REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE IBN KHALDOUN - TIARET

MEMOIRE

Présenté à :

FACULTÉ MATHEMATIQUES ET INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

MASTER

Spécialité : Réseaux et Télécommunications

Par :
Beneddine Mustapha Mohamed Amine
Abid Malika

Sur le thème

Conception d'un IDS basé sur le Deep Learning et RBN

Soutenu publiquement le 18 / 07 / 2021 à Tiaret devant le jury composé de :

Mr Mostefaoui Kadda	M.A.A Université Ibn-khaldoun	Président
Mr Mostefaoui sid Ahmed	M.C.B Université Ibn-khaldoun	Encadreur
Mr Meghazi lhadj madani	M.A.A Université Ibn-khaldoun	Examinateur

2020-2021

Dédicace

Je dédie ce modeste travail

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leurs prières tout au long de mes études.

A toute ma famille.

A tous ceux qui, de près ou de loin, ont participé à mon éducation, et m'ont aidé Dans les moments difficiles à surmonter mes problèmes.

A tous mes amis du plus proche au plus loin.

***Sans oublier** tous les professeurs qui ont contribué à ma formation de l'enseignement primaire, moyen et secondaire jusqu'à l'enseignement supérieur.*

Beneddine Mustapha

Dédicace

Je dédie ce modeste travail

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leurs prières tout au long de mes études.

A toute ma famille.

A tous ceux qui, de près ou de loin, ont participé à mon éducation, et m'ont aidé Dans les moments difficiles à surmonter mes problèmes.

A tous mes amis du plus proche au plus loin.

***Sans oublier** tous les professeurs qui ont contribué à ma formation de l'enseignement primaire, moyen et secondaire jusqu'à l'enseignement supérieur.*

Abid Malika

Remerciement

En premier lieu, nous remercions Dieu le tout puissant qui nous a donné le courage et la volonté de réaliser ce modeste travail.

Nous remercions nos parents qui nous ont suivis pendant nos études.

Nous remercions Mr. ***Mostefaoui sid Ahmed Mokhtar****, notre encadreur, pour son aide, son encouragement, son orientation, et pour ses précieux conseils durant la réalisation de ce travail.*

Nos remerciements aux membres de jury qui ont accepté de juger notre travail.

Nous adressons aussi nos remerciements à tous les professeurs qui nous ont enseignés durant ce cursus universitaire.

Enfin Nous tenons à saisir cette occasion et adresser nos profonds remerciements et nos profondes reconnaissances à toutes personnes qui nous ont aidés de près ou de loin dans la réalisation de ce mémoire.

Résumé

Les réseaux jouent un rôle important dans la vie moderne et le cyber sécurité est devenu domaine de recherche. Un système de détection d'intrusion (IDS) qui est une technique de cyber sécurité importante, surveille l'état des logiciels et du matériel fonctionnant sur le réseau. Malgré des décennies de développement, les IDS existants sont toujours confrontés à des défis pour améliorer la précision de détection et réduire le taux de fausses alarmes et détecter les attaques inconnues. Pour résoudre les problèmes ci-dessus, de nombreux chercheurs se sont concentrés sur le développement d'IDS qui capitalisent sur les méthodes d'apprentissage automatique. L'approche proposée a été testée sur la base publique NSL-KDD. Nous avons effectué la réduction de la dimensionnalité de la base DBN. Ensuite, nous avons effectué une classification (multi-classes) avec RB. Les résultats obtenus sont très satisfaisants car on réussit à obtenir un taux minimum des fausses alertes (faux positive) et un taux maximum de détection d'attaques, avec une réduction du temps d'apprentissage et de prédiction.

Mots Clés Sécurité des Réseaux, IDS, Deep Learning, Machine Learning, DBN, RB, NSL-KDD.

Sommaire

Liste des Figures

Liste des Tableaux

Liste des abréviations

DL: Deep Learning

ML: Machines Learning

CD : Contrastive Divergence.

CNN : Convolution Neural Network.

CPU: Central Processor Unit.

DBN: Deep Belief Network.

RB: Réseau Bayesians

RBM: Restraints Boltzmann Machines

DDoS: Distributed Deny Of Service.

DL: Deep Learning.

DNN: Deep Neural Network.

DoS: Deny Of Service.

HIDS: Host Intrusion Detection System.

IDMEF: Intrusion Detection Message Exchange Format.

IDS: Intrusion Detection System.

ML: Machine Learning.

MLP: Multi-Layer Perceptron's.

NIDS: Network Intrusion Detection System.

NLP: Natural Language Processing.

PC: Personal Computer.

IDWG: Intrusion Detection Working Group.

IETF: Internet Engineering Task Force

Introduction Général

La détection d'intrusions consiste à scruter le trafic réseau, collecter tous les événements, les analyser et générer des alarmes en cas d‘identification de tentatives malveillantes, Ces systèmes sont devenus jour après jour très utilisés dans les stratégies de sécurité des réseaux et systèmes informatiques. Le domaine de la détection d'intrusion est très ouvert à la recherche et au développement. En général, les IDS peuvent être divisés en deux techniques (approches) : la détection des abus (approche par scénario), et la détection des anomalies (approche comportemental).

La détection des abus (approche par scénario) : utilise des modèles bien définis basé sur des signatures d'attaques déjà connues et qui peuvent être fournies par l‘expert du domaine. C’est à dire, en cas de nouvelle attaque il ne déclenche pas une alerte.

La détection des anomalies (approche comportemental) : La détection d'anomalies suppose que l‘activité normale est différente de l’activité intrusive en élaborant un profil de comportement normale et un mécanisme permettant de comparer le comportement courant au profil établi pour détecter des écarts sensibles qui seront considérés comme des éventuelles intrusions.

Les techniques de détection d'anomalies (approche comportemental) ont l'avantage de détecter les attaques inconnues par rapport à la technique de détection d'abus (approche par scénario). Ce résultat et dériver d’une phase d’apprentissage sur une grande base de données.

Dans ce mémoire, nous essayons de proposer un modèle pour un IDS comportemental, capable de détecter les nouvelles attaques avec le maximum de taux de réussite et avec le minimum de fausses alertes,

Notre modèle utilise les Réseaux de croyances profondes qui s’appellent en anglais (Deep Belief Network DBN) basé sur la Machine Boltzmann restreinte (Restricted Boltzmann Machine RBM). DBN est utilisé comme un moyen de réduction de caractéristiques qui est suivie par le classifier RBN (réseau bayésien). Ensuite nous allons essayer d’examiner les performances en évaluant l'efficacité de notre modèle hybride DBN-RB par la comparaison avec le modèle proposé dans [Salam]. Tout ça après plusieurs expériences sur l'ensemble de données KDD99.

Le reste de Notre mémoire est organisé comme suit :

La première partie : Consacrée à la présentation des différents aspects de la sécurité informatique ainsi que les concepts relatifs aux réseaux d'une manière générale. Une description bien détaillée des systèmes de détection d'intrusions, leurs différents types, leurs principes de fonctionnement et leurs avantages et inconvénients.

La deuxième partie : Réservée à la présentation des différents types d'apprentissage automatique et spécialement l'apprentissage profond, leur principe, fonctionnement, apprentissage, test et évaluation.

La troisième partie : Consiste à détailler la méthode d'apprentissage élaborée, les modules et dépendances à prendre en compte dans l'application de cette approche pour la détection d'intrusions.

La dernière partie : dans cette partie nous allons détailler notre approche proposée DBN-RBN ainsi présenter les résultats obtenus et la comparaison de ces derniers avec le travail de **[40]**.

Chapitre 1 : Sécurité informatique et systèmes de détection d'intrusions

Introduction

Le progrès technologique, le développement des moyens de communications, l'ouverture du monde sur les nouvelles technologies, et la transmission de divers types de données à travers les réseaux, ainsi que d'autre facteurs, apportent un danger d'accès et de manipulation des données par des personnes non autorisés, ou des concurrents. Donc la sécurité de l'information par une gamme de techniques et mécanismes d'authentification et de contrôle d'accès est devenue un besoin crucial afin de construire un système sécurisé déterminant et éliminant ces vulnérabilités.

Le système de détection des intrusions (IDS) est l'une de ces techniques qui offre un contrôle permanent des attaques rn permettant de détecter toute tentative de violation de la politique de sécurité, c'est-à-dire toute intrusion.

Dans ce premier chapitre nous présentons deux parties, la première présente les principales notions de base de la sécurité informatique et les systèmes de détection d'intrusions, en commençant par les définitions des différentes notions de la sécurité informatique, puis les attaques informatiques et leurs classifications avec exemples. La deuxième partie présente les systèmes de détection d'intrusions, leur historique, définition, principe de fonctionnement, et critères de classification, ...etc.

I. La sécurité informatique

1. Définitions

La sécurité informatique : C'est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles. Elles caractérisent ce à quoi s'attendent les utilisateurs de systèmes informatiques en regard de la sécurité. La sécurité informatique vise généralement les cinq principaux objectifs suivants : (appelés aussi les propriétés de la sécurité informatiques) **[1].**

- **Disponibilité :** demande que l'information sur le système soit disponible aux personnes autorisées.
- **Confidentialité :** permet d'assurer que l'information sur le système ne puisse être lue que par les personnes autorisées.
- **Intégrité :** L'intégrité permet de certifier que l'information sur le système ne puisse être modifiée que par les personnes autorisées (pas de divulgation à des tiers non autorisés).
- **Non-répudiation :** pour éviter la contestation par l'émetteur de l'envoi de données, la non répudiation est une propriété qui assure que l'auteur d'un acte ne peut ensuite nier l'avoir effectué (signature de l'acte) et que le récepteur ne peut ultérieurement dénier avoir reçu un message (exemple exécution d'un ordre boursier, d'une commande...).
- **L'authentification :** l'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

On retrouve aussi dans le domaine de la sécurité informatique l'utilisation notamment des termes suivants, vulnérabilité, intrusion, menace, attaque**. [2][3]**

- **Vulnérabilité :** faute créé durant le développement du système, ou durant l'opération, pouvant être exploitée afin de créer une intrusion.
- **Intrusion :** faute malveillante externe résultant d'une attaque qui a réussi à exploiter une vulnérabilité.

- **Menace :** possibilités et probabilités d'attaque contre la sécurité. Une menace est définie par le processus d'attaque, par la cible et par le résultat (conséquences de la réussite d'une attaque).
- **Attaque :** C'est n'importe quelle action qui a le but de menacer la sécurité des informations et de nuire au moins à l'une des propriétés de la sécurité informatique (disponibilité, Confidentialité, Intégrité, L'authentification). Il s'agit d'une tentative d'intrusion, nous abordons dans ce qui suit les différents buts et classes de ces attaques (tentatives d'intrusion).

2. Buts des attaques informatiques

Il existe plusieurs objectifs pour les attaques :

- **Interruption :** vise la disponibilité des informations (Dos, . . .)
- **Interception :** vise la confidentialité des informations (capture de contenu, analyse de trafic,. . .).
- **Modification :** vise l'intégrité des informations (modification, rejet,. . .).
- **Fabrication :** vise l'authenticité des Informations (Masque rade). **[4]**

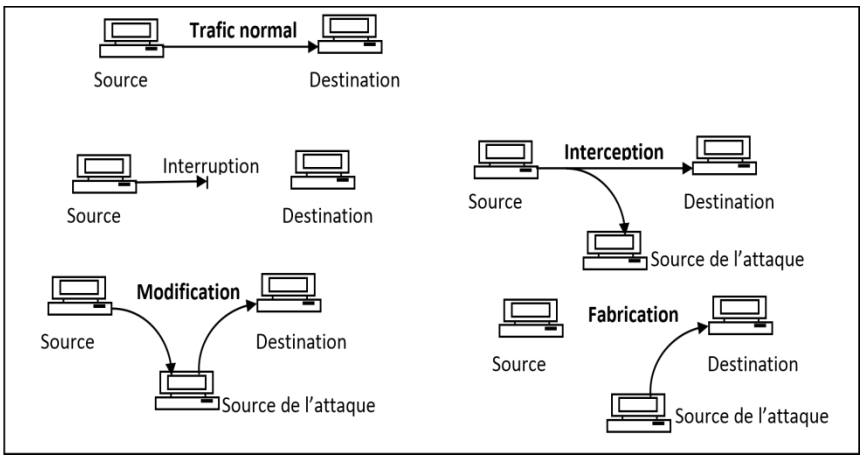


Figure 1.1 : Buts des attaques Informatiques [5]

3. Les différentes classes d'attaques informatiques

Du point de vue de la sécurité informatique, une menace est une violation potentielle de la sécurité. Cette menace peut être accidentelle, intentionnelle (attaque), active ou passive, Ils

existent dans la littérature plusieurs classifications d'attaques informatiques selon des critères différents, parmi lesquelles :

3.1. Classification selon l'effet de l'attaque

Selon les effets résultant de l'attaque on peut classifier les attaques en deux groupes principaux, les attaques passives et les attaques actives.

- **Les attaques passives :** consistent à accéder, utiliser ou à observer le système cible sans modifier les données ou dysfonctionner les ressources de ce dernier, elles sont généralement indétectables (ex. : capture de contenu, analyse de trafic).
- **Les attaques actives :** consistent à effectuer des changements non autorisés sur les données des systèmes, à s'introduire dans des équipements réseau ou à perturber leurs fonctionnements, les attaques de ce type sont bien évidemment plus dangereuses.(ex. : mascarade et déni de service).

3.2. Classification selon la source de l'attaque

En termes de relation intrusion-victime, les attaques sont classées comme suit :

- **Les attaques internes :** provenant des employés de leur entreprise ou de leurs partenaires commerciaux ou clients,
- **Les attaques externes :** venant de l'extérieur, fréquemment via Internet.

3.3. Classification selon la cible de l'attaque

- **Les attaques réseaux :** Les attaques réseaux s'appuient sur des vulnérabilités liées directement aux protocoles ou à leur implémentation.
- **Les attaques applicatives :** Les attaques applicatives s'appuient principalement sur des vulnérabilités spécifiques aux applications utilisées.

4. Exemples d'attaques

Il existe un nombre énorme d'attaques qui menacent les systèmes et les réseaux informatiques, néanmoins, la plupart d'entre elles ne sont que des variantes des autres. Voici des exemples d'attaques les plus connues aujourd'hui ciblant les réseaux informatiques.

4.1. Attaques de Dénis de Services (Denial Of Service [DOS])

Est une attaque informatique ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service offert. Il peut s'agir de :

- L'inondation d'un réseau afin d'empêcher son fonctionnement.

- La perturbation des connexions entre deux machines, empêchant l'accès à un service particulier.
- L'obstruction d'accès à un service à une personne en particulier.
- Également le fait d'envoyer des milliards d'octets à un box internet.

L'attaque par déni de service peut ainsi bloquer un serveur de fichiers, rendre impossible l'accès à un serveur web ou empêcher la distribution de courriel dans une entreprise. Les principales attaques qu'on peut trouver sont Apache2, Back, Land, Mailbomb, SYN Flood, Ping of death, Process table, Smurf, Syslogd, Teardrop, Udp storm.

4.2. **Probing (Sondage)**

L'attaquant de cette classe commence par un sondage de la future victime, ce que l'on appelle scan, ce sondage va balayer chaque port IP afin de connaitre les services offerts par le système (topologie du réseau, protections employées,..) une fois terminé, la machine de l'intrus (celui qui réalise l'intrusion) tente alors d'identifier le système d'exploitation utilisé par cette victime et d'exploiter les informations qu'elle a récolté. Cette classe d'attaque est la plus étendue et qu'elle requit une expertise technique minime. Les exemples de ce type d'attaque sont : Ipsweep, Mscan, Nmap,….

4.3. **Attaques User to Root**

L'objectif de cette classe d'attaques est d'obtenir la main de l'administrateur système (Root) à partir d'un simple compte utilisateur par l'exploitation des vulnérabilités, Les exploits les plus connus sont les débordements réguliers des Buffers (buffer overflows) dus aux erreurs de programmation, Les principales attaques de ce type sont : Eject, Ffbconfig, Fdformat, Load module, Perl, Ps, Xterm.

4.4. **Attaque Remote to User**

Dans cette classe d'attaque, l'attaquant essaye d'exploiter les vulnérabilités d'une machine distante afin d'avoir un accès illégal à cette dernière. Pour réussir cette attaque, l'attaquant exploite les bugs des applications installées dans la machine cible, les mauvaises configurations de celles-ci et du système qui les héberge, etc.

4.5. **L'usurpation d'adresse IP (IP Spoofing)**

Le principe de fonctionnement de cette attaque est d'envoyer des paquets IP en utilisant une IP source qui n'a pas été allouée à l'ordinateur qui émet ces paquets pour le but de masquer l'identité de l'attaquant lors d'une attaque d'un serveur ou n'importe quel cible dans le réseau, ou d'usurper l'identité d'un autre équipement du réseau pour bénéficier des services auxquels il a accès.

4.6. **Les analyseurs réseau (sniffer)**

Est un dispositif permettant d'écouter le trafic d'un réseau, c'est à-dire de capturer les informations qui y circulent, vu que les données dans un réseau non commuté sont envoyées à toutes les machines du réseau et dans une utilisation normale les machines ignorent les paquets qui ne leur sont pas destinés. Le sniffer peut également servir cette propriété à une personne malveillante ayant un accès physique au réseau pour collecter des informations (ex : les mots de passes), mais un sniffer peut aussi être utilisé comme un outil positif pour le but d'étudier et de capturer le trafic d'un réseau par les administrateurs réseaux et les détecteurs d'intrusion (IDS).

4.7. **Balayage des ports (port scanning)**

Est une des activités considérées comme suspectes servant par les pirates informatiques pour découvrir les faiblesses potentiellement exploitables et chercher les ports ouverts sur un serveur de réseau en balayant les ports disponibles de la victime qui potentiellement exécute de nombreux services qui écoutent des ports connus. les balayages de ports se font habituellement sur le protocole TCP pour le but d'ouvrir des connexions pour effectuer une intrusion, la même technique de balayage des ports est aussi utilisée par les administrateurs des systèmes informatiques pour contrôler la sécurité des serveurs de leurs réseaux.

4.8. **TCP Session Hijacking**

Le « **vol de session TCP** » est une technique consistant à intercepter une session TCP initiée entre deux machines afin de la détourner, dans la mesure où le contrôle d'authentification s'effectue uniquement à l'ouverture de la session, un pirate réussissant cette attaque parvient à prendre possession de la connexion pendant toute la durée de la session.

4.9. **Les trappes (backdoor)**

C'est une fonction ou un programme permettant à un pirate de prendre le contrôle d'un ordinateur à distance. Il peut être placé dans un cheval de Troie[1] ou un virus.

4.10. **Attaque par virus**

Il s'agit d'un programme autoreproductible et généralement destructeur qui contamine le disque dur ainsi que tous autres supports de stockage utilisés et qui peut faire exécuter à l'ordinateur des actions non désirées. Le virus informatique peut donc se propager à l'intérieur même de l'ordinateur, en infectant petit à petit tous les fichiers. Il est donc destiné à modifier à notre insu le fonctionnement de l'ordinateur, certains virus peuvent simplement faire «beeper» le PC, d'autres peuvent détruire les données (formater, effacer le secteur de démarrage, voir détruire le matériel). **[6]**

5. Mécanismes de défense contre les attaques réseaux

C'est l'ensemble de procédures ou dispositifs qui sont conçu pour détecter, prévenir ou contrer les attaques qui menacent la sécurité informatique, il existe plusieurs outils de prévention contre ces attaques réseaux, Nous allons citer ci-dessous quelques mécanismes.

- **Chiffrement :** Algorithmes généralement basés sur des clefs en transformant les données. son efficacité est dépendante du niveau de sécurité des clefs.
- **Signature numérique :** Données ajoutées pour vérifier l'intégrité ou l'origine des données.
- **Bourrage de trafic :** Données ajoutées pour assurer la confidentialité, notamment au niveau du volume du trafic.
- **Notarisation :** Utilisation d'un tiers de confiance pour assurer certains services de sécurité.
- **Contrôle d'accès :** Vérifie les droits d'accès d'un acteur aux données. N'empêche pas l'exploitation d'une vulnérabilité.
- **Antivirus :** Logiciel censé à protéger l'ordinateur contre les logiciels (ou fichiers potentiellement exécutables) néfastes. Ne protège pas contre un intrus qui emploie

[1]**Chevaux de Troie** : sont des programmes qui en plus d'une fonction classique, ont une fonction cachée nuisible, récupérer vos mots de passe, détruire le disque dur, etc.

un logiciel légitime, ou contre un utilisateur légitime qui accède à une ressource alors qu'il n'est pas autorisé à le faire.

- **Le pare-feu :** Un élément (logiciel ou matériel) du réseau informatique contrôlant les communications qui le travers. Il a pour fonction de faire respecter la politique de sécurité du réseau, celle-ci définissant quelles sont les communications autorisées ou interdites. Le pare-feu n'empêche pas un attaquant d'utiliser une connexion autorisée pour attaquer le système, ainsi ne protège pas contre une attaque venant du réseau intérieur (qui ne le traverse pas).
- **Système de détection d'intrusion :** Repère les activités anormales ou suspectes sur le réseau surveillé. Ne détecte pas les accès incorrects autorisés par un utilisateur légitime. Son inconvénient est la mauvaise détection : taux de faux positifs, faux négatifs.
- **Journalisation ("logs") :** Enregistrement des activités de chaque acteur. Permet de constater que des attaques ont eu lieu (audit), de les analyser et potentiellement de faire en sorte qu'elles ne se reproduisent pas.
- **Analyse des vulnérabilités ("Security audit") :** Identification des points de vulnérabilité du système. Ne détecte pas les attaques ayant déjà eu lieu, ou lorsqu'elles auront lieu. **[6]**

Mais Aucun des mécanismes de sécurité ne suffit par lui-même, et pour cela dans la plupart du temps en vue d'atteindre un niveau acceptable de sécurité informatique plusieurs mécanismes sont utilisés en même temps.

II. Systèmes de détection et de prévention des intrusions

1. Historique

Les systèmes de détection ont connu une croissance rapide, le concept de système de détection d'intrusions a été introduit en 1980 par **James Anderson** dans l'effort d'amélioration de la vérification de la sécurité informatique et la capacité de surveillance, puis complété par le premier modèle de détection d'intrusions établit par **Denning Dorothy** en 1987, ensuite de nombreux prototypes sont apparus depuis 1988, et des grands budgets sont investis pour la recherche dans ce domaine. **[7]**

2. Systèmes de détection intrusion (IDS)

La détection d'intrusion concerne l'ensemble des pratiques et mécanismes utilisés pour la détection d'erreurs pouvant conduire à une défaillance de sécurité, et/ou pour la détection d'attaques. Un IDS est l'implémentation des pratiques et mécanismes de détection d'intrusion. Les IDS ont donc pour rôle de détecter et/ou bloquer (on parle dans ce cas d'**I**ntrusion **P**révention **S**ystem) des attaques surmenantes au sein d'un système ou d'un réseau. Ils peuvent être déployés sur l'hôte surveillé, on parle alors de **H**ost Based **I**ntrusion **D**etection **S**ystem (**HIDS**), ou bien sur le réseau surveillé, on parle alors de **N**etwork Based **I**ntrusion **D**etection **S**ystem (**NIDS**) **[2] [8]**.

3. Architecture des IDS et Principes de fonctionnement

3.1. Architecture des IDS

Plusieurs architectures ont été proposées pour décrire les différents éléments intervenants dans un système de détection d'intrusion. L'architecture la plus simple est composée de trois modules :

- la source de données,
- l'analyseur des données
- le module des réponses.

L'architecture générale d'un IDS proposée par **IDWG** de l'IETF[2] **(I**ntrusion **D**étection **W**orking **G**roup) est montrée dans la (**Figure 1.2**).

Dans cette architecture on trouve les trois modules cités précédemment couplés avec d'autres composants, l'objectif été la définition d'un standard de communication entre les composants du système de détection d'intrusion. Cette architecture définit un format d'échange de message pour les IDS : **I**ntrusion **D**étection **M**essage **E**xchange **F**ormat (**IDMEF**), qui contient implicitement un modèle de données.

[2]L'**I**nternet **E**ngineering **T**ask **f**orce, (**IETF)** : est un groupe informel, international, ouvert à tout individu, qui participe à l'élaboration des standards L'IETF produit la plupart des nouveaux standards d'Internet.

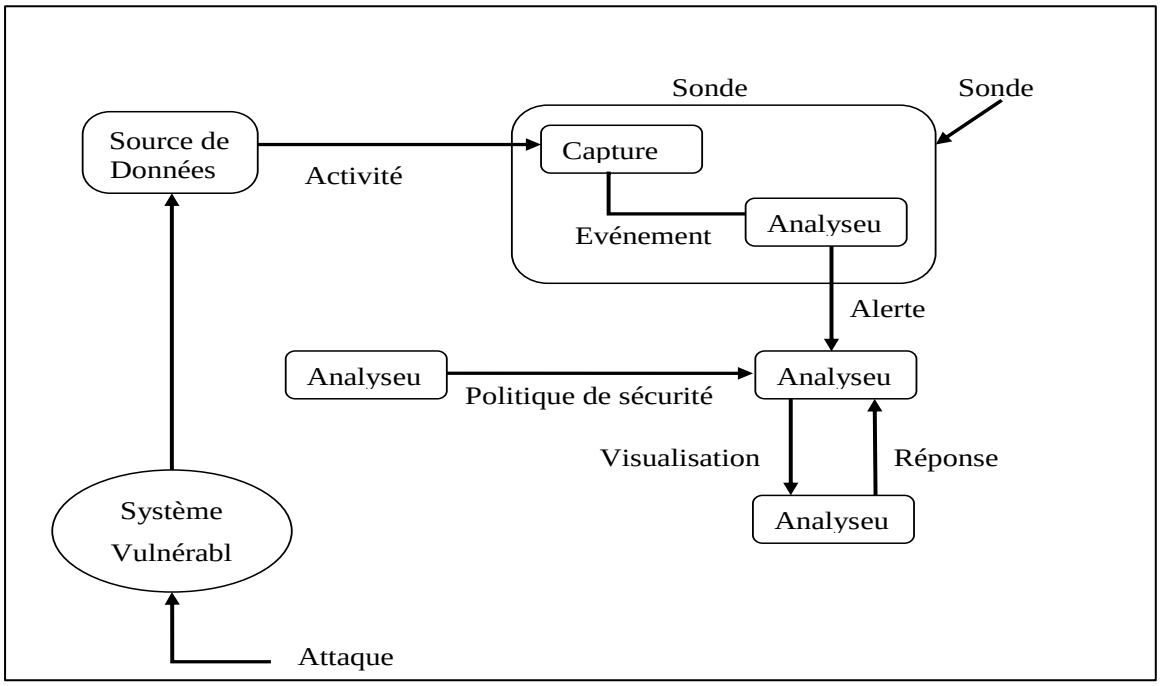


Figure 1.2 : Architecture d'un IDS proposée par IDWG **[9]**

Cette architecture est composée des modules suivants :

• **Source de données :** C'est l'interface entre le système surveillé et l'IDS, elle fait la collecte d'informations sur les activités du système.

• **Capteur :** Chargé de filtrer et formater les informations brutes envoyées par la source de données. Le résultat de ce traitement sera un message formaté, appelé aussi événement, il représente l'unité de base dans un scenario d'attaque.

• **Analyseur :** Permet d'analyser les événements générés par le capteur. S'il détecte une activité intrusive il émet une alerte, qui est un message sous un format standard. Dans cette architecture, le capteur et l'analyseur forment ensemble une sonde.

• **Alertes :** Lorsqu'un IDS détecte une intrusion, il doit la signaler à l'administrateur à travers les alertes, Ces dernières générées par les IDS sont généralement stockées dans les journaux du système ou utilisés pour prendre des actions contre les attaques (cela dépend du type d'IDS : à réaction active ou passive). Cependant il existe une norme qui permet d'en formaliser le contenu, afin de permettre à différents éléments de sécurité d'inter-opérer. Ce format s'appelle IDMEF[3] (**I**ntrusion **D**étection **M**essage **E**xchange **F**ormat), où il est possible de les visualiser ultérieurement par un expert de sécurité.

[3]**IDMEF** (**I**ntrusion **D**étection **M**essage **E**xchange **F**ormat) Utilisé dans le cadre de la sécurité informatique, est un format de données servant à échanger des informations de sécurité collectées par

• **Manager :** En plus de la notification des alertes, il offre à l'administrateur la possibilité de configurer une sonde et de gérer les alertes envoyées par l'analyseur.

3.2. Principe de fonctionnement des IDS

La Figure 1.3 illustre le fonctionnement d'un IDS et l'enchainement de ses actions lors de la détection des intrusions**.**

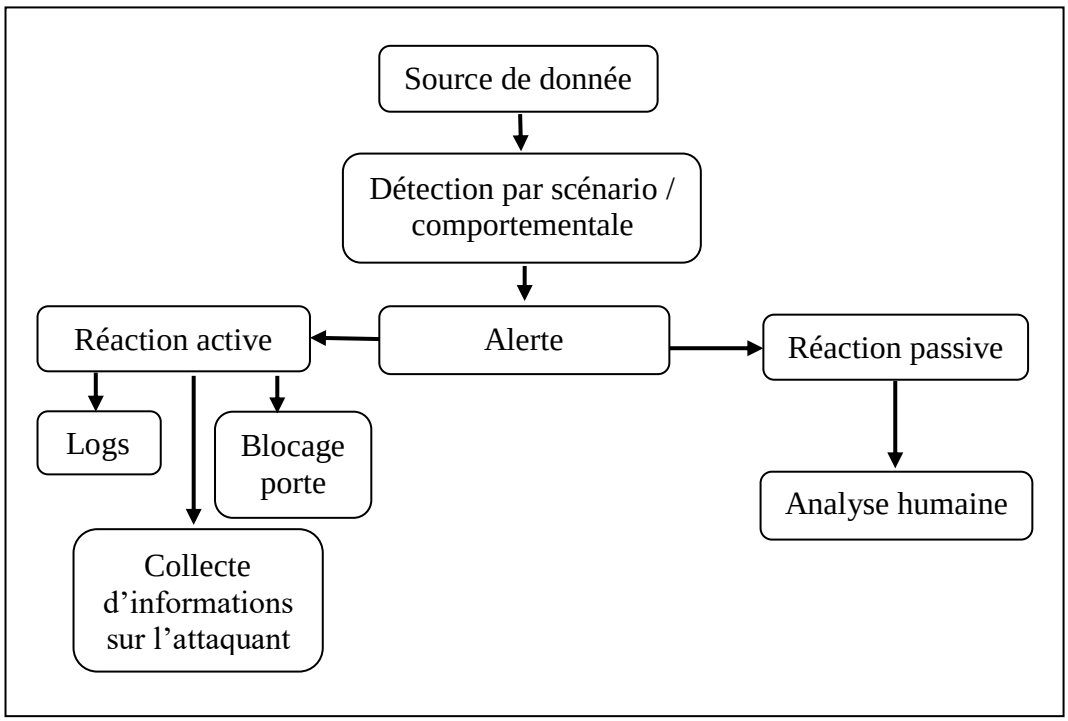


Figure1.3 : Fonctionnement d'un IDS **[10]**

4. Emplacement de l'IDS

Il est très important de faire bien positionner le système de détection d'intrusion, cela nécessite de bien identifier les ressources à protéger et ce qui est le plus susceptible d'être attaqué, Il convient alors d'implémenter précautionneusement dans la zone convenable. Il existe plusieurs endroits stratégiques où il convient de placer un IDS.

les logiciels de détection d'intrusion et de prévention d'intrusion avec les systèmes de management qui communiquent avec eux.

La **Figure 1.4** illustre un réseau local ainsi que les trois positions que peut y prendre un IDS :

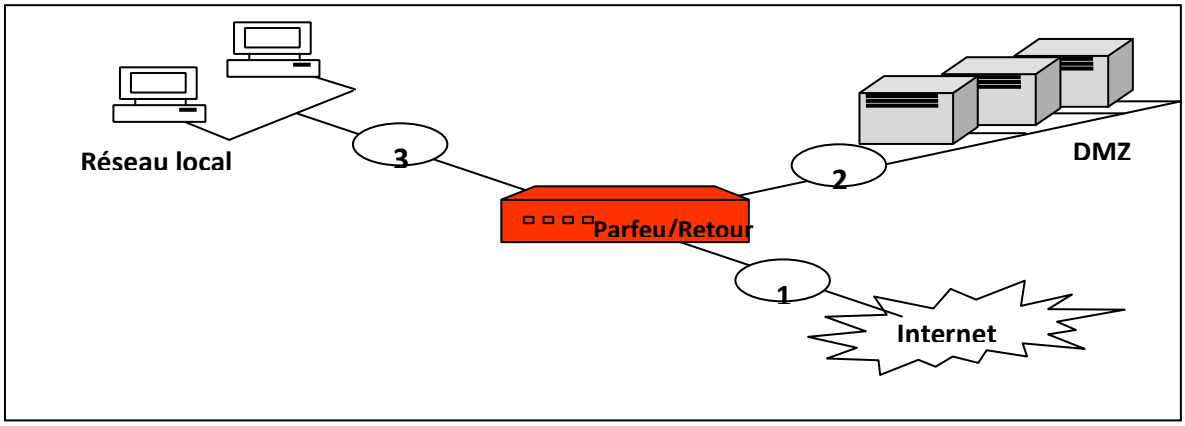


Figure1.4 : Emplacements des IDS

- **Position (1) :** L'IDS Sur cette position sert à détecter l'ensemble des attaques frontales, provenant de l'extérieur, vers le firewall. Dans ce cas beaucoup d'alertes seront remontées ce qui rendra les logs difficilement consultables.
- **Position (2) :** L'IDS placé sur la DMZ[4], utilisé pour détecter les attaques qui n'ont pas été filtrées par le firewall et qui relèvent d'un certain niveau de compétence. Les logs seront ici plus clairs à consulter puisque les attaques bénins ne seront pas recensées.
- **Position (3) :** L'IDS dans cette position a pour objectif de rendre compte des attaques internes, provenant du réseau local de l'entreprise. Il peut être judicieux d'en placer un à cet endroit étant donné le fait que 80% des attaques proviennent de l'intérieur.

5. Classification des IDS

Il existe plusieurs classifications des systèmes de détection des intrusions, nous avons opté pour le modèle apparu dans la **Figure 1.5** :

[4]**DMZ** est un terme informatique désigne (zone démilitarisée), est un sous-réseau séparé du réseau local et isolé de celui-ci et d'Internet (ou d'un autre réseau) par un pare-feu. Ce sous-réseau contient les machines étant susceptibles d'être accédées depuis Internet comme les serveurs.

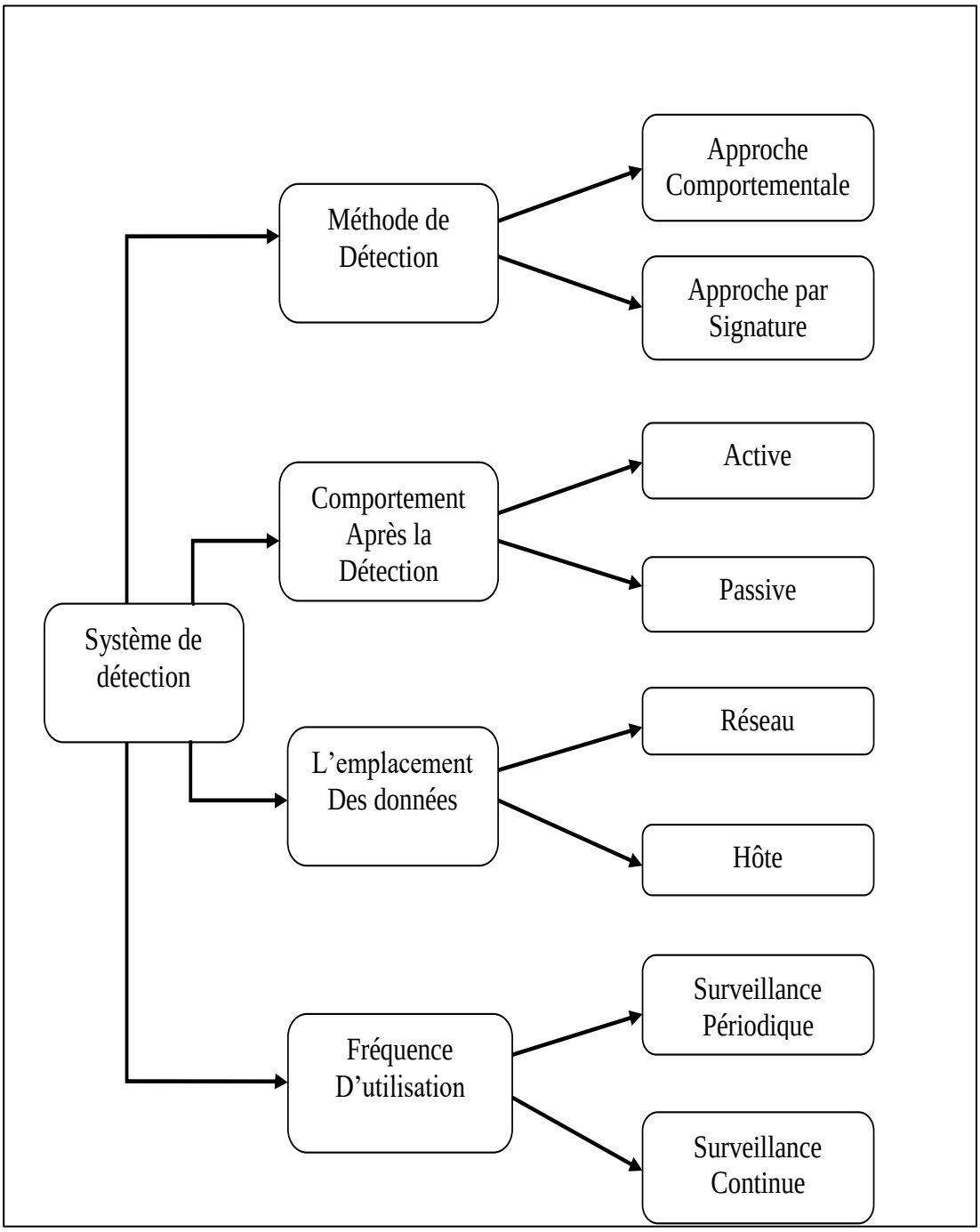


Figure1.5 : Classification d'IDS **[11]**

5.1. Méthodes de détection :

Comme le montre la Figure 1.6, il existe deux techniques principales de détection : par signatures ou par comportements.

L'approche par signatures consiste à détecter des attaques en vérifiant si les observations correspondent à des attaques connues, tandis que l'approche par comportements (ou par détection d'anomalies) consiste à détecter une attaque en vérifiant que les observations ne correspondent pas à des comportements légitimes de référence. Certains IDS combinent les deux approches afin d'obtenir de meilleurs résultats.

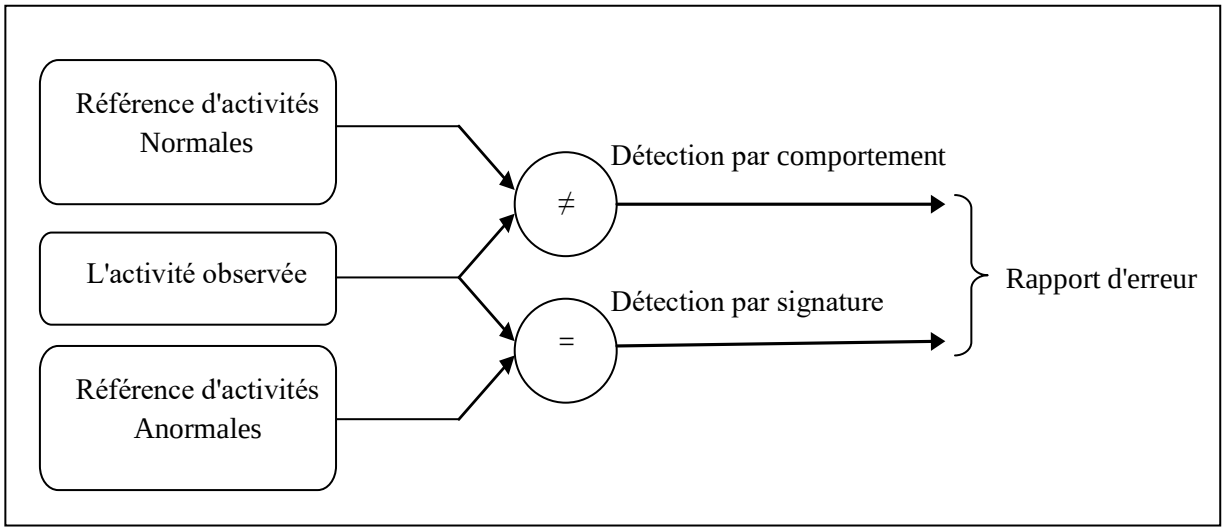


Figure 1.6 : Techniques de détection d'intrusions **[8]**

5.1.1. Approche comportementale il défaut débuter par l'approche à signature

Les modèles comportementaux sont apparus bien plus tard que les IDS à signatures. Ils ont pour principe la détection d'anomalies. Leur mise en œuvre comprend toujours une phase d'apprentissage au cours de laquelle ils vont " découvrir " le fonctionnement " normal " des éléments surveillés. Une fois cet apprentissage effectué ces IDS signaleront les divergences par rapport au fonctionnement de référence. Les modèles comportementaux peuvent être élaborés à partir d'analyses statistiques ou de techniques proches de l'intelligence artificielle. La principale caractéristique des IDS comportementaux est la détection des nouveaux types d'attaque, en effet ces IDS ne sont pas programmés pour reconnaître des attaques spécifiques mais signalent toute activité " **anormale** ". De ce fait une attaque ne doit pas nécessairement être connue d'avance ; dès lors qu'elle représente une activité anormale elle peut être détectée par l'IDS comportemental. Du fait même de leur conception ces IDS sont incapables de qualifier

la criticité des attaques. De plus, ces IDS signaleront par exemple tout changement dans le comportement d'un utilisateur qu'il soit hostile ou non. De fréquents ajustements sont nécessaires afin de faire évoluer le modèle de référence de sorte qu'il reflète l'activité normale des utilisateurs et réduire le nombre de fausses alertes générées.

5.1.2. Approche par signature

Le concept de bibliothèque de signatures d'attaque est l'approche la plus basique et la plus ancienne. Cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Cette démarche appliquée à la détection d'intrusion, est très similaire à celle des outils antivirus et présente les mêmes inconvénients que celle-ci. Il est aisé de comprendre que ce type d'IDS est purement réactif ; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite des mises à jour quotidiennes. De plus, ce système de détection est aussi bon qu'il est la base de signature. Si les signatures sont erronées ou incorrectement conçues, l'ensemble du système est inefficace. C'est pourquoi ces systèmes sont souvent contournés par les pirates qui utilisent des techniques dites " d'évasion " qui consistent à maquiller les attaques utilisées. Ces techniques de maquillage tendent à faire varier les signatures des attaques qui ainsi ne sont plus reconnues par l'IDS. Ce modèle est par contre très aisé à implémenter et à optimiser. Il permet la séparation du moteur logiciel de la base de signature qui peut ainsi être mise à jour indépendamment. Il permet également une classification relativement facile de la criticité des attaques signalées.

5.2. Comportement après la détection

Ils existent deux types d'IDS, actifs et passifs :

5.2.1. IDS à réponse passive

La réponse passive d'un IDS consiste à enregistrer les intrusions détectées dans un fichier de log qui sera analysé par le responsable de sécurité ou générer des alarmes, envoi d'un E-mail à un ou plusieurs utilisateurs, etc. Ceci permet de remédier aux failles de sécurité pour empêcher les attaques enregistrées de se reproduire, mais elle n'empêche pas directement une attaque de se produire.

5.2.2. IDS à réponse active

La réponse active au contraire a pour but de stopper une attaque au moment de sa détection sans attendre l'intervention humaine. Pour cela on dispose de deux techniques : la reconfiguration du firewall et l'interruption de la session TCP courante. La reconfiguration du firewall permet de bloquer le trafic malveillant au niveau du firewall, en fermant le port utilisé ou en interdisant l'adresse de l'attaquant. Cette fonctionnalité dépend du modèle de firewall utilisé, tous les modèles ne permettant pas la reconfiguration par un IDS. De plus, cette reconfiguration ne peut se faire qu'en fonction des capacités du firewall.

L'IDS peut également interrompre une session établie entre un attaquant et sa machine cible, de façon à empêcher le transfert de données ou la modification du système attaqué **[12].**

5.3. Emplacement des données

Il existe des IDS qui surveillent l'état de la sécurité au niveau du réseau par la capture et l'analyse des paquets qui circulent à travers le réseaux (NIDS : Network Intrusion Détection System) , et d'autres surveillent l'état de la sécurité au niveau des hôtes et analysent les informations produites par le système d'exploitation ou par des applications installées dans les machines locales (HIDS :Host Intrusion Détection System), quelques IDS hybrides utilisent les NIDS et HIDS pour avoir des alertes plus pertinentes. **[13].**

6. Systèmes de détection d'intrusion réseaux (NIDS)

L'IDS réseau ou (NIDS : Network Intrusion Détection System) surveille le trafic réseau. Il se place sur un segment réseau et écoute le trafic. Ce trafic sera ensuite analysé afin de détecter les signatures d'attaques ou les différences avec le fonctionnement de référence.

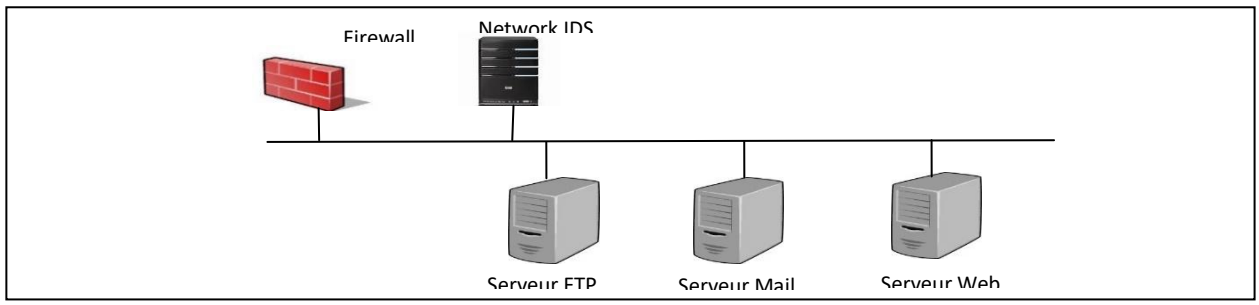


Figure 1.7 : Exemple d'un IDS dans un réseau (NIDS) **[13]**

6.1. Les avantages des NIDS

- Le NIDS peut surveiller un grand réseau (un grand nombre d'hôte).
- Le déploiement de NIDS a peu d'impact sur un réseau existant. Les NIDS sont habituellement des dispositifs passifs qui écoutent sur un fil de réseau sans interférer l'opération normale de ce dernier. Ainsi, il est habituellement facile de monter en rattrapage un réseau pour inclure IDS avec l'effort minimal.
- NIDS peut être très sûr contre l'attaque et être même caché a beaucoup d'attaquants.

6.2. Les inconvénients des NIDS

- Il est difficile à traiter tous les paquets circulant sur un grand réseau. De plus il ne peut pas reconnaître des attaques pendant le temps de haut trafic.
- Plusieurs avantages de NIDS ne peuvent pas être appliqués pour les commutateurs modernes. La plupart des commutateurs ne fournissent pas des surveillances universelles des ports et limitent la gamme de surveillance de NIDS. Même lorsque les commutateurs fournissent de tels ports de surveillance, souvent le port simple ne peut pas refléter tout le trafic traversant le commutateur.
- NIDS ne peuvent pas analyser des informations chiffrées (dans le cas d'utilisation des VPN). La plupart de NIDS ne peuvent pas indiquer si une attaque est réussie ou non. Il reconnaît seulement qu'une attaque est initialisée. C'est-à-dire qu'après le NIDS détecte une attaque, l'administrateur doit examiner manuellement chaque host s'il a été en effet pénétré.
- Quelques NIDS provoquent des paquets en fragments. Ces paquets mal formés font devenir l'IDS instable.

7. Systèmes de détection d'intrusion sur hôte (HIDS)

- HIDS : Host Intrusion Détection System ou L'IDS Système : accomplir le travail de surveillance du trafic sur une machine locale par l'analyse des journaux, les appels systèmes, analyse de la base de registre et des logs en provenance de firewalls hétérogènes et vérifie l'intégrité des systèmes de fichiers. Le principe de fonctionnement des HIDS dépend du système sur lequel ils sont installés. L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levées par nature.

- HIDS peut aussi observer les paquets réseaux de l'hôte (de la machine locale) pour la découverte des signaux d'intrusions (Déni de Services, Backdoor, chevaux de Troie, etc.).

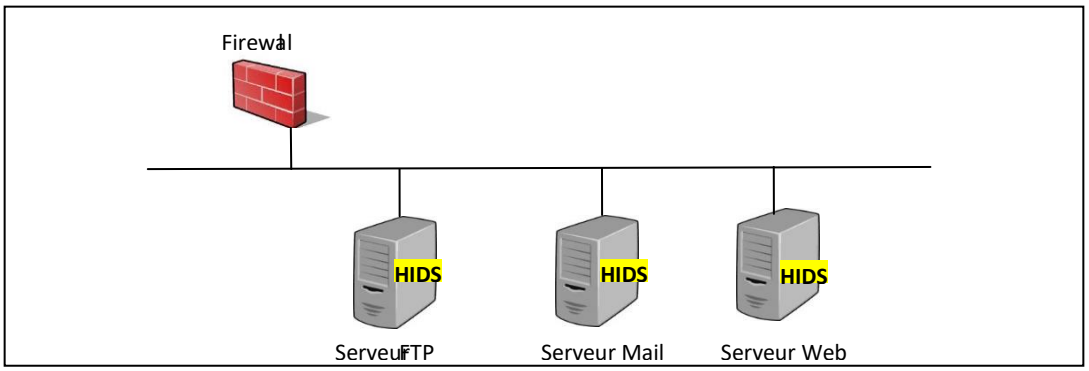


Figure 1.8 : Exemple d'un HIDS (L'IDS –Niveau Système) **[14]**

7.1. Les avantages des HIDS

- Pouvoir surveiller des événements locaux jusqu'au host, détecter des attaques qui ne sont pas vues par NIDS.
- Analyse des flux cryptés (ce que ne peut réaliser un NIDS).
- lorsque les sources des informations de host-based sont générées avant l'encrypte des données ou après le décrypte des données au host de la destination.
- Les HIDS peuvent détecter le cheval de Troie ou les autres attaques relatives à la brèche intégrité de logiciel.

7.2. Les inconvénients des HIDS

- HIDS est difficile à gérer, et des informations doivent être configurées et gérées pour chaque host surveillé.
- Puisque au moins des sources de l'information pour HIDS se résident sur l'host de la destination par les attaques, l'IDS peut être attaqué et neutralisé comme une partie de l'attaque.
- HIDS n'est pas bon pour la surveillance qui s'adresse au réseau entier parce que le HIDS ne voit que les paquets du réseau reçus par ses hosts.
- HIDS peut être neutralisé par certaine attaque de Dos.

8. Les avantages d'utilisation des IDS

- **Déjouer les attaques attendues sur le réseau :** Les IDS protègent les systèmes contre les attaques réseaux par : détection de porte dérobée, détection d'usurpation

d'adresse IP, Dos, les vers, les chevaux de Troie, virus, Botnet, rootkit, Spyware, et autres menaces qui pourraient nuire au réseau, Les IDS actifs prennent des mesures automatiques contre les menaces de sécurité et les risques auxquels font face,

- **Avertis Administrateur réseau d'alerte pour les événements de sécurité potentiels :** la fonction de base des systèmes de détection d'intrusions est de générer des avertissements là où existent des menaces externes, internes ou de violations de la politique de sécurité réseau, et aussi de fournir à l'administrateur des informations détaillées sur le mouvement des données au sein du réseau.
- **Gagnez du temps :** L'utilisation des IDS fournit beaucoup de temps et d'effort pour connaitre de ce qui se passe dans le réseau, peut aussi tourner en permanence sans superviseur humain.
- **Contrôle des Programmes utilisés par les employés pour surveiller l'Internet :** IDS peut aider à découvrir les programmes qui traitent de l'internet, cela permet de mieux contrôler et de protéger le réseau.
- **Avoir la confiance des clients :** Les IDS aident les organisations de protéger les données de ses clients contre le vol et la violation de la sécurité, Cela permet d'avoir la confiance des clients et partenaires et garder une bonne réputation sur l'organisation.
- **Économisez de l'argent :** Grace aux IDS les organisations peuvent déterminer les mouvements suspects dans le réseau et signaler les responsables pour prendre des mesures proactives en protéger le réseau et gagner l'argent qui sera dépensé si la violation de la sécurité est arrivé dans le réseau ou si le vol de renseignements personnels a eu lieu.

Conclusion

Dans ce chapitre nous avons expliqué différentes notions de la sécurité informatique, on a expliqué les attaques informatiques, leurs classifications et les mécanismes utilisés dans la prévention contre ces attaques, parmi ces mécanisme on a détaillé les systèmes de détection des intrusions vu que c'est notre objectif dans ce mémoire, qui jouent un rôle complémentaire aux mécanismes de sécurité traditionnels.

Chapitre 2 : Apprentissage automatique

Introduction

Selon Arthur Samuel Machine Learning (ou apprentissage automatique) est un domaine d'études qui permet aux ordinateurs d'apprendre sans être explicitement programmés. Selon Tom Mitchell Un programme informatique est dit apprendre de l'expérience E en ce qui concerne une certaine tâche T et une certaine mesure de rendement P, si son rendement sur T, tel que mesuré par P, s'améliore avec l'expérience E. Il existe plusieurs types d'apprentissages automatiques [w1], Permis lesquels on peut citer :

- Supervisé
- Non supervisé

Dans ce chapitre, nous allons parler des différents types d'apprentissage automatique et par la suite nous essayons de détailler leur principe de façon très brève, enfin nous passerons à mettre en évidence **l'apprentissage profond (**Deep Learning **ou** DL**)** et les réseaux bayésiens naïves.

I. L'apprentissage automatique (Machine Learning)

1. Définition

L'apprentissage automatique est un sous-domaine de l'intelligence artificielle (IA) qui se concentre sur la conception de systèmes qui apprennent – ou améliorent le rendement – en fonction des données qu'ils consomment. L'intelligence artificielle et l'apprentissage automatique sont souvent évoqués ensemble. **[14]**

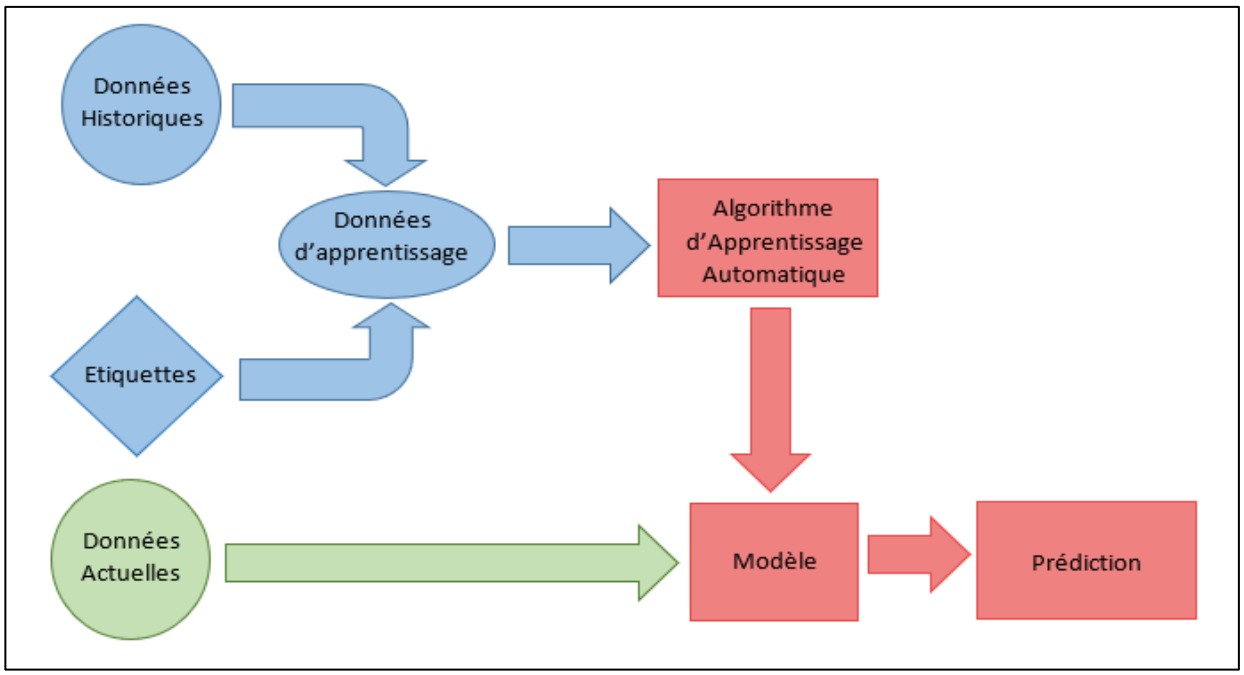


Figure.2.1 : Méthodes permettant d'apprendre et de prédire des données. **[w4]**

2. Types d'apprentissage

- **Supervisé :** avec un corpus d'apprentissage étiqueté. Il existe deux types de modèles d'apprentissages supervisés : les modèles de régressions et les modèles de classifications.
- **Non Supervisé :** apprendre des modèles dans des données sans étiquettes. Il doit extraire automatiquement les catégories à associer aux données qu'on lui soumet. Reformuler ce paragraphe.
- **Apprentissage par renforcement :** Dans ce cas, bien que les sorties idéales ne soient pas connues directement, il y a un moyen quelconque de connaître si les sorties du RNA s'approchent ou s'éloignent du but visé. Ainsi, les poids sont ajustés

de façons plus ou moins aléatoire et la modification est conservée si l'impact est positif ou rejetée sinon. **[16]**

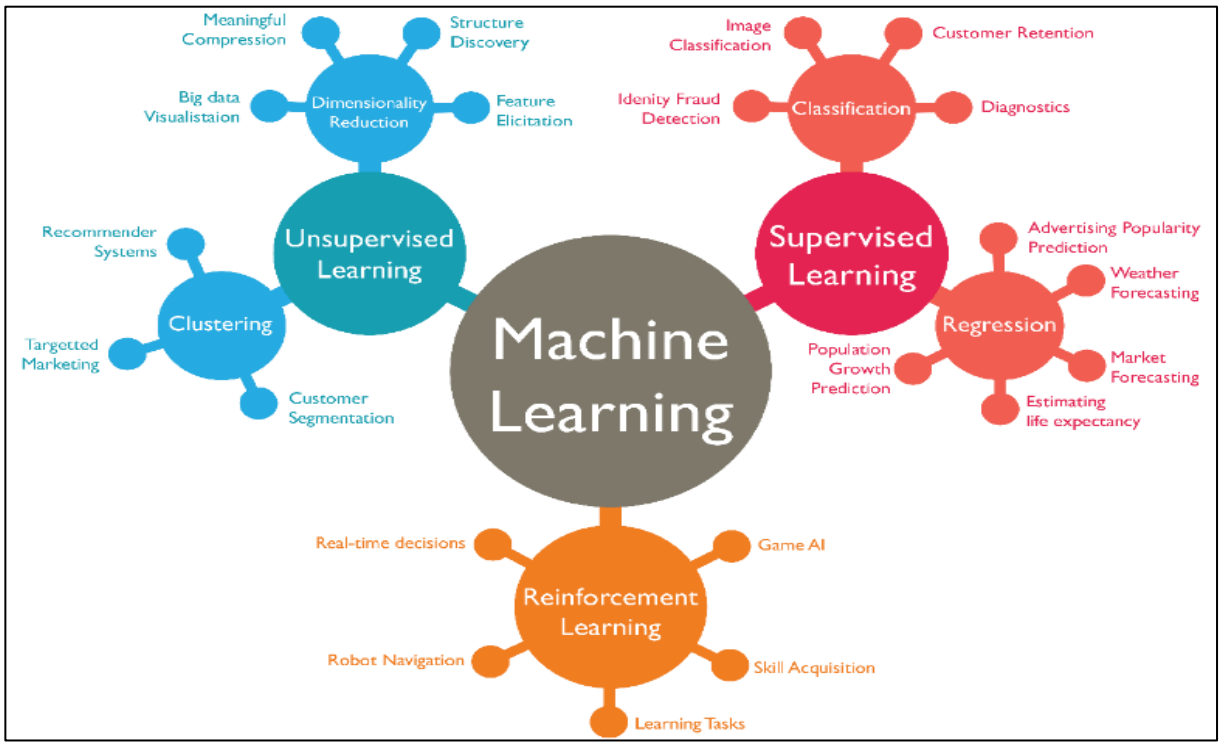


Figure.2.2 : Schéma des différents cas d'utilisation pour un type d'entrainement donné **[w5]**

3. L'Apprentissage automatique vers apprentissage profond

La plupart des méthodes d'apprentissage automatique (Machine Learning) fonctionnent Bien en raison de représentations et de fonctions de saisie conçues par l'homme. L'apprentissage automatique devient simplement une optimisation des poids pour mieux faire une prédiction finale.

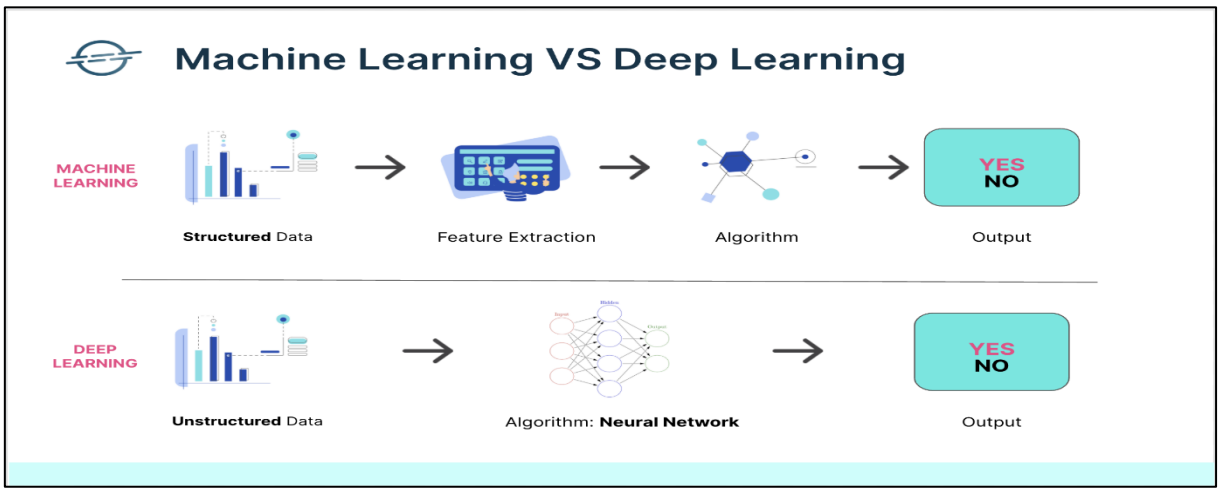


Figure.2.3: Machine-Learning vers Deep Learning [w6]

4. Introduction au réseau neuronal (Neural Network)

Un réseau de neurones artificiels est composé de nombreux neurones artificiels reliés entre eux selon une architecture de réseau spécifique. L'objectif du réseau de neurones est de transformer les entrées en sorties significatives. **[19]**

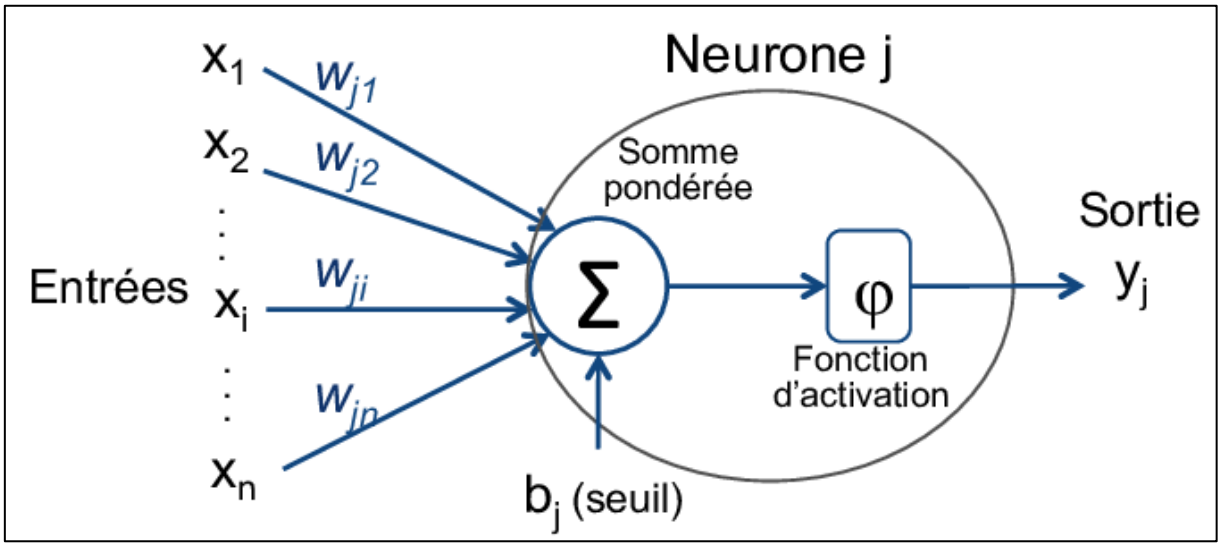


Figure.2.4 : La-structure d'un neurone artificiel

II. L'apprentissage en profondeur (Deep Learning)

1. Définition

Nous définissons l'apprentissage profond (DL) comme une classe de techniques d'apprentissage automatique (ML), dans lesquelles les informations sont traitées en couches hiérarchiques pour comprendre les représentations et les caractéristiques des données à des niveaux de complexité croissants. L'apprentissage profond est appelé, également apprentissage hiérarchique ou apprentissage profond structuré **[14]**.

Tous les algorithmes d'apprentissage profond sont des réseaux de neurones (Neural networks), qui partagent certaines propriétés de base communes. Ils sont tous constitués de neurones interconnectés, organisés en couches. Ce qui les différencie, c'est l'architecture du réseau (ou la manière dont Les neurones sont organisés dans le réseau) et parfois la manière dont ils sont formés. **[15]**

2. Principe

Les modèles d'apprentissage approfondi reposent également sur l'extraction des entités de niveau supérieur des entités de niveau inférieur afin d'obtenir une représentation stratifiée des données d'entrée via une approche d'apprentissage non supervisée sur les différents niveaux des entités [4]. Un classement des notions et des théories est obtenu en apprenant différentes couches de représentations des données représentant différents niveaux d'absorption des données. Certains des cadres d'apprentissage profond de la littérature sont les réseaux de croyances profondes (DBN) [10], les réseaux d'auto-encodeurs profonds / empilés (DAE / SAE) et les réseaux de neurones convolution els (CNN). Ces cadres d'apprentissage profond ont été utilisés dans divers secteurs, tels que le traitement du langage naturel, la reconnaissance de la parole, la reconnaissance audio, la reconnaissance et la détection d'objets et la vision par ordinateur. L'apprentissage profond est une branche des algorithmes d'apprentissage automatique qui : **[14]**

- Utilise plusieurs couches de nœuds de traitement non linéaires pour l'extraction et la transformation d'entités. Les couches successives utilisent les sorties des couches précédentes en entrée.
- Apprends plusieurs niveaux de représentations liés à différents niveaux d'abstraction. Ces niveaux représentent une hiérarchie de concepts.

3. Quelques méthodes d'apprentissage profond

Nous présentons les principales méthodes d'apprentissage profond. La liste suivante n'est pas exhaustive, mais elle représente la grande majorité des algorithmes utilisés.

3.1. Machines de Boltzmann Restreintes

Une machine de Boltzmann est composée d'une couche de neurones qui reçoit l'entrée, ainsi que d'une couche de neurones cachée. Si nous supposons que les neurones d'une même couche sont indépendants entre eux, nous appelons cette configuration une machine de Boltzmann restreinte (RBM) **[18]**. Les machines de Boltzmann restreintes sont un type particulier de réseau de neurones génératifs, où les neurones sont organisés en deux couches, à savoir visible et masquée. Contrairement aux réseaux à retransmission directe, les données d'une RBM peuvent circuler dans les deux sens des unités visibles aux unités cachées, et inversement. La RBM est l'un des outils d'apprentissage en profondeur les plus populaires en raison de sa capacité à connaître

la distribution de la probabilité des entrées de manière supervisée et non supervisée. Elle a été introduite par PaulSmolensk en 1986 avec le nom Harmonium. Cependant, elle est popularisée par Hinton en 2002 avec l'avènement de l'algorithme d'apprentissage amélioré pour RBM. Après cela, elle a eu une large application dans diverses tâches telles que l'apprentissage de la représentation, la réduction de la dimensionnalité, les problèmes de prédiction **[w3]**, la classification, la régression, le filtrage collaboratif (collaborative filtering), l'‘apprentissage des fonctionnalités (feature Learning) et modélisation des sujets (topicmodeling). En 2002, le professeur Hinton a introduit la divergence contrastive (CD), un algorithme non supervisé pour l'apprentissage des RBM. **[15]**

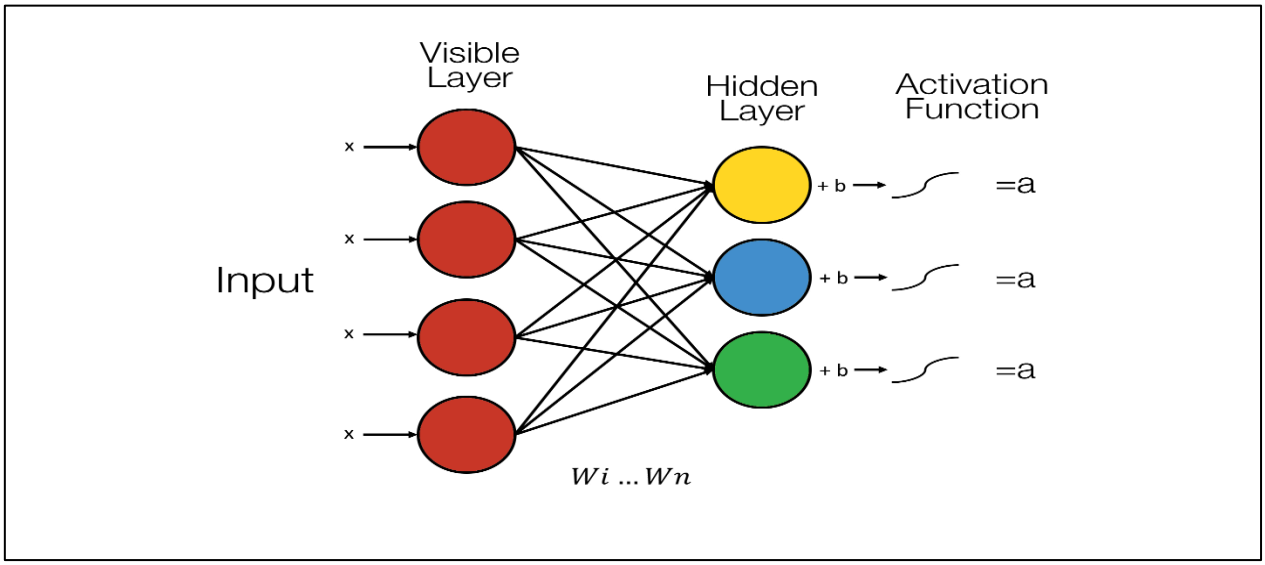


Figure.2.5 : Machine de Boltzmann restreinte

L'énergie (ou fonction) d'activation pour une Machine de Boltzmann Restreinte est définie de la manière suivante :

$$\boldsymbol{E} = \left[\sum_{\boldsymbol{ij}} \left(\boldsymbol{w_{ij} x_i\, h_j}\right) + \sum_{\boldsymbol{i}} \left(\boldsymbol{b_i\, x_i}\right) + \sum_{\boldsymbol{i}} \left(\boldsymbol{c_j\, h_j}\right)\right]$$

Avec :

- $\boldsymbol{w_{ij}}$ la matrice de poids entre le neurone **j** et le neurone **i** ;
- $\boldsymbol{x_i}$ est l'état du neurone visible **i** , $\boldsymbol{x_i} \in \{\boldsymbol{0}, \boldsymbol{1}\}$;
- $\boldsymbol{h_j}$ est l'état du neurone caché **j** ;
- $\boldsymbol{b_i}$ et sont $c_j$respectivement les biais des neurones $\boldsymbol{x_i}$**et**$\boldsymbol{h_j}$.

3.2. Perceptrons multicouches

Des réseaux neuronaux organisés en plusieurs couches (au moins une couche cachée) au sein des quelles une information circule de la couche d'entrée vers la couche de sortie uniquement, il s'agit donc d'un réseau à propagation directe (feedforward) avec propagation anticipée et couches entièrement connectées. Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global. Le concept de base du perceptron singulier a été introduit par Rosenblatt en 1958. Le perceptron calcule une sortie unique à partir de multiples entrées à valeurs réelles en formant une combinaison linéaire en fonction de ses poids d'entrée, puis en plaçant éventuellement la sortie via une fonction d'activation non linéaire. Mathématiquement, cela peut être écrit comme **[15] :**

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) = \varphi(w^T x + b)$$

Avec :

- **w** désigne le vecteur des poids.
- **x** est le vecteur des entrées.
- **b** le biais.
- φ Représente la fonction d'activation.

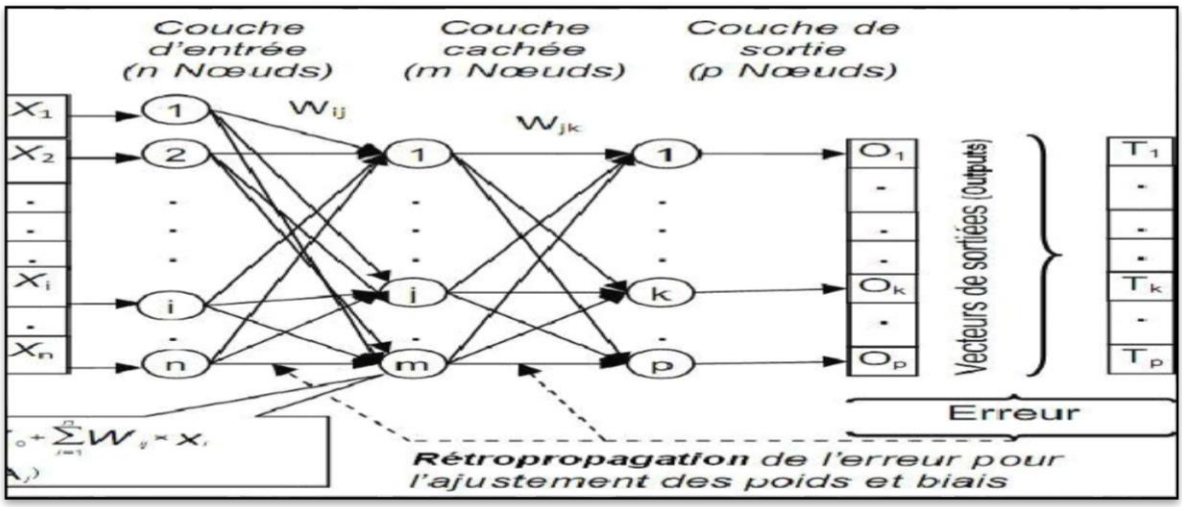


Figure 2.6 : Perceptron multicouches

3.3. Réseaux de croyance profonde (DBN)

Un réseau DBN est un type de réseau de neurones profonds qui est essentiellement un modèle génératif probabiliste comprenant plusieurs couches de variables cachées. Ces réseaux ont à la fois des bords dirigés et non dirigés. Il est formé à l'aide d'une série de RBM, souvent d'auto-encodeurs, avec une couche supplémentaire formant un réseau bayésien. L'utilisation de RBM signifie la présence d'aucune connexion intra-couche. De plus, les performances d'un DBN dépendent en grande partie de l'initialisation des nœuds. Par conséquent, les couches utilisent un apprentissage préalable non supervisé à l'aide de la procédure d'empilement de RBM, qui intègre une divergence contraste (CD). Un réseau de croyance (BN) est un graphe acyclique dirigé constitué de couches d'unités binaires stochastiques, chaque couche connectée ayant une pondération. Ces unités binaires stochastiques ont l'état 0 ou 1 et la probabilité d'être activé (devenant 1) est déterminée par un biais et une entrée pondérée provenant d'autres unités **[15]**, représentée par :

$$\boldsymbol{p(u_i = 1) = \frac{1}{1 + e^{-(b_i + \sum_j u_i w_{ij})}}}$$

Où :

- **u** est une unité stochastique.
- **b** le biais associé à cette unité.
- **w** le paramètre pondéré.

Certaines de ces unités sont des unités visibles. Celles-ci posent deux problèmes principaux qui doivent être résolus. Ce sont :

- **Le problème d'inférence :** où les états des unités non observées doivent être inférés.
- **Le problème d'apprentissage :** les interactions entre les unités sont ajustées pour que le réseau soit capable de générer les données observées dans les unités visibles.

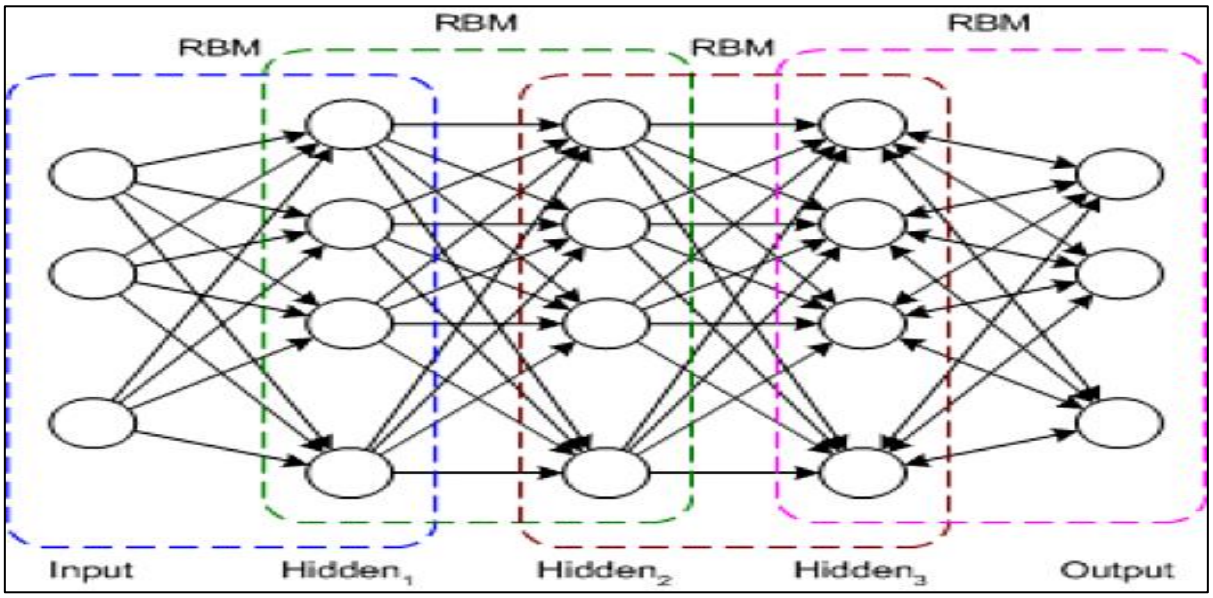


Figure 2.7 : Schéma d'un réseau DBN.

Un réseau DBN est un réseau de croyances multicouches dans lequel chaque couche est un RBM empilé les uns sur les autres pour former le réseau DBN. La première étape de l'apprentissage d'un DBN en utilisant une approche de base consiste à apprendre une couche de caractéristiques des unités visibles à l'aide de l'algorithme CD. L'étape suivante consiste à traiter les activations des caractéristiques précédemment formées comme des unités visibles, puis à en apprendre davantage à partir des caractéristiques des couches cachées suivantes. La dernière étape consiste à former l'ensemble du réseau DBN de manière supervisée, ce qui permet d'affiner les paramètres du réseau. Comme indiqué précédemment, les RBM peuvent être empilés et formés de manière gourmande pour former ces DBN qui sont des modèles graphiques permettant d'apprendre à extraire des représentations hiérarchiques profondes des données d'apprentissage en entrée. Ces Framework modélisent comme suit la distribution commune entre le vecteur de données observé **x** et les **l** couches cachées h^k: **[15]**

$$P(x,h^1,\dots\dots,h^l) = \left(\prod_{k=0}^{l-1} p(h^k|h^{k+1})\; p(h^{l-1},h^l)\right)$$

Où

x=h^0 et p($h^{k-1}|h^k$) représente la distribution conditionnelle de l'unité visible conditionnée par les unités cachées de la RBM au niveau **k**. De plus, la distribution conjointe visible-cachée dans le niveau supérieur du RBM est $p(h^{l-1}|h^l)$.

III. Les Réseaux Bayésiens (RB)

Les réseaux bayésiens sont un formalisme de raisonnement probabiliste introduit en autre par [Kim & Pearl, 1987, Lauritzen & Speigelhalter, 1988, Jensen, 1996, Jordan, 1998, Naïmet al. 2004]. **[16]**

Les RB sont la combinaison de la théorie des probabilités et de la théorie des graphes. Les RB peuvent adopter une structure causale une structure aléatoire (cas peu fréquent du fait de la complexité des calculs et du non praticité des résultats obtenus). Les RB sont surtout employés pour décrire des situations où l'information est incertaine qui peut également être méconnue ou ignorée. Les probabilités obtenues donnent une représentation de la connaissance que nous avons de la situation. **[17]**

1. Définition

B = (G,$\boldsymbol{\theta}$) est un réseau bayésien si G = (X, E) est un graphe acyclique dirigé dont les sommets représentent un ensemble de variables aléatoires X = {$\mathbf{x_1}$, · · ·,$\mathbf{x_n}$}, et si $\boldsymbol{\theta_i}$= [P($\mathbf{x_i}$/$\mathbf{x_{Pa(x_i)}}$)] est la matrice des probabilités conditionnelles du nœud i connaissant l'état de ses parents Pa ($\boldsymbol{x_i}$) dans G. Une hypothèse imposée par la théorie des réseaux bayésiens est que pour chaque variables$\boldsymbol{x_i}$, l'ensemble de variables Pa ($\boldsymbol{x_i}$) doit être tel que $\boldsymbol{x_i}$ est conditionnellement indépendants à $\boldsymbol{x_j}$ (j≠ $\boldsymbol{i}$) sachant Pa ($\boldsymbol{x_i}$) noté $\boldsymbol{x_i}\perp\boldsymbol{x_i}$| Pa ($\boldsymbol{x_i}$) .Un réseau bayésien B représente une distribution de probabilité sur X dont la loi jointe peut se simplifier de la manière suivante : **[16]**

$$P(x_1, x_2, \dots\dots, x_n) = \prod_{i=1}^{n} p\left(x_i \middle| x_{Pa(x_i)}\right)$$

Cette décomposition de la loi jointe permet d'avoir des algorithmes d'inférence puissants qui font des réseaux bayésiens des outils de modélisation et de raisonnement très pratiques lorsque les situations sont incertaines ou les données incomplètes. Ils sont alors utiles pour les problèmes de classification lorsque les interactions entre les différentes variables peuvent être modélisées par des relations de probabilités conditionnelles. Lorsque la structure du réseau bayésien n'est pas fournie a priori par un expert, il est possible d'en faire L'apprentissage à partir d'une base de données. La recherche de structure de réseaux bayésiens n'est pas simple, principalement à cause de la taille super-exponentielle de l'espace de recherche en fonction du nombre de variables. Nous allons commencer par introduire quelques notions générales sur la structure. Des réseaux bayésiens, la façon d'associer un score à cette structure et les propriétés intéressantes de ces scores. Ensuite nous détaillerons les méthodes de

recherche de structure les plus couramment utilisées, de la recherche de la causalité, au parcours heuristique de l'espace des réseaux bayésiens avec les différents problèmes d'initialisation que cela pose. Nous comparerons alors ces méthodes grâce à deux séries de tests. La première série de tests concerne la capacité des méthodes à retrouver une structure connue. L'autre série de tests permet d'évaluer l'efficacité de ces méthodes à trouver un bon réseau bayésien pour des problèmes de classification en utilisant éventuellement certaines connaissances a priori sur la tâche à résoudre. Nous conclurons alors sur les avantages et inconvénients des différentes méthodes utilisées et évoquerons plusieurs perspectives.

2. Étapes du développement d'un réseau Bayésien

Les trois étapes du développement d'un RB sont décrites ci-après.

2.1. Choix de la granularité des nœuds

Pour décrire un RB, il faut tout d'abord définir le type des nœuds nécessaires : continu ou discret. Par la suite, nous nous focaliserons sur les nœuds discrets qui sont décrits par un nombre d'états finis et mutuellement exclusifs. Cette étape est nommée choix de la granularité et correspond au nombre d'états décrivant le nœud.

2.2. Détermination de la structure du réseau Bayésien

Plusieurs types de raisonnement existent pour la recherche de structure, par exemple les méthodes d'apprentissage du réseau comme la méthode K2, de recherche gloutonne ou d'algorithme génétique. Mais pour utiliser ces méthodes, il faut être en possession de données statistiques importantes et précises afin d'obtenir le réseau le plus représentatif. Il est également possible, si nous possédons un modèle probabiliste d'y associer une structure qui correspond à ce modèle. La dernière possibilité est une connaissance des relations de cause à effet. Pour la construction de notre réseau bayésien, nous avons opté pour une représentation causale qui permet une représentation lisible et facilement exploitable par différentes personnes.

2.3. Détermination des modèles probabilistes ou le paramétrage des tableaux de probabilités

Le paramétrage des tableaux de probabilités fixe les probabilités a priori et conditionnelles du réseau. Si nous disposons d'exemples représentatifs et précis alors nous pouvons utiliser des méthodes d'apprentissage de paramètres telles que l'algorithme de maximum de vraisemblance proposé par Dempster, qui auront pour but de remplir les tableaux de probabilités **[17].**

3. Un exemple dans la modélisation des risques

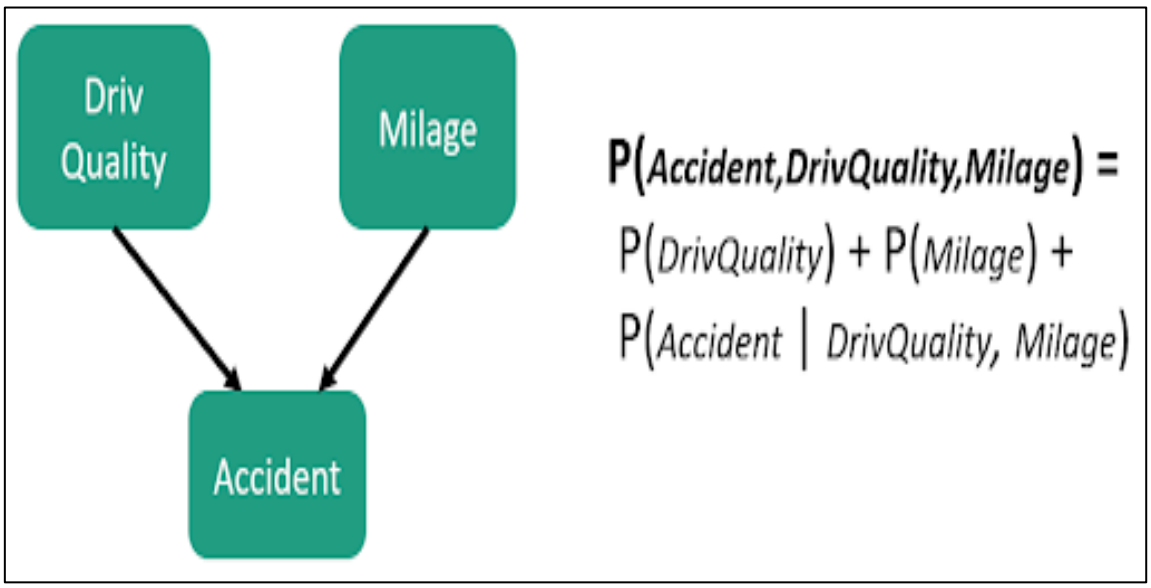


Figure 2.8 : Une modélisation des risques

Conclusion

Le machine Learning est un domaine de recherche très actif, qui ne cesse de progresser afin d‘améliorer les performances des résultats.

L‘apprentissage profond (Deep Learning) est un outil très puissant qui permet d'effectuer de multiples actions et révolutionner plusieurs domaines technologiques. Traduction automatique moderne, moteurs de recherche, assistants informatiques et plusieurs applications de notre vie quotidienne sont tous alimentés par un apprentissage profond.

Ce chapitre a présenté les architectures des réseaux de neurones les plus connues et réseau Bayésien ainsi que les nouvelles architectures qui semblent avoir un avenir prometteur dans différents domaines d‘application de la technologie.

Chapitre 3 : L’Etat de l’art

Introduction

Les incidents de piratage augmentent de jour en jour à mesure que la technologie se déploie, Un grand nombre d'incidents de piratage sont signalés par les entreprises chaque année. Et pour éviter ces incidents et avoir un environnement plus sécurisé.

Les solutions de sécurité actuelles comprennent l'utilisation des boîtes centrales comme le pare-feu, les Antivirus et les systèmes de détection d'intrusion (IDS). Un IDS est un type d'outil de sécurité qui surveille le trafic réseau et analyse le système pour détecter les activités suspectes et alerte le système ou l'administrateur réseau**.** En 1980 Anderson proposé le premier système de détection d'intrusion (IDS), et avec le développement actuel les chercheurs aussi sont développer les systèmes de sécurité comme les IDS qui sont aussi développer. Et pour avoir plus des incidents de piratage voilà quelque exemple :

En 2007 Distributed Denial of Service (DDoS) attaque a été lancée contre les sites Web estoniens, prétendument par la Russie, le 17 juin 2008, Amazona commencé à recevoir des demandes authentifiées de plusieurs utilisateurs dans l'un de ses emplacements. Janvier 2013, Agence européenne pour la sécurité des réseaux et de l'information (ENISA) signalé que Dropbox a été attaquée par DDoS et a subi une perte substantielle de service pendant plus de 15 heures affectant tous les utilisateurs à travers le monde. 28 septembre 2014 Facebook a été attaquée par DDoS. **[40]**

1. Historique de l'apprentissage automatique dans la détection d'intrusion

Dans la figure ci-dessous, nous résumons quelques études réalisées ces dix dernières années, en précisant les algorithmes et leur année.

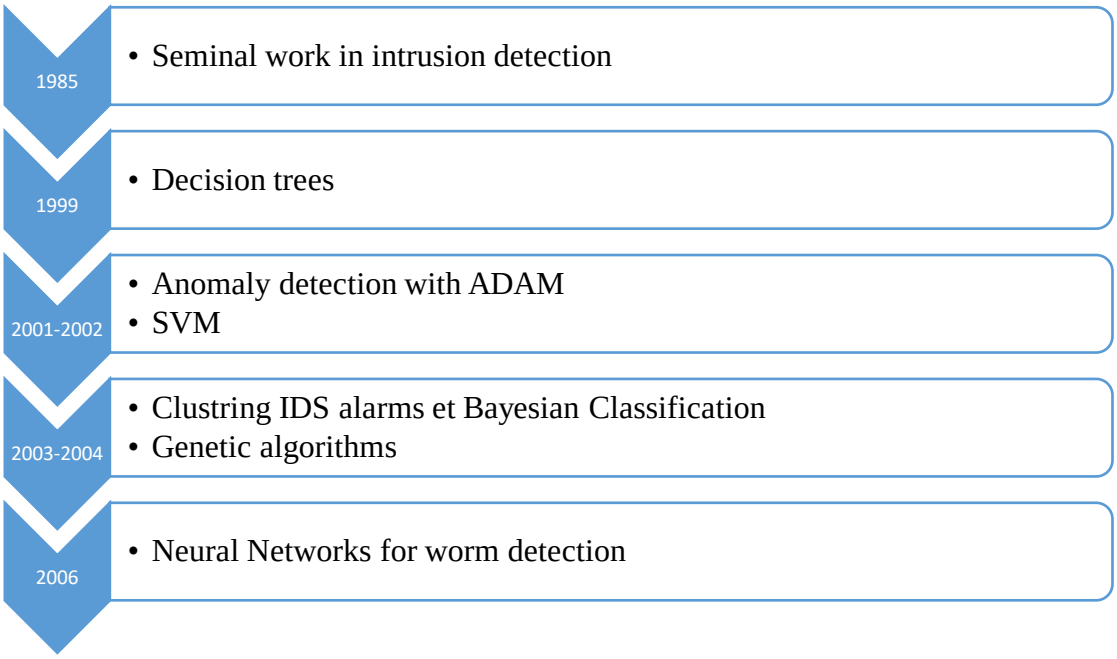


Figure 3.1 : Historique de l'apprentissage automatique [39]

2. Mécanismes des IDS

À haut niveau, les mécanismes de détection utilisés par les IDS sont trois types :

- détection d'abus.
- détection d'anomalies.
- détection hybride.

3. La détection des abus

Les techniques de détection des abus peuvent être globalement classées en techniques basées sur la connaissance et basées sur l'apprentissage automatique. Dans la technique basée sur les connaissances, le trafic réseau ou les données d'audit de l'hôte (telles que les traces d'appels système) sont comparés à des règles ou des modèles d'attaque prédéfinis. Les techniques basées sur les connaissances peuvent être classées en trois types :

3.1. Correspondance des signatures

Analysent les paquets entrants par rapport à des modèles fixes. Si l'un des modèles correspond à l'en-tête du paquet, le paquet est marqué comme anormal.

3.2. Analyse de la transition des états

Maintiennent un modèle de transition d'état du système pour les modèles suspects connus. Différentes branches du modèle conduisent à un état final compromis de la machine.

3.3. Systèmes experts basés sur des règles

Basés sur des règles maintiennent une base de données de règles pour différents scénarios intrusifs. L'IDS basé sur les connaissances nécessite une maintenance régulière de la base de données de connaissances de manière dynamique et ne peut pas détecter les nouvelles variantes des attaques.

La détection des abus peut également être effectuée à l'aide d'algorithmes d'apprentissage automatique supervisé tels que le réseau neuronal artificiel à propagation arrière (BP-ANN), l'arbre de décision (DT) C4.5, et la machine vectorielle de support multi-classes (SVM).

4. Les techniques de détection des anomalies

Les IDS basés sur la détection d'anomalies reposent sur l'hypothèse que le comportement de l'attaquant diffère du comportement normal de l'utilisateur. Il aide à détecter les attaques en évolution. Les IDS basés sur des anomalies modélisent le comportement normal du système et continuent de le mettre à jour pendant un certain temps. Par exemple, chaque connexion réseau est identifiée par un ensemble de fonctionnalités telles que le protocole, le service, le nombre de tentatives de connexion, les paquets par flux, les octets par flux, l'adresse source, l'adresse de destination, le port source, le port de destination, etc. Les statistiques comportementales de ces fonctionnalités sont enregistrées sur une période. Tout écart anormal dans les valeurs des fonctionnalités pour tout flux de connexion sera marqué comme anormal par le moteur de détection d'anomalies. Elles sont largement classées en trois types :

4.1. Techniques basées sur la machine à états finis (FSM)

Une machine à états finis (FSM) produit un modèle comportemental composé d'états, de transitions et d'actions. La détection d'anomalies peut également être effectuée à l'aide d'algorithmes d'apprentissage automatique semi-supervisés et non supervisés tels que le réseau neuronal SOM (Self Organizing Map), algorithmes de regroupement (clustering), et Machine à vecteurs de support à une classe (SVM).

4.2. Techniques basées sur l'apprentissage automatique

Les IDS basés sur l'apprentissage automatique pour la détection des anomalies fournissent un système basé sur l'apprentissage pour découvrir les attaques zero-day. Une attaque Zero-day fait référence à l'exploitation d'une vulnérabilité qui n'était pas connue auparavant. Ils ont été classés en quatre types :

- Classificateurs uniques avec toutes les caractéristiques de l'ensemble de données.
- Classificateurs uniques avec des caractéristiques limitées de l'ensemble de données.
- Classificateurs multiples avec toutes les caractéristiques de l'ensemble de données.
- Classificateurs Multiples avec des caractéristiques limitées de l'ensemble de données.

5. Les approches de détection hybrides

Les approches de détection hybrides intègrent une approche de détection des abus et des anomalies pour détecter les attaques. En général, certains des avantages de l'utilisation de l'IDS basé sur l'apprentissage automatique par rapport à l'IDS classique basé sur les signatures sont les suivants :

- Il est facile de contourner l'IDS basé sur la signature en effectuant de légères variations dans un modèle d'attaque, tandis que l'IDS basé sur l'apprentissage automatique basé sur des techniques supervisées peut facilement détecter les variantes d'attaque à mesure qu'ils apprennent le comportement du flux de trafic.
- La charge du processeur est faible dans les IDS basés sur l'apprentissage automatique, car ils n'analysent pas toutes les signatures de la base de données de signatures

- Certains IDS basés sur l'apprentissage automatique, en particulier basés sur des algorithmes d'apprentissage non supervisés, peuvent détecter de nouvelles attaques.
- Les IDS basés sur l'apprentissage automatique peuvent capturer les propriétés complexes du comportement d'attaque et par la suite améliorer la précision et la vitesse de détection par rapport aux IDS conventionnels basés sur les signatures Différents types d'attaques continuent d'évoluer. L'IDS basé sur les signatures nécessitera la maintenance de la base de données de signatures de temps en temps et la maintiendra à jour, tandis que l'IDS basé sur l'apprentissage automatique basé sur le cluster ING et la détection des valeurs aberrantes ne nécessitera pas une telle mise à jour.

6. Différence entre la détection des abus et la détection d'anomalie

Détection des abus	**Détection d'anomalie**
Il modélise les systèmes/ signatures d'attaque bien connues existantes pour détecter une activité malveillante .Une correspondance du modèle entrant avec les profils d'attaque existants est déclarée aussi suspecte.	Il utilise le profil de comportement normal établi du système. Une incompatibilité du modèle entrant avec le profil normal existant est déclarée suspecte.
La « correspondance de signature » est une approche très populaire de détection des abus, qui a un succès commercial.	«L'apprentissage statistique » est une approche de détection d'anomalies très populaire et les chercheurs travaillent toujours dans ce sens
Les approches d'apprentissage automatique supervisé sont bien adaptées aux utilisations abusives détection telle que DT, NB, BP-ANN.	Les approches d'apprentissage automatique semi ou non supervisées sont bien adaptées à la détection d'anomalies telles que le regroupement, SOM-ANN, SVM à une classe, etc.

Impossible de détecter les attaques inconnues.	Bon pour détecter les attaques inconnues.
Faible occurrence de faux positifs	Occurrences élevées de faux positifs.
Très bonne précision pour détecter les attaques connues.	Fournit une bonne précision pour les attaques inconnues
alléguer réside dans le maintien d'une base de données à jour de toutes les signatures d'attaques connues	Le défi consiste à différencier l'attaque et à faire évoluer le comportement normal.
Exp. SNORT , Suricata	Exp. IDES , MINDS

Tableau 3.1 : Différence entre la détection des abus et la détection d'anomalie [40]

7. Algorithmes d'apprentissage automatique dans la conception des IDS

Il existe deux principaux types d'apprentissage automatique : l'apprentissage supervisé et non supervisé. L'apprentissage Supervisé repose sur des informations utiles contenues dans des données étiquetées. La classification est la tâche la plus courante dans apprentissage supervisé (et est également utilisé le plus fréquemment dans l'IDS), cependant, l'étiquetage manuel des données est coûteux et chronophage. Par conséquent, le manque de données étiquetées suffisantes constitue le principal goulot d'étranglement à l'apprentissage supervisé En revanche, l'apprentissage non supervisé extrait des fonctionnalités précieuses des informations à partir de données non étiquetées, ce qui facilite grandement l'obtention de données d'apprentissage. Cependant, la détection des performances des méthodes d'apprentissage non supervisé sont généralement inférieures à celles de l'apprentissage supervisé.

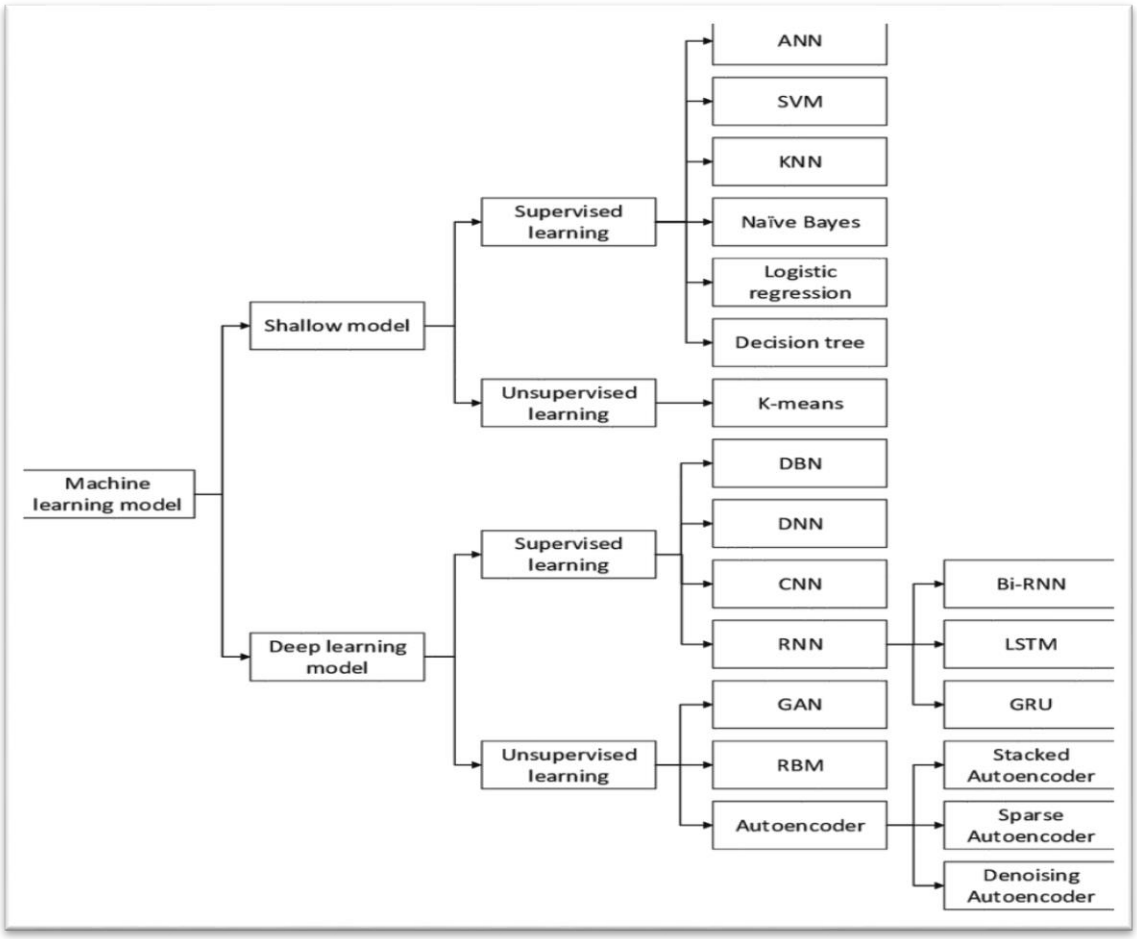


Figure 3.2 : taxonomie des algorithmes d'apprentissage automatique.

L'apprentissage en profondeur utilise les couches suivantes de traitement de l'information dans une hiérarchie pour la classification ou la fonctionnalité représentation. Il utilise les réseaux profonds ayant plusieurs couches de traitement. Il se compose d'un niveau d'entrée fournissant les données de base et suivies de couches cachées consécutives qui analysent les données et la sortie est produite. Il a gagné popularité ces dernières années. L'IDS existant peut être amélioré en adoptant cette dernière technique. Deng et al fourni la catégorisation des méthodes de deep Learning en fonction de leur architecture dans les types suivants : générative (non supervisée), discriminant (supervisé) et hybride. Profondeur non surveillée les architectures d'apprentissage ou génératives utilisent les méthodes : Auto Encoder (AE) et Boltzmann Machine (BM).

Semblable à ANN, AE utilise des couches cachées ; Cependant, il n'a que trois couches cachées. Les nœuds de la couche d'entrée et couche de sortie sont les mêmes. Les nœuds cachés sont utilisés pour réduire la fonctionnalité dimensionnalité et fournir le nouvel ensemble de fonctionnalités. UN différent ensemble de caractéristiques sont appris dans les profondeurs en

cascade pour s'entraîner le plus précisément. BM prend la décision stochastique en utilisant le la structure des unités binaires du neurone. Deep BM (DBM) a une structure en cascade alors que BM restreint n'a pas de connexions parmi les unités cachées. Les multiples couches qui s'empilent un par un forment un réseau de croyances profondes (DBN). Supervisé l'apprentissage est utilisé pour distinguer certaines parties des données et a été utilisé pour les classifications de modèles. Réseau de neurones à convolution (CNN) est un exemple d'apprentissage supervisé qui fournit apprentissage rapide. CNN utilise trois champs : les champs récepteurs locaux, poids partagés et mise en commun. L'approche hybride utilise les deux méthodes. Un exemple d'architecture hybride est Deep Réseaux de neurones (DNN). DNN fournit une connexion entièrement couche cachée formant des réseaux multicouches en cascade. L'utilisation de l'apprentissage en profondeur pour la classification des images est assez populaire. Cependant, le défi réside dans l'adoption de la profondeur apprentissage pour la détection des attaques dans le trafic réseau. Au cours des dernières années, il a été appliqué l'apprentissage en profondeur pour la détection d'intrusion comme indiqué dans le tableau XII. Seok et al. Ont employé profondément l'apprentissage pour la détection d'attaque qui est basé sur la conversion de code malveillant dans l'image et en appliquant CNN qui prend ces images en entrée pour apprendre les caractéristiques d'attaque. Aussi, la plupart du travail pour l'IDS basé sur l'apprentissage profond utilise cette approche pour réduire la dimensionnalité des caractéristiques. Il a de nombreux avantages. Combiner le supervisé et le non supervisé les approches d'apprentissage en profondeur améliorent les résultats de détection des approches traditionnelles. Il aide à développer de nouvelles méthodes sur la sécurité du réseau qui sont plus sûres qu'approches traditionnelles d'apprentissage automatique. L'apprentissage en profondeur est adaptable au contexte changeant des données car il effectue l'analyse exhaustive des données. Cependant, l'utilisation de l'apprentissage profond pour l'analyse des attaques est toujours un domaine difficile et ouvert pour les chercheurs travailler sur. Les ressources nécessaires à la formation du réseau sont aussi assez énormes. L'apprentissage en profondeur peut être appliqué lorsqu'il est difficile de trouver la corrélation entre l'entrée brute et la cible classé. Un système de détection d'intrusion fiable doit pouvoir gérer les entrées bruyantes et les grandes données discrètes ou continues. L'apprentissage par renforcement (RL) est un autre domaine intéressant De la recherche. L'apprentissage par renforcement (RL) est l'un des algorithmes d'apprentissage automatique où plusieurs agents et machines travailler/interagir ensemble pour apprendre le comportement au sein d'un contexte et améliorer les performances de détection des attaques. Des capteurs ou des agents détectent l'environnement à des intervalles

de temps discrets et l'entrée est mappée pour localiser les informations d'état. Une fois que Les agents RL exécutent l'action et la rétroaction est observée à partir de l'environnement. Les actions correctes des agents sont récompensées par le environnement, appelé signal de renforcement. Les agents tirent ensuite parti les récompenses et améliorer les connaissances sur l'environnement pour sélectionner l'action suivante. Certains chercheurs ont appliqué RL pour détecter les attaques par déni de service distribué [27].

8. Modèles d'apprentissage profond

Les modèles d'apprentissage profond se composent de divers réseaux profonds. Parmi eux, des réseaux de croyances profondes (DBN), réseaux de neurones profonds (DNN), réseaux de neurones consolatifs (CNN) et les réseaux de neurones récurrents (RNN) sont des modèles d'apprentissage supervisé, tandis que les autos encodeurs, les restreignent Boltzmann machines (RBM) et les réseaux antagonistes génératifs (GAN) sont des modèles d'apprentissage non supervisés. Le nombre d'études sur les IDS basés sur l'apprentissage profond a augmenté rapidement de 2015 à aujourd'hui. Les modèles d'apprentissage en profondeur apprennent directement les représentations d'entités à partir des données d'origine, telles que les images et les textes, sans nécessiter une ingénierie manuelle des fonctionnalités. Ainsi, les méthodes d'apprentissage en profondeur peuvent exécuter d'une manière de bout en bout. Pour les grands ensembles de données, les méthodes d'apprentissage en profondeur présentent un avantage significatif sur des modèles peu profonds. Dans l'étude du Deep Learning, les principaux axes sont l'architecture réseau, sélection d'hyper paramètres et stratégie d'optimisation. Une comparaison de divers modèles d'apprentissage en profondeur est présentée dans le tableau 2.

Algorithmes	Types de données appropriés	Supervisé ou Non supervisé	Les fonctions
RBM	Vecteurs de caractéristiques	Non supervisé	Extraction de caractéristiques, Fonctionnalité réduction, Dé bruitage

DBN	Vecteurs de caractéristiques	supervisé	Extraction de caractéristiques Classification
DNN	Vecteurs de caractéristiques	supervisé	Extraction de caractéristiques Classification
CNN	Données brutes, Vecteurs de caractéristiques Matrices	supervisé	Extraction de caractéristiques Classification
RNN	Données brutes, Vecteurs de caractéristiques Données de séquence	supervisé	Extraction de caractéristiques Classification
GAN	Données brutes, Vecteurs de caractéristique	Non supervisé	Augmentation des données, Formation contradictoire

Tableau 3.2 : comparaison entre les modèles d'apprentissage en profondeur [41].

9. Ensembles de données de référence dans IDS

9.1. DARPA1998

L'ensemble de données DARPA1998 **[36]** a été construit par le laboratoire Lincoln du MIT et est un ensemble de données de référence dans les études IDS. Pour le compiler, les chercheurs ont collecté le trafic Internet sur neuf semaines, les sept premières semaines forment l'ensemble de formation et les deux dernières semaines forment l'ensemble de test. Le jeu de données contient à la fois des paquets bruts et des étiquettes. Il existe cinq types d'étiquettes, normal, déni de service (DOS), Sonde, utilisateur à racine (U2R) et distant à local (R2L). Parce que les paquets bruts ne peuvent pas être appliqués directement à modèles d'apprentissage automatique traditionnels, l'ensemble de données KDD99 a été conçu pour surmonter cet inconvénient.

9.2. KDD99

Le KDD99 [37] est le jeu de données de référence IDS le plus répandu à l'heure actuelle. Ses compilateurs extraits des entités en 41 dimensions à partir de données dans DARPA1998. Les étiquettes dans KDD99 sont les mêmes que les DARPA1998. Il existe quatre types de fonctionnalités dans KDD99 à savoir, les fonctionnalités de base, les fonctionnalités de contenu basées sur l'hôte des fonctionnalités statistiques et des fonctionnalités statistiques basées sur le temps. Malheureusement, le jeu de données KDD99 comprend de nombreux défauts. Premièrement, les données sont gravement déséquilibrées, ce qui fait que les résultats de la classification sont biaisés classes majoritaires. En outre, il existe de nombreux enregistrements en double et des enregistrements redondants. Beaucoup des chercheurs doivent filtrer soigneusement l'ensemble de données avant de pouvoir l'utiliser. En conséquence, l'expérimentation les résultats de différentes études ne sont pas toujours comparables. Dernier point mais non le moindre, les données KDD sont trop anciennes pour représentent l'environnement réseau actuel.

9.3. NSL-KDD

Pour surmonter les lacunes du jeu de données KDD99, le NSL-KDD [38] a été proposé. Les enregistrements du NSL-KDD ont été soigneusement sélectionnés sur la base du KDD99. Registres de différents les classes sont équilibrées dans le NSL-KDD, ce qui évite le problème biais de classification. Le NSL-KDD aussi suppression des enregistrements en double et redondants, par conséquent, il ne contient qu'un nombre modéré d'enregistrements. Par conséquent, les expériences peuvent être implémentées sur l'ensemble de données et les résultats de différents les articles sont cohérents et comparables. Le NSL-KDD atténue les problèmes de biais et de données redondance dans une certaine mesure. Cependant, le NSL-KDD n'inclut pas de nouvelles données, donc, classe minoritaire les échantillons font toujours défaut et ses échantillons sont toujours obsolètes.

9.4. UNSW-NB15

L'UNSW-NB15 **[39]** ensemble de données a été compilé par l'Université du Pays de Galles du Sud, où les chercheurs configuré trois serveurs virtuels pour capturer le trafic réseau et extraire des fonctionnalités en 49 dimensions en utilisant l'outil nommé Bro. L'ensemble de données comprend plus de types d'attaques que l'ensemble de données KDD99, et ses caractéristiques sont plus abondantes. Les catégories de données comprennent des données normales et neuf types d'attaques. Les fonctionnalités comprennent des fonctionnalités de flux, des fonctionnalités de base, des fonctionnalités de contenu, des fonctionnalités de temps, des fonctionnalités supplémentaires et caractéristiques étiquetées. L'UNSW-NB15 est représentatif des nouveaux ensembles de données IDS, et a été utilisé dans certains des études récentes. Bien que l'influence de l'UNSW-NB15 soit actuellement inférieure à celle du KDD99, elle est nécessaire pour construire de nouveaux ensembles de données pour développer de nouveaux IDS basés sur l'apprentissage automatique. **[27]**

10. Validation et mesure performance

10.1. Matrice de confusion

En Machine Learning, l'évaluation de la qualité de la classification est faite avec différentes mesures comme les faux positifs et les taux négatifs, la matrice de confusion, la précision, rappel et F-Mesure. La matrice de confusion est une technique d'évaluation appliquée à tout type de problème de classification. Elle affiche les quatre valeurs (vrai positif, vrai négatif, faux positif et faux négatif) d'une manière dont la relation entre elles est facilement compréhensible comme le montre le **tableau 6**.

		Prédiction de la classe	
		Classe négative (Normale)	**Classe positive (Attaque)**
Classe actuel	**Classe négative (Normale)**	Vrai négative (VN)	Faux positive (FP)
	Classe positive (Attaque)	Faux négative (FN)	Vrai positive (VP)

Tableau 3.3 : Matrice de confusion

10.2. Métrique d'évaluation

Les métriques de performance d'un IDS comprennent le taux d'exactitude, le taux de fausse alerte et le taux de détection des attaques, elles sont définies comme suit :

- **Taux d'Exactitude :** montre à qu'elle point le système est exacte, c'est le nombre des cas bien classés sur le nombre de type de tous les cas.

$$Exactitude = \frac{VP + VN}{VP + VN + FP + FN}$$

- **Taux de détection :** mesure le taux des attaques détectées par un IDS dans un environnement donné et pendant une durée donnée. C'est le nombre des attaques détecté sur le nombre des attaques existants dans le corpus.

$$\mathrm{DR} = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FN}}$$

- **Les fausses alarmes :** ce critère mesure le taux de fausses alertes générées par un IDS dans un environnement donné et pendant une durée donnée. C'est le nombre des alertes généré comme attaque sur le nombre des types classés comme normal existants dans le corpus.

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{VN} + \mathrm{FP}}$$

- **la précision** : c'est-à-dire la proportion de prédictions correctes parmi les points que l'on a prédits positifs. C'est la capacité de notre modèle pour qu'il ne déclenche d'alarme que pour un vrai incendie.

$$\text{Précision} = \frac{\mathrm{VP}}{\mathrm{VP} + \mathrm{FP}}$$

Pour évaluer un compromis entre rappel et précision, on peut calculer la "F-mesure", qui leur moyenne harmonique.

$$\mathrm{F-mesure} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} = \frac{2\mathrm{VP}}{2\mathrm{VP} + \mathrm{FP} + \mathrm{FN}}$$

Telle que : Rappel =DR. **[28] [29] [30]**

Conclusion

Dans ce chapitre nous avons présenté un état de l'art sur les IDS conçus par les techniques de l'apprentissage automatique pour avoir une vue globale de ce qui se fait de nos jours dans ce domaine. Les techniques de croyance profonde et de codage approfondi ont permis l'analyse de grands ensembles de données et une analyse des systèmes plus approfondie.

Chapitre 4 : Implémentation et discussion des résultats

Introduction

Apres avoir donné toute la théorie que nous voyons nécessaire pour le développement de notre système. Il est temps de mettre en œuvre les différentes fonctionnalités ainsi les algorithmes proposés.

Dans ce chapitre, nous explorons les réseaux bayésiens naïves DBN () et les RBNs pour le développement d'un système de détection d'intrusion déjà décrit dans le chapitre précédent pour faire une classification supervisée de la base NSL_KDD afin de concevoir un IDS basé sur l'analyse du comportement de ces connexions et permet de les classifier en deux types : attaque ou normale , et les attaques se répartissent en quatre classes : R2L,DoS, U2R et sondage. Premièrement, nous utilisons un réseau de croyances profondes pour réduire la dimensionnalité des ensembles de caractéristiques. Ceci est suivi d'un RBN pour classer l'intrusion en cinq issues, Normal, R2L, Dos, U2R, et Sonder.

I. Implémentation :

1. Matériel et logiciels utilisés

Dans notre travail, nous avons utilisé un environnement caractérisé par un PC (HP) de 8 Go de RAM, 1 CPU i7-3110M cadencé à 2.40GHz, tournant sous Windows 10 64 bits. En ce qui concerne le coté logiciel, notre développement est fait sur Anaconda qui est doté de Python 3.8 ainsi que l'éditeur jupyter 6.2.0 qui a été choisi pour sa simplicité et son efficacité. Plusieurs bibliothèques ont été utilisées : Tensorflow, keras, scipy, numpy, pandas, matplotlib, sklearn, Environnement Weka 3.8.5. Afin d'afficher les résultats obtenus nous avons développé une petite interface avec le langage Java script, HTML et CSS.

Environnement de programmation

1.1. Présentation de Weka

Weka (Waikato Environment for Knowledge Analysis) est une suite populaire de logiciels d'apprentissage automatique. Écrite en Java, Développée à l'université de Waikato, Nouvelle-Zélande. Weka est un Logiciel libre disponible sous la Licence publique générale GNU (GPL). L'espace de travail Weka contient une collection d'outils de visualisation et d'algorithmes pour l'analyse des données et la modélisation prédictive, allié à une interface graphique pour un accès facile de ses fonctionnalités **[31].**

1.2. Présentation de langage Python en informatique

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages. **[32]**

1.3. Définition jupyter

Jupyter se présente comme un outil extrêmement simple à mettre en œuvre qui nous permettra de transformer nos Jupyter Notebooks en applications web ou en Dashboard. **[33]**

1.4. Des bibliothèques python utilisées

- **Pandas**

Est une librairie Python qui a pour objectif de faciliter la vie en matière de manipulation de données. Les structures de données gérées par Pandas peuvent contenir tout type d'éléments à savoir (dans le jargon Pandas) des Séries et Data Frame et des Panel. Dans le cadre de nos expérimentations nous avons utilisé plutôt les Data frame car ils offrent une vue bidimensionnelle des données (comme un tableau Excel), et c'est exactement ce que l'on va chercher à utiliser pour nos modèles. **[34]**

- **Keras**

Est une API de réseaux de neurones de haut niveau, écrite en Python et ineffaçable avec Tensorflow, CNTK et Theano. Elle a été développée avec pour objectif de permettre des expérimentations rapides. Être capable d'aller de l'idée au résultat avec le plus faible délai possible étant la clef d'une recherche efficace. **[w4]**

- **Tensorflow**

Open source destinée au calcul numérique de haute performance. Son architecture flexible permet un déploiement facile du calcul sur diverses plates-formes (processeurs CPU, processeurs graphiques GPU, processeurs tenseurs TPU), et des ordinateurs de bureau aux clusters de serveurs, aux périphériques mobiles. La bibliothèque Tensorflow5 a été développée à l'origine par des chercheurs et des ingénieurs de l'équipe Google Brian au sein de l'organisation Google AI, elle prend en charge l'apprentissage automatique et l'apprentissage profond **[w5]**

- **Sklearn**

Scikit-learn est une bibliothèque libre destinée à l'apprentissage automatique et à la data-science. **[w6]**

2. Description de la base NSL-KDD et présentation du modèle de classification

2.1. Description de la base NSL-KDD

La base NSL-KDD (Network Security Layer-Knowledge Discovery in Databases) a été fondé sur l'ensemble de données KDD99, Cette dernière est une base de données qui contient des connexions TCP/IP extraites de l'ensemble de données d'évaluation des systèmes de détection d'intrusions. KDD99 été réalisées en 1998 par l'agence de L'armé américain DARPA (Défense Advanced Research Project Agency) et AFRL (Laboratoire de recherche de l'armée de l'air), ensuite MIT Lincoln Labs8 a collecté et distribué les ensembles de données pour l'évaluation du système de détection d'intrusions de réseau informatique. La base NSL-KDD est un ensemble de données qui représente une version réduite de l'originale KDD 99, proposé en 2010 par les chercheurs dans le domaine de détection d'intrusions réseaux afin de résoudre certains problèmes qui ont apparu dans la base KDD 99. NSL-KDD considérée comme un ensemble de données de référence pour aider les chercheurs à comparer les différentes méthodes de détection d'intrusions. Le NSL-KDD présente les différences suivantes par rapport à l'originale KDD 99 :

- Il n'inclut pas les enregistrements redondants dans les données d'apprentissage, ce qui améliore la performance de classification.
- Il n'y a pas d'enregistrements en double dans les ensembles de test proposés, ce qui aide à l'obtention de meilleurs taux de détection.
- Le nombre d'enregistrements dans les données d'apprentissage et les ensembles d'essais sont raisonnables, ce qui il est abordable d'exécuter les expériences sur l'ensemble complet sans la nécessité de choisissez au hasard une petite portion. Par conséquent, les résultats d'évaluation des travaux de recherche seront cohérents et comparables.

Les données NSL-KDD contiennent des enregistrements de connexion TCP/IP, dont chaque enregistrement est constitué de 41 attributs caractérisant la connexion, et un attribut représente 5 classes qui sont : normales et 4 types d'attaques. Les principaux types d'attaques de l'ensemble de données NSL-KDD sont (Probing, Dénis de Services, User to Root, Remote to User), Ces attaques ont été abordés en détail dans le premier chapitre de ce mémoire. **[36][37]**

Distribution des attaques de la base KDD99

Classes	**Normal**	**Probe**	**DOS**	**R2L**	**U2L**	**Total**
Nombre	97278	4107	391458	1126	52	494021
Pourcentage	19.69%	0.8313%	79.24%	0.2279%	0.0105%	100%

Tableau 4.1 : Répartition des attaques dans l'ensemble d'apprentissage KDD99

Classes	**Normal**	**Probe**	**DOS**	**R2L**	**U2L**	**Total**
Nombre	60593	4166	229853	16189	228	311029
Pourcentage	19.48%	1.34%	73.90%	5.20%	0.0 733%	100%

Tableau 4.2 : Répartition des attaques dans l'ensemble de Test KDD99

2.2. Le contenue de l'ensemble de données NSL-KDD

- **KDDTrain + .ARFF:** Ensemble complet NSL-KDD avec étiquettes binaires en format ARFF
- **KDDTrain + .TXT:** Ensemble complet de trains NSL-KDD incluant les étiquettes d'attaque et le niveau de difficulté au format CSV
- **KDDTrain + _20Percent.ARFF :** Un sous-ensemble de 20% du fichier KDDTrain + .arff
- **KDDTrain + _20Percent.TXT :** Un sous-ensemble de 20% du fichier KDDTrain + .txt
- **KDDTest + .ARFF:** Le test complet NSL-KDD avec des étiquettes binaires au format ARFF
- **KDDTest + .TXT:** Ensemble de test complet NSL-KDD incluant les étiquettes d'attaque et le niveau de difficulté au format CSV
- **KDDTest-21.ARFF :** Un sous-ensemble du fichier KDDTest + .arff qui n'inclut pas les enregistrements avec un niveau de difficulté de 21 sur 21
- **KDDTest-21.TXT :** Sous-ensemble du fichier KDDTest+ .txt qui n'inclut pas les enregistrements ayant un niveau de difficulté de 21 sur 21.

2.3. Attributs de la base NSL-KDD

Les 41 attributs de la base NSL-KDD et leurs types de données. Ces attributs peuvent être classés en trois groupes :

- **Les attributs de base :** ces attributs décrivent les informations de base d'une connexion, telles que la durée, les hôtes source et destination, port et flag.
- **Les attributs du trafic :** ces attributs sont basés sur des statistiques, tels que le nombre de connexions vers la même machine.
- **Les caractéristiques du contenu :** ces attributs sont construits à partir de la charge utile (Data) des paquets du trafic tels que nombre d'échec de connexion et le nombre d'accès aux fichiers de contrôle.

3. Processus de génération du modèle de classification :

Notre sujet traite la classification des connexions TCP/IP pour la détection d'intrusions et Comme pour tous modèles de classification, notre modèle respecte les phases principales suivantes : Phase de prétraitement, phase réduction des caractéristiques (l'utilisation de l'algorithme Deep Belief Network) et phase de classement. (Avec l'utilisation de l'algorithme Réseau Naïve Bayésienne). Comme l'indique le diagramme ci-dessous. **(Figure 4.2)**.

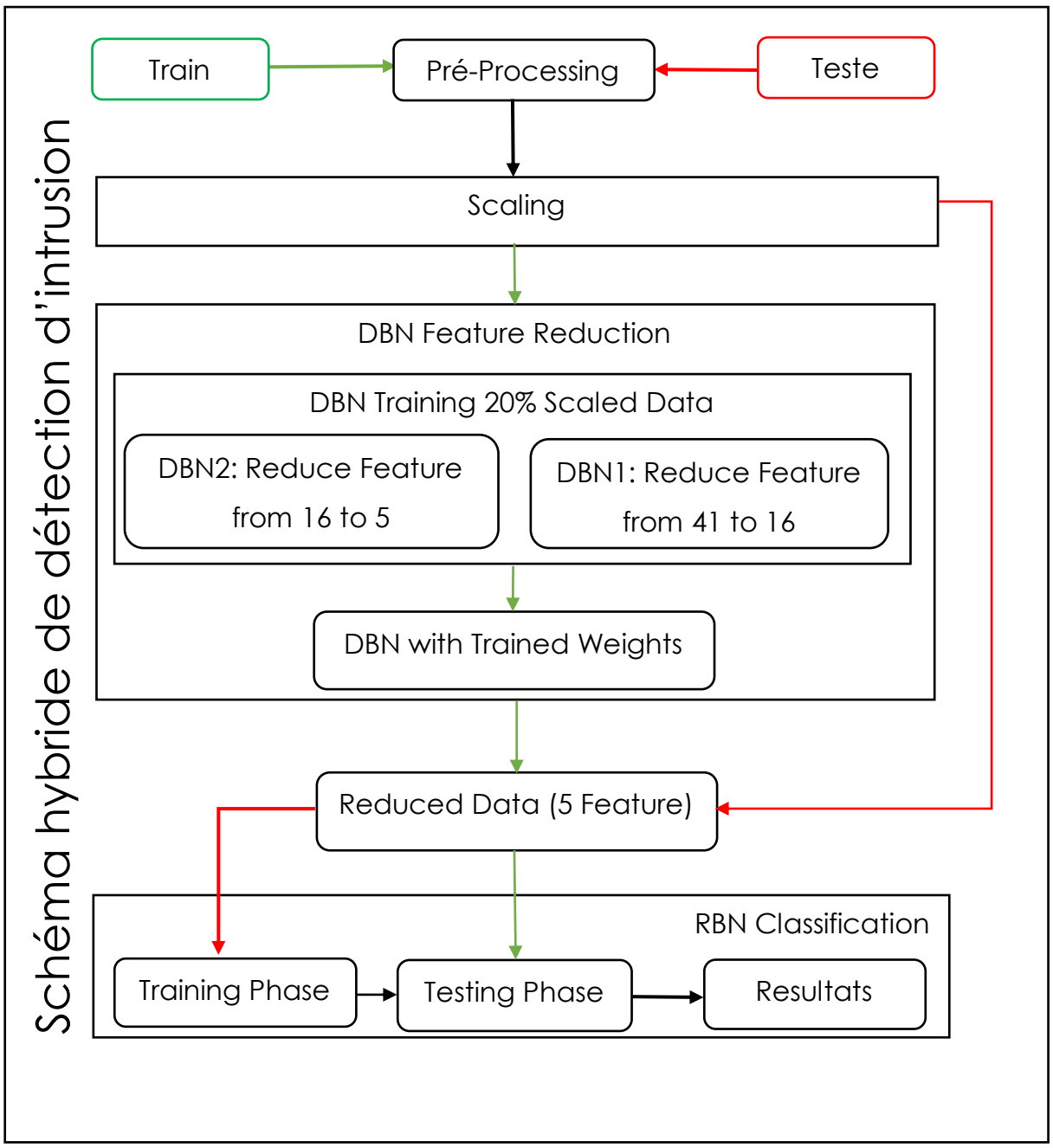


Figure 4.1 : Architecture des IDS basé sur les techniques d'apprentissage profonde

4. Prétraitement de l'ensemble de données de la base NSL-KDD

Le rôle de cette étape est de préparer les données pour qu'elles soient directement exploitables par les différents modules de traitement (apprentissage, validation et classification). Le jeu de données de traitement sont trop volumineux et consomment du temps, Et les attributs de l'ensemble de données NSL-KDD sont un mélange d'attribues continus, discrets et symboliques avec l'intervalle des valeurs très variés pour cela le

prétraitement est nécessaire .il est constitué plusieurs étapes qui s'exécutent successivement.

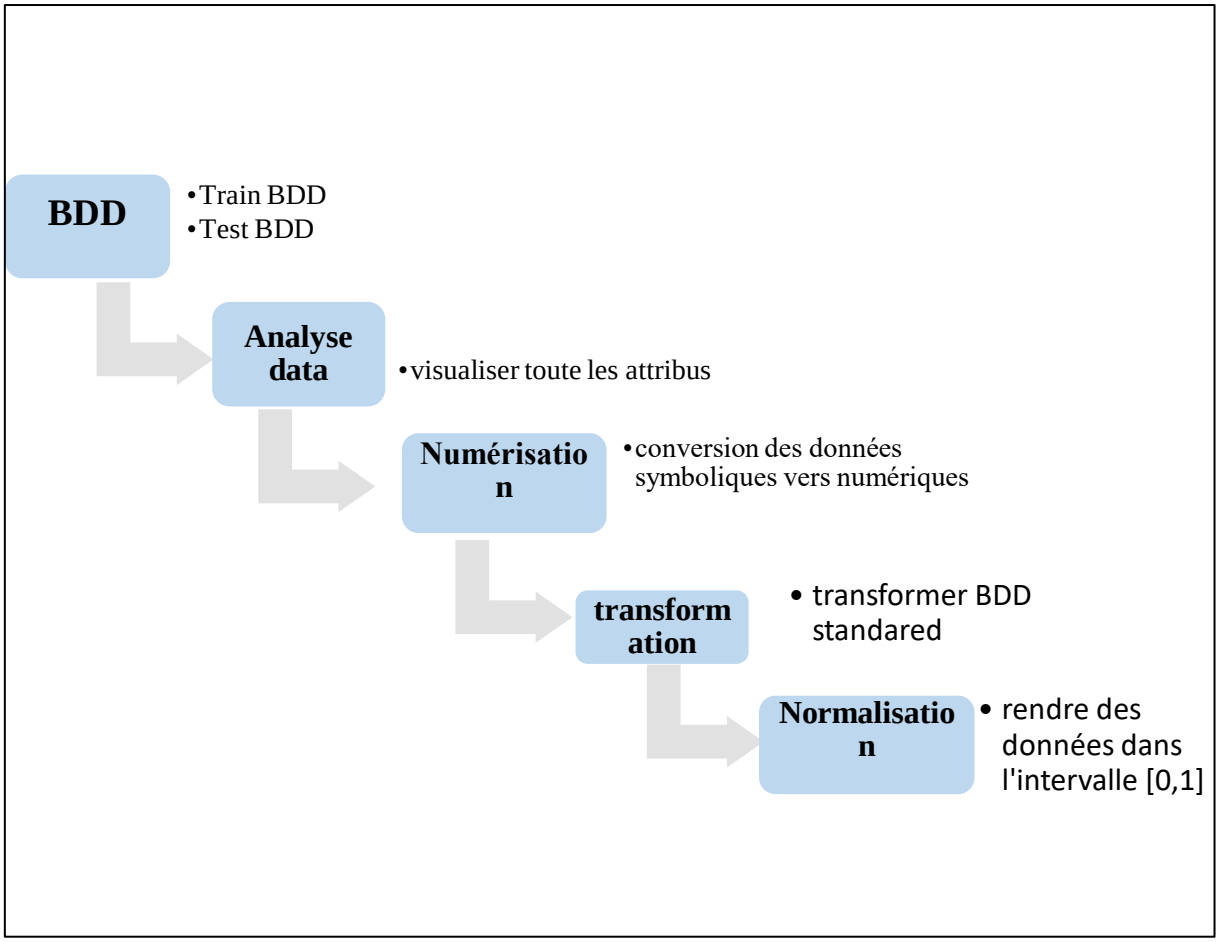


Figure 4.2 : les étapes de Prétraitement

4.1. Etape 1

La Base de données NLS KDD original

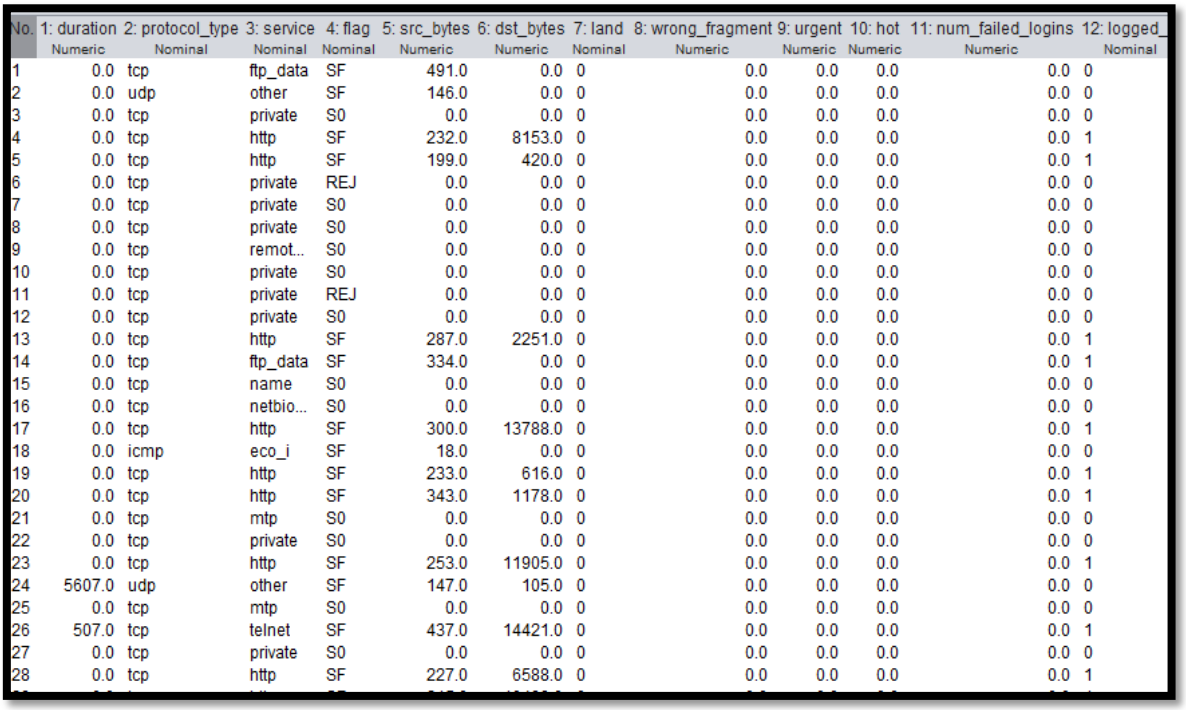
```
5]:  1 print('Dimensions of the Train set:',df.shape)
     2 print('Dimensions of the test set:',test_df.shape)

Dimensions of the Train set: (125972, 42)
Dimensions of the test set: (22543, 42)
```

Figure 4.3 : les dimensions de base de données NSL-KDD (Train, Test)

4.2. Etape 2 :

Les Données de NSL-KDD sont de trois types : numérique, Nominale et binaire. Les attributs 2, 3 et 4 sont nominaux, 7, 12, 14, 15, 21 et 22 sont binaires, et le reste des attributs sont de type numérique comme le montre le tableau (1) de l'annexe.



No.	1: duration	2: protocol_type	3: service	4: flag	5: src_bytes	6: dst_bytes	7: land	8: wrong_fragment	9: urgent	10: hot	11: num_failed_logins	12: logged_
	Numeric	Nominal	Nominal	Nominal	Numeric	Numeric	Nominal	Numeric	Numeric	Numeric	Numeric	Nominal
1	0.0	tcp	ftp_data	SF	491.0	0.0	0	0.0	0.0	0.0	0.0	0
2	0.0	udp	other	SF	146.0	0.0	0	0.0	0.0	0.0	0.0	0
3	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
4	0.0	tcp	http	SF	232.0	8153.0	0	0.0	0.0	0.0	0.0	1
5	0.0	tcp	http	SF	199.0	420.0	0	0.0	0.0	0.0	0.0	1
6	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	0.0	0.0	0
7	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
8	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
9	0.0	tcp	remot...	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
10	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
11	0.0	tcp	private	REJ	0.0	0.0	0	0.0	0.0	0.0	0.0	0
12	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
13	0.0	tcp	http	SF	287.0	2251.0	0	0.0	0.0	0.0	0.0	1
14	0.0	tcp	ftp_data	SF	334.0	0.0	0	0.0	0.0	0.0	0.0	1
15	0.0	tcp	name	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
16	0.0	tcp	netbio...	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
17	0.0	tcp	http	SF	300.0	13788.0	0	0.0	0.0	0.0	0.0	1
18	0.0	icmp	eco_i	SF	18.0	0.0	0	0.0	0.0	0.0	0.0	0
19	0.0	tcp	http	SF	233.0	616.0	0	0.0	0.0	0.0	0.0	1
20	0.0	tcp	http	SF	343.0	1178.0	0	0.0	0.0	0.0	0.0	1
21	0.0	tcp	mtp	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
22	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
23	0.0	tcp	http	SF	253.0	11905.0	0	0.0	0.0	0.0	0.0	1
24	5607.0	udp	other	SF	147.0	105.0	0	0.0	0.0	0.0	0.0	0
25	0.0	tcp	mtp	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
26	507.0	tcp	telnet	SF	437.0	14421.0	0	0.0	0.0	0.0	0.0	1
27	0.0	tcp	private	S0	0.0	0.0	0	0.0	0.0	0.0	0.0	0
28	0.0	tcp	http	SF	227.0	6588.0	0	0.0	0.0	0.0	0.0	1

Figure 4.4 : capture de chargement du corpus de train

4.3. Etape 3

Numérisation (conversion des données symboliques vers numériques), la base de données de KDD99 contient trois attributs ("protocol_type", "service" et "flag".) ont des données nominales. Sachant que les modèles de réseaux de neurones n'acceptent que des attributs numériques. Nous avons converti les trois attributs mentionnées ci-dessus vers des données numériques, Il existe plusieurs méthodes de conversion, parmi lesquelles la conversion alphabétique simple que nous avons choisi pour numériser les attributs de type nominal de la base de données. La conversion simple consiste à remplacer les valeurs des données catégoriques en ordre alphabétique par des nombres, l'attribut "type_protocole" par exemple, contiennent trois valeurs catégorielles distinctes :"tcp", "icmp" et "udp". Ces valeurs sont d'abord classées par ordre alphabétique puis on les attribuant un nombre pour chaque catégorie distinct de valeurs comme le montre **(le tableau 4)**.

Avant conversion	Apres conversion
Tcp	0
Icmp	1
udp	2

Tableau 4.3 : La Conversion alphabétique simple des valeurs de l'attribut "protocol_type"

Feature map : Attribuer des noms d'attaque à l'un des cinq classes, 0 pour normal, 1 pour Dos (Denial of Service), 2 pour U2R (User to Root), 3 pour R2L (Remote to Local), et 4 pour Probe.

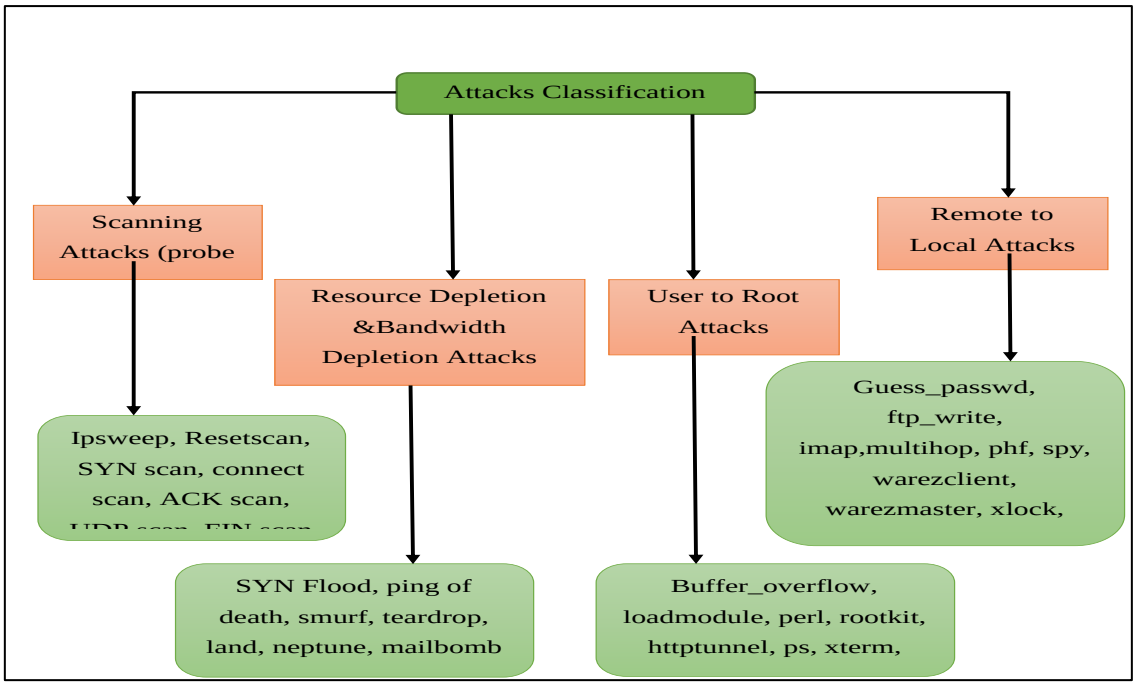


Figure 4.5 : une taxonomie de devers Attaques à base de NSL_KDD

4.4. Etape 4

Dans cette étape on a Transposé un ensemble de nombres vers un range [0,1].

4.5. Etape 5

Qui appels **« Normalisation»** Les valeurs obtenues après l'opération de la numérisation sont très variées et constituent un grand intervalle, Certains attributs prennent de grandes valeurs (src_bytes, dst_bytes, etc.), alors que d'autres ne prennent que des petites valeurs (serror_rate, same_srvrate, etc.), et cela peut nuire à la rentabilité du modèle de détection d'intrusions. Afin d'éviter ce problème et garantir l'efficacité du modèle généré, les valeurs de la base données doivent être ajustées ou normalisées, dans notre cas les données de la base NSL-KDD sont normalisés dans l'intervalle de [0, 1].

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_diff_srv_rate	dst_host_same_src_port_rat
0	0	2	44	9	146	0	0	0	0	0	...	0.60	0.8
1	0	1	49	5	0	0	0	0	0	0	...	0.05	0.0
2	0	1	24	9	232	8153	0	0	0	0	...	0.00	0.0
3	0	1	24	9	199	420	0	0	0	0	...	0.00	0.0
4	0	1	49	1	0	0	0	0	0	0	...	0.07	0.0
5	0	1	49	5	0	0	0	0	0	0	...	0.05	0.0
6	0	1	49	5	0	0	0	0	0	0	...	0.07	0.0
7	0	1	51	5	0	0	0	0	0	0	...	0.05	0.0
8	0	1	49	5	0	0	0	0	0	0	...	0.06	0.0
9	0	1	49	1	0	0	0	0	0	0	...	0.07	0.0
10	0	1	49	5	0	0	0	0	0	0	...	0.07	0.0
11	0	1	24	9	287	2251	0	0	0	0	...	0.00	0.1
12	0	1	20	9	334	0	0	0	0	0	...	0.00	1.0
13	0	1	36	5	0	0	0	0	0	0	...	0.07	0.0
14	0	1	38	5	0	0	0	0	0	0	...	0.06	0.0
15	0	1	24	9	300	13788	0	0	0	0	...	0.00	0.0
16	0	0	14	9	18	0	0	0	0	0	...	0.00	1.0
17	0	1	24	9	233	616	0	0	0	0	...	0.00	0.0
18	0	1	24	9	343	1178	0	0	0	0	...	0.00	0.0
19	0	1	35	5	0	0	0	0	0	0	...	0.05	0.0
20	0	1	49	5	0	0	0	0	0	0	...	0.06	0.0

Figure 4.6 : capture de chargement du corpus de train après la normalisation

5. Apprentissage et établissement du modèle de classification

La plupart des travaux de la détection d'intrusion utilisent les classificateurs du même niveau de façon isolée. Dans cette proposition, nous proposons une approche qui est Représentée dans un modèle de détection d'intrusion hybride et hiérarchique.

5.1. Réduction des caractéristiques avec DBN

Dans cette partie, le DBN a été utilisé comme méthode de réduction de dimensionnalité pour améliorer la sortie de données réduite. On utilisant une structure DBN doubler, la première phase **DBN 1** réduit efficacement les données (par exemple de 41 à 13 caractéristiques et le seconde de 13 caractéristiques à 5 caractéristiques de sortie basées sur les données NSL-KDD [d'après l'article de Mostafa A. Salma, Heba F. Eid, Rabie A. Ramadan, Ashraf Darwish, et Aboul Ella Hassanien])[**]

Algorithm 1: TrainsupervisedDBN (p̂, ϵ, ℓ, W, b, c Mean_Fiel_computation)

1. Entraînez un DBN de manière purement supervisée, avec la procédure gloutonne par couche dans laquelle chaque couche ajoutée est entraînée en tant que RBM (par exemple par Contrastive Divergence).
2. p̂ est la distribution d'apprentissage d'entrée pour le réseau
3. ϵ est le taux d'apprentissage pour la formation
4. ℓ est le nombre de couches à entraîner
5. W^k est la matrice de poids pour le niveau k, pour k de 1 à ℓ
6. b^k est le vecteur de décalage des unités visibles pour RBM au niveau k, pour k de 1 à ℓ
7. c^k est le vecteur de décalage des unités cachées pour RBM au niveau k, pour k de 1 à ℓ Mean_field_computation est un booléen qui est vrai données d'entraînement à chaque niveau supplémentaire obtenu par une approximation de champ moyen au lieu d'un échantillonnage stochastique.

For k = 1 to ℓ **do**

- initialize W^k =0, b^k = 0, c^k =0

While not stopping criterion **do**

- sample h^0 =x from p̂

For *i* =1 to k – 1 **do**

If Mean_field_computation **then**

- assign h_j^i to $Q(h_j^i = 1|h^{i-1})$, for all elements *j* of h^i

Else

- sample h_j^i from $Q(h_j^i = 1|h^{i-1})$, for all elements *j* of h^i

End if

End for

- RBMupdate(h^{k-1}, ϵ, W^k, b^k, c^k)

End while

End for

Algorithm 1: RBMupdate(x_1, ϵ, W, b, c)

1. Il s'agit de la procédure de mise à jour RBM pour les unités binomiales. Il peut facilement s'adapter à d'autres types d'unités.
2. x_1 est un échantillon de la distribution d'apprentissage pour le RBM.
3. ϵ est un taux d'apprentissage pour la descente de gradient stochastique dans Contrastive Divergence.
4. W est la matrice de poids RBM, de dimension (nombre d'unités cachées, nombre d'entrées)
5. b est le vecteur de décalage RBM pour les unités d'entrée.
6. C'est le vecteur de décalage.
7. Notation : $Q(h_2 = 1| x_2)$ est le vecteur d'éléments $Q(h_{2i} = 1|x_2)$.

For all hidden units i **do**

- Compute $Q(h_{1i} = 1|x_1)$ (for binomial units, $sigm(c_i + \sum_j W_{ij} x_{ij})$)
- Sample $h_{1i} \epsilon \{0,1\}$ from $Q(h_{1i} |x_1)$

End for

For all visible units j **do**

- Compute $Q(x_{2j} = 1|h_1)$ (for binomial units, $sigm(b_j + \sum_i W_{ij} h_{ij})$)
- Sample $x_{2j} \epsilon \{0,1\}$ from $P(x_{2j} = 1|h_1)$

End for

For all hidden units i **do**

- Compute $Q(h_{2i} = 1|x_2)$(for binomial units, $sigm(c_i + \sum_j W_{ij} x_{2j})$)

End for

- $W \leftarrow W + \epsilon (h_1 x'_1 - Q (h_2 = 1|x_2) x'_2)$
- $b \leftarrow b + \epsilon (x_1 - x_2)$
- $c \leftarrow c + \epsilon (h_1 - Q (h_2 = 1|x_2))$

5.2. Classification des intrusions

Les 5 caractéristiques qui résultent du DBN seront transmises au classifier bayésien pour la classification. RBN est une technique de classification basée la théorie des probabilités et de la théorie des graphes.

6. Résultats expérimentaux et discussion

Dans notre approche nous utilisent le DBN comme une méthode de la réduction dimensionnalité pour réduire le volume de la base de données NSL-KDD avant d'appliquer le l'algorithme de classification RBN.

6.1. Schéma DBN

Nous avons effectué plusieurs tests dans notre étude expérimentale en réglant à chaque fois les Paramètres d'apprentissage de réseaux de neurones (taux d'apprentissage, nombres de couches cachés, neurones par couche, les bornes d'intervalle des poids initiaux, nombres d'itérations) dont l'objectif est de trouver les valeurs de ces paramètres qui donnent les meilleurs résultats en termes du taux de réussite (accuracy), etc. Pour le nombre de couches cachées, nous avons réalisé le test sur différents valeurs, Les résultats obtenus pour les trois tests sont montrés dans les tableaux ci-dessous.

Le première Test (A)

	Learning rate	Nombre itération	Nombre components	Temps (min)	actuel
RBM1	0.01	50	256	17:07	0.87
RBM2	0.04	50	256	59:58	0.90
Moyenne / Total	/	/	/	77 :04	0,89

Tableau 4.4 : Résultat de Test A

Le deuxième Test (B)

	Learning Rate rbm	**Learning Rate ANN**	**Nombre Epochs rbm**	**Nombre itération ANN**	**Hidden layers**	**Temps (min)**	**actuel**
DBN 1	0.01	0.1	10	5	250	62:17	0.95
DBN 2	0.01	0.1	10	5	250	26:12	0.97
Moyenne	/	/	/	/	/	88,29	0,96

Tableau 4.5 : les résultats de test B

Le dernière Test (C)

	Learning Rate rbm	**Learning Rate ANN**	**Nombre Epochs rbm**	**Nombre itération ANN**	**Hidden layers**	**Temps (min)**	**actuel**
DBN 1	0.01	0.1	20	20	250	107:49	0.97
DBN 2	0.01	0.1	15	12	250	46 :40	0.83
Moyenne	/	/	/	/	/	/	0,90

Tableau 4.6 : Résultat de test C

6.2. Comparution entre les trois tests

Apres apprentissage des trois tests ont obtenus les résulte suivantes de réduction des caractéristiques :

	Test A	**Test B**	**Test C**
Itération 1	41-17	41-19	41-16
Itération 2	17-11	19-8	16-5

Tableau 4.7 : Tableau comparative entre les trois tests

Donc d'après l'analyse du **Tableau 4.7** on a vu que le dernier test (C) donne un résultat prometteur en terme d'extraction de caractéristiques, les attribues qui résulte de cette phase seront des entrés dans les prochaines étapes de classification Schéma RBN.

7. Schéma réseau bayésien naïve

Nous avons créé notre ensemble de données contenant 5 caractéristiques pour la classification. Avec un ensemble contient 20% des données utilisées pour l'apprentissage du modèle. Dans cette opération on a utilisé Environnement WEKA avec deux types de classification binaire et multiple Comme indiqué dans les graphes suivants :

7.1. Une classification binaire

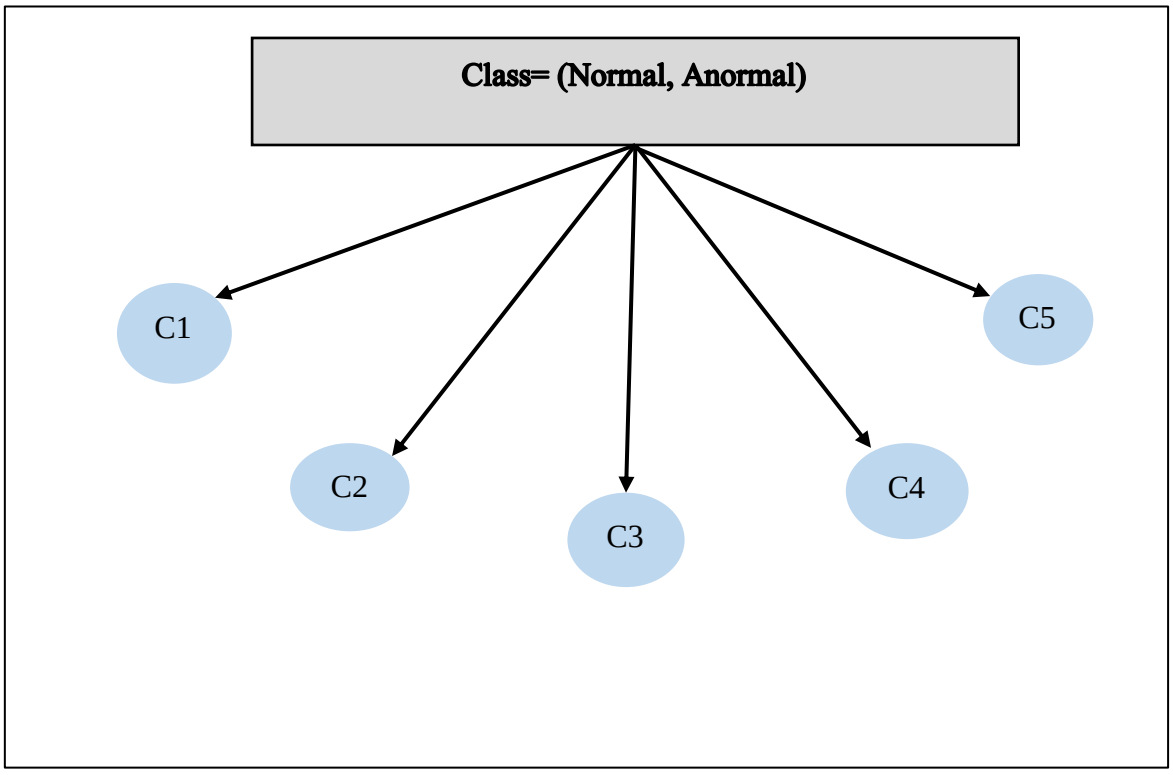


Figure 4.7 : Modèle bayésien de la classification binaire

Classification modèle

La distribution des probabilités conditionnelles des variables dans le contexte de la classe peut être calculée comme suit :

	normal	anormal
La distribution des probabilités	0.43	0,57

Tableau 4.8 : La distribution des probabilités des deux class

7.1.1. Confusion Matrix

normal	**anormal**	
9623	**88**	**normal**
45	**12787**	**anormal**

Tableau 4.9 : Matrice de confusion de classification binaire

7.1.2. Précision détaillée par classe

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
normal	0,991	0,004	0,995	0,991	0,993	0,988	0,995	0,993
anormal	0,996	0,009	0,993	0,96	0,995	0,988	0,994	0,962
Moyennes	0,994	0,007	0,994	0,994	0,994	0,988	0,994	0,972

Tableau 4.10 : Précision détaillée de classification binaire

On résume Stratification de cross-validation :

- Instances correctement classes 22410 99.41 %
- Incorrectly Classified Instances 133 0.59 %
- Erreur absolue moyenne 0.0059
- Erreur quadratique moyenne 0.075
- Erreur absolue relative 1.1986 %
- Total Number of Instances 22543

7.2. Une classification multiple

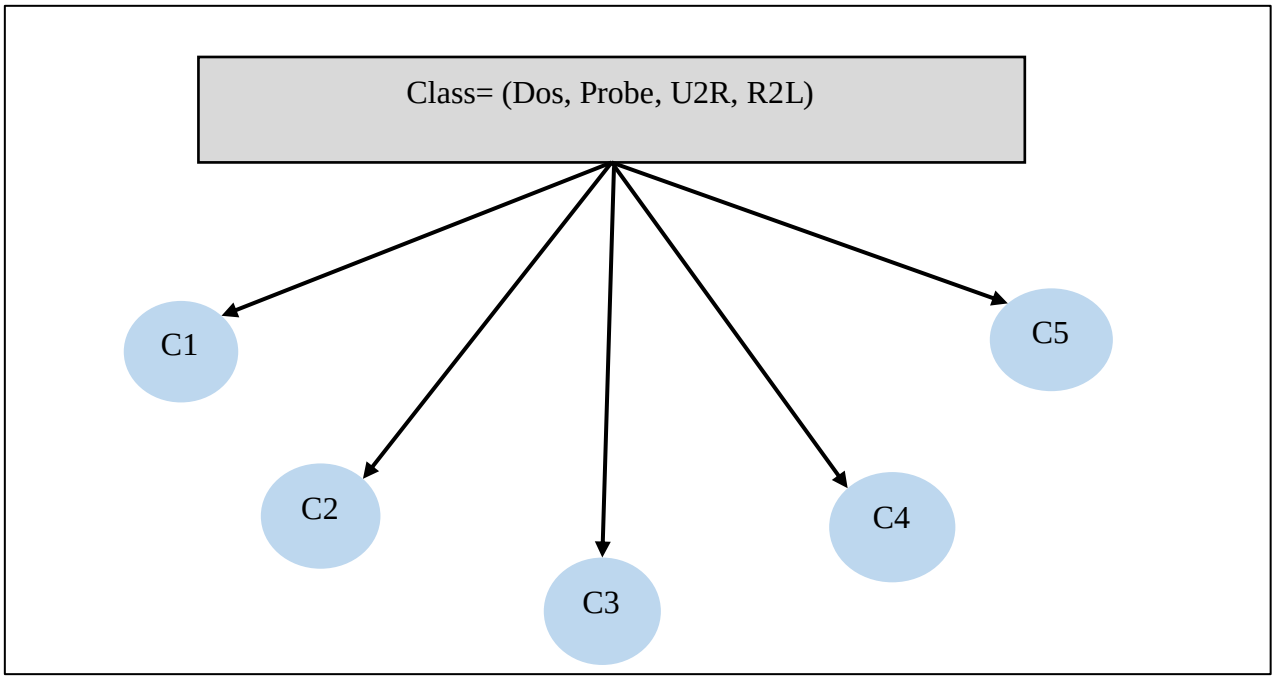
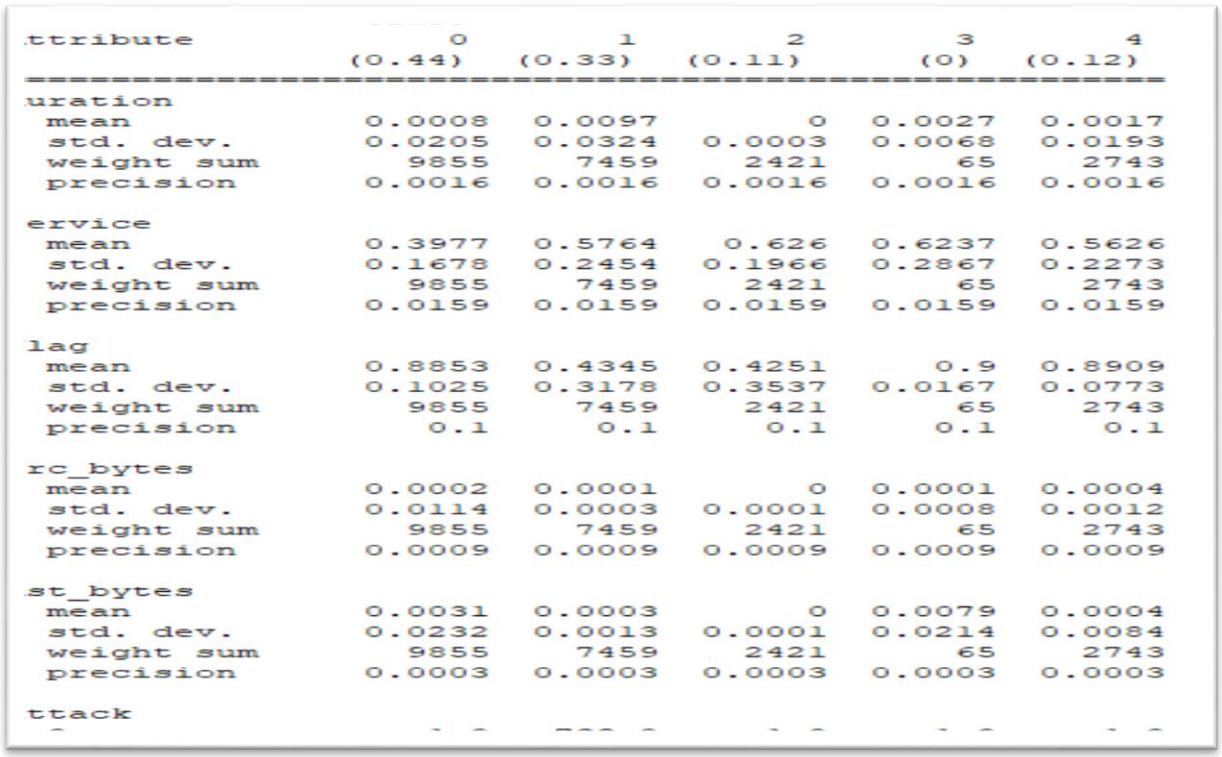


Figure 4.8 : Modèle bayésien de la classification multiple

7.2.1. Classification modèle

La distribution des probabilités conditionnelles des variables dans le contexte de la classe.



ttribute	0 (0.44)	1 (0.33)	2 (0.11)	3 (0)	4 (0.12)
uration					
mean	0.0008	0.0097	0	0.0027	0.0017
std. dev.	0.0205	0.0324	0.0003	0.0068	0.0193
weight sum	9855	7459	2421	65	2743
precision	0.0016	0.0016	0.0016	0.0016	0.0016
ervice					
mean	0.3977	0.5764	0.626	0.6237	0.5626
std. dev.	0.1678	0.2454	0.1966	0.2867	0.2273
weight sum	9855	7459	2421	65	2743
precision	0.0159	0.0159	0.0159	0.0159	0.0159
lag					
mean	0.8853	0.4345	0.4251	0.9	0.8909
std. dev.	0.1025	0.3178	0.3537	0.0167	0.0773
weight sum	9855	7459	2421	65	2743
precision	0.1	0.1	0.1	0.1	0.1
rc_bytes					
mean	0.0002	0.0001	0	0.0001	0.0004
std. dev.	0.0114	0.0003	0.0001	0.0008	0.0012
weight sum	9855	7459	2421	65	2743
precision	0.0009	0.0009	0.0009	0.0009	0.0009
st_bytes					
mean	0.0031	0.0003	0	0.0079	0.0004
std. dev.	0.0232	0.0013	0.0001	0.0214	0.0084
weight sum	9855	7459	2421	65	2743
precision	0.0003	0.0003	0.0003	0.0003	0.0003
ttack					

Figure 4.9 : Modèle bayésien de la classification multiple

7.2.2. Confusion Matrix :

normal	Dos	Probe	U2R	R2L	Class
9633	**11**	**191**	**15**	**5**	**Normal**
0	**6871**	**278**	**275**	**35**	**Dos**
0	**0**	**2399**	**19**	**3**	**Probe**
1	**0**	**8**	**49**	**7**	**U2L**
0	**4**	**611**	**7**	**2121**	**R2L**

Tableau 4.11 : Matrice de confusion de classification multiple

7.2.3. Précision détaillée par classe :

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Normal	0,977	0.000	1,000	0,977	0,989	0,980	0,991	0,993
Dos	0,921	0.001	0,998	0,921	0,958	0,940	0,996	0.988
Probe	0,991	0.054	0,688	0,991	0,812	0,802	0,995	0,992
U2R	0,754	0.014	0,134	0,754	0,228	0,314	0,953	0,312
R2L	0,773	0.003	0.977	0,773	0,863	0,854	0,986	0,950
moyenne	0,935	0,007	0,960	0,935	0,942	0,930	0,992	0,984

Tableau 4.12 : Précision détaillée de classification multiple

On résume Stratification de cross-validation :

- Correctly Classified Instances 21073 93.4791 %
- Incorrectly Classified Instances 1470 6.5209 %
- Kappa statistic 0.9046
- Mean absolute error 0.0495
- Root mean squared error 0.1611
- Relative absolute error 18.388 %
- Root relative squared error 43.9181 %
- Total Number of Instances 22543

8. Etude comparative

Tout d'abord, nous avons comparé nos résultats DBN-RBN en tant que méthode de réduction de caractéristiques et les RBN comme classificateurs avec Mostafa A. Salama DBN-SVM (20%) qui pondre DBN avec deux couche RBM pour réduire les caractéristiques et le SVM comme classificateur [38] Résumés dans le tableau (**Tableau 4.13**) suivant :

	DBN-SVM	DBN-RBN
DBN	89,63	90,06
Schéma hybride	90.06	96,00

Tableau 4.13 : Précision des performances du DBN

II. Description des différentes interfaces

Nous présentons dans cette partie du document quelques captures d'écrans issues de notre interface. Dans cette interface, nous pouvons voir la présence de 05 pages web :

1. Page home qui contient le titre de notre PFE
2. La page BDD contient les base de donnée train et test qui on utilisant pour la Rédaction et la classification
3. Page tests pour afficher les trois tests et leur résulte
4. La page RBN-classification pour afficher le résultat de classification
5. La dernière page à propre sur notre PFE

1. La page Home :

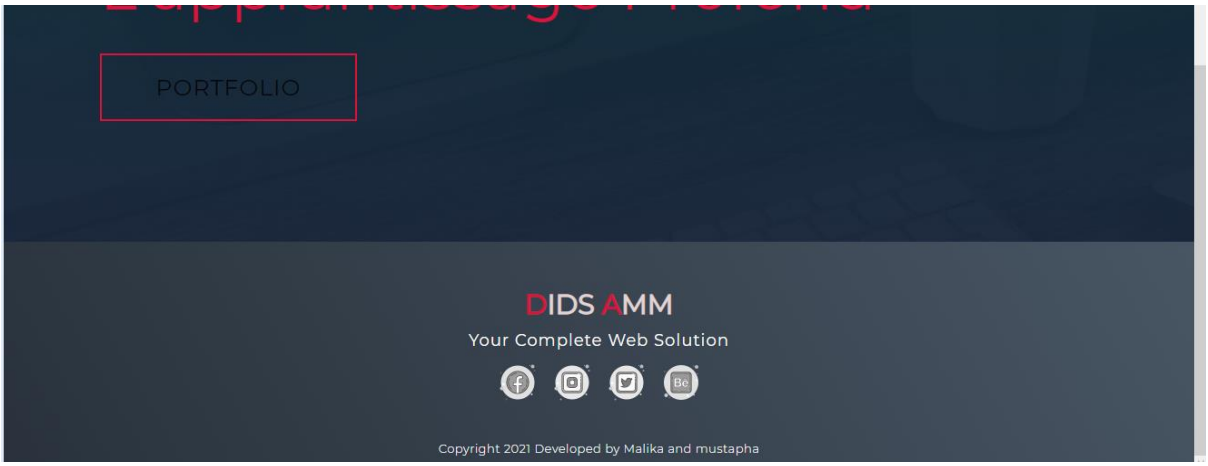


Figure .4.10 : home interface

2. La page BDD :

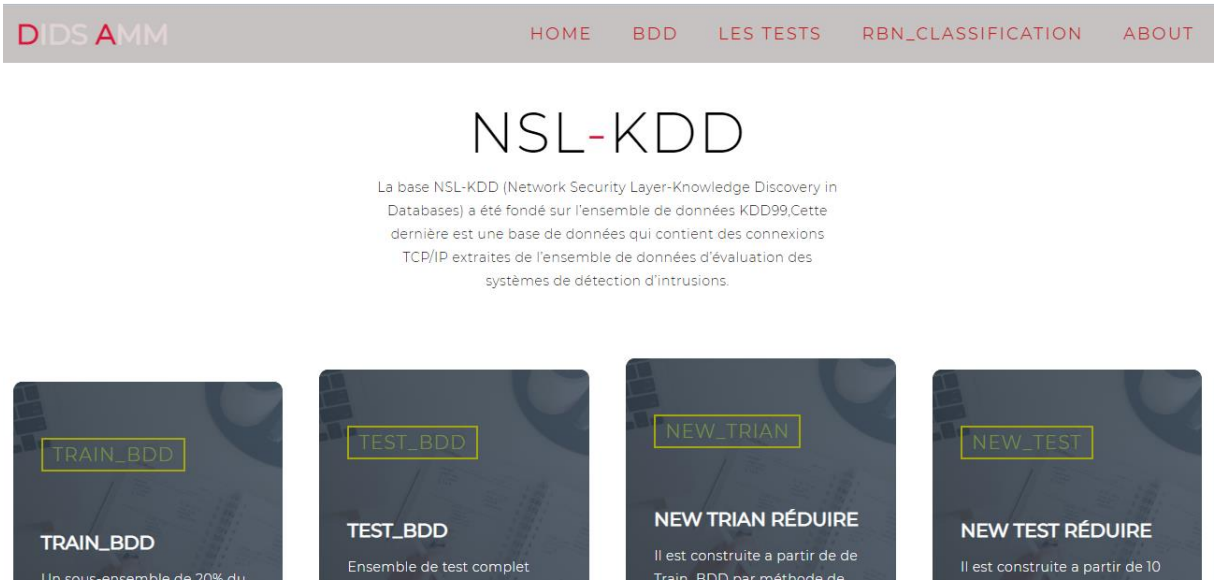


Figure 4.11 : la deuxième page qui représente les BDD

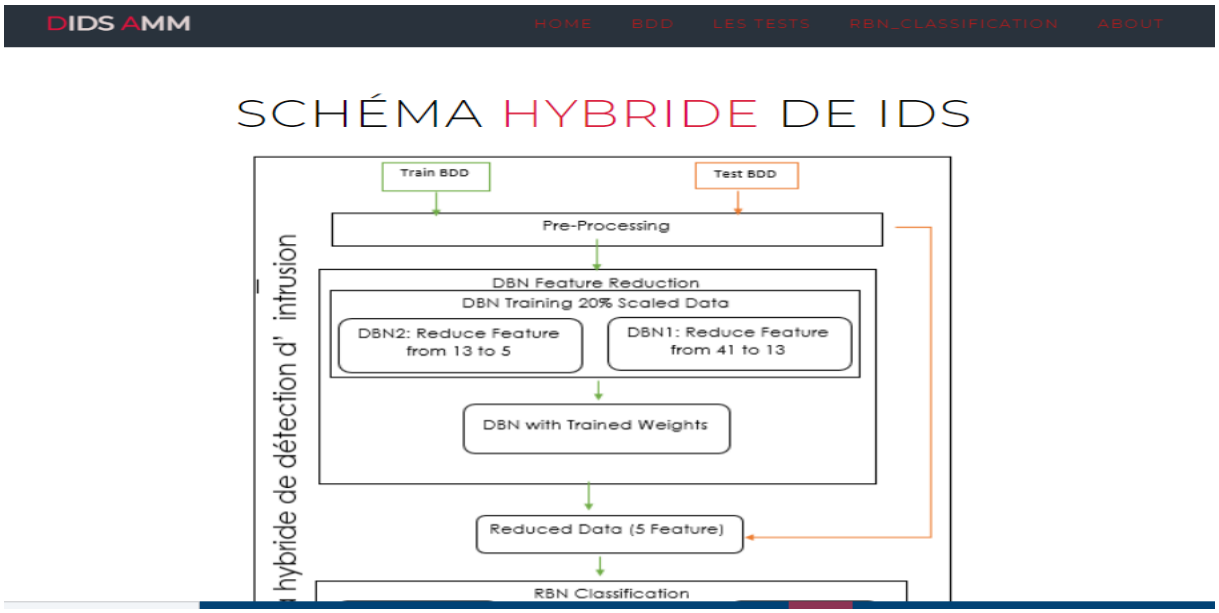


Figure 4.12 : la deuxième page qui représente schéma utilisée

3. La page Tests :

Figure 4.13 : la troisième page qui représente le test 1

Figure 4.14 : la troisième page qui représente le code source de test 1

4. La page RBN_CLASSIFICATION :

Figure 4.15 : la quatrième page qui représente la classification

5. La page ABOUT :

Figure 4.16 : la dernière page qui représente à propre de PFE

Conclusion

Dans ce chapitre, nous avons présenté une approche hybride comportementale contre les attaques TCP /IP, dans la premier partie nous avons proposé une méthode de la rédaction des dimensionnalité de base de donnée NSL-KDD, et comme une deuxième partie nous avons faites une classification binaire et multiple avec les Réseaux Bayésiennes.

La réalisation de ce travaille faite par (Python, Weka, HTML, CSS et java script) comme des outils de programmation, et pour l'affichage des résultats obtenu nous avons créé une interface web

Conclusion Général

Nous rappelons que l'objectif de notre travail est l'étude et la conception d'un modèle pour un IDS comportementale En se basant sur la technique de l'apprentissage automatique telles que les DBNs et les RBNs.

Dans la première partie de notre travail, nous avons proposé une méthode pour la réduction de la dimensionnalité de bases de données volumineuses. La réduction est faite en employant un Réseau de croyances profondes Deep Belief Network avec lequel nous avons réussi à réduire la dimensionnalité des différentes bases de données avec un taux de perte très réduit.

Le deuxième objectif visé est la classification des nouvelles instances avec beaucoup moins d'attributs, de même que la réduction, la prédiction des classes a été faite avec un Réseaux Bayésiens naïfs (RBN) pour prédire les classes des nouvelles instances issues de la phase de réduction de la dimensionnalité.

Ces méthodes ont été testées a la base de données du benchmark NSL-KDD comme source de données, et nous avons fait un modèle de classification binaire (deux classes : normale et attaque) pour classifier les connexions TCP/IP en deux classes : attaque ou normal

Le travail sur ce type de projet nous a permis d'approfondir notre connaissance dans le domaine de l'apprentissage profond, il nous a permis, aussi, de connaître et expérimenter de nouvelles bibliothèques python dédiées à ce domaine d'avant-garde et de savoir le mode de fonctionnement de plusieurs algorithmes de l'apprentissage automatique tels que les différentes architectures du Deep Learning. Bien que les objectifs visés, au préalable, ont été atteints, mais il reste toujours des perspectives et des améliorations possibles qui peuvent encore être réalisés dans le future, telles que :

- L'amélioration de ce travail en utilisant d'autres méthodes du Deep Learning et faire une étude comparative entre ces dernières au niveau des résultats, performance et rapidité.
- Penser à la création d'un modèle capable de capturer le trafic réseau en temps réel sans avoir besoin d'utiliser l'ensemble de données NSL-KDD ou autre.
- La réalisation d'un IDS basé sur notre modèle proposé.

Annexe 1

Les attributs de NSL KDD : Les détails des attributs sont répertoriés dans les tableaux suivants. **[36]**

N°	**Nom de l'attribut**	**Type**	**Description**
1	duration	Numérique	La durée de connexion
2	protocol_type	Nominal	Protocole utilisé dans la connexion (tcp, udp,icmp)
3	service	Nominal	Service réseau de destination, (http,telnet, ftp_data, etc.)
4	flag	Nominal	Statut de la connexion Normal ou Erreur (SF, REJ, S0, S1, etc.)
5	src_bytes	Numérique	Nombre d'octets de donnée transférés de la source à la destination (491, etc.)
6	dst_bytes	Numérique	Nombre d'octets de données transférés de destination à la source (0, etc.)
7	land	Binaire	**Si** l'adresse IP de source et destination et le nombre de port sont les mêmes **alors**, land=1 **sinon** land=0
8	wrong_fragment	Numérique	Nombre total de fragments erronés dans cette connexion
9	urgent	Numérique	Nombre de paquets urgents
10	Hot	Numérique	Nombre d'indicateurs «Hot »
11	num_failed_logins	Numérique	Nombre de tentatives de connexion échouées
12	logged_in	Binaire	**Si** connecté avec succès **alors** logged_in=1 **sinon** logged_in=0
13	num_compromised	Numérique	Nombre de conditions compromises
14	root_shell	Binaire	1 si le root shell est obtenu, 0autrement
15	su_attempted	Binaire	1 **si** la commande "suroot" a été tentée ou utilisée, **sinon** 0
16	num_root	Numérique	Nombre d'accès " root " ou nombre d'opérations effectuées comme racine dans la connexion

17	num_file_creations	Numérique	Nombre d'opérations de création de fichiers
18	num_shells	Numérique	Nombre d'invités du shell
19	num_access_files	Numérique	Nombre d'opérations sur les fichiers de contrôle d'accès
20	num_outbound_cmds	Numérique	Nombre de commandes sortantes dans une session FTP
21	is_host_login	Binaire	1 **si** la connexion appartient à la liste du «hot» (root ou admin) ; **sinon** 0
22	is_guest_login	Binaire	1 **si** le login est un login «guest» ; sinon 0
23	count	Numérique	Nombre de connexions vers le même hôte de destination que la connexion en cours dans les deux dernières secondes
24	srv_count	Numérique	Nombre de connexions vers le même service (N° Port) que la connexion en cours dans les deux dernières secondes
25	serror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag s0, s1, s2 ou s3, parmi les connexions agrégées dans count

26	srv_serror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag s0, s1, s2 ou s3, parmi les connexions agrégées dans srv_count
27	rerror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag REJ, parmi les connexions agrégées dans count
28	srv_rerror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag REJ, parmi les connexions agrégées dans srv_count

29	same_srv_rate	Numérique	Le pourcentage de connexions qui sont au même service, parmi les connexions agrégées dans count
30	diff_srv_rate	Numérique	Le pourcentage de connexions qui sont aux différents services, parmi les connexions agrégées dans count
31	srv_diff_host_rate	Numérique	Le pourcentage de connexions qui sont à différentes machines de destination, parmi les connexions agrégées dans srv_count
32	dst_host_count	Numérique	Nombre de connexions ayant la même adresse IP de l‟hôte de destination
33	dst_host_srv_count	Numérique	Nombre de connexions ayant le même numéro de port
34	dst_host_same_srv_rate	Numérique	Le pourcentage de connexions qui sont au même service, parmi les connexions agrégées dans dst_host_count
35	dst_host_diff_srv_rate	Numérique	Le pourcentage de connexions qui sont aux différents services, parmi les connexions agrégées dans dst_host_count
36	dst_host_same_src_port_rate	Numérique	Le pourcentage de connexions qui sont au même port de source, parmi les connexions agrégées dansdst_host_srv_count
37	dst_host_srv_diff_host_rate	Numérique	Le pourcentage de connexions qui sont à différentes machines de destination, parmi les connexions agrégées dans dst_host_srv_count
38	dst_host_serror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag s0, s1, s2 ou s3, parmi les connexions agrégées dans dst_host_count

39	dst_host_srv_serror_rate	Numérique	Le pourcentage de connexions quiont activé le flag s0, s1, s2 ou s3, parmi les connexions agrégées dans dst_host_srv_count
40	dst_host_rerror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag REJ, parmi les connexions agrégées dans dst_host_count
41	dst_host_srv_rerror_rate	Numérique	Le pourcentage de connexions qui ont activé le flag REJ, parmi les connexions agrégées dans dst_host_srv_count

Tableau1 : Les 41 attributs de la base NSL-KDD

Bibliographie

[1] Gunadiz Safia, Algorithmes d'intelligence artificielle pour la classification d'attaques réseaux à partir de données TCP, université de boumerdes, 2011.

[2] M Thibaut Probst : évaluation et analyse des mécanismes de sécurité des réseaux dans les infrastructures virtuelles de Cloud computing, 2015.

[3] Mme LABED Ines : Proposition d'un système immunitaire artificiel pour la détection d'intrusions, université de Constantine, 2006.

[4] Rodrigue Mpyana Mise en place d'un système de sécurité basé sur l'authentification dans un réseau IP. Cas de Mecelco, 2011.

[5] William Stallings, network security essentials: applications and standards fourth edition, 2011.

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

[7] Philippe Biondi, Architecture expérimentale pour la détection d'intrusions dans un système informatique, 2001.

[8] David Powell et Robert Stroud: Conceptual Model and Architecture of MAFTIA. Technical Report Series-University of Newcastle upon Tyne Computing Science, 2003.

[9] Abdelhalim Zaidi. Recherche et détection des patterns d'attaques dans les réseaux IP _a hauts débits. Réseaux et télécommunications [cs.NI]. Université d'Evry-Val d'Essonne, 2011.

[10] Nathalie Dagorn. Détection et prévention d'intrusion : présentation et limites. [Rapport de recherche], Université de Nancy1, France, 2006.

[11] SLIMANI Ahmed, Application des systèmes immunitaires artificiels à la détection d'intrusion, USTOMB : 2011

[12] Cédric Michel, Langage de description d'attaques pour la détection d'intrusions par corrélation d'événements ou d'alertes en environnement réseau hétérogène, Université de Rennes 1,2003.

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

[14] C. Llorens, L. Levier, D. Valois, B. Morin, ―Tableaux de bord de la sécurité réseau,‖ Paris, France, Editions Eyrolles, 2010.

[15] S. Martin, ―Anti-IDS Tools and Tactics,‖ SANS Technology Institute, Août 2001.

[16] C. Michel ―Langage de description d'attaques pour la détection d'intrusions par corrélation d'événements ou d'alertes en environnement réseau hétérogène,‖ Thèse de doctorat, Université de Rennes 1, Rennes, France, Déc 2003.

[17] L. Mohammadpour, T. Chaw Ling, C. Sun Liew, C. Yong Chong, ―A Convolutional Neural Network for Network Intrusion Detection System,‖ Proceedings of the APAN – Research Workshop 2018, pp. 50–55, 2018.

[18] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. A. Uusitalo, B. Timus, and M. Fallgren, ―Scenarios for 5G mobile and wireless communications: The vision of the metis project,‖ IEEE Commun. Mag., vol. 52, no. 5, pp. 26–35, Mai 2014.

[19] S. Park, H. Park, ―ANN Based Intrusion Detection Model,‖ Suisse, Springer, pp. 433–437, 2019.

[20] B. Ramsundar, R. Bosagh Zadeh, ―TensorFlow pour le Deep learning - De la régression linéaire à l'apprentissage par renforcement,‖ Paris, France, Editions First-O'Reilly, pp. 9, Oct 2018.

[21] J. R. Quinlan, C4. 5: programs for machine learning. Elsevier, 2014.

[22] C. Cortes and V. Vapnik, "Support vector machine," Machine learning, vol. 20, no. 3, pp. 273–297, 1995.

[23] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fern ´ andez, and ´ E. Vazquez, "Anomaly-based network intrusion detection: Techniques, ´systems and challenges," computers & security, vol. 28, no. 1, pp. 18–28, 2009

[25] T. Kohonen, "The self-organizing map," Proceedings of the IEEE, vol. 78, no. 9, pp. 1464–1480, 1990.

[26] P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," Computer Communications, vol. 35, no. 7, pp. 772–783, 2012

[27] Preeti Mishra, Member, IEEE, Vijay Varadharajan, Senior Member, IEEE, Uday Tupakula, Member, IEEE and Emmanuel S. Pilli, Senior Member, IEEE, "A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection ,"

[28] "Apprentissage Supervisé." https://machinelearnia.com/apprentissage-supervise-4-

etapes/?fbclid=IwAR2TI466KdZSzo8ljjvX5pob1TsojLg_UMFp1X1g6aM3BW7TPCkug2SMeE (accessed Oct. 16, 2020).

[29] "Apprentissage Non-Supervisé." https://www.lemagit.fr/definition/Apprentissagenon-supervise.

[30] Adrien Haccoun, “Comparaison de méthodes de classifications.” [Online]. Available:

[31]Weka.”https://www.cs.waikato.ac.nz/ml/weka/?fbclid=IwAR2ZaIuZyVXMv51C3YvCs7NMnuxGLJTeEoHEq67KdxXX3bU9ByLuJqxnhs (accessed Oct. 16, 2020).

[32] “Python.” https://www.journaldunet.fr/web-tech/dictionnaire-duwebmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/.

[33] “Jupyter.” https://jupyter.org/.

[34] “Pandas.” https://pandas.pydata.org/.

[35] P. Porras, D. Schnackenberg, S. Staniford-Chen, M. Stillman, and F. Wu, “The common intrusion detection framework architecture (CIDF),” Univ. Calif., 1998.

[36] L.Dhanabal& Dr. S.P. Shantharajah, «A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms», International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015.

[37] Rebecca Petersen, Data Mining for Network Intrusion Detection,Université de MID SWEDEN,2015

[38] Mostafa A. Salama ’’ Hybrid Intelligent Intrusion Detection Scheme’’

Web Bibliographie

W1 : li3, __apprantissage sur les IDS [En ligne].disponible :https://www.li3-uasz.sn/seminaire/génération-automatique-de-signatures-par-apprentissage-pour-les-systèmes-de-détection-d [Consulté le 9 avril 2021].

W2 : Wikipédia, — Apprentissage automatique IDS. [En ligne].Disponible: https://fr.wikipedia.org/wiki/Apprentissage_automatique_appliqué_aux_systèmes_de_détection. [Consulté le 9 avril 2021].

W4 : passereaux, __ Intelligence artificielle : maintenance prédictive et brevets. [En ligne].Disponible: https:/www.plass.com/fr/articles/intelligence-artificielle-maintenance-predictive-et-brevets on [Consulté le 28 avril 2021].

W5: Deeply Learning, types d’apprentissage. [En ligne].Disponible: https://deeplylearning.fr/cours_theoriques_deep_learning/les-differents-types-dapprentissage on [Consulté le 28 avril 2021].

W6: LAWTOMATED,__ A.I. Technical: Machine vs. Deep Learning. [En ligne].Disponible: https://lawtomated.com/a-i-technical-machine-vs-deep-learning on [Consulté le 28 avril 2020].