthode de pandas pour écrire un DataFrame dans un fichier CSV.
patchify()	Fonction d'une bibliothèque appelée "patchify" qui découpe une grande image en petits patches, souvent utilisée en imagerie.
imshow()	Fonction de la bibliothèque matplotlib pour afficher une image.

Normalization()	Couche de normalisation dans TensorFlow/Keras pour normaliser les données d'entrée.
fit_resample()	Fonction de la bibliothèque imbalanced-learn pour effectuer l'échantillonnage des données déséquilibrées en adaptant les méthodes de suréchantillonnage et de sous-échantillonnage.
Resizing()	Couche de redimensionnement dans TensorFlow/Keras pour redimensionner une image à une taille spécifiée.
Dense()	Couche dense (entièrement connectée) dans TensorFlow/Keras pour créer une couche de neurones entièrement connectés.
Dropout()	Couche de régularisation dans TensorFlow/Keras pour appliquer une régularisation de type "dropout" aux neurones d'une couche.
Embedding()	Couche d'incorporation dans TensorFlow/Keras pour créer une représentation vectorielle dense des données catégorielles.
Position_embedding()	Couche d'encodage de position dans TensorFlow/Keras pour ajouter des informations de position aux données d'entrée.
create_vit_classifier()	Fonction pour créer un modèle de classification basé sur le modèle "Vision Transformer" (TVI).
run_experiment()	Fonction pour exécuter une expérience d'apprentissage automatique ou d'apprentissage profond avec un modèle spécifié.

Bibliographie

[Rivet et Alain, 2007]. Rivet, A. (2007). Normes de qualité et systèmes d' information. Les Journées Réseaux-JRES, Strasbourg.

[Sang et Fernand, 2012]. Sang, F. L. (2012). Protection des systèmes informatiques contre les attaques par entrées-sorties (Doctoral dissertation, INSA de Toulouse).

[Salma et Tidjani, 2015]. Salma, B. A. H. A. Z. Déploiement d'une solution Anti-virus (Doctoral dissertation).

[Dumont, 2009]. Dumont, R. (2009). Cryptographie et sécurité informatique. Cours provisoires, Université de liege Faculté des sciences appliqués 2010.

[Han et al., 2020]. Han, S. H., Lee, H. K., Gim, G. Y., Kim, S. J. (2020). Empirical study on anti-virus architecture for container platforms. IEEE Access, 8, 134940-134949.

[SUNGHWAN et AL., 2020]. Kim, S., Yoon, S., Narantuya, J., Lim, H. (2020). Secure collecting, optimizing, and deploying of firewall rules in software-defined networks. IEEE Access, 8, 15166-15177.

[Herrb, 2022] . Herrb, M. Installation de pare-feu redondants avec OpenBSD.

[Prieur et Benoît, 2020]. Prieur, B. (2020). Fondamentaux de la sécurité informatique utilisateur (Doctoral dissertation, IT-Akademy).

[Kocaleva et AL., 2022]. Kocaleva, M., Zlatanovska, B., Karamazova Gelova, E., Stojanova, A., Miteva, M. (2022). Apache http server as forward proxy server.

[Muller et Al., 2004]. Müller, J., Fischer, S., Gorlatch, S., Mauve, M. (2004). A proxy server-network for real-time computer games. In Euro-Par 2004 Parallel Processing : 10th International Euro-Par Conference, Pisa, Italy, August 31-September 3, 2004. Proceedings 10 (pp. 606-613). Springer Berlin Heidelberg.
[Ferguson et AL., 1998]. Ferguson, P., Huston, G. (1998). What is a VPN ?.

[Antonio et Al., 2022]. Gentile, A. F., Macrì, D., De Rango, F., Tropea, M., Greco, E. (2022). A

VPN Performances Analysis of Constrained Hardware Open Source Infrastructure Deploy in IoT Environment. Future Internet, 14(9), 264.

[Sistani et Patel, 2016]. Sistani, A. K. R., Patel, A. M. (2016). Design and evaluation of a virtual private network architecture for collaborating specialist users. Asia-Pacific Journal of Information Technology and Multimedia, 5(1), 15-30.

[MAHREZ et AL., 2020]. MAHREZ, D., ZERROUKI, A. N. (2020). Pratique de solution tunnel VPN IPsec dans un environnement Virtuel et open source (Doctoral dissertation, Université Ibn Khaldoun-Tiaret-).

[Liao et AL., 2013]. Liao, H. J., Lin, C. H. R., Lin, Y. C., Tung, K. Y. (2013). Intrusion detection system : A comprehensive review. Journal of Network and Computer Applications, 36(1), 16-24.

[DeVivo et AL., 1999]. De Vivo, M., Carrasco, E., Isern, G., De Vivo, G. O. (1999). A review of port scanning techniques. ACM SIGCOMM Computer Communication Review, 29(2), 41-48.

[Kemal et Bulent, 2009]. Bicakci, K., Tavli, B. (2009). Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks. Computer Standards Interfaces, 31(5), 931-941.

[Calvet et Joan, 2013]. Calvet, J. (2013). Analyse dynamique de logiciels malveillants (Doctoral dissertation, École Polytechnique de Montréal).

[Bouzayani et Hatem, 2012]. Bouzayani, H. (2012). Modèle quantitatif pour la détection d'intrusion : une architecture collaborative IDS-HONEYPOT (Doctoral dissertation, Université du Québec en Outaouais).

[Gurney et Kevin, 1997]. Gurney, K. (1997). An introduction to neural networks. CRC press.

[Borne et al., 2007]. Borne, P., Benrejeb, M., Haggège, J. (2007). Les réseaux de neurones : présentation et applications (Vol. 15). Editions OPHRYS.

[Goodfellow et al., 2016]. Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT press.

[Salehinejad et al., 2017]. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S. (2017). Recent advances in recurrent neural networks. arXiv preprint arXiv :1801.01078. [Medsker et al. 2001]. Medsker, L. R., Jain, L. C. (2001). Recurrent neural networks. Design and Applications, 5, 64-67.

[Hochreiter et al., 1997]. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[Li et al., 2021]. Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. (2021). A survey of convolutional neural networks : analysis, applications, and prospects. IEEE transactions on neural networks and learning systems.

[Kattenborn et al., 2021]. Kattenborn, T., Leitloff, J., Schiefer, F., Hinz, S. (2021). Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. ISPRS journal of photogrammetry and remote sensing, 173, 24-49.

[He et Kaiming, 2016]. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[Imrana et al., 2021]. Imrana, Y., Xiang, Y., Ali, L., Abdul-Rauf, Z. (2021). A bidirectional LSTM deep learning approach for intrusion detection. Expert Systems with Applications, 185, 115524.

[Altunay et Albayrak, 2023]. Altunay, H. C., Albayrak, Z. (2023). A hybrid CNN+ LSTMbased intrusion detection system for industrial IoT networks. Engineering Science and Technology, an International Journal, 38, 101322.

[Mushtaq et al., 2022]. Mushtaq, E., Zameer, A., Umer, M., Abbasi, A. A. (2022). A two-stage intrusion detection system with auto-encoder and LSTMs. Applied Soft Computing, 121, 108768.

[ElSayed et al., 2021]. ElSayed, M. S., Le-Khac, N. A., Albahar, M. A., Jurcut, A. (2021). A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique. Journal of Network and Computer Applications, 191, 103160.

[Ho et al., 2022]. Ho, C. M. K., Yow, K. C., Zhu, Z., Aravamuthan, S. (2022). Network Intrusion Detection via Flow-to-Image Conversion and Vision Transformer Classification. IEEE Access, 10, 97780-97793.

[Belgrana et al., 2020]. Belgrana, F. Z., Benamrane, N., Hamaida, M. A., Chaabani, A. M., Taleb-Ahmed, A. (2021, January). Network intrusion detection system using neural network and condensed nearest neighbors with selection of NSL-KDD influencing features. In 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS) (pp. 23-29). IEEE.

[Dhanabal et Shantharajah, 2015]. Dhanabal, L., Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. International journal of advanced research in computer and communication engineering, 4(6), 446-452.

[Dosovitskiy et al., 2020]. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2020). An image is worth 16x16 words : Transformers for image recognition at scale. arXiv preprint arXiv :2010.11929.

[Han et al., 2016]. Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., ... Tao, D. (2022). A survey on vision transformer. IEEE transactions on pattern analysis and machine intelligence, 45(1), 87-110.

[Kulkami et al., 2020]. Kulkarni, A., Chong, D., Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. In data democracy (pp. 83-106). Academic Press.

Webographie

NSL KDD. "UCI Knowledge Discovery in Databases Archive 'KDD Cup 1999 Data'". Available at : https://www.unb.ca/cic/datasets/nsl.html [Accessed : June 18, 2023].

CIC. "Canadian Institute for Cybersecurity datasets". Available at : https://www.unb.ca/cic/datasets/ids-2017.html [Accessed : June 18, 2023].

Résumé

Résumé

Au fil du temps, les réseaux informatiques ont connu un développement de manière exponentielle, avec lequel les attaques réseaux et informatiques ont connu une augmentation à leur tour, ce qui représente un grave danger pour les personnes et les entreprises, par conséquent il faudrait assurée la sécurité des systèmes d'information. Plusieurs systèmes de détection d'intrusion (SDI) ont été proposés au cours des années et qui sont une solution efficace contre les attaques. Différentes approches, stratégies et techniques ont été réalisées afin de contrer ces attaques qui sont toujours en évolution, l'objectif de ces SDIs est d'atteindre une haute précision.

Dans ce mémoire de fin d' études, nous proposons une méthode de détection d'intrusion qui combine les transformateurs de vision avec les techniques de sélection des caractéristiques qui sont la variance et les forêts aléatoires. Ces algorithmes de selection aident à déterminer les attributs les plus importants dans les bases de données « NSL KDD » et « CIC 2017 » pour mieux repérer les intrusions sur le réseau, tout en équilibrant les classes existantes dans les deux bases de données via un sur échantillonnage. Exceptionnellement, nous avons procédé à une réduction de dimensionnalité de la base CIC 2017, pour gagner en temps d' exécution via un sous-échantillonnage aléatoire. Afin d' adapter nos bases a notre transformateur de vision, nous convertissons les données brutes en images, ce qui facilite la manipulation de ces données-là dans notre modèle.

Pour évaluer les performances de notre approche proposées nous avons utilisé trois paramètres, le taux de bonne détection, la précision et les fausses alarmes, les résultats obtenus sont très satisfaisants. Pour prouver l' efficacité de notre approche nous avons effectué une comparaison avec un travail de la littérature.

Mots clés : Système de détection d'intrusion (SDI), apprentissage profond, transformateurs de vision, sélection des caractéristiques, taux de variance, forêts aléatoires, réduction de dimensionnalité, échantillonnage aléatoire, NSL KDD, CIC 2017.

Abstract

Over time, computer networks have experienced exponential development. The increase in network and computer attacks, which has also experienced similar development, poses a serious danger to the security of information systems. Several intrusion detection systems (IDS) have been proposed over the years, using a lot of strategies and techniques to counter these attacks which are still evolving. The main objective of these intrusion detection systems is to achieve high accuracy.

In this end of studies dissertation, we propose a method of intrusion detection that combines vision transformers with feature selection techniques, namely variance and random forests. These algorithms help determine the most important attributes in the "NSL KDD" and "CIC 2017" databases for better identification of network intrusions, while balancing the existing classes in both databases through oversampling. Additionally, we performed dimensionality reduction on the CIC 2017 database using random undersampling to improve execution time. To adapt our databases to the vision transformer, we converted the raw data into images, which facilitated data manipulation in our model.

To evaluate the performance of our proposed approach, we used three parameters: detection rate, precision, and false alarms. The results obtained are very satisfactory. To demonstrate the effectiveness of our approach, we compared it with a previous work in the literature.

Keywords: Intrusion Detection System (IDS), deep learning, vision transformers, feature selection, variance rate, random forests, dimensionality reduction, random sampling, NSL KDD, CIC 2017.

ملخص

بمرور الوقت ، تطورت شبكات الكمبيوتر بشكل كبير ، مما أدى بدوره إلى زيادة هجمات الشبكات والكمبيوتر ، مما يمثل خطرًا خطيرًا على الأشخاص والشركات ، لذلك يجب ضمان أمن أنظمة المعلومات. تم اقتراح العديد من أنظمة كشف التسلل على مر السنين والتي تعد حلاً فعالاً ضد الهجمات. تم تنفيذ أساليب واستراتيجيات وتقنيات مختلفة لمواجهة هذه الهجمات التي لا تزال قيد التطوير ، والهدف من هذه هو تحقيق دقة عالية.

في هذه الرسالة ، نقترح طريقة للكشف عن التسلل تجمع بين محولات الرؤية وتقنيات اختيار الميزات التي تكون متباينة وغابات عشوائية. تساعد هذه الخوارزميات في تحديد أهم السمات في قاعدتي بيانات NSL" "KDD و CIC" "2017 لتحديد عمليات اقتحام الشبكة بشكل أفضل ، مع موازنة الفئات الموجودة في قاعدتي البيانات من خلال الإفراط في أخذ العينات. بشكل استثنائي ، قمنا بتقليل أبعاد قاعدة بيانات CIC 2017 ، لتوفير وقت التنفيذ عبر أخذ العينات الفرعية العشوائية. من أجل تكييف قواعدنا مع محول الرؤية الخاص بنا ، نقوم بتحويل البيانات الأولية إلى صور ، مما يسهل معالجة هذه البيانات في نموذجنا.

لتقييم أداء نهجنا المقترح استخدمنا ثلاث معاملات ، معدل الكشف الجيد ، الدقة والإنذارات الكاذبة ، النتائج التي تم الحصول عليها مرضية للغاية. لإثبات فعالية نهجنا قمنا بإجراء مقارنة مع عمل الأدب.

الكلمات الرئيسية: نظام كشف التسلل ، التعلم العميق ، محولات الرؤية ، اختيار الميزات ، معدل التباين ، الغابات العشوائية ، تقليل الأبعاد ، أخذ العينات العشوائية ، KDD NSL CIC 2017 .