République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Ahmed Draia - Adrar

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

Option Systèmes intelligents.

Thème

Développement d'une Architecture Basée sur l'Apprentissage Profond (Deep Learning) pour la Détection d'Intrusion dans les Réseaux.

Préparé par

Mr. MIMOUNE Zakarya.

Encadré par

Mr. OUAHAB Abdelwhab.

Soutenu Publiquement le 01/07/2019.

Devant le jury

Président	**Mr CHERAGUI Mohamed Amine.**	**M. A. A**	**Univ. Adrar**
Promoteur	**Mr DEMRI Mohamed.**	**M. A. A**	**Univ. Adrar**
Examinateur	**Mr KABOU Salah El Dine.**	**M. C. B**	**Univ. Adrar**

Année Universitaire 2018/2019.

Remerciements :

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon travail et qui m'ont aidée lors de la rédaction de ce mémoire.

Je voudrais dans un premier temps remercier, mon Encadreur de mémoire Dr OUAHAB Abdelwhab, Enseignant au niveau de l'Université Ahmed Draia - Wilaya d'Adrar, Faculté des Sciences et de la Technologie

Département des Mathématique et Informatique

, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je remercie également toute l'équipe pédagogique de l'université Ahmed Draia - Wilaya d'Adrar, Faculté des Sciences et de la Technologie

Département des Mathématique et Informatique, et les intervenants professionnels responsables de ma formation, pour avoir assuré la partie théorique de celle-ci.

Je tiens à témoigner toute ma reconnaissance aux personnes suivantes, pour leur aide dans la réalisation de ce mémoire

Ma maman qui m'a beaucoup appris sur les défis à relever dans le monde de notre vie. Elle a partagé ses connaissances et expériences dans la vie, tout en m'accordant sa confiance et une large indépendance dans l'exécution de missions valorisantes, et mon Père aussi.

Et Messieurs, Les Enseignants de l'Université de Mohamed Boudiaf de la Wilaya de M'Sila, pour m'avoir accordé des entretiens et avoir répondu à mes questions sur la culture du monde des Travail, ainsi que leur expérience personnelle. Ils ont été d'un grand soutien dans l'élaboration de ce mémoire.
Mes parents, Mes Frères et mes amies, pour leur soutien constant et leurs encouragements.

Et Exceptionnellement Les Etudiants Master Informatique Promos 2018/2019.

Dédicace

Je dédié ce travail

A ma maman qui m'a soutenu et encouragé durant ces années d'études.

Qu'elle trouve ici le témoignage de ma profonde reconnaissance.

A mes Frères, et ceux qui partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail. Ils m'ont chaleursement supporté et encouragé tout au long de mon parcours.

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.

A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès.

A tous ceux que j'aime.

Table des Matières

Listes des Figures

-

Listes des Tableaux

RESUMER

Le système détection d'intrusion jouer un rôle important dans la sécurité de l'information au niveau du réseau informatique, la technologie consiste à identifier avec précision diverses attaques sur le réseau. Dans ce mémoire, nous explorons comment modéliser un système de détection d'intrusion basé sur l'apprentissage en profondeur, et les techniques d'apprentissage en profondeur sont réputées pour leur capacité à gérer des données à grande échelle ces jours-ci. Ils ont été étudiés dans diverses applications, par exemple la langue, la modélisation graphique, la parole, l'audio, la reconnaissance d'image, la vidéo, le langage naturel et les zones de traitement de signal, et nous utilisons une approche d'apprentissage en profondeur pour l'intrusion détection par l'apprentissage automatique (ML-IDS). Et, nous étudions la performance du type en apprentissage en profondeur supervisé, ainsi que, nous avons utilisé un réseau neurone convolutional (CNN) une architecture en apprentissage profond a obtenu des résultats significatifs dans le domaine de la vision par ordinateur, pour chercher sur le contenu du DataSet de référence les données normale et anormale. Les résultats expérimentaux montrent que ML-IDS convient parfaitement à la modélisation d'un modèle de classification avec une grande précision et que ses performances sont supérieures à celles des modèles de la méthode de classification par les différents types apprentissage profond, Le modèle ML-IDS, améliore la précision de la détection d'intrusion et fournit une nouvelle méthode de recherche pour la détection d'intrusion.

Mots Clés

Sécurité des Réseaux, IDS, Deep Learning, Machine Learning, Réseau de neurones et CNN, KDD99.

Introduction Générale

Internet est devenu l'outil le plus essentiel et l'une des meilleures sources d'informations sur le monde actuel. Internet peut être considéré comme l'une des composantes majeures de l'éducation et des activités commerciales. Par conséquent, les données sur Internet doivent être sécurisées. La sécurité Internet est l'une des principales préoccupations de nos jours. Étant donné qu'Internet est menacé par diverses attaques, il est essentiel de concevoir un système pour protéger ces données, ainsi que les utilisateurs qui les utilisent. Le système de détection d'intrusion (IDS) est donc une invention répondant à cette exigence. Les administrateurs réseau adaptent le système de détection d'intrusion afin d'empêcher les attaques malveillantes. Par conséquent, le système de détection d'intrusion est devenu un élément essentiel de la gestion de la sécurité. Il y a différentes façons de découvrir les anomalies. Techniques d'apprentissage profond différente sont introduits afin d'identifier les anomalies, ces dernières années, les réseaux de neurones artificiels, appelés apprentissage en profondeur, ont remporté de nombreux concours en reconnaissance de formes et en apprentissage automatique. L'apprentissage en profondeur appartient à une classe de méthodes d'apprentissage automatique, qui emploie des couches consécutives d'étapes de traitement de l'information de manière hiérarchique pour la classification des motifs et l'apprentissage des caractéristiques ou des représentations. Généralement, l'apprentissage en profondeur joue un rôle important dans les résultats de la classification des images. En outre, l'apprentissage en profondeur est également couramment utilisé pour le langage, la modélisation graphique, la reconnaissance de formes, la parole, l'audio, les images, la vidéo, le langage naturel et le traitement du signal, les progrès sur les algorithmes d'apprentissage pourraient améliorer la capacité de l'IDS à atteindre un taux de détection plus élevé et un taux de fausse alarme plus faible. Cependant, les implémentations d'apprentissage approfondi dans les applications de détection d'intrusion présentent certaines limites. Nous sommes conscients qu'il est difficile d'adopter correctement l'apprentissage en profondeur dans l'application IDS puisque les différentes approches ont été adoptées par les précédentes. Et nous utilisons des méthodes d'apprentissage automatique superviser par les réseaux de neurones conventionnels (CNN), on fin la complexité de la méthode d'apprentissage en profondeur peut être l'une des raisons.

CHAPITRE I.I SECURITE des SYSTEMES INFORMATIQUE Et IDS

Introduction

Aujourd'hui, élaborez des mécanismes pour protéger les informations. Qui sont devenus menacés, infiltrés et volés. Nous parlons d'une multiplication de programmes qui endommageront le système d'information au niveau les réseaux informatiques, poussant les utilisateurs à acheter aujourd'hui le marché des logiciels de sécurité pour identifier et réduire les risques et les menaces pesant sur la sensibilité, la vulnérabilité et l'intégration des données.

Principaux Concepts
- Risques Et Menaces.
- Sensibilité Et Vulnérabilité.
- Intégrité des Données, Authenticité des Correspondants.

I-I Sécurité des Systèmes Informatique

I-I.1 Risques et Menaces

I-I.1.1 Les Risques

A) Perte des Données Par Destruction ou Altération du Support

- dégâts dus aux éléments naturels.
- destruction volontaire.
- destruction involontaire ou désordre.
- altération du support.

B) Destruction ou Falsification due À un Traitement Erroné

- Destruction involontaire due à l'utilisateur.
- Destruction ou falsification par des tiers.
- Force externe sur l'installation.
- Traitement correct de données fausses.
- Dysfonctionnement des matérielles.
- Erreurs dans les programmes.

La vulnérabilité degré d'exposition au danger (si l'on peut facilement rentrer dedans)
La sensibilité caractère stratégique (valeur de l'information) = confidentialité.
Deux types de risques
Le risque structurel dépend de l'organisation de l'entreprise.
Le risque accidentel indépendant de tous les facteurs de l'entreprise.

Quatre niveaux de risque

- **Acceptables** pas de conséquences graves pour les utilisateurs du réseau

Ex panne électrique, perte de liaison, engorgement...

- **Courants** pas de préjudices graves au réseau, on peut réparer facilement

Ex gestion du réseau, mauvaise configuration, erreur utilisateur...

- **Majeurs** dus à des facteurs graves et qui causent de gros dégâts mais récupérables

Ex foudre qui tombe sur un routeur…

- **Inacceptables** fatals pour l'entreprise, ils peuvent entraîner son dépôt de bilan

Ex Perte ou corruption des informations importantes **[1].**

I-I.1.1.1- **Gestion des risques**

- **Vulnérabilité + menace = risque**

  - Menace cible + agent + conséquence
    - Fichier source + employé + "bug"
  - Vulnérabilité cible + agent + procédé
- Fichier source + employé + altération (In) volontaire

- **Contre-mesures**

  - Exemple Authentification + contrôle des droits de modification
    - Compromis efficacité/coût des contre- mesures
  - coût de l'incident versus coût des contre-mesures.

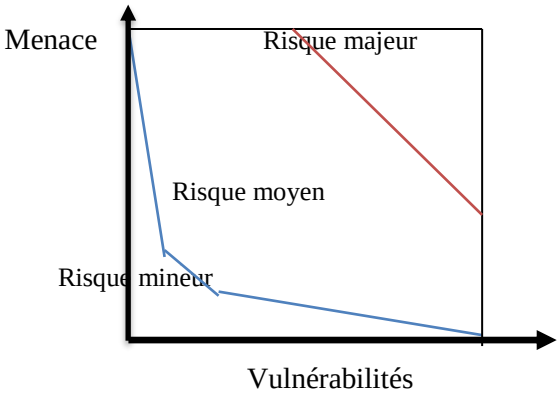


Figure I.I-1. Evolution des vulnérabilités par rapport les Menaces.

I-I.1.1.2- **Les Objets de Sécurité**

- Les informations, **l**e system -les réseaux -Etc……

I-I.1.1.3- **Attaque, Service et Mécanisme**

• **Une Attaque** n'importe quelle action qui compromet la sécurité des informations.

• **Mécanismes de Sécurité** un mécanisme qui est conçu pour détecter, prévenir et lutter contre une attaque de sécurité.

• **Service de Sécurité u**n service qui augmente la sécurité des traitements et des échanges de données d'un système. Un service de sécurité utilise un ou plusieurs mécanismes de sécurité **[2]**.

I-I.1.1.4- **Les modèles d'attaques (1)**

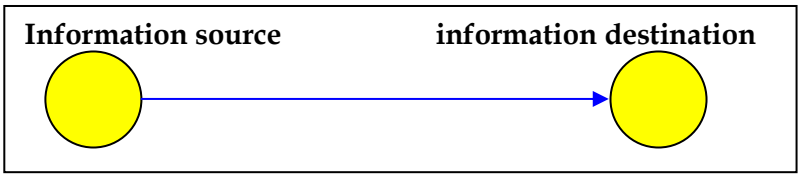


Figure I.I-2. Flux normal de transmission de l'information(a)

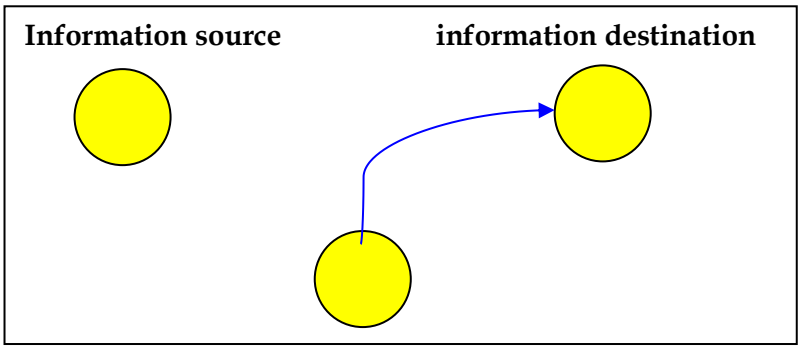


Figure I.I-3. Fabrication(b).

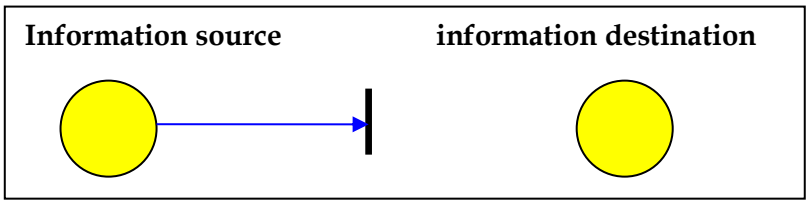


Figure I.I-4. Interruption(c).

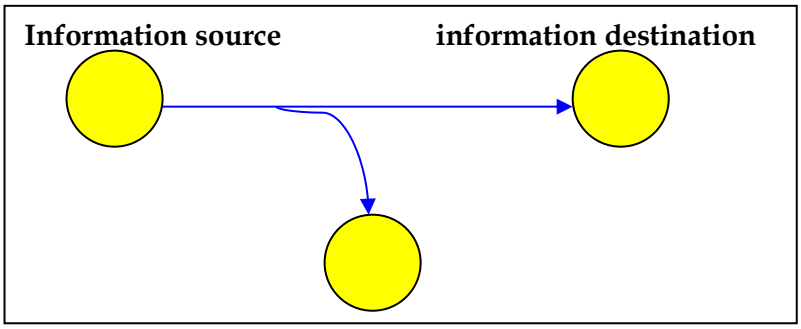


Figure I.I-5. Interception(d).

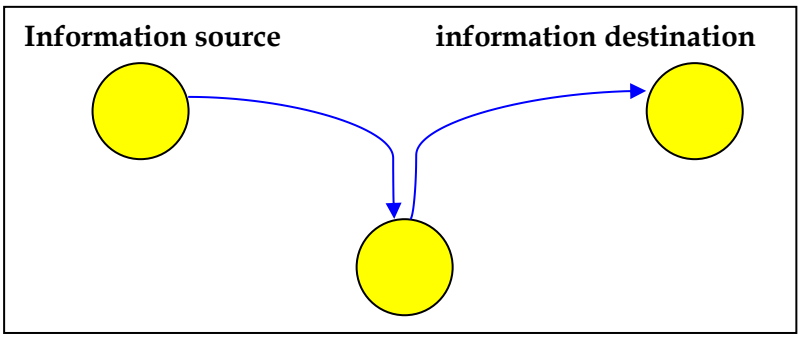


Figure I.I-6. Modification(e).

▪ **Buts des attaques**

• Interruption vise la disponibilité des informations
• Interception vise la confidentialité des informations
• Modification vise l'intégrité des informations
• Fabrication vise l'authenticité des informations.

I-1.1.5- **Analyse des risques**

- Le risque « zéro » n'existe pas
  - On peut donc
    - Ignorer les risques (Ne pas les traiter)
      - Partir du principe que « ça n'arrivera jamais »
  - Accepter les risques
    - C'est pouvoir agir afin de limiter leur portée
    - Les risques existeront toujours mais avec une portée limitée
  - Deux facteurs influencent le fait d'ignorer ou d'accepter
  - La probabilité qu'ils se produisent est vraiment très faible
  - Le coût pour limiter les risques est trop élevé

I-I.1.1.6- **Accepter les risques**

Qu'est-ce qu'un risque acceptable ?

- Un risque dont le coût de l'antidote est minime
- Un risque induit par la nature même de l'activité de l'entreprise

•Exemple Un site web marchand.

Savoir lutter contre le risque

- Éviter qu'un évènement se produise

•Interdire l'utilisation d'Internet !!!?

- Mettre en œuvre des solutions techniques

•**Firewall** par exemple.

Exemple d'analyse De Risque

- **Une inondation se produit**
  - Cela peut nuire gravement à l'activité de l'entreprise
- **Vous pouvez espérer que cela n'arrive jamais**
  - Vous ignorez alors le risque
- **Vous prenez une bonne assurance**
  - On ne traite pas le risque mais on en limite les conséquences
- **On peut installer une salle étanche**
  - On lutte contre le risque
- **Il y a toujours des risques résiduels**
  - Quelqu'un laisse la porte ouverte.
  - On prend quand même une assurance **[3].**

I-I.1.2- Les Mesure de Sécurité

- Copies de sécurité (supports différents, décentralisation, journal des opérations).
- Limitations d'accès au système et aux copies (physiques, à distance, mots de passe).
- Protection physique (eau, feu, magnétisme).
- Contrôles (consistance).
- Maîtrise des codes sources des programmes.
- Formation des utilisateurs.
- Documentation sur matériel et logiciel.
- Organisation et gestion des responsabilités.

I-I.1.3- Problème de Sécurité sur Internet

Problèmes dus à des failles notamment dans les protocoles de communication

-Toute information circulant sur Internet peut être capturée et enregistrée et/ou modifiée.

***Problème de confidentialité et d'intégrité**

-Toute personne peut falsifier son adresse IP (*spoofing*) ce qui engendre une fausse identification.

*** Problème d'authentification & Problème d'absence de traçabilité**

-Aucune preuve n'est fournie par Internet quant à la participation dans un échange électronique.

I-I.1.4- Les Menaces

Ce sont les résultantes d'actions et d'opérations du fait d'autrui. Deux catégories

- **Passives** atteinte à la confidentialité (prélèvement par copie, écoute de l'information sur les voies de communication) souvent indétectables.
- **Actives** nuisent à l'intégrité des données (brouillage, déguisement (se faire passer pour quelqu'un d'autre), interposition (vol de session)).

Niveaux de compétence *le débutant devine les mots de passe, le professionnel les décrypte.*

Statistiquement, ces menaces se répartissent de cette façon
26% accidents (incendies, inondations, pannes, catastrophes naturelles…)
17% erreurs (défauts de qualité, erreurs humaines…)
57% malveillance (d'origine interne à 80% vols des équipements, copies illicites de logiciels, sabotages + attaques logiques, intrusions et écoutes, acte de vengeance **[4].**

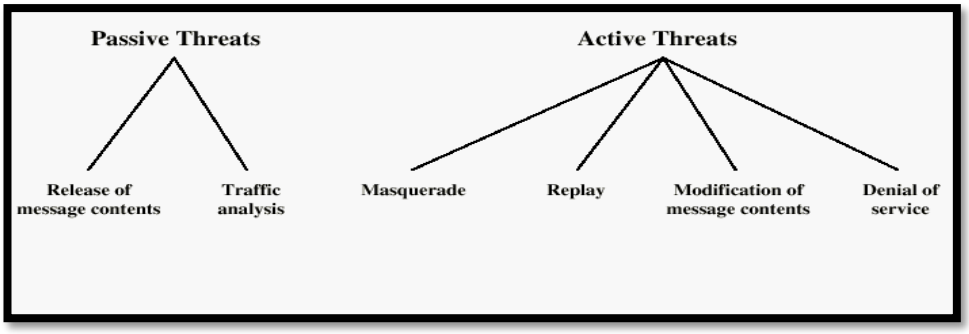


Figure I.I-7. Attaques passives ou actives.

I-I.2- Enjeux pour les entreprises

I-I.2.1- Quelque Chiffres

En 1988, le Computer Emergency Réponse Team (CERT) signalait 6 incidents de sécurité informatique En 1995, il recensait 2412 incidents et 137529 en 2003. En 1995, le Computer Emergency Réponse Team (CERT) signalait 171 vulnérabilités ; En 2000, il en recensait 1 090 et 4 129 en 2002.

Voici les différents nombre d'incidents répertoriés selon les années

Year	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003
Incidents	6	132	252	406	773	1,334	2,340	2,412	2,573	2,134	3,734	9,859	21,756	52,658	82,094	137,529

Tableau I.I-1. Différents nombre d'incidents répertoriés selon les années**.**

De nos jours les réseaux d'entreprise s'étendent de plus en plus, et le nombre de données critiques sur les SI augmente. Ce qui implique aussi une augmentation du nombre d'attaques dues à de nombreuses vulnérabilités. Il est donc nécessaire à l'heure actuelle de miser sur la détection et la prévention des attaques.

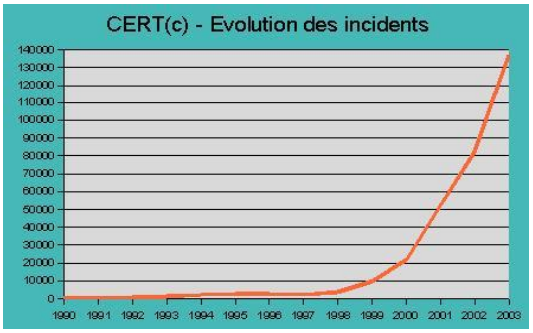


Figure I.I-8. Evolution des incidents.

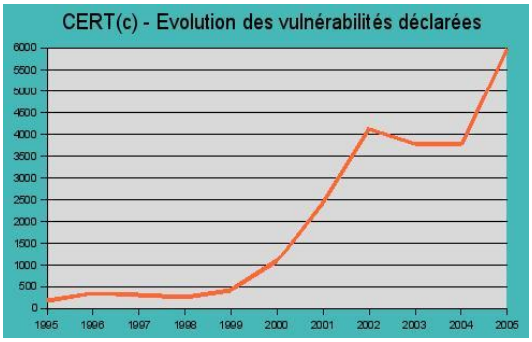


Figure I.I-9. Evolution des vulnérabilités déclarées.

Parmi les entreprises / agences gouvernementales US ayant répondu à l'enquête 2009 du Computer Security Institute (443)

- 1/3 des entreprises ont été frauduleusement représentés comme émetteurs de courriels « Phishing »;
- 19,5 % font part de fraudes financières, 64,3% d'infections par malware;
- Les pertes moyennes 2009 des entreprises ayant répondu s'élèvent à 234 244 $ (par entreprise).
- 15% font part d'accès non autorisés depuis l'interne
- 14% font part de pénétration de système depuis l'extérieur (Enquête CSI 2008)
- 25% des interrogés signalent les intrusions aux autorités légales
- Vol ou perte d'ordinateurs ou appareils mobiles 6,7m$
- Vol d'informations propriétaires 6m$ **[5].**

Voici les principaux types d'attaques que l'on peut observer

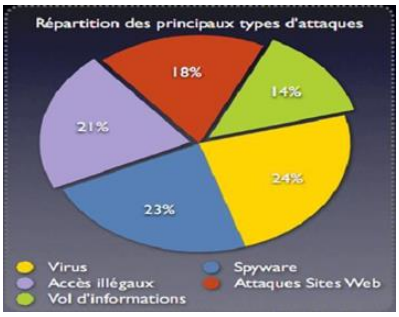


Figure I.I-10. Répartition des principaux types d'attaque.

I-I.2.2- Sécurisation des SI

La sécurité des systèmes d'informations vise donc à garantir

- **Confidentialité de l'information**
- **Intégrité de l'information**
- **Disponibilité de l'information**
- **Non répudiation de l'information.**

✓ **Confidentialité de l'information**

A pour objectif de garantir le secret de l'information, et à en garantir le périmètre de publication.

✓ **Intégrité**

L'intégrité existe quand les personnes autorisées à modifier l'information sont réellement les seules à pouvoir le faire.

✓ **Disponibilité**

La disponibilité des accès aux ressources informatiques consiste à maintenir le système opérationnel, par la mise en place de systèmes de redondance ou de plans de secours et de restauration. Ces actions d'urgence étant à prévoir suite à

- La panne du système
- L'erreur de manipulation
- La pénétration et à la destruction
- L'attaque par refus d'accès.
- Tout arrêt du SI

* Non répudiation

La non répudiation permet de certifier que lors de l'échange d'un message entre deux personnes, A et B, il s'agit bien de A qui a envoyé l'information et que c'est bien la personne B qui a reçu l'information.

La non répudiation va permettre de se prémunir contre les attaques de type Man in the Middle **[6]**.

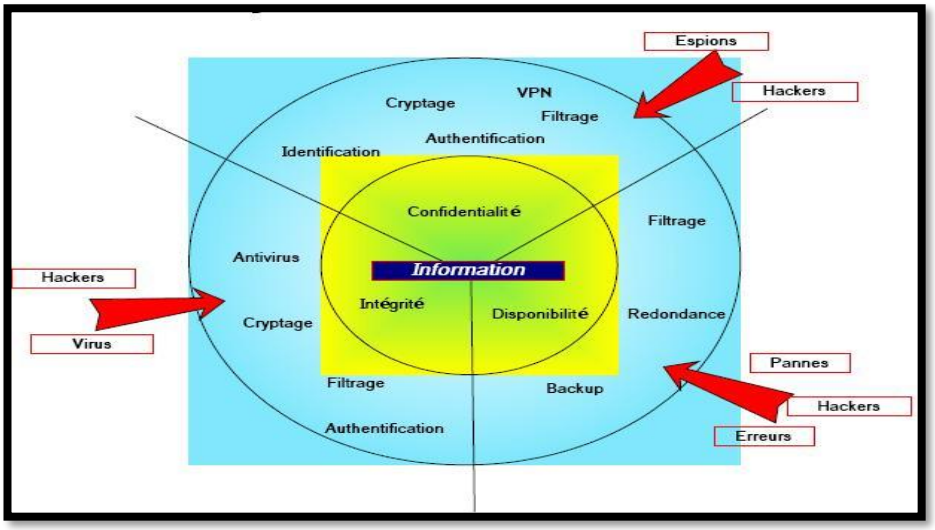


Figure I.I-11. Prémunir contre les attaques de type Man in the Middle

I-I.2.3- Se prémunir contre les attaques

Il est nécessaire pour les entreprises de protéger leur système d'information, pour diminuer la perte de capitaux **[6]**.

Le palmarès des équipements utilisés pour se prémunir contre ces attaques est représenté par le schéma suivant

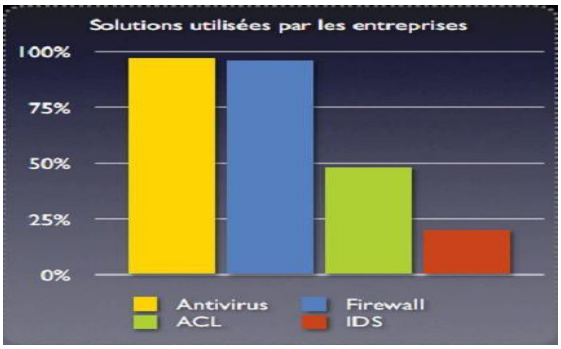


Figure I.I-12. Solutions utiliser pour l'entreprise.

I-I.3- Anatomie d'une attaque

- **Probe :** Collecte d'Information**.**
- **Penetrate :** Pénétrer le Réseau**.**
- **Persist :** Persister l'Infiltration.
- **Propagate :** Observer les autres ressources disponibles.
- **Paralyze :** Mener une attaque.

I-I.4- Type de sécurité

I-I.4.1 **Sécurité Active**

Filtrer et bloquer des flux (IPS).

Figure I.I-13. Logo Firewall.

I-I.4.2 **Sécurité Passive**

Détection/Reconnaissance d'intrusions (IDS)

Figure I.I-14. Logo Spyware.

I-II.1 Detection d'Intrusion

I-II.1.1- Presentation IDS

Les IDS, ou systèmes de détection d'intrusions, sont des systèmes software ou hardware conçus afin de pouvoir automatiser le monitoring d'événements survenant dans un réseau ou sur une machine particulière, et de pouvoir signaler à l'administrateur système, toute trace d'activité anormale sur ce dernier ou sur la machine surveillée. L'IDS est un système de détection passif.

L'administrateur décidera ou non de bloquer cette activité.

Ces systèmes de surveillance du réseau sont devenus pratiquement indispensables dû à l'incessant accroissement en nombre et en dangerosité des attaques réseaux depuis quelques années.

I.II.1.2- Principe de détection

Nous classons les IDS en deux grandes catégories de principe de détection d'intrusion

I.II.1.2.1- Approche par scénario

Les systèmes à base de signatures qui consistent à rechercher dans l'activité de l'élément surveillé les signatures (empreintes) d'attaques répertoriées et donc connues. Ce principe de détection d'intrusion est réactif et pose plusieurs contraintes, en effet il ne détecte que les attaques répertoriées dont il possède l'empreinte. De ce fait il nécessite des mises à jour fréquentes. Ce principe de détection implique aussi que les pirates peuvent contourner celui-ci en maquillant leurs attaques, il modifie en fait la signature connue par les IDS et de ce fait l'attaque devient invisible par l'IDS **[7]**.

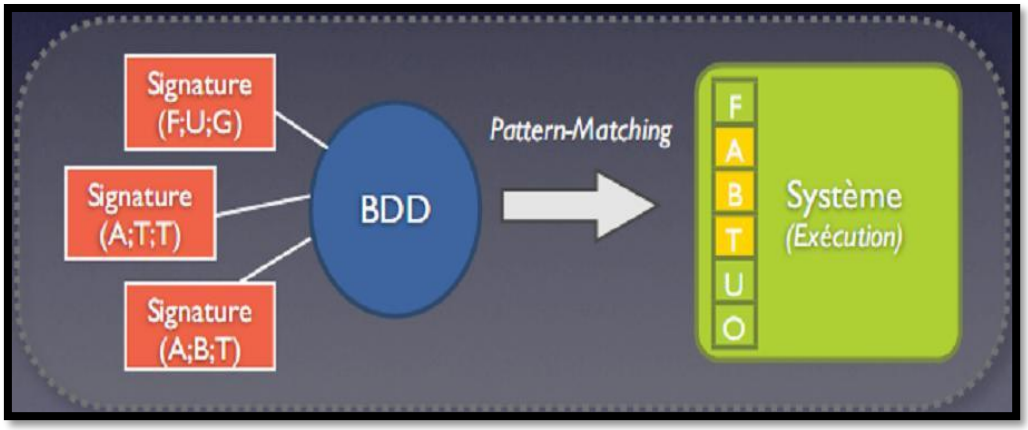


Figure I.I-15. Approche par scénario.

Il existe différentes méthodes pour repérer les attaques

- **Analyse de Motif**

  La plus simple et là plus couramment utilisée pour détecter une intrusion. Une base de connaissance contient toutes les chaînes alphanumériques caractéristiques d'une intrusion.

- **Recherches Génériques**

  Adaptée pour les virus. On regarde dans le code exécutable les commandes qui sont potentiellement dangereuses. Par exemple, une commande DOS non référencée est détectée, des émissions de mails, des instructions liées à des attaques connues.

- **Contrôle d'Intégrité**

  Effectue une photo de tous les fichiers d'un système et génère une alerte en cas d'altération de l'un des fichiers. MD-5 est le plus fréquemment utilisé mais les spécialistes recommandent maintenant le SHA-256 et SHS on signe les hash et on les met dans un coffre-fort (stocké sur un périphérique externe en lecture seule physique) Et on compare périodiquement les nouveaux hash au hash signés. Aujourd'hui l'exemple le plus connu utilisant cette approche est l'IDS SNORT.

I.II.1.3- Les différents types d'IDS

Les IDS disposent de deux approches différentes, afin de déceler les intrusions

I.II.1.3.1- Les IDS à signature

Généralement, les IDS réseaux se basent sur un ensemble de signatures qui représentent chacune le profil d'une attaque. Cette approche consiste à rechercher dans l'activité de l'élément surveillé (un flux réseau) les empreintes d'attaques connues, à l'instar de l'antivirus. Une signature est habituellement définie comme une séquence d'événements et de conditions relatant une tentative d'intrusion. La reconnaissance est alors basée sur le concept de "Pattern Matching" (analyse de chaînes de caractères présente dans le paquet, à la recherche de correspondance au sein d'une base de connaissance). Si une attaque est détectée, une alarme peut être remontée (si l'IDS est en mode actif, sinon, il se contente d'archiver l'attaque).

I.II.1.3.2- Les IDS comportementaux

Les IDS comporte aux ont pour principale fonction la détection d'anomalie. Leur déploiement nécessite une phase d'apprentissage pendant laquelle l'outil va apprendre le comportement "normal" des flux applicatifs présents sur son réseau.

Ainsi, chaque flux et son comportement habituel doivent itères déclarés ; l'IDS se chargera d'émettre une alarme, si un flux anormal est détecté, et ne pourra bien entendu, spécifier la criticité de l'éventuelle attaque.

Les IDS comportementaux sont apparus bien plus tard que les IDS à signature et ne bénéficient pas encore de leur maturité. Ainsi, l'utilisation de tels IDS peut s'avérer délicate dans le sens où les alarmes remontées contiendront une quantité importante de fausses alertes. Ce problème peut être résolu en généralisant la déclaration des flux mais cette opération peut entrainer une transparence de l'IDS face à la détection de certaines attaques **[8].**

Conclusion

Ce type de diagnostic nous donne un diagnostic clair de la menace d'intrusion, il est donc possible de réagir et de contre-attaquer, si la politique de sécurité le permet. Cependant, ils ne peuvent détecter les attaques que dans la base de connaissances. Vous devez mettre à jour cette base de données à tout moment. Il est possible de désactiver IDS en utilisant cette méthode avec une attaque par déni de service DoS.

CHAPITRE I.II IDS EN PRATIQUE

Introduction

Système de détection d'intrusion est devenu un élément important dans la structure de sécurité de votre ordinateur.

L'objectif de la sécurité informatique :

- Protection réseau connecté aux autres réseaux non sécurisés,
- Pour permettre l'accès à certains services uniquement. (Web, mail, FTP, etc..).
- Méfiez-vous des attaques et intrusions.

I-II- IDS en Pratique

I-II.1- Qu'est-ce qu'une attaque ?

- Découverte systématique d'**informations** du réseau par des **scans** de port et **balayage** du réseau.
- Tentative réelle d'intrusion dans un réseau.

I-II.2- Qu'est-ce qu'une intrusion ?

- Prise de contrôle à distance (totale ou partielle) d'un ou de plusieurs serveurs ou hôtes.
- Dans 9 cas sur 10, une intrusion est précédée d'une attaque.

I-II.2.1-Qu'est-ce qu'un IDS ?

L'IDS (Intrusion Détection System)

– Surveiller
– Contrôler
– Détecter

Justificatif

1- **Nombre de failles élevés –** 3273 nouvelles entre Janvier et Juillet 2007.
2- **Coût d'une attaque est élevé –** Code Red/Nimda est estimé à 3.2 Milliards $ par Computer Economics.
3- **Les outils pour lancer les attaques sont facilement disponibles et exploitables.**

I-II.2.2-De quoi est constitué un IDS?

-Un IDS est essentiellement un sniffer couplé avec un moteur qui analyse le trafic selon des règles.

- Ces règles décrivent un trafic à signaler.

- L'IDS peut analyser

- Couche Réseau (IP, ICMP).
- Couche Transport (TCP, UDP).
- Couche Application (HTTP, Telnet).

- Selon le type de trafic, l'IDS accomplit certaines actions.

I-II.2.3- Qu'est un Internet Détection System (IDS) ?

- C'est un système qui détecte (tente de détecter) les intrusions.
- C'est un processus de découverte et d'analyse de comportements hostiles dirigé contre un réseau.

I-II.2.4- Pourquoi un IDS (1) ?

- **La sécurité active n'est pas suffisante**
  - Les pare-feu ne contrent pas toutes les menaces.
  - Innovation constante des techniques d'hacking.
  - Faille potentielle selon les fonctionnalités des systèmes.
  - Failles inhérentes de certains OS **[9]**.

I-II.2.5-Pourquoi un IDS (2) ?

- Remonter la source de l'attaque.
- Détecte les techniques employées.
- En cas d'intrusion, les traces sont preuves tangibles **[10]**.

I-II.3- Avenir et L'Actions d'un IDS

I-II.3.1- Avenir des IDS

- LIDS (Linux IDS).
- Evolution de Snort.
- Port sentry, Shadow.

I-II.3.2- L'Actions d'un IDS

- Journaliser l'événement - Source d'information et vision des menaces courantes
- Avertir un système avec un message
  - Exemple appel SNMP.
- Avertir un humain avec un message
  - Courrier électronique, SMS, interface web, etc.
- Amorcer certaines actions sur un réseau ou hôte
  - Exemple mettre fin à une connexion réseau, ralentir le débit des connexions, etc. (rôle actif) **[11]**.

I-II.4- HIDS & HIPS

Les IDS peuvent se classer en deux catégories qui correspondent à ce que s'attache à surveiller l'IDS.

I-II.4.1- Network based IDS (NIDS)

L'IDS réseau ou Network based IDS (NIDS) surveille comme son nom l'indique 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.

On notera une contrainte à ce système, en effet le cryptage du trafic sur les réseaux commutés rend de plus en plus difficile l' "écoute" et donc l'analyse du segment réseau à analyser, car le contenu des paquets est crypté. De plus, un trafic en constante augmentation sur les réseaux contraint les NIDS à être de plus en plus performants pour analyser le trafic en temps réel.

Enfin, avec sa ou ses cartes d'interface réseau en mode promiscues (permet à celle-ci d'accepter tous les paquets qu'elle reçoit, même si ceux-ci ne lui sont pas adressés.), qui n'ont donc pas d'adresses IP, ni de pile de protocole attaché, il peut écouter tout le trafic qui arrive à l'interface en restant invisible **[12]**.

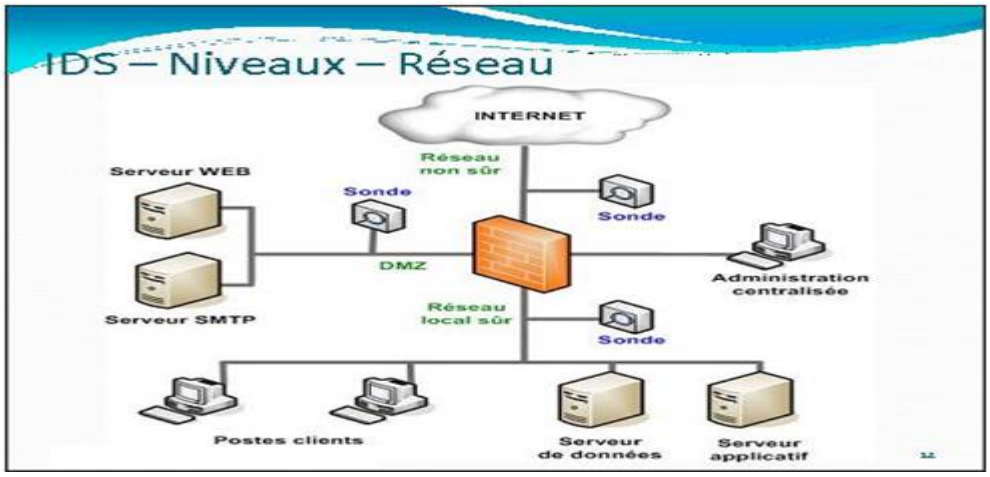


Figure I.II-1. Placement de la sonde sur le réseau (niveaux-réseaux).

On peut placer les NIDS à différents endroits sur le réseau, mais bien sûr la politique de sécurité menée définira leur emplacement.

- On peut les mettre par exemple dans la zone démilitarisée ou DMZ afin de contrer les attaques contre les systèmes publics.
- On peut les mettre aussi derrière un firewall donnant accès à l'extérieur du réseau (Internet) afin de détecter des signes d'attaque parmi le trafic entrant/sortant du réseau.

- Prendre l'option de mettre l'IDS derrière le firewall permet de réduire les paquets à analyser par celui-ci; le firewall bloque le plus "gros " du trafic et décongestionne par ce fait l'IDS et le travail de l'administrateur. Les logs se trouvent sinon trop "parasités" et rendent difficile la détection de vrai risque.
- Derrière le firewall, il y a deux positions possibles pour la sonde
  - En coupure.

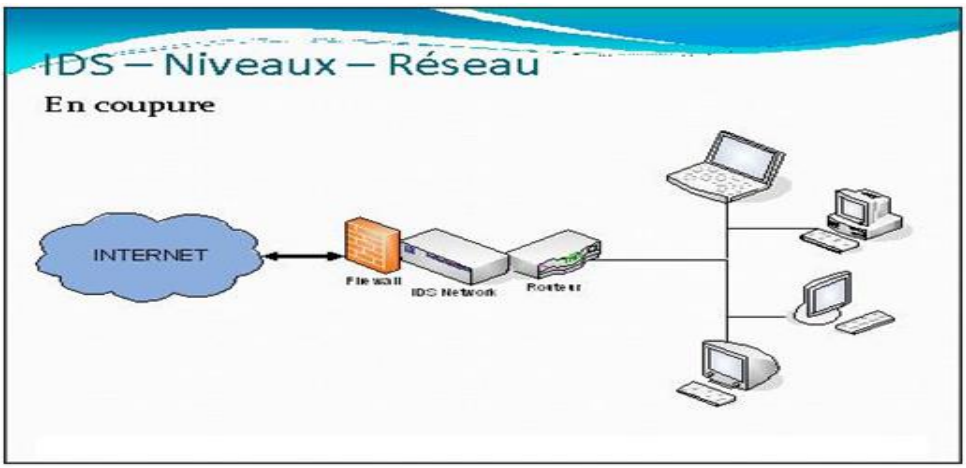


Figure I.II-2. Placement de la sonde sur le réseau (niveaux-réseaux en coupure).

Ici il a une faiblesse d'architecture, si la sonde tombe (par exemple à cause d'une attaque de dénis de service) c'est tout le réseau qui tombe.

- **En recopie de port**

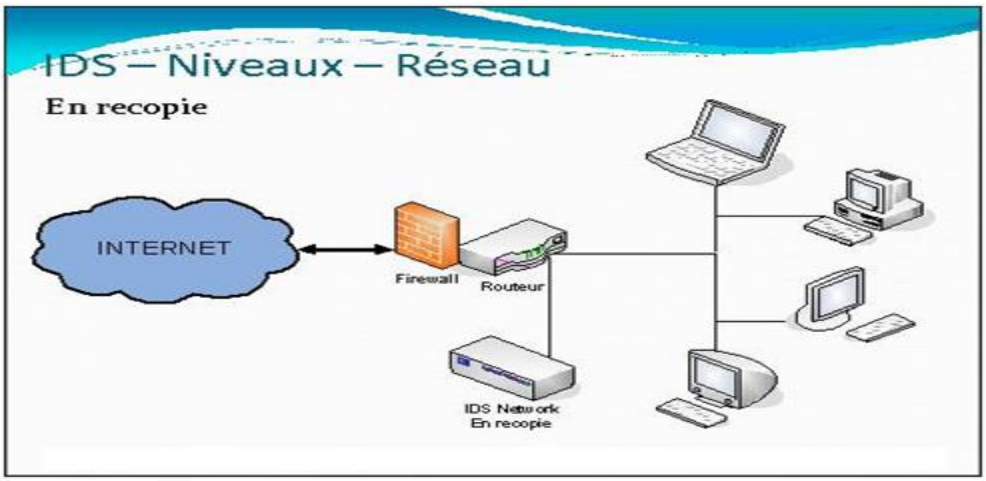


Figure I.II-3. Placement de la sonde sur le réseau (niveaux-réseaux en recopie).

Ici la sonde analyse aussi bien le réseau qu'en mode coupure sauf que si elle tombe due à une attaque cela ne pose aucun problème à l'architecture réseau. Et la sonde étant passive cette solution est la meilleure **[13]**.

I-II.4.2- Host Based IDS (HIDS)

L'IDS Systèmes ou Host Based IDS (HIDS) surveille le trafic sur une seule machine. Il analyse les journaux systèmes, les appels systèmes et enfin vérifie l'intégrité des systèmes de fichiers.

Les HIDS sont de par leur principe de fonctionnement dépendant du système sur lequel ils sont installés. Ce système peut s'appuyer ou non sur le système propre au système d'exploitation pour en vérifier l'intégrité et générer des alertes.

Il peut aussi capturer les paquets réseaux entrant/sortant de l'hôte pour y déceler des signaux d'intrusions (Déni de Services, Backdoors, chevaux de Troie, tentatives d'accès non autorisés, exécution de codes malicieux, attaques par débordement de buffers…). Il permet

- Détection de compromission de fichiers (contrôle d'intégrité)
- Analyse de la base de registre (Windows) ou des LKMs (Linux)
- Analyse et corrélation de logs en provenance de firewalls hétérogènes
- Analyse des flux cryptés (ce que ne peut réaliser un NIDS !).

L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levées. Par nature, ces IDS sont limités et ne peuvent détecter les attaques provenant des couches réseaux.

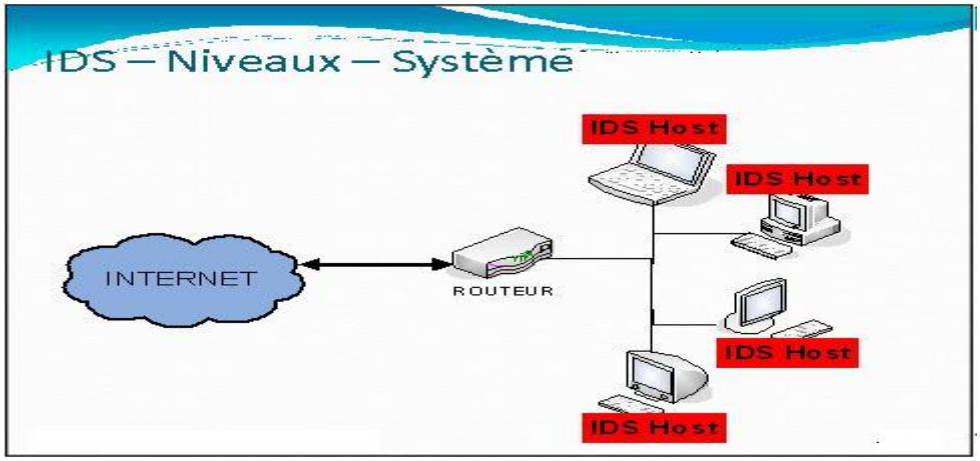


Figure I.II-4. Placement de la sonde sur le réseau (niveaux-système).

Récapitulatif

- **Network based IDS (NIDS)**
  - **Positif**
    -N'affecte pas les performances du réseau.
    - N'est pas visible
  - **Négatif**
    - Faible devant les attaques de dénis de services
    - Un point unique de défaillance
- **Host Based IDS (HIDS)**
  - **Positif**
    - Surveille les intrusions qui s'appliquent uniquement à l'hôte
  - **Négatif**
    - Besoin de HIDS spécifique pour des systèmes spécifiques.
    - Utilise la ressource du système **[14]**.

I-II.5- Type de réponses aux attaques

On a vu en quoi consistait une attaque, ainsi qu'une intrusion au sein d'un SI. D'autre part, les enjeux de se prémunir contre ces intentions de nuire peuvent s'exprimer en milliers de dollars par année pour une entreprise.

La DSI (Direction des Systèmes d'Information) d'un établissement pourra mettre en place une politique de sécurité pour faire face à ces dangers. Deux politiques (pouvant être complémentaire) seront alors mises en œuvre

I-II.5.1- Réponse Active

Les réponses actives consistent à répondre directement à une attaque, la plupart du temps en gén01érant des requêtes de fin de connexion vers la source de manière à la contraindre à cesser son activité intrusive sur le champ.

Dans le cas de données TCP, ceci se traduit par l'envoi de paquets RST qui marquent la fin d'une session aussi bien vers la source que la destination. Dans le cas des protocoles ICMP ou UDP qui n'implémentent pas de machines d'états, il peut s'avérer plus complexe de marquer une fin de session dans la mesure où la notion même de session n'existe pas. Une méthode couramment utilisée consiste à générer des requêtes ICMP Network Unreachable ou UDP Port Unreachable en espérant que la source reçoive ces requêtes et cesse d'émettre.

Si ces techniques permettent d'interrompre le flux intrusif, elles présentent toute fois des inconvénients majeurs.

Le fait de générer des paquets de réponse à une intrusion peut fournir à l'attaquant d'éventuelles informations révélant la présence d'un système de protection actif, tel un IPS.

L'IPS utilise comme je l'ai dit ci-dessus la génération de paquets pour couper la connexion

- TCP Reset.
- ICMP Network Unreachable (inaccessible).
- UDP Port Unreachable.
- Drop.

I-II.5.2- Réponse Passive

Cette technique consiste à bloquer les flux associés à une activité intrusive sans en informer la source, c'est-à-dire sans générer de paquets spécifiques à destination du pirate. Les réponses passives se traduisent la plupart du temps par des opérations de reconfiguration automatique d'un firewall (Snort(NIDS) le fait avec packet filtrer (firewall)) afin de bloquer les adresses IP source impliquées dans les intrusions.

Le problème n'est pas le même que la réponse active, n'ayant aucune action vis à vis de l'attaquant celui-ci n'est pas aux courant de la présence de l'IPS. En revanche le problème de l'authenticité de la source de l'attaque est le même. Avec un firewall on a aussi la possibilité de ce coupé d'un réseau important.

En effet, si le pirate usurpe une adresse IP sensible telle qu'un routeur d'accès ou un serveur DNS, l'entreprise qui implémente une reconfiguration systématique d'un firewall risque tout simplement de se couper elle-même du monde extérieur.

- **Exemple de problème réponse Active**

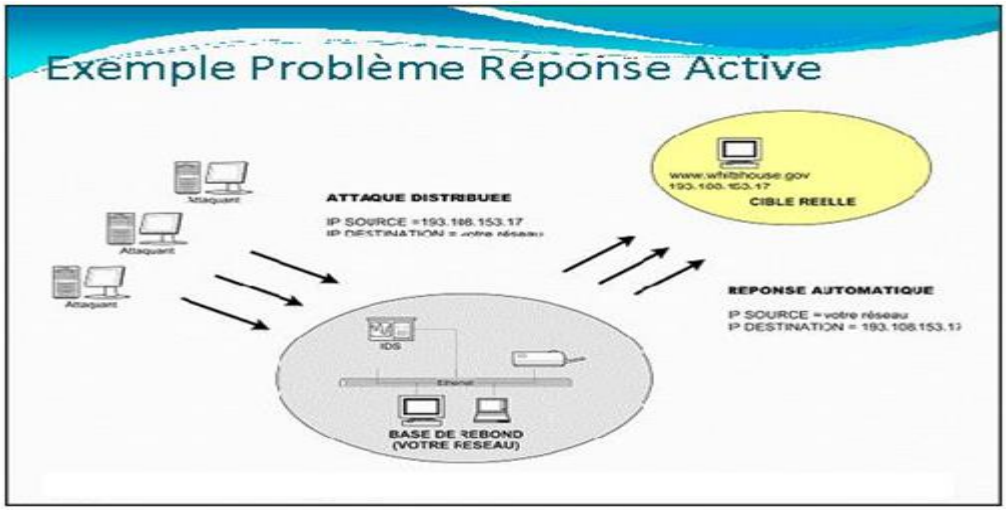


Figure I.II-5. Exemple de problème réponse Active**.**

I-II.6- Complémentarité & Contournement des IDS

- NIDS détecte-les (tentatives) d'intrusions grâce à une base de signatures
- Mais il n'ARRETE pas l'intrusion !!
- Capture les infos et les techniques de l'attaquant
- Aide indispensable pour se prémunir des intrusions en prenant les mesures de sécurité adéquates **[15]**.

I-II.6.1- Quelques techniques

I-II.6.1.1- **Déni de service contre un IDS**

Un pirate pourra essayer d'effectuer un DoS sur l'IDS pour qu'il ne puisse plus remplir pleinement son rôle.

I-II.6.1.1.1- **Attaque par insertion**

Il s'agit là de l'ajout de paquets superflus

Ex fragmentation IP et recouvrement de fragments (modification des champs « longueur » et « décalage »).

Ex Exploiter le Timeout pour le réassemblage (~60s sur machines, <60s sur IDS)

I-II.6.1.1.2- **Attaque par évasion**

Cette technique a pour but de ne pas faire détecter un paquet par l'IDS.

Ex modification des chaînes de caractères

GET /etc/rc.d/././././/.//./passwd

Ou

GET %65%74%63/%70%61%73%73%77%64 (codage en hexadécimal).

Où

URL longues, Remplacer espaces par tabulations, '/' par '\' etc.

I-II.6.1.1.3- **Shellcode**

Il s'agit là d'une technique permettra de faire du débordement de la pile. En général, un pirate utilisera plusieurs instructions assembleur NOP pour pouvoir atteindre des zones mémoires. Il pourra alors remplacer les instructions NOP par d'autres instructions n'altérant pas le fonctionnement de son programme, et qui ne seront pas détectées **[16]**.

I-II.7- Etude d'un NIDS Basic TCP-DUMP

Un NIDS de base Tcp dump.

- Outil de base indispensable pour l'analyse réseau.
- Disponible sur toutes les plateformes UNIX.
- Permet l'écriture de filtres simples pour surveiller des types de trafic.
- Exemple de mise en situation dans le scan suivant

Exemple simple de scan

¦Starting nmap V. 2.53 by fyodor@insecure.org (www.insecure.org/nmap/)

- **¦Interesting ports on (192.168.1.30)**
- **¦Port State Service**
- **¦53/udp open domain**
- **¦**
- **¦Nmap run completed -- 1 IP address (1 host up) scanned in 1 second**

Exemple commande

Tcpdump host 192.168.1.30 and ip[9]=17,

Qui signifie

« Affiche moi sur la sortie standard tous les datagrammes dont le 9ème octet est à 17, protocole UDP (cf. Datagramme IP) » **[17]**.

I-II.8- Étude des fonctionnalités se Snort(NIDS) (gratuite)

Snort est un projet Open Source de détection d'intrusion sur le réseau open-source fonctionnant sur les systèmes Windows et Linux. Capable d'analyser en temps réel le trafic et de consigner le transit des paquets de données sur le réseau IP, il peut réaliser une analyse de protocole, une recherche sur le contenu et peut être utilisé pour détecter un nombre important d'attaques réseau connues et permet ainsi d'alerter quant aux tentatives d'intrusion sur votre réseau.

Figure I.I-6. Logo Snort.

Vous pouvez télécharger **Snort**, paquet source ou binaire, à partir du site officiel

http//www.snort.org/

Exemples de NIDS Snort (IDPS, le plus répandu), OpenSnort, Shadow, etc.

I-II.8.1- Mise en place d'un IDS
Où positionner son IDS ??

Il existe plusieurs endroits stratégiques où il convient de placer un IDS.
Le schéma ma suivant illustre un réseau local ainsi que les trois positions que peut y prendre un IDS

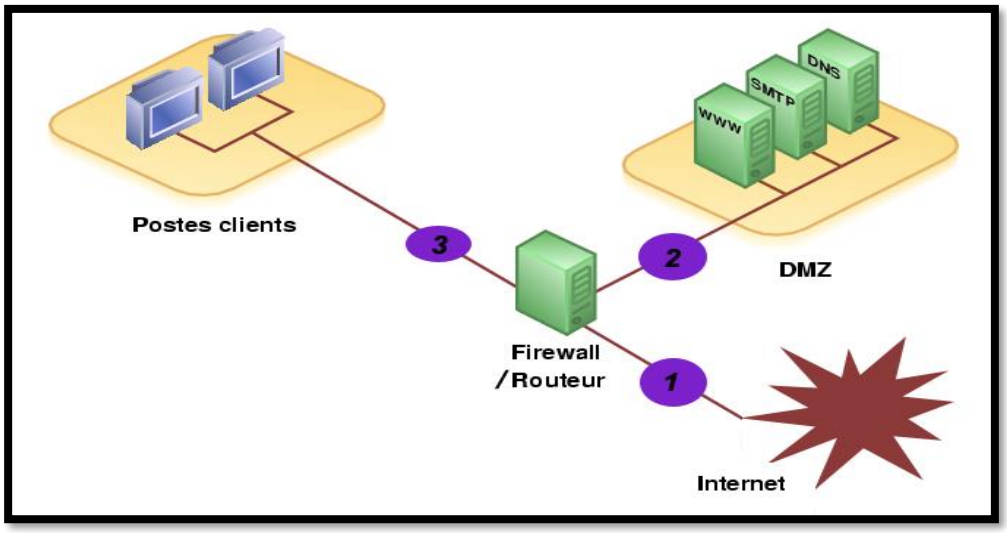


Figure I.II-7. Positionnement du NIDS au sein du réseau.

- **Position (1)** Sur cette position, l'IDS va pouvoir détecter l'ensemble des attaques frontales, provenant de l'extérieur, en amont du firewall. Ainsi, beaucoup (trop?) d'alertes seront remontées ce qui rendra les logs difficilement consultables.
- **Position (2)** Si l'IDS est placé sur la DMZ, il détectera 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 bénines ne seront pas recensées.
- **Position (3)** L'IDS peut ici 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. De plus, si des trojans ont contaminé le parc informatique (navigation peu méfiante sur internet) ils pourront être ici facilement identifiés pour être ensuite éradiqués **[18]**.

I-II.8.2- Lancement de Snort

Snort dispose de plusieurs modes de fonctionnements qui sont les suivants

- **Mode écoute** Ce mode permet de lancer snort en mode sniffer et permet d'observer les paquets que l'IDS perçoit ("snort -v")
- **Mode "log de paquets"** Le log de paquet permet l'archivage des paquets circulant sur le réseau de l'IDS. Il permet, grâce à ses arguments des opérations intéressantes permettant de limiter les logs à certains critères, comme une plage d'adresse IP (ex "snort -l ../log/snort -h 192.168.0.0/24") .

- **Mode "détection d'intrusion"** Le mode IDS permet à Snort d'adopter un comportement particulier en cas de détection d'une (succession) de chaînes de caractères dans les paquets interceptés ; selon les règles définies dans les fichiers d'extension ".rules" du répertoire /rules ("Snort -A full -d -l ../log -c $SNORTPATH/snort.conf").

I-II.8.3- Fonctionnement des règles de Snort

Les règles de Snort sont décrites dans un langage simple et suivent le schéma suivant

A. **L'en-tête de règle** qui contient

- L'action de la règle (la réaction de Snort);
- Le protocole qui est utilisé pour la transmission des données (snort en considère trois TCP, UDP et ICMP);
- Les adresses IP source et destination et leur masque;
- Les ports source et destination sur lesquels il faudra vérifier les paquets.

B. **Les options de la règle** (entre parenthèse) qui contiennent

- Le message d'alerte;
- Les conditions qui déterminent l'envoi de l'alerte en fonction du paquet inspecté.

L'exemple de règle suivant est simple et permet de détecter les tentatives de login sous l'utilisateur root, pour le protocole ftp (port 21)

Alert tcp any any -> 192.168.1.0/24 21 **(**content "USER root"; nocase;
msg "Tentative d'accès au FTP pour l'utilisateur root";**)**

Les messages en direction de cette plage d'adresse IP effectuant une tentative de login root ("USER root" contenu dans le paquet) auront pour conséquence la génération de l'alerte "Tentative d'accès au FTP pour l'utilisateur root".

Ainsi, il s'agit de renseigner ces variables par les champs que l'on pourrait trouver dans les paquets propres à l'intrusion telle que les "Shell code" que les "exploits" utilisent afin d'insérer des instructions malicieuses dans des programmes sujets aux "buffer overflows". Ainsi, ils obtiennent des accès privilégiés sur la machine et peuvent en prendre le contrôle.

La trace suivante montre un paquet typique provenant d'un tel ping

[root@localhost etc]# tcpdump icmp -vv -X

tcpdump listening on eth0, link-type EN10MB (Ethernet),

capture size 96 bytes 142741.472192

IP (tos 0x0, ttl 128, id 12102, offset 0,

flags [none], length 60) windows > 192.168.0.101 icmp 40

echo request seq 24300

0x0000 4500 003c 2f46 0000 8001 895e c0a8 0064 E..</F.....^...d

0x0010 c0a8 0068 0800 ea5b 0400 5f00 6162 6364 ...h...[.._.**abcd**

0x0020 6566 6768 696a 6b6c 6d6e 6f70 7172 7374 **efghijklmnopqrst**

0.0030 7576 7761 6263 6465 6667 6869 **uvwabcdefghi [19**].

…

I-II.9- Etude d'un HIDS Tripwire

HIDS

Figure I.II-8. Logo Tripwire.

- **Host IDS**
- **S'installe sur un serveur**
- **Avertit l'administrateur en cas de compromission de l'hôte**
- **Est basé sur l'intégrité du système**

Un HIDS Tripwire

- Tripwire compare les résultats avec la base de référence générée préalablement (md5)
- Parce que l'empreinte générée sur un fichier ne peut être reproduite deux fois à l'identique.

Tripwire

-Vérifie les Emprunte de fichiers (MD5, SHA, etc.) – périodicité via le cron.

-Base de référence des fichiers à analyser.

-Emprunte des fichiers de configuration et base de données de Tripwire.

Exemples de HIDS Logsufer, Swatch, Nocol, Osiris, Prelude (Hybride), etc **[20**].

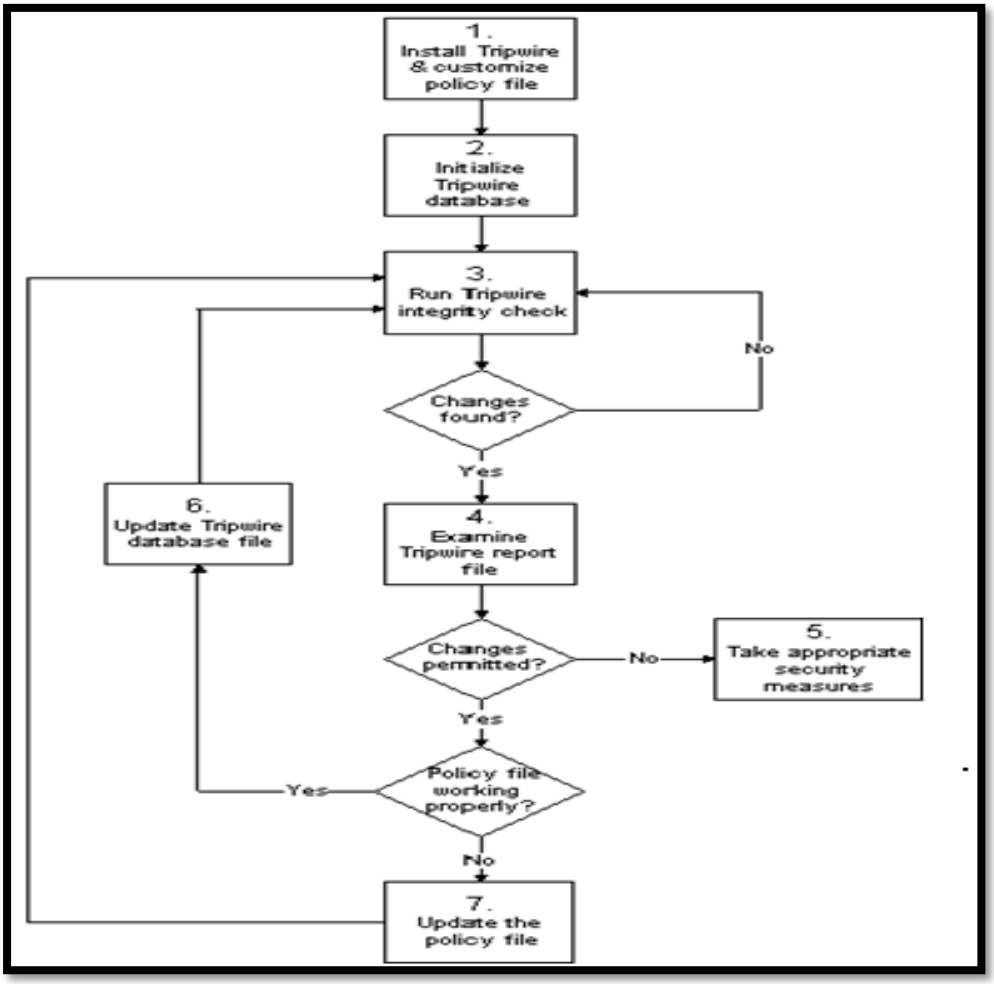


Figure I.II-9. Tripwire en schéma.

Conclusion

Le principe de reportant après infiltration a rapidement évolué pour rechercher un IDS capable de réagir en temps réel. Les dégâts ne sont pas suffisants pour réagir et empêchent le trafic suspect détecté. Les technologies de réponse incluent les IDS ou IPS actifs.

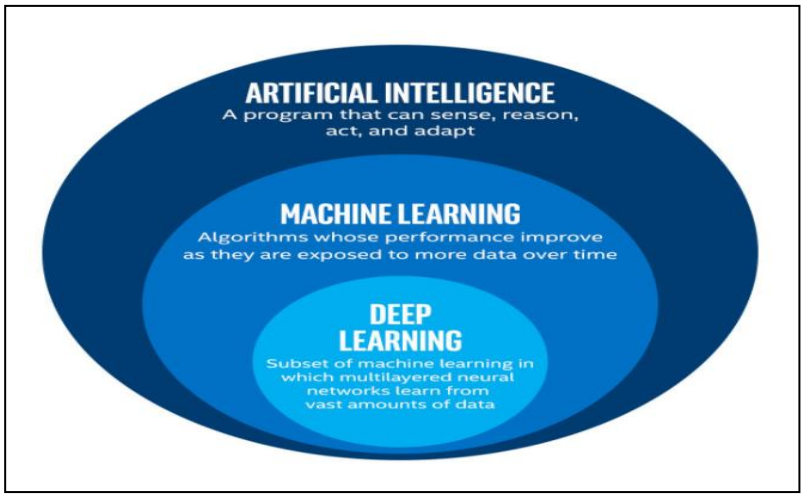
CHAPITRE II DEEP LEARNING (Apprentissage Profondeur).

Introduction

Dans ce chapitre, nous présenterons **l'apprentissage en profondeur (DL)** et les **réseaux de neurones profonds (DNN),** c'est-à-dire des réseaux de neurones avec plusieurs couches cachées. Vous pouvez vous demander quel est le but d'utiliser plus d'une couche cachée est, étant donné le théorème d'approximation universel. C'est dans question naïve, et pendant longtemps les réseaux de neurones ont été utilisés de cette façon.

Sans entrer trop dans les détails, une des raisons est que se rapprocher d'une fonction complexe pourrait nécessiter un nombre énorme de neurones dans la couche cachée, rendant son utilisation peu pratique.

Il est capable d'apprendre des abstractions de caractéristiques d'exemples d'entrée, de comprendre les caractéristiques de base des exemples et de faire des prédictions basées sur celles les caractéristiques. C'est un niveau d'abstraction qui manque dans d'autres **machines de base apprentissage (ML)** et dans les réseaux neuronaux peu profonds.



II.1- Notions de base sur l'apprentissage automatique (Machine Learning Basics)

II.1.1- L'apprentissage automatique est un domaine de l'informatique qui donne aux ordinateurs la possibilité d'apprendre sans être explicitement programmé

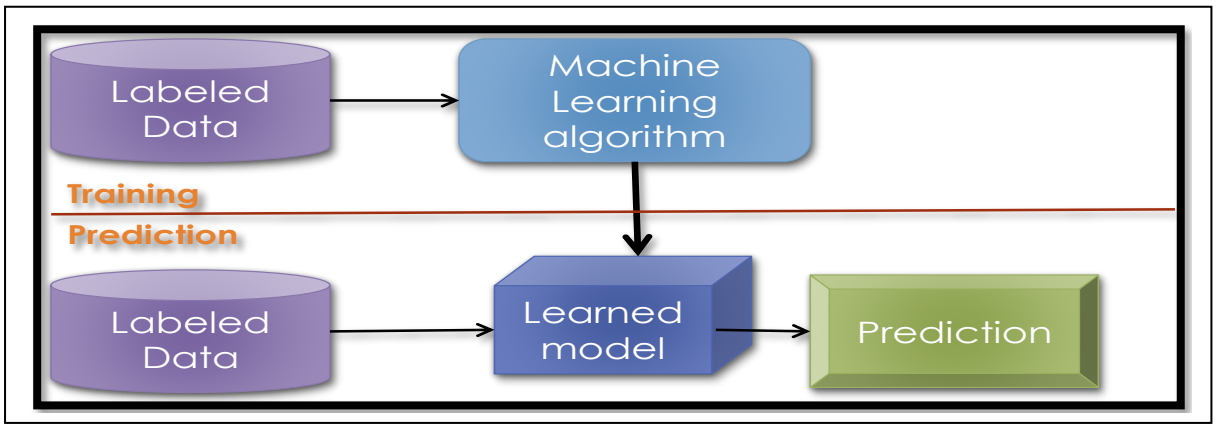


Figure.II.1. Méthodes permettant d'apprendre et de prédire des données.

II.1.2- Types d'apprentissage (Types of Learning)

Supervisé Apprentissage avec un ensemble de **formation étiqueté.**

Exemple classification des emails avec des emails déjà étiquetés.

Non Supervisé Découvrez des **modèles** dans des **données sans étiquette.**

Exemple regrouper des documents similaires basés sur du texte.

Apprentissage par renforcement apprendre à agir sur la base du **retour / récompense.**

Exemple apprendre à jouer Go, récompense gagner ou perdre **[21]**.

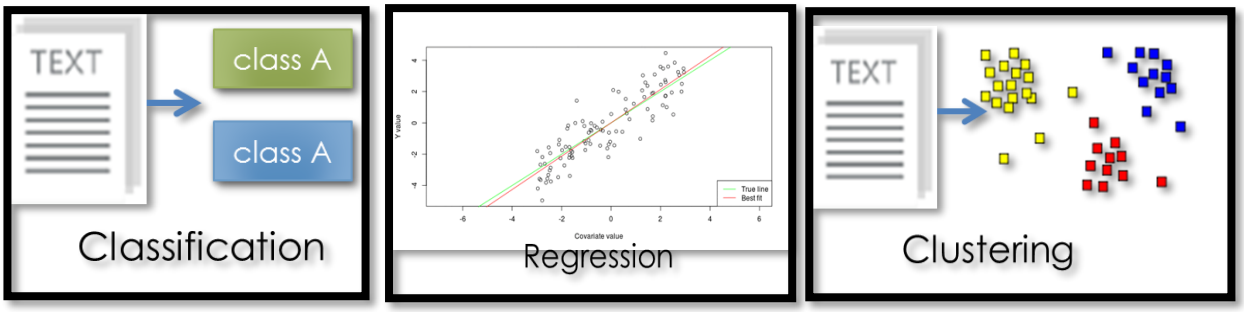


Figure.II.2. Exemple classification. **Figure.II.3.** Exemple Régression. **Figure.II.4.** Exemple Clustering.

Détection d'une anomalie.

Marquage de séquence.

……

II.1.3- ML vers Deep Learning

La plupart des méthodes **d'apprentissage automatique (Machine Learning)** fonctionnent bien en **raison de représentations** et de **fonctions de saisie conçues par l'homme**.

ML devient simplement une **optimisation des poids** pour mieux faire une prédiction finale.

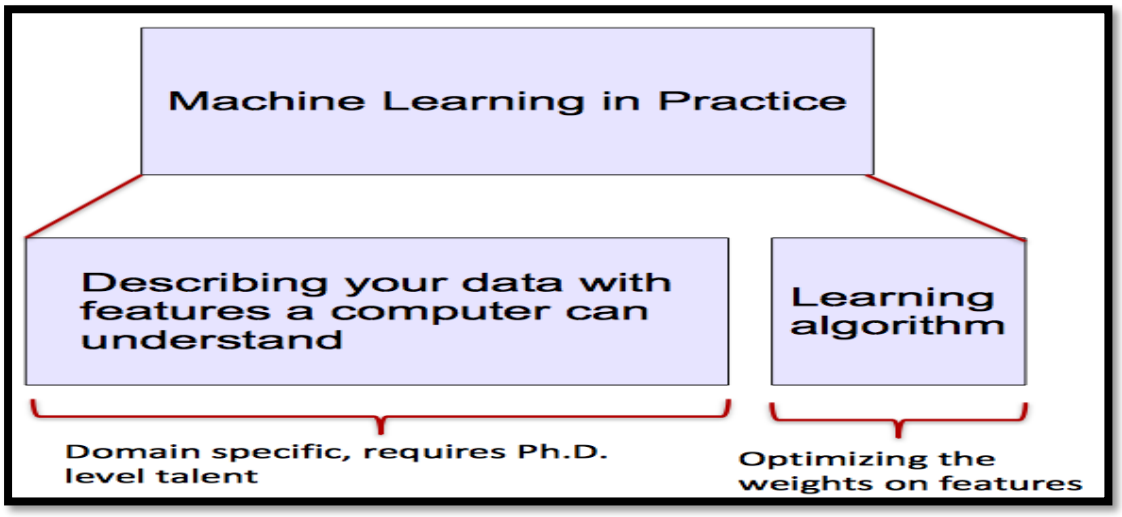


Figure.II.5. Machine Learning en pratique.

II.2- Introduction à l'apprentissage en profondeur (Deep Learning)

II.2.1- Qu'est-ce que l'apprentissage en profondeur (DL) ?

Un sous-champ d'apprentissage automatique des **représentations** de données d'apprentissage. Exceptionnel efficace pour **l'apprentissage des modèles**.

Les algorithmes d'apprentissage approfondi tentent d'apprendre (à plusieurs niveaux) la représentation en utilisant une **hiérarchie de plusieurs couches**, Si vous fournissez au système des **tonnes d'informations**, il commence à le comprendre et à réagir de manière utile.

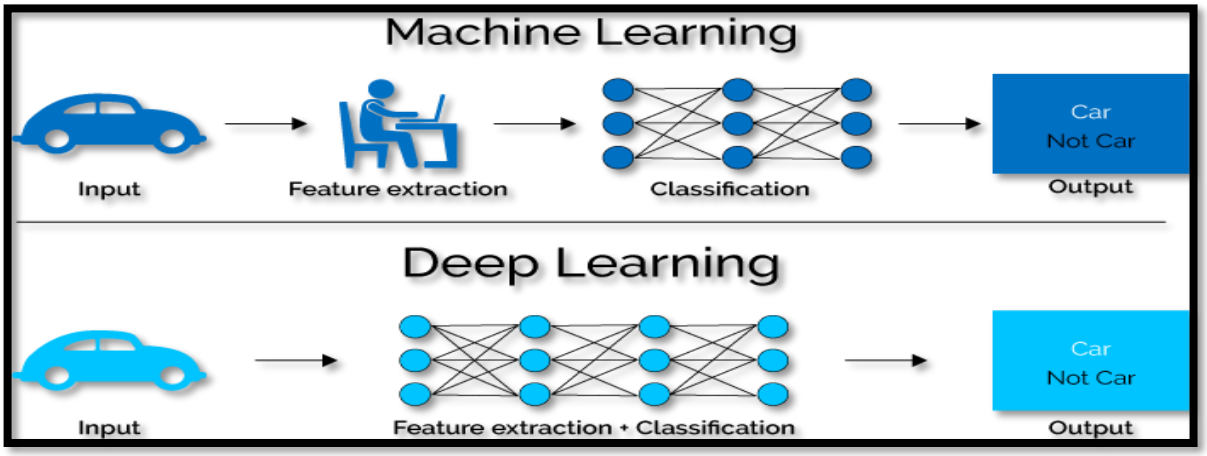


Figure.II.6. ML vers Deep Learning.

II.2.2- Pourquoi DL est-il utile ?

1. Les fonctionnalités conçues manuellement sont **souvent sur-spécifiées, incomplètes** et prennent beaucoup de temps à concevoir et à valider.
2. Les fonctionnalités **apprises sont faciles à adapter**, **rapides** à apprendre.
3. L'apprentissage en profondeur fournit un cadre très **souple**, **universel** (presque ?) Et pouvant être appris pour représenter des informations **mondiales**, visuelles et linguistiques.
4. Peut apprendre à la fois sans surveillance et sous surveillance.
5. Apprentissage efficace du système conjoint de **bout en bout**.
6. Utiliser de grandes quantités de données d'entraînement **[22]**.

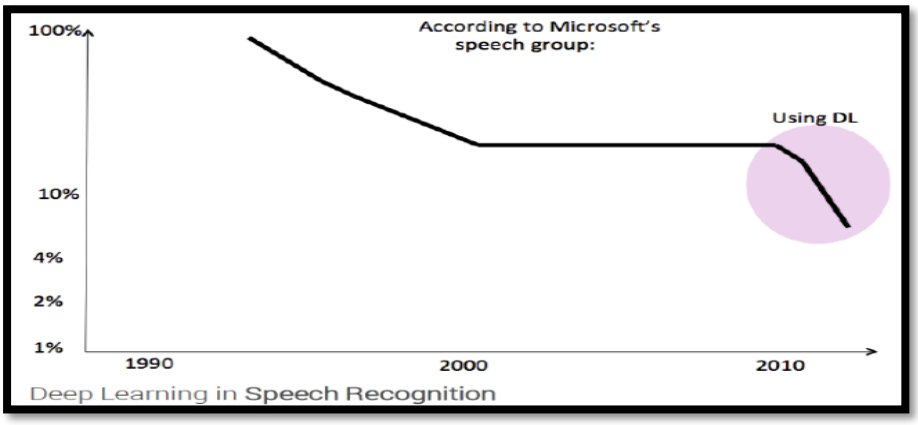


Figure.II.7. Apprentissage en profondeur dans la reconnaissance vocale

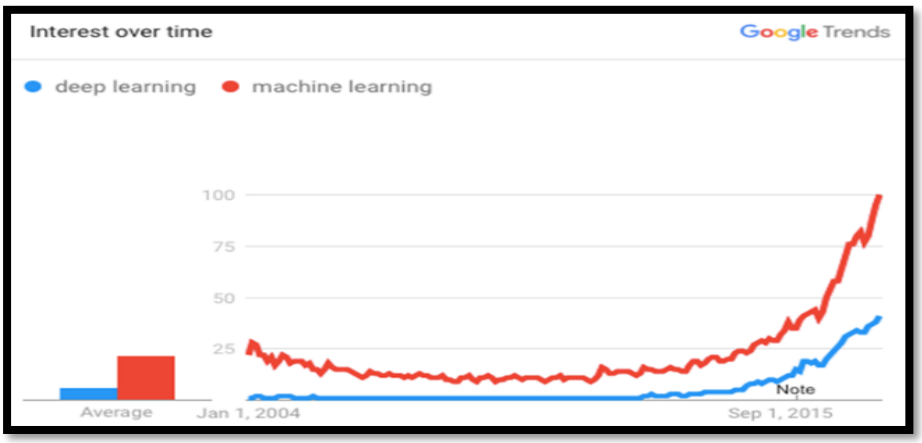


Figure.II.8. Google Trends l'Utilisation DL vers ML.

- **Plusieurs améliorations majeures de la PNL au cours des dernières années**
  - ✓ Traduction automatique
  - ✓ Analyse des sentiments
  - ✓ Agents de dialogue
  - ✓ Question répondant
  - ✓ Classification du texte…
- **Tirer parti des différents niveaux de représentation**
  - ✓ Mots et caractères.
  - ✓ Syntaxe et sémantique.

II.3- Introduction du 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.

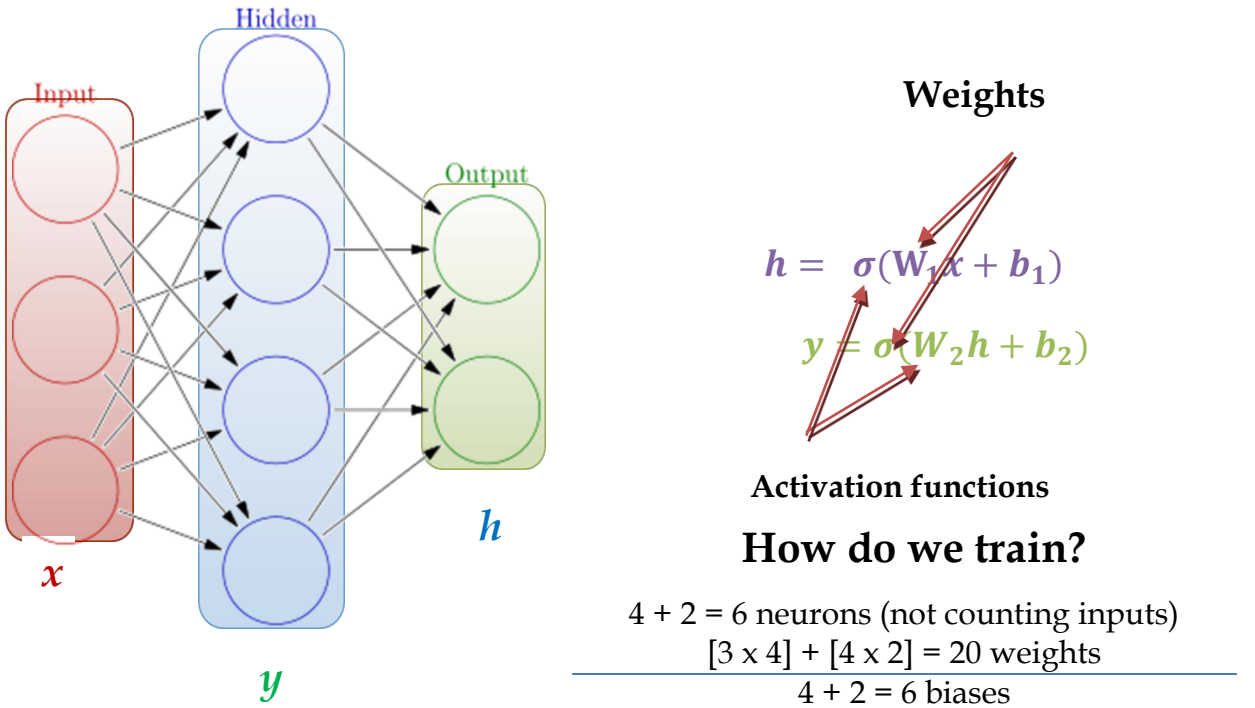


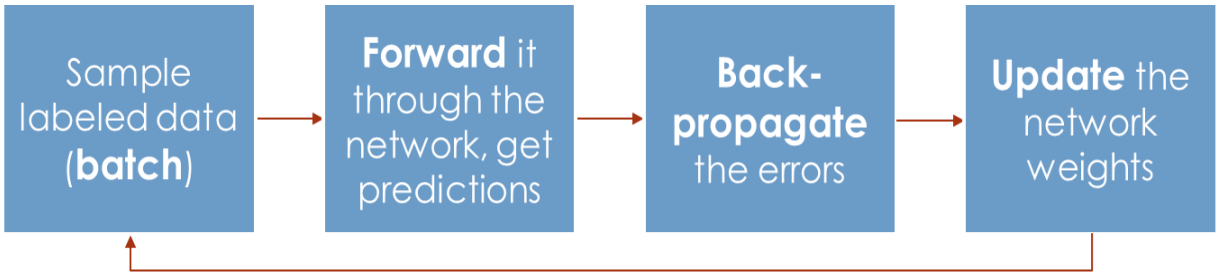
Figure.II.9. Simple Neural Network**.**

II.3.1- Réseaux de neurones artificiels

Tâches à résoudre par des réseaux de neurones artificiels

- ✓ Contrôler les mouvements d'un robot en fonction de sa propre perception et d'autres informations (par exemple, des informations visuelles);
- ✓ Décider de la catégorie de produits alimentaires potentiels (par exemple, comestibles ou non) dans un monde artificiel;
- ✓ Reconnaître un objet visuel (par exemple, un visage familier);
- ✓ Prédire où un objet en mouvement va, quand un robot veut l'attraper **[23]**.

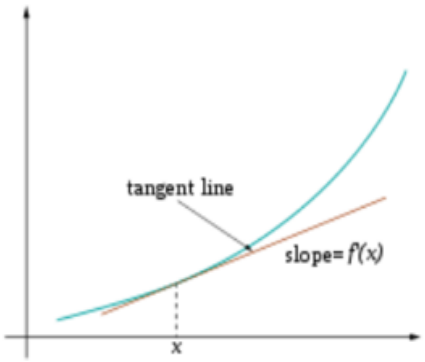
Entraînement (Training)



Optimiser (min. Ou max.) **Fonction objectif / coût $J(\theta)$**

Générer **un signal d'erreur qui mesure** la différence entre les prévisions et les valeurs cibles.

Utilisez le signal d'erreur pour modifier les poids et obtenir des prévisions plus précises, La soustraction d'une fraction du gradient vous déplace vers le minimum (local) de la fonction de coût.



Adaptive Gradient Algorithme (AdaGrad)

Objective/cost function $J(\theta)$ **Review of backpropagation**

$$\theta_j^{new} = \theta_j^{old} - \alpha \frac{d}{d\theta_i^{old}} J(\theta)$$ **Update each element of θ**

$$\theta^{new} = \theta^{old} - \alpha \nabla_\theta J(\theta)$$ **Matrix notation for all parameters**

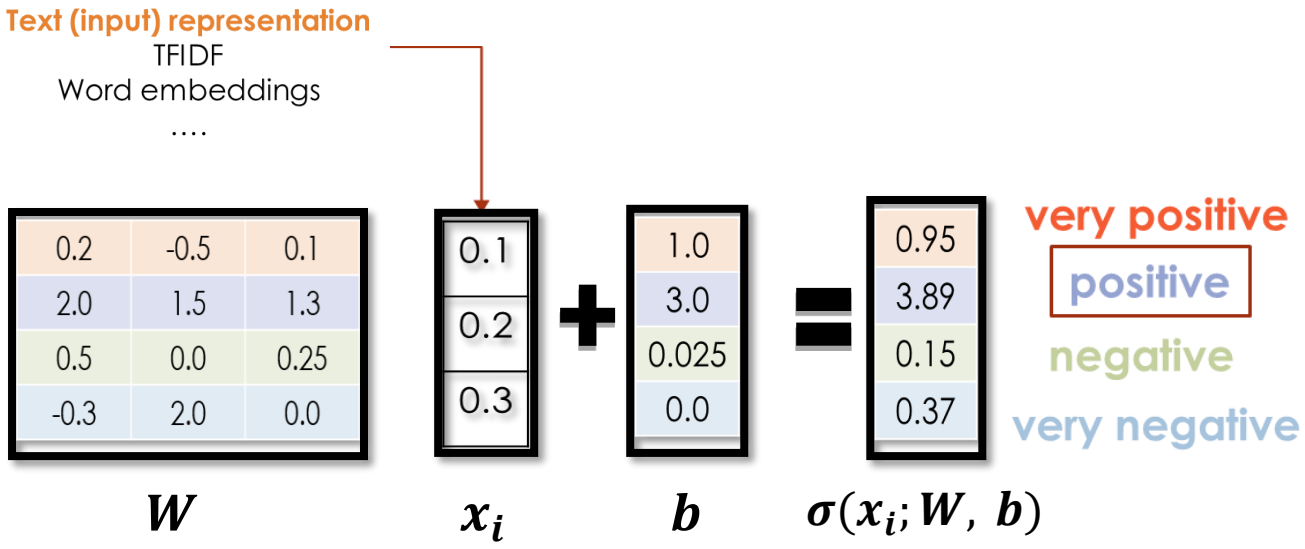
Learning rate

Applique récursivement une **règle de chaîne**

À travers chaque nœud.

N.B **Algorithme de gradient adaptatif (AdaGrad)** qui maintient un taux **d'apprentissage** par paramètre améliorant les performances sur les problèmes de gradients clairsemés (par exemple, problèmes de langage naturel et de vision par ordinateur).

Un passage en avant



II.3.2- Principaux composants / hyper-paramètres RN

$$W_1 W_2 x = Wx$$

II.3.2.1- fonctions d'activation NN

Non-linéarités nécessaires pour apprendre des représentations complexes (non linéaires) de données, sinon le NN ne serait qu'une fonction linéaire **[24]**.

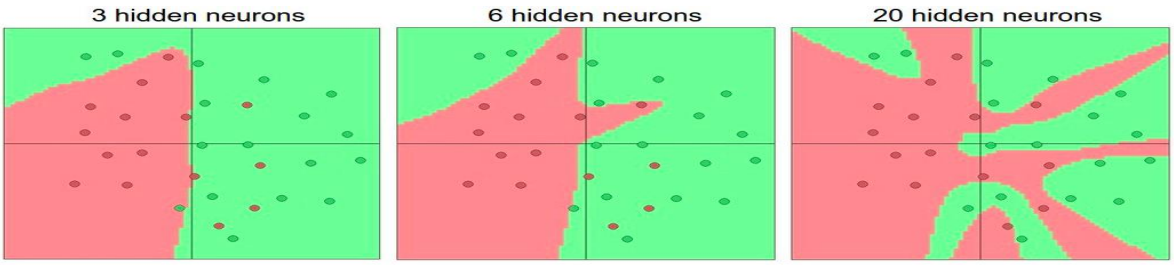


Figure.II.10. Plus de couches et de neurones peuvent approximer des fonctions plus complexes.

II.3.2.1.1- Activation Sigmoid

Prend un nombre réel et le réduit "squashes"

à zéro 0 et 1.

$$R^n \rightarrow [0,1]$$

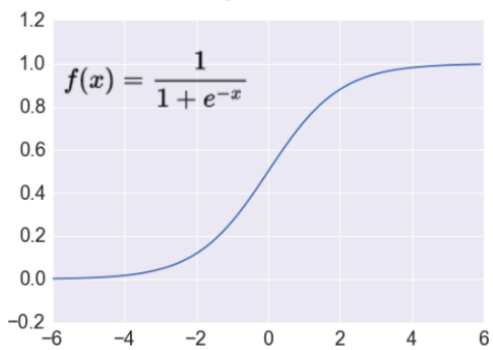


Figure.II.11. Activation RN par fonction Sigmoid

+ Belle interprétation comme le **taux de déclenchement** (**firing rate**) d'un neurone

- 0 = pas de tir du tout.
- 1 = tir complet.

- Les neurones sigmoïdes **saturent** et **tuent les gradients**, ainsi **NN** apprendra à peine lorsque les neurones sont activés à 0 ou 1 (saturer)

- Gradient à ces régions presque zéro.
- Presque aucun signal ne coulera à ses poids.
- Si les poids initiaux sont trop importants, la plupart des neurones satureraient.

II.3.2.1.2- Activation Tanh

Prend un nombre réel et le réduit "squashes"

à -1 et 1.

$R^n \rightarrow [-1,1]$

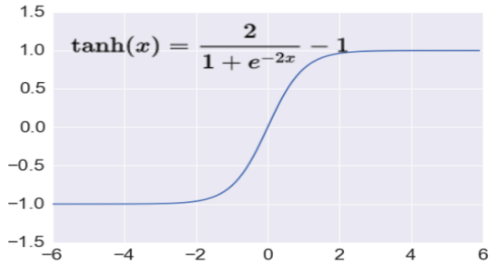


Figure.II.12. Activation RN par fonction Tanh

- Like sigmoid, Tanh neurons **saturate.**
- Unlike sigmoid, output is **zero-centered.**
- Tanh is a **scaled sigmoid** $\tanh(x) = 2sigm(2x) - 1.$

II.3.2.1.3- Activation ReLU

Prend un nombre de valeurs réelles
et le limite à zéro $f(x) = \max(0, x).$

N.B La plupart des réseaux profonds utilisent ReLU de nos jours.

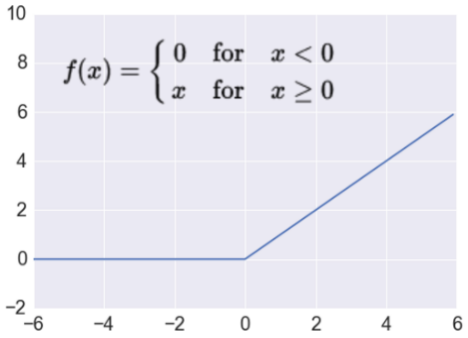


Figure.II.13. Activation RN par fonction ReLU

- Trains beaucoup plus **vite**
  - accélère la convergence de SGD.
  - dû à une forme linéaire, non saturante.
- Operations moins coûteuses
  - comparé à sigmoïde / tanh (exponentielles, etc.).
  - mis en œuvre simplement en seuillant une matrice à zéro.Plus **expressif**.
- Empêche le **problème de disparition du gradient** **[25]**.

Surapprentissage (Overfitting)

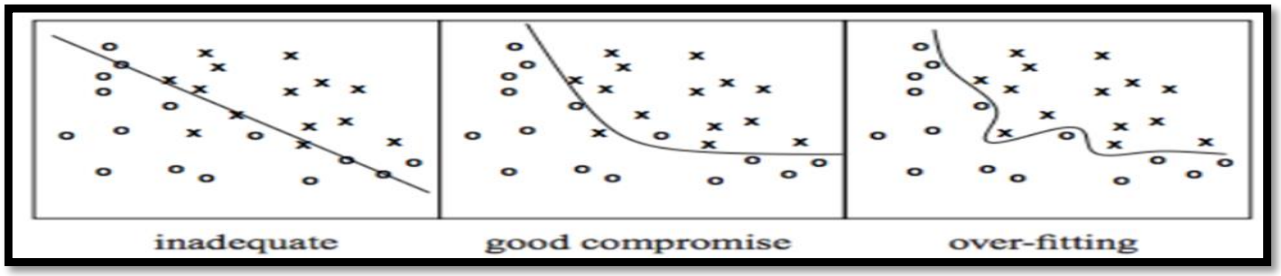
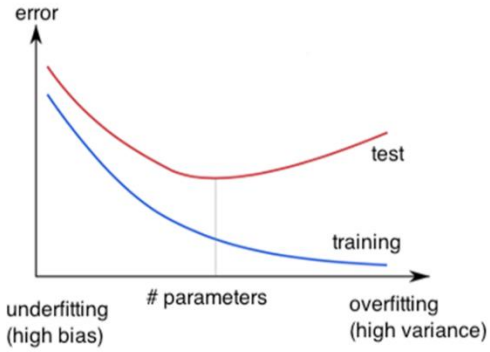


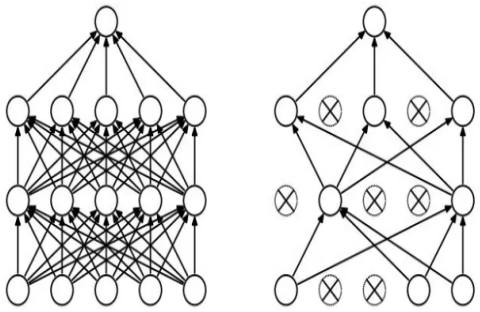
Figure.II.14. Suradaptation des données.

<u>Remarque</u>-Nous disons qu'une hypothèse « surpasse » les données si nous pouvons trouver une hypothèse différente avec plus d'erreur d'apprentissage, mais moins d'erreur de données réelle.



L'hypothèse apprise peut très bien correspondre aux données d'apprentissage, même aux valeurs aberrantes (bruit) mais ne pas généraliser à de nouveaux exemples (données de tes **[26]**.

II.3.2.2- Régularisation NN



- **Abandonner**
  - ✓ Larguer au hasard des unités (pendant que leurs connexions)
  - ✓ Chaque unité de paiement avec une probabilité fixe, indépendante of the other units.
  - ✓ Hyper-paramètre p à choisir (syntoniser)

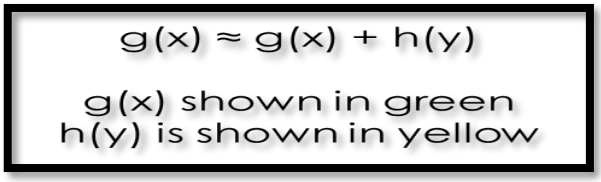
Figure.II.15. Exemple **Régularisation NN.**

- **L2 = perte de poids**
  - ✓ Terme de régularisation qui pénalise les gros poids, ajouté à l'objectif.
  - ✓ La valeur de décroissance du poids détermine la régularisation dominante lors du calcul du gradient.
  - ✓ Coefficient d'affaiblissement du poids ⟶ grosse pénalité pour les gros poids.

$$J_{reg}(\theta) = J(\theta) + \lambda \sum_{k} \theta_k^2$$

- **Arrêt précoce**
  - ✓ Utilisez une erreur de validation pour décider quand arrêter votre entraînement.
  - ✓ Arrêtez-vous lorsque la quantité contrôlée ne s'est pas améliorée après n époques suivantes.
  - ✓ n s'appelle la patience.

II.3.2.3- Réglage des hyper-paramètres (Tuning)



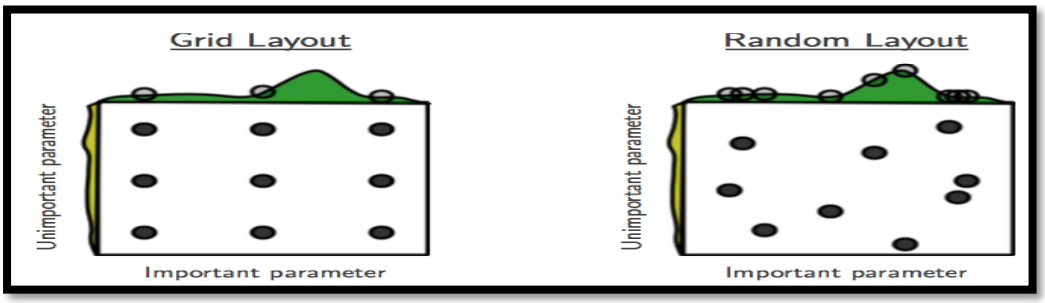


Figure.II.16. Réglage des hyper-paramètres (Tuning).

"Recherche quadrillée et aléatoire de 9 essais d'optimisation de la fonction **g(x) ≈ g(x) + h(y).**

Avec la recherche sur grille, neuf essais ne testent que g (x) à trois endroits distincts.

Avec la recherche aléatoire, les neuf essais explorent les valeurs distinctes de g. "

- Les deux essayent des conFigureurations au hasard et à **l'aveuglette**.
- L'essai suivant est indépendant de tous les essais effectués avant.

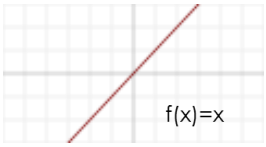
Optimisation bayésienne pour le réglage hyper-paramètre

Faites un choix plus judicieux pour le prochain essai, minimisez le nombre d'essais

1. Recueillir les performances dans plusieurs configurations.
2. Faites des déductions et décidez quelle configuration essayer ensuite **[27]**.

II.3.2.4- Classification vers les Tâches de Régression

II.4.2.4.1- Fonctions de perte et sortie (Loss functions and output) [28].

	Classification	Régression
Training examples	R^n x {class_1, ..., class_n} (One-Hot Encoding)	R^n x R^m
Output Layer	Soft-max [map R^n to a probability distribution]	**Linear (Identity) or Sigmoid** 
Cost (loss) function	Cross-entropy	**Mean Squared Error**

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\top \mathbf{w}_k}}$$

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\left[y_k^{(i)} \log \hat{y}_k^{(i)} + \left(1 - y_k^{(i)}\right) \log\left(1 - \hat{y}_k^{(i)}\right)\right]$$

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

Mean Absolute Error

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left|y^{(i)} - \hat{y}^{(i)}\right|$$

List of loss functions

N.B

- **MSE** a de belles propriétés mathématiques qui facilitent le calcul du gradient en raison du carré, les erreurs importantes ont une influence relativement plus grande sur les MSE.
- Par conséquent, **MAE** est plus résistant aux valeurs aberrantes car il n'utilise pas de carré.

II.4- Convolutional Neural Networks (CNNs)

En **apprentissage automatique**, un réseau de **neurones convolutifs ou réseau** de neurones à convolution (en anglais **CNN** ou Convolutional Neural Networks) est un type de réseau de neurones artificiels acycliques (feed-forward), dans lequel le motif de connexion entre les neurones est inspiré par le cortex visuel des animaux. Les neurones de cette région du cerveau sont arrangés de sorte qu'ils correspondent à des régions qui se chevauchent lors du pavage du champ visuel. Leur fonctionnement est inspiré par les processus biologiques, ils consistent en un empilage multicouche de perceptrons, dont le but est de prétraiter de petites quantités d'informations. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

II.4.1- Présentation

Considérons l'analyse d'une image monochrome (en 2 dimensions, largeur et hauteur) ou en couleur (en 3 dimensions, en considérant l'image **RVB** avec 3 unités de profondeurs, dont la troisième correspond à l'empilement de 3 images selon chaque couleur, rouge, verte et bleue).

Un réseau neuronal convolutif se compose de deux types de neurones artificiels, agencés en « couches » traitant successivement l'information

- **les neurones de traitement**, qui traitent une portion limitée de l'image (appelée « champ réceptif ») au travers d'une fonction de convolution.
- **les neurones de mise en commun** des sorties dits de *pooling* (totale ou partielle) [(1)].

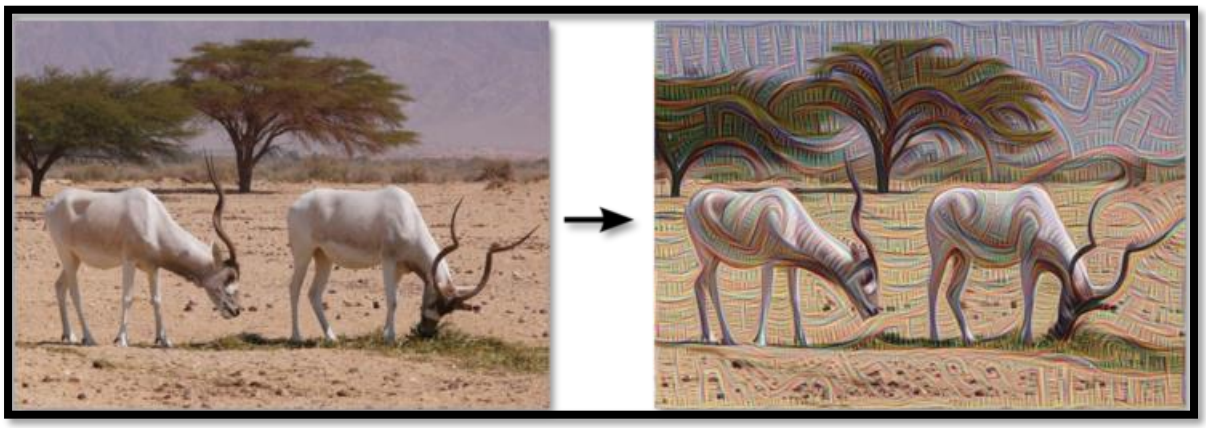


Figure.II.17. Exemple d'image (gauche) et de son traitement par *DeepDream* (droite).

- Un traitement correctif non-linéaire et ponctuel peut être appliqué entre chaque couche pour améliorer la pertinence du résultat.
- L'ensemble des sorties d'une couche de traitement permet de reconstituer une image intermédiaire, qui servira de base à la couche suivante **[29]**.

II.4-2- Nombre de filtres

Comme la taille des textes et des images intermédiaires diminue avec la profondeur du traitement, les couches proches de l'entrée ont tendance à avoir moins de filtres tandis que les couches plus proches de la sortie peuvent en avoir davantage. Pour égaliser le calcul à chaque couche, le produit du nombre de caractéristiques et le nombre de pixels traités est généralement choisi pour être à peu près constant à travers les couches. Pour préserver l'information en entrée, il faudrait maintenir le nombre de sorties intermédiaires (nombre d'images intermédiaires multiplié par le nombre de positions de pixel) pour être croissante (au sens large) d'une couche à l'autre, Le nombre d'images intermédiaires contrôle directement la puissance du système, dépend du nombre d'exemples disponibles et la complexité du traitement.

Convolutional Neural Networks (CNNs) (1)

Idée principale de CNN pour le texte

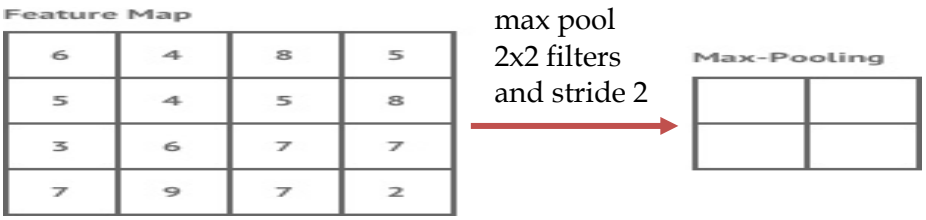
Calculer les vecteurs pour n-grammes et les regrouper ensuite.

Exemple «cela prend trop de temps», calculez les vecteurs pour

Cela prend, prend aussi, trop longtemps, cela prend aussi, prend trop de temps, cela prend trop de temps.

Convolutional Neural Networks (CNNs) (2)

Idée principale de **CNN** pour le texte
Calculer les vecteurs pour n-grammes et **les regrouper ensuite.**



II.4.2.1- Forme du filtre

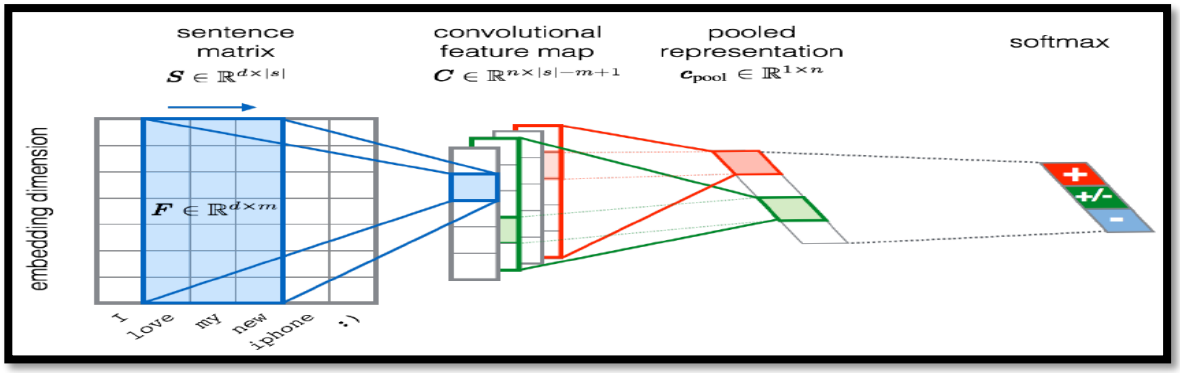
Les formes de filtre varient grandement dans la littérature. Ils sont généralement choisis en fonction de l'ensemble de données. Les meilleurs résultats sur les images de MNIST (28 x 28) sont habituellement dans la gamme de 5 × 5 sur la première couche, tandis que les ensembles de données d'images naturelles (souvent avec des centaines de pixels dans chaque dimension) ont tendance à utiliser de plus grands filtres de première couche de **12 × 12**, voire **15 × 15**.

Le défi est donc de trouver le bon niveau de granularité de manière à créer des abstractions à l'échelle appropriée et adaptée à chaque cas **[30]**.

II.4.2.2- Forme du Max Pooling

Les valeurs typiques sont 2 × 2. de très grands volumes d'entrée peuvent justifier un pooling 4 × 4 dans les premières couches. Cependant, le choix de formes plus grandes va considérablement réduire la dimension du signal, et peut entraîner la perte de trop d'information.

II.4.3.A- CNN pour la Classification du Texte



FigureII.18. CNN pour la classification du texte.

N.B

La largeur m des filtres de convolution est fixée à 5; n = 300 cartes de caractéristiques
La première couche incorpore des mots dans des vecteurs de faible dimension.

II.4.3.B- CNN avec plusieurs filtres

En fonction des valeurs et de la taille de la matrice du filtre, nous obtenons une nouvelle image plus ou moins modifiée. Le but de ce procédé est de faire ressortir certaines caractéristiques de l'image. Par exemple, avec une première image comme celle affichée ci-dessous, on peut la transformer dans le but de produire plusieurs variantes. Voici le résultat de différentes transformations à l'aide de plusieurs filtres couramment utilisés **[31]**.

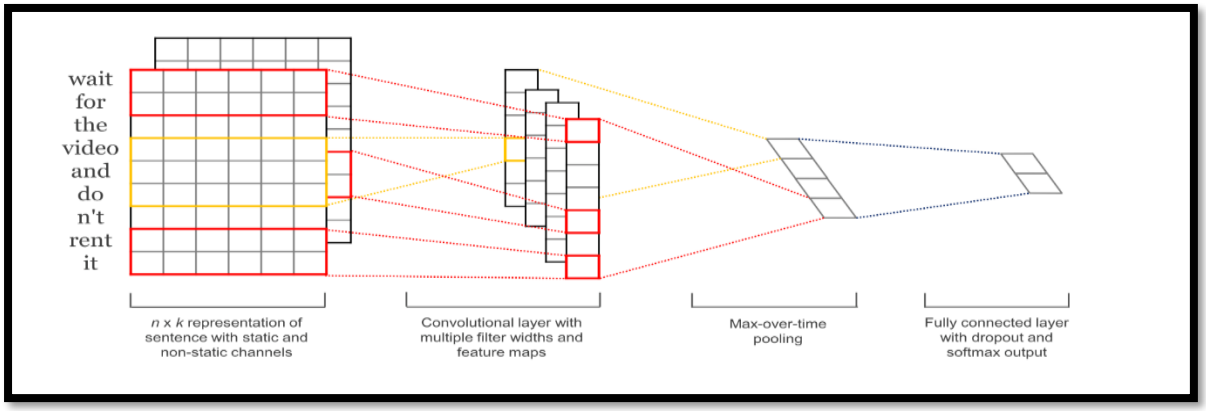


Figure.II.19. CNN avec plusieurs filtres.

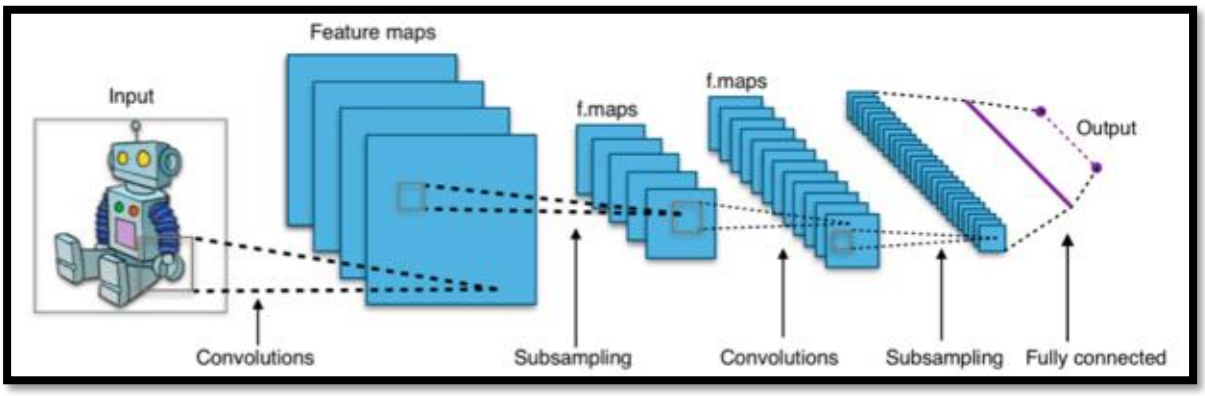
II.4.4- Exemples de modèles de CNN

La forme la plus commune d'une architecture de réseau de neurones convolutifs empile quelques couches Conv-ReLU, les suit avec des couches Pool, et répète ce schéma jusqu'à ce que l'entrée soit réduite dans un espace d'une taille suffisamment petite. À un moment, il est fréquent de placer des couches entièrement connectées (FC). La dernière couche entièrement connectée est reliée vers la sortie. Voici quelques architectures communes de réseau de neurones convolutifs qui suivent ce modèle

- INPUT -> FC implémente un classifieur linéaire
- INPUT -> CONV -> RELU -> FC
- INPUT -> [CONV -> RELU -> POOL] * 2 -> FC -> RELU -> FC Ici, il y a une couche de CONV unique entre chaque couche POOL
- INPUT -> [CONV -> RELU -> CONV -> RELU -> POOL] * 3 -> [FC -> RELU] * 2 -> FC Ici,

il y a deux couches CONV empilées avant chaque couche POOL.

L'empilage des couches CONV avec de petits filtres de pooling (plutôt un grand filtre) permet un traitement plus puissant, avec moins de paramètres. Cependant, avec l'inconvénient de demander plus de puissance de calcul (pour contenir tous les résultats intermédiaires de la couche CONV) **[32]**.



FigureII.20. Architecture standard d'un réseau à convolutions.

Conclusion

Le machine Learning est un outil très puissant qui permet d'effectuer de multiples actions comme classifier des données, faire apprendre à un programme à partir d'expérimentations ou encore de créer un programme évolutionnaire qui s'améliore sans cesse. Ainsi, même avec un échantillon peu fourni (le machine Learning nécessite habituellement des échantillons avec 50 spécimens) et des données influencées par la subjectivité de celui qui les mesure, le machine learning reste relativement précis malgré quelques lacunes, néanmoins, le machine Learning n'a pas que des qualités, il doit être constamment adapté au problème qu'il tente de résoudre, enfin, le machine learning doit être utilisé comme un outil car tous les problèmes ne nécessitent pas un programme complexe en machine learning, les réseaux à convolution (**RCNN**), malgré leur complexité, sont fortement conseillés pour la classification d'images ou la reconnaissance visuelle. Ce sont des domaines où ils surpassent toutes les autres méthodes existantes.

CHAPITRE III Les Systemes de Detection d'Intrusion Bases sur la Machine Learning

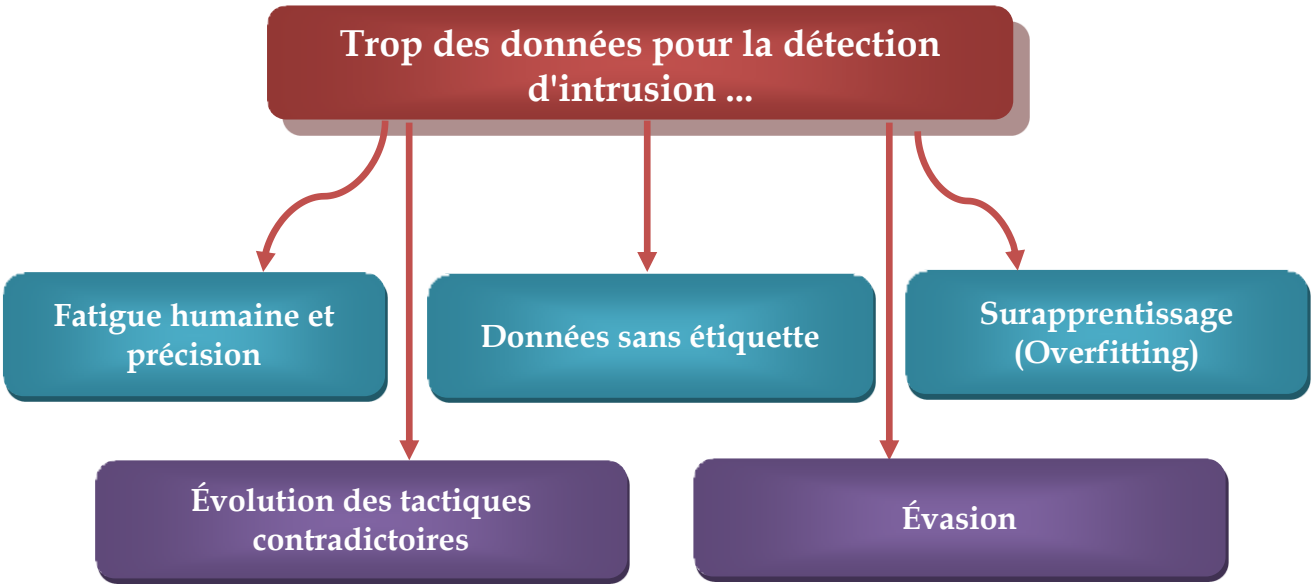
Introduction

Les techniques **d'apprentissage en profondeur (DL)** sont connues pour leur capacité à gérer des données à grande échelle ces jours-ci. Ils ont été étudiés dans diverses applications, par exemple la langue, la modélisation graphique, la parole, l'audio, la reconnaissance d'image, la vidéo, le langage naturel et les zones de traitement de signal.

De plus, des recherches approfondies sur les méthodes d'apprentissage automatique (**ML**) dans le système de détection d'intrusion (**IDS**) ont été menées dans les milieux universitaire et industriel. Cependant, les énormes données et les difficultés à obtenir des instances de données sont des défis de taille pour les IDS basés sur l'apprentissage automatique.

Nous montrons certaines limitations des systèmes **IDS** antérieurs qui utilisaient des apprenants classiques et introduisons l'apprentissage des fonctionnalités, y compris la construction, l'extraction et la sélection des fonctionnalités, afin de relever les défis. Nous discutons de certaines techniques d'apprentissage en profondeur distinguées et de leur application aux fins de **l'IDS**. Les orientations futures de la recherche utilisant des techniques d'apprentissage en profondeur à des fins **d'IDS** sont brièvement résumées.

- **Problème**

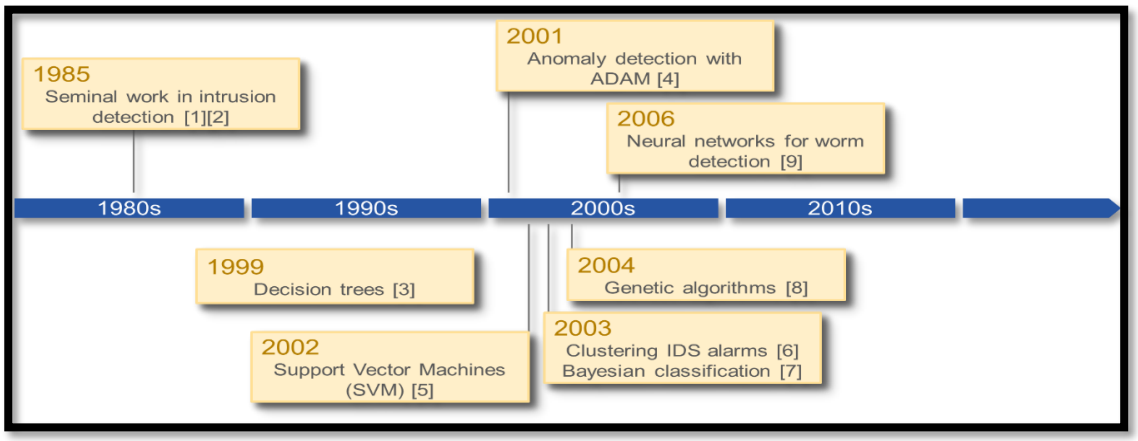


- **Les défis de l'apprentissage automatique pour la détection d'intrusion**

Evasion can reduce detection accuracy to 33% [10].

Overfitting makes machine learning like signature detection [11].

- **Machine Learning in Intrusion Détection (ML-DS Historique)**



III.1- Deep Learning de la Sécurité des Réseaux

La sécurité du réseau est un élément important du domaine des technologies de l'information, car il fournit des stratégies de sécurité préventives à l'infrastructure physique et logicielle du système. Notamment, la quantité de données générées par les réseaux est en augmentation dans le secteur des technologies de l'information, qui comprend les machines réseau, les systèmes d'exploitation et les applications. En revanche, le secteur a toujours recours à des approches manuelles pour prévenir et atténuer les dysfonctionnements du système.

La structure de l'apprentissage en profondeur est décrite à la Figure. 1. Les méthodes d'apprentissage en profondeur inspirées par la profondeur de la structure du cerveau humain apprennent des caractéristiques de niveau inférieur au concept de niveau supérieur.

C'est en raison de l'abstraction de plusieurs niveaux que DBN aide à apprendre les fonctions mappant de l'entrée à la sortie. Le processus d'apprentissage ne dépend pas de caractéristiques créées par l'homme. DBN utilise un algorithme d'apprentissage non supervisé, une machine Boltzmann restreinte (RBM) pour chaque couche **[33]**.

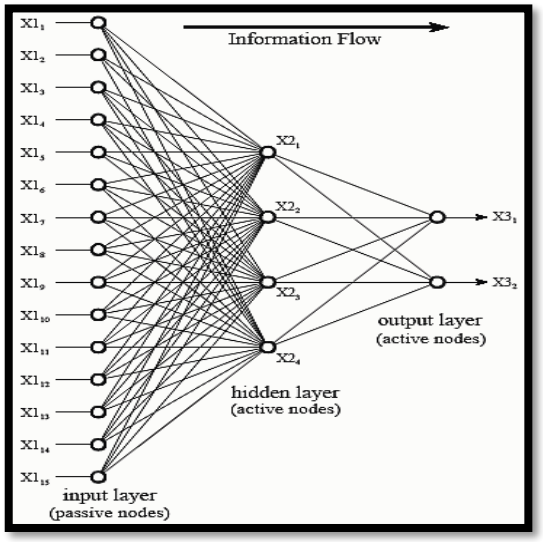


Figure III.1. Structure of deep learning.

III.1.1- Identification de la Circulation

L'identification du trafic est un élément clé de la sécurité du réseau car elle déclenche le drapeau rouge en cas d'intrusion dans le réseau. Le système s'appuie notamment sur les méthodes de détection traditionnelles, qui sont de plus en plus utilisées. Inefficace en raison de l'augmentation proportionnelle des données. Les approches traditionnelles incluent, par exemple, l'identification du port, le protocole HTTP standard qui ne fonctionne pas comme prévu en raison du nombre réduit de protocoles suivant le système. Un autre système implique la méthode basée sur la signature qui repose sur les données utiles. Il est important de noter que cette approche peut être utilisée dans plusieurs applications.

Les chercheurs ont mis au point de nombreux algorithmes d'exploration de données pour l'identification du trafic. Les méthodes traditionnelles d'exploration de données, telles que Native Bayes, Random Forests, Decision Tree, ont été largement utilisées pour classer le trafic réseau. Jun et al**, o**nt utilisé **SVM** et **RBM** ensemble pour résoudre le problème de la détection du trafic réseau.

Néanmoins, l'apprentissage automatique a permis d'améliorer les méthodes d'analyse des problèmes de sécurité des réseaux. En pratique, l'apprentissage en profondeur peut être utilisé pour analyser la sécurité d'un réseau de différentes manières. Pour commencer, la méthode s'est révélée utile pour détecter toute anomalie sur le système de réseau car elle utilise des données statistiques pour calculer les problèmes de réseau en évaluant l'interrelation des neutres dans le système. Afin de comprendre les anomalies dans la sécurité du réseau, il est important d'évaluer plusieurs variables dans la sécurité du réseau **[34]**.

III.1.2- Facteurs de la sécurité du réseau

Premièrement, pour tenter de détecter une intrusion dans le système, il est toujours difficile de différencier les données normales des données anormales. Pour ce faire, la méthode de détection devrait définir les caractéristiques des données malveillantes sur le système. En autre, la technique devrait être capable de concevoir un système de classification capable de différencier avec précision les deux ensembles d'informations authentiques et malveillantes. Ce système est connu sous le nom de technique de réduction de dimensionnalité qui utilise des approches de codage automatique pour calculer la distance entre les nœuds au sein du réseau. Il est important de noter que la technique part du principe que la normalité des données est déterminée par la cohérence de la distance entre les nœuds. En tant que tel, plus la distance entre les nœuds est longue, ce qui indique l'anomalie de l'information, agissant ainsi comme un pointeur sur la présence de données malveillantes. À cet égard, il existe deux systèmes de mesure la distance de Manhattan, qui est la distance totale entre les dimensions du réseau et la distance euclidienne, qui correspond principalement à la taille du vecteur en cours d'évaluation.

Deuxièmement, un autre défi de l'apprentissage en profondeur est le caractère sacré des données utilisées pour détecter une anomalie, dans un processus connu sous le nom d'empoisonnement à la normalité. Fondamentalement, les fonctionnalités utilisées pour extraire les données sont importantes car elles déterminent le résultat, en particulier dans l'apprentissage en profondeur non supervisé. En tant que telle, l'approche doit garantir que les informations normales ne sont pas affectées [**35]**.

III.1.3- Techniques utilisées pour détecter les anomalies dans la sécurité du réseau

Plus important encore, plusieurs approches peuvent être utilisées pour détecter des anomalies dans la sécurité du réseau en utilisant l'apprentissage en profondeur méthode. Pour commencer, la réduction de la dimensionnalité de l'encodeur automatique est un système qui repose sur des composants d'encodeur et de décodeur qui incluent en outre des couches d'entrée, de sortie et cachées, de plus, le système utilise trois étapes qui comprennent une formation préalable, un déroulement et un système de réglage précis.

Le réseau de croyances profondes (DBN) est **également une méthode d'apprentissage en profondeur qui utilise une couche non supervisée** de GAR et un autre niveau supervisé de réseaux de tiers. La méthode DBN est divisée en deux, à savoir que le RBM est conditionné

séparément de manière non supervisée. En outre, le second processus implique que le processus neuronal de BP implique l'utilisation de la dernière sortie du RBM en tant qu'entrée du nouveau niveau de BP, qui sont ensuite classés en utilisant une approche **supervisée**.

Par conséquent, les deux techniques sont combinées pour former le système de détection hybride malveillant. L'approche repose sur la réduction de la dimensionnalité via l'utilisation de la méthode de codage automatique pour fournir l'espace entre les vecteurs. Par conséquent, ces données sont classées via l'utilisation du système DBN par apprentissage en profondeur. Enfin, la précision de la détection est également améliorée afin de réduire les complexités associées au temps dans le système hybride.

L'identification de protocole inconnue est un défi majeur pour les méthodes de détection traditionnelles en matière de sécurité de réseau, car les recherches ont montré que près de 17% des flux de trafic sur les réseaux sont inconnus. Cependant, l'apprentissage en profondeur a tenté d'atténuer le problème, car il est capable d'identifier plus de la moitié du flux de trafic inconnu dans les méthodes traditionnelles.

Mieux encore, l'approche permet de placer une probabilité sur les flux inconnus, augmentant ainsi la précision **[36]**.

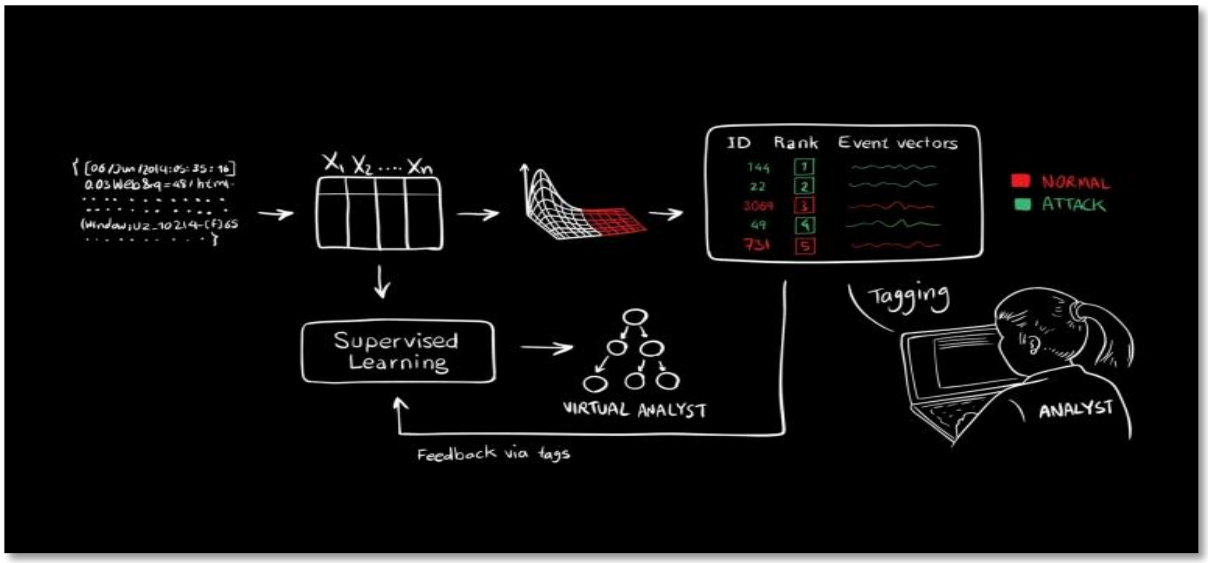


Figure III.2. Analyse les données et détecte les activités suspectes à l'aide d'un apprentissage automatique non supervisé.

III.2- Deep Learning Par L'analyse des Motifs des Données dans la Sécurité du Réseau

La croissance du secteur des technologies de l'information a rendu nécessaire la nécessité de méthodes plus récentes et plus performantes d'analyse du fonctionnement de ces systèmes informatiques.

À cette fin, il existe plusieurs méthodes d'apprentissage automatique permettant d'enquêter sur les principes sous-jacents aux appareils. Le domaine de l'apprentissage en profondeur est notamment dynamique en raison du développement de nouvelles techniques dans plusieurs sous-branches, notamment la reconnaissance d'images, la sécurité informatique et la reconnaissance vocale. Les méthodes classiques d'apprentissage en profondeur utilisées dans la sécurité des réseaux ne parviennent pas à détecter les intrusions dans les systèmes de réseau en raison de l'augmentation proportionnelle de la production de données.

En tant que telle, l'analyse de données volumineuses utilisant un système de croyances profondes est la dernière innovation qui tente d'étudier les modèles d'information en vue de détecter les entrées non autorisées dans les réseaux informatiques.

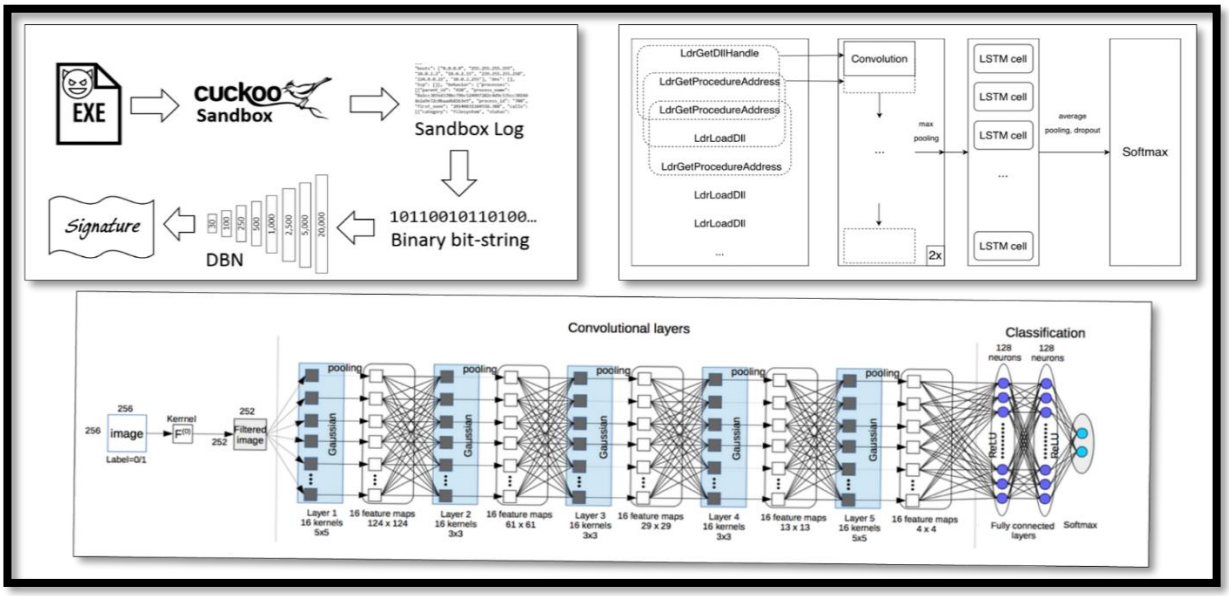


Figure III.3. Deep Learning L'Analyse Papers on Security.

III.2-1 Analyse Déterministe

En substance, le système fonctionne sur des principes connus, notamment l'apprentissage automatique déterministe et probabiliste. L'apprentissage machine déterministe utilise de petits ensembles de données, qui sont analysés pour détecter les écarts par rapport aux modèles normaux.

Ces informations sont ensuite évaluées par des experts en informatique qui formulent des modèles à utiliser pour des investigations ultérieures des données.

Normalement, les informations obtenues sont comparées à une base, de sorte que toute donnée dépassant les niveaux normaux est considérée comme une action intrusive.

III.2-2 Analyse Probabiliste

D'autre part, l'apprentissage automatique probabiliste va encore plus loin puisqu'il évalue les schémas impliqués dans l'évaluation qui auraient pu échapper à l'analyse déterministe. Il est important de noter que le système s'appuie sur le clustering afin détecter tout caractère anormal concernant les données. Le système repose sur une action non supervisée dans laquelle le système s'exécute indépendamment pour générer une carte qui est finalement analysée par la même machine pour tout comportement anormal.

En conséquence, la démarche est plus efficace puisque le L'évaluation est concluant et permet de cerner le problème exact grâce à des estimations prudentes qui le situent à 90%.

III.2-3 Réseaux de Codage en Profondeur

Les réseaux de codage en profondeur ont également gagné en popularité ces derniers temps en raison des avantages qu'il possède dans les techniques d'apprentissage en profondeur. L'approche est dynamique car le système est prédictif et s'adapte aux nouveaux environnements de données.

La méthode utilise notamment les produits d'approches descendantes et les utilise comme intrants pour les approches ascendantes.

En outre, le modèle extrait des entités à l'aide de modèles linéaires, qui sont à leur tour utilisés comme éléments de construction pour les calques.

III.3- Problèmes de la Détection d'Intrusion

III.3.1- Ensemble de données (DataSet KDD1999) pour évaluation

Le principal problème de la détection d'intrusion est d'assurer une communication de sécurité dans des réseaux multi-nœuds différents d'intrus de réseau éventuels par la méthode de classification du trafic entrant en classes anormales normales et différentes.

Les méthodes de détection d'intrusion existantes reposent sur de faux détection positive ou problème de détection négative en raison du fait que de nombreuses actions d'intrusion restent non détectées et que les utilisateurs légitimes sont détectés comme intrus. Dans cet article, nous comparons la précision avec différentes méthodes de l'ensemble de données **KDD-99**.

Depuis **1999, le KDD-99** est l'ensemble de données le plus utilisé au monde pour l'évaluation des méthodes de classification du trafic. Les attaques dans l'ensemble de données KDD-99 appartiennent à quatre catégories dans **[37]**

III.3.1.1 Attaque par déni de service (DOS) attaque dans laquelle l'attaquant crée également des ressources de calcul ou de mémoire occupé ou trop plein pour traiter des demandes légitimes, ou refuse aux utilisateurs légitimes l'accès à une machine.

III.3.1.2 Attaque d'utilisateur à utilisateur (U2R) est une classe d'exploits dans laquelle l'attaquant commence par accéder à un compte utilisateur normal sur le système (obtenu peut-être en reniflant des mots de passe, par une attaque par dictionnaire ou par une ingénierie sociale) et est capable d'exploiter une vulnérabilité pour obtenir un accès root au système.

III.3.1.3 Attaque distante par rapport à l'attaque locale (R2L) survient lorsqu'un attaquant capable d'envoyer des paquets à une machine via un réseau mais ne disposant pas d'un compte sur cette machine exploite une vulnérabilité pour obtenir un accès local en tant qu'utilisateur de cette machine.

III.3.1.4 Attaque de détection tentative de collecte d'informations sur un réseau d'ordinateurs dans le but apparent de contourner ses contrôles de sécurité.

III.3.2- Prétraitements Dataset

En raison du déséquilibre du jeu de données KDD-99 dans le Tableau I, toutes les méthodes permettant de classer le trafic dans celui-ci deviennent difficiles.

Title of Dataset	Data Classified					
	Normal	*DOS*	*Probe*	*U2R*	*R2L*	*Total*
10% KDD Data	97278	391458	4107	52	1126	494021
10% KDD Data For Test	60591	223298	2377	39	5993	292298

Tableau III.1. Répartition des Attaques dans Data Set-IDS 99 KDD CUP.

Pour résoudre ce problème, la technique de sur échantillonnage des minorités synthétiques (**SMOTE**), qui est une technique très populaire, a été appliquée pour traiter ce problème **[38]**.

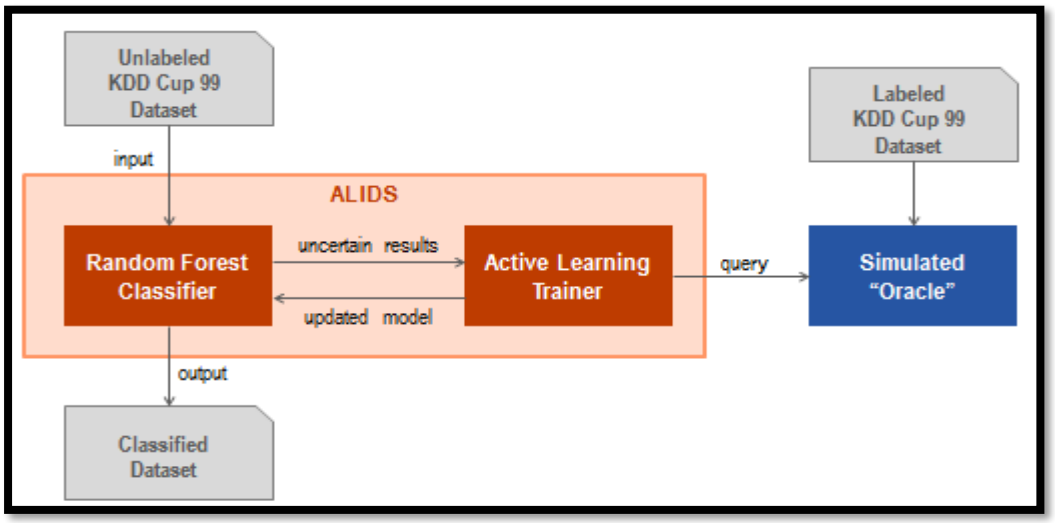


Figure III.4. Active Learning Intrusion Detection System (ALIDS) Prototye.

III.4- Évaluation des Performances Après la procédure technique SMOTE, le problème de déséquilibre a été résolu. Les résultats sont présentés dans le Tableau III.2.

Title of Dataset	Data Classified					
	Normal	*DOS*	*Probe*	*U2R*	*R2L*	*Total*
10% KDD Corrected Data	559186	391458	726993	671372	735472	3084481

Tableau III.2. Distribution d'attaques après l'application de Smote.

Le problème de la répartition des déséquilibres a été réduit au Tableau II. Ensuite, nous utilisons les données pour nos expériences. Certaines des caractéristiques sont énumérées dans le Tableau III.3.

No.	*Name*	*Type*
1	HTTP response code	Number
2	HTTP request type	Text
3	HTTP packet length	Number
4	Contain attachment	Number
5	Attachment type	Text
6	Attachment size	Number
7	Download/upload	Boolean
8	Number of HTTP links with same IMSI in 2 min	Number
9	Number of HTTP packet sent with same IMSI in 2 min	Number
10	Number of HTTP packet received with same IMSI in 2 min	Float
11	Send-to-receive ratio of packets with same IMSI in 2 min	Float
12	Number of bytes sent with same IMSI in 2 min	Number
13	Number of bytes received with same IMSI in 2 min	Number
14	Send-to-receive ratio of bytes with same IMSI in 2 min	Float
15	Ratio of packet with the same destination IP in 3 min	Float

Tableau III.3. Quelques fonctionnalités utilisées dans les expériences.

C'est un problème de classification pour la détection du trafic. Nous sélectionnons certaines méthodes de classification classiques, telles que Décision Tree (C4.5), Naïve Bayes, Support Vector Machine (SVM) et SVM-RBMS **[39,40]**.

III.4.1 Approche d'essai

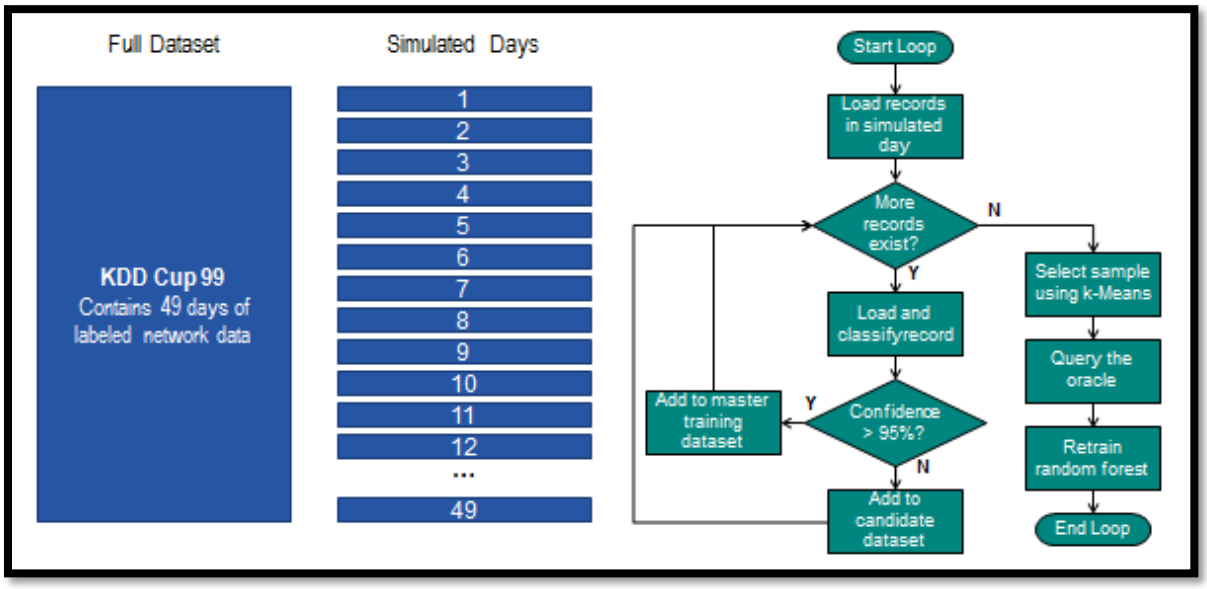
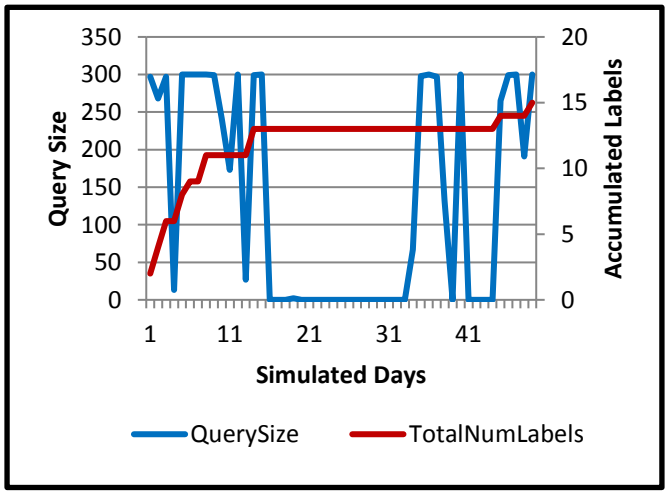


Figure.III.5. Approche d'essai DataSet KDD Cup99 par jour simulé envoyé à Oracle .

III.4.1.1 Résultats

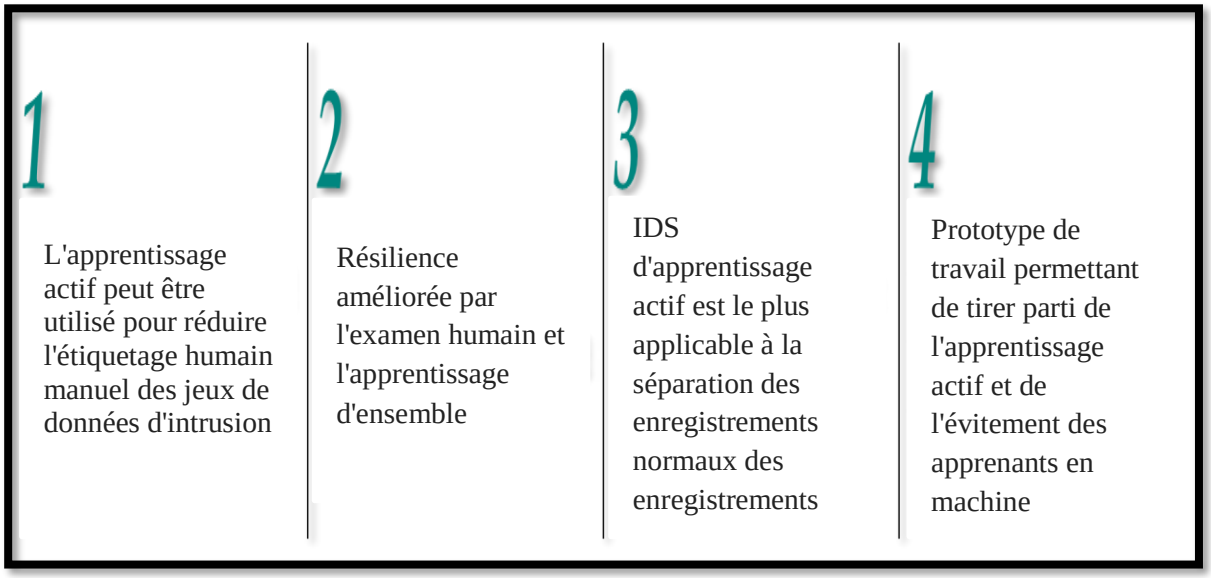


Résumé

- Jusqu'à 300 enregistrements par jour simulé envoyé à Oracle.
- 0,31% du nombre total d'enregistrements envoyés à Oracle pour étiquetage.
- 91% des enregistrements normaux identifiés.
- Identifié 15 des 23 étiquettes possibles **[41]** .

Figure.III.5. Résultats Approche d'essai DataSet KDD Cup99 par jour simulé .

III.4.1.2 Contributions [42]



Conclusion

L'apprentissage en profondeur a pris de l'importance ces derniers temps en raison de son efficacité à évaluer les réseaux Sécurité, le système a permis l'évaluation exhaustive et concluante de la sécurité du réseau. Il est à noter que les méthodes traditionnelles de sécurité des réseaux ne fonctionnent de plus en plus efficacement en raison du traitement accru des données. Néanmoins, **l'apprentissage en profondeur (ML)** a révolutionné l'évaluation des défis en matière de sécurité des réseaux, le système utilise plusieurs approches pour détecter les anomalies dans le système, notamment la détection des anomalies, l'identification du trafic, néanmoins, le système est confronté à certaines limitations, notamment le caractère sacré des données utilisées pour générer des entrées et des sorties. De même, de nouvelles méthodes d'apprentissage en profondeur gagnent du terrain en raison de la nécessité d'une évaluation des données plus rapide et efficace, les techniques de croyance profonde et de codage approfondi ont permis l'analyse de grands ensembles de données et une analyse système plus approfondie, respectivement.

CHAPITRE IV Implantation et des Outils de Développement

Outil de Développement

Introduction

Les systèmes de détection d'intrusion ont fait l'objet de nombreuses recherches, mais la plupart des changements concernent l'ensemble de données collectées, qui contient de nombreux exemples de techniques d'intrusion telles que la force brute, le déni de service ou même une infiltration à partir d'un réseau, à mesure que les comportements et les modèles de réseau changent et que les intrusions évoluent, il est devenu indispensable de passer des ensembles de données statiques et ponctuels à des ensembles de données générés de manière plus dynamique, qui non seulement reflètent la composition du trafic et les intrusions de cette époque, mais sont également modifiables et extensibles, et reproductible, Pour ce faire, nous allons former un modèle d'apprentissage en profondeur afin d'identifier une anomalie à partir d'un ensemble de données donné, en raison de l'application de l'apprentissage automatique (Machine Learning) au sein du système, la détection basée sur les anomalies est rendue la plus efficace parmi les systèmes de détection d'intrusion, car ils n'ont pas besoin de rechercher un motif d'anomalie spécifique, mais plutôt de traiter tout ce qui ne correspond pas au profil comme « anormal ».

IV.1- An End-to-End Open Source Machine Learning Platform (TensorFlow)

IV.1.1- Qu'est-ce que la plateforme d'intelligence machine TensorFlow?

En savoir plus sur la bibliothèque open source développée par Google pour l'apprentissage par la machine et la recherche sur les réseaux de neurones profonds.

A. TensorFlow **TensorFlow** est une bibliothèque de logiciels open source pour le calcul numérique à l'aide de graphiques de flux de données. Il a été développé à l'origine par Google Brain Team au sein de l'organisation de recherche Machine Intelligence de Google pour l'apprentissage par la machine et la recherche sur les réseaux neuronaux profonds, mais le système est suffisamment général pour pouvoir s'appliquer à une grande variété d'autres domaines également. Il a atteint la version 1.0 en février 2017 et a poursuivi son développement rapide, avec plus de 21 000 engagements jusqu'à présent, dont de nombreux contributeurs extérieurs.

Cet article présente TensorFlow, sa communauté open source et son écosystème, et présente quelques modèles intéressants à source ouverte **TensorFlow**.

TensorFlow est Multi-Plateforme. Il fonctionne sur presque tout les processeurs graphiques et les processeurs, y compris les plates-formes mobiles et intégrées, et même les unités de traitement de tenseurs (TPU), qui sont du matériel spécialisé permettant d'effectuer des calculs de tenseurs. Ils ne sont pas encore largement disponibles, mais nous avons récemment lancé un programme alpha.

Figure.IV.1. Logo TensorFlow.

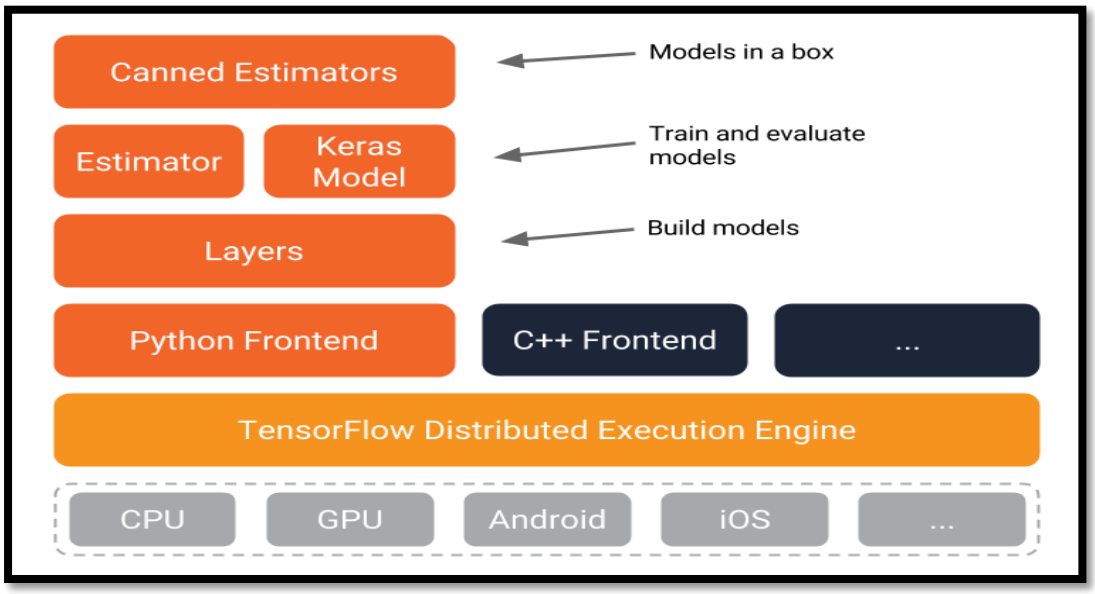


Figure.IV.2. Hiérarchie des TensorFlow. L'API Estimators est au sommet.

Le moteur d'exécution distribué TensorFlow extrait les nombreux périphériques pris en charge et fournit un cœur hautes performances implémenté en C ++ pour la plate-forme TensorFlow, En plus de cela, vous trouverez les interfaces **Python et C ++** (avec d'autres à venir). L'API de couches fournit une interface plus simple pour les couches couramment utilisées dans les modèles d'apprentissage approfondi.

De plus, des API de niveau supérieur, notamment **Keras** (**plus d'informations sur le site Keras.io**) et l'API Estimator, facilitent la formation et l'évaluation des modèles distribués.

B. TensorFlow in Anaconda

Un certain nombre de méthodes peuvent être utilisées pour installer TensorFlow, telles que l'utilisation de pip pour installer les roues disponibles sur PyPI. L'installation de TensorFlow à l'aide de packages **conda** offre de nombreux avantages, notamment un système de gestion de packages complet, une prise en charge plus étendue de la plate-forme, une expérience GPU plus simple et de meilleures performances du processeur. Ces packages sont disponibles via le référentiel **Anaconda** et leur installation est aussi simple que d'exécuter « conda Install TensorFlow » ou « **conda** Install tensorflow-gpu » à partir d'une interface de ligne de commande.

Figure.IV.3. Logo Keras.

Figure.IV.4. Logo Anaconda.

L'un des principaux avantages de l'installation de TensorFlow à l'aide de conda plutôt que de pip provient du système de gestion des paquets conda. Lorsque TensorFlow est installé à l'aide de conda, conda installe également toutes les dépendances nécessaires et compatibles pour les packages. Cela se fait automatiquement.

Les utilisateurs n'ont pas besoin d'installer de logiciel supplémentaire via les gestionnaires de packages système ou d'autres moyens.

En Autre, n'importe lequel des quelque 1400 packages construits par des professionnels dans le référentiel **Anaconda** peut être installé à côté de TensorFlow afin de fournir un environnement de données complet.

Ces packages sont installés dans un environnement **Conda** isolé dont le contenu n'a aucune incidence sur les autres environnements.

IV.2- Langage de Programmation Python version 3.7 et JetBrains PyCharm Community Edition 2018.3.4

IV.2.1- Langage de Programmation Python version 3.7

(Date de première version 20 février 1991)

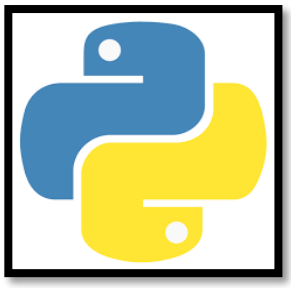


Figure.IV.5. Logo Python v3.7.

Python est un langage de programmation interprété, multi-paradigme et multiplateformes.

Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence BSD4 et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux5, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation6.

A. Utilisation

Python est un langage qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. On l'utilise également comme langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. Il est particulièrement répandu dans le monde scientifique, et possède de nombreuses bibliothèques optimisées destinées au calcul numérique.

B. Bibliothèque standard

Python possède une grande bibliothèque standard, fournissant des outils convenant à de nombreuses tâches diverses. Le nombre de modules de la bibliothèque standard peut être augmenté avec des modules spécifiques écrits en C ou en Python.

La bibliothèque standard est particulièrement bien conçue pour écrire des applications utilisant Internet, avec un grand nombre de formats et de protocoles standards gérés (tels que MIME et HTTP). Des modules pour créer des interfaces graphiques et manipuler des expressions rationnelles sont également fournis. Python inclut également un framework de tests unitaires (unittest, anciennement PyUnit avant version 2.1) pour créer des suites de tests exhaustives.

C. Interfaces graphiques

Python possède plusieurs modules disponibles pour la création de logiciels avec une interface graphique. Le plus répandu est Tkinter. Ce module convient à beaucoup d'applications et peut être considéré comme suffisant dans la plupart des cas. Néanmoins, d'autres modules ont été créés pour pouvoir lier Python à d'autres bibliothèques logicielles (« toolkit »), pour davantage de fonctionnalités, pour une meilleure intégration avec le système d'exploitation utilisé, ou simplement pour pouvoir utiliser Python avec sa bibliothèque préférée. En effet, certains programmeurs trouvent l'utilisation de Tkinter plus pénible que d'autres bibliothèques.

Les principaux modules donnant accès aux bibliothèques d'interface graphique sont Tkinter et Pmw (Python megawidgets)44 pour Tk, wxPython pour wxWidgets, PyGTK pour GTK+, PyQt et PySide pour Qt, et enfin FxPy pour le FOX Toolkit. Il existe aussi une adaptation de la bibliothèque SDL Pygame, un binding de la SFML PySFML, ainsi qu'une bibliothèque écrite spécialement pour Python Pyglet (en).

D. Implémentations du langage

Outre la version de référence, nommée CPython (car écrite en langage C), il existe d'autres systèmes mettant en œuvre le langage Python

- Stackless Python, une version de CPython n'utilisant pas la pile d'appel du langage C ;
- Jython, un interprète Python pour machine virtuelle Java. Il a accès aux bibliothèques fournies avec l'environnement de développement Java ;
- IronPython, un interprète / compilateur (expérimental) pour plateforme .Net / Mono ;
- Brython, une implémentation de Python 3 pour les navigateurs web ;
- MicroPython, variante légère pour microcontrôleurs ;
- PyPy un interprète Python écrit dans un sous-ensemble de Python compilable vers le C ou LLVM ;
- un compilateur (expérimental) pour Parrot, la machine virtuelle de Perl 6 ;
- Shed Skin, un compilateur d'un sous-ensemble de Python produisant du code en C++ ;
- Unladen Swallow (en), une version de CPython optimisée et basée sur LLVM, maintenant abandonnée (la dernière version remonte à octobre 2009).

Ces autres versions ne bénéficient pas forcément de la totalité de la bibliothèque de fonctions écrites en C pour la version de référence, ni des dernières évolutions du langage.

IV.2.2- JetBrains PyCharm Community Edition 2018.3.4

PyCharm Community Edition est un environnement de programmation léger et open source dédié au développement de programmes en Python uniquement.

Ergonomique, l'interface de **PyCharm** est divisée en deux blocs et vous permet d'avoir une vue d'ensemble des projets par arborescence dans la fenêtre de gauche, ainsi que de travailler via un éditeur de code dans l'encart de droite. Par ailleurs, notez que l'éditeur a été pensé dans le but de faciliter la programmation et fait état de fonctionnalités intelligentes telles que la vérification de bugs à la volée, la correction rapide des erreurs, ou encore le nettoyage de votre code.

Figure.IV.6. Logo PyCharm v2018.3.4.

IV.3- Utilisation d'environnements virtuels dans Jupyter Notebook et Python

The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Figure.IV.7. Logo Jupyter Notebook.

IV.4- Description l'Application DL-IDS

C'est en fait le projet, mis en œuvre et réalisé en réseau informatique.

Systèmes de détection d'intrusion ont fait l'objet de nombreuses recherches, mais la plupart des changements concernent l'ensemble de données collectées, qui contient de nombreux exemples de techniques d'intrusion telles que la force brute, le déni de service ou même une infiltration à partir d'un réseau.

À mesure que les comportements et les modèles de réseau changent et que les intrusions évoluent, il est devenu indispensable de passer des ensembles de données statiques et ponctuels à des ensembles de données générés de manière plus dynamique, reflétant non seulement la composition du trafic et les intrusions de cette époque, mais également modifiables, et reproductible.

Pour ce faire, nous allons former un modèle d'apprentissage approfondi afin d'identifier une anomalie à partir d'un ensemble de données donné.

En raison de l'application de l'apprentissage automatique au sein du système, la détection basée sur les anomalies est rendue la plus efficace parmi les systèmes de détection d'intrusion, car ils n'ont pas besoin de rechercher un motif d'anomalie spécifique, mais ils traitent simplement tout ce qui ne correspond pas au profil comme « anormal ».

- **DataSet**

Pour ce faire, nous allons utiliser le jeu de données ISCX 2012 collecté par l'Institut canadien pour la cyber sécurité. Vous pouvez le trouver dans le lien ci-dessous Avant de commencer, le jeu de données réel visualisé en utilisant la bibliothèque Matplotlib pour vérifier la quantité de données contenue dans chaque classe (Normal et Anomalie).

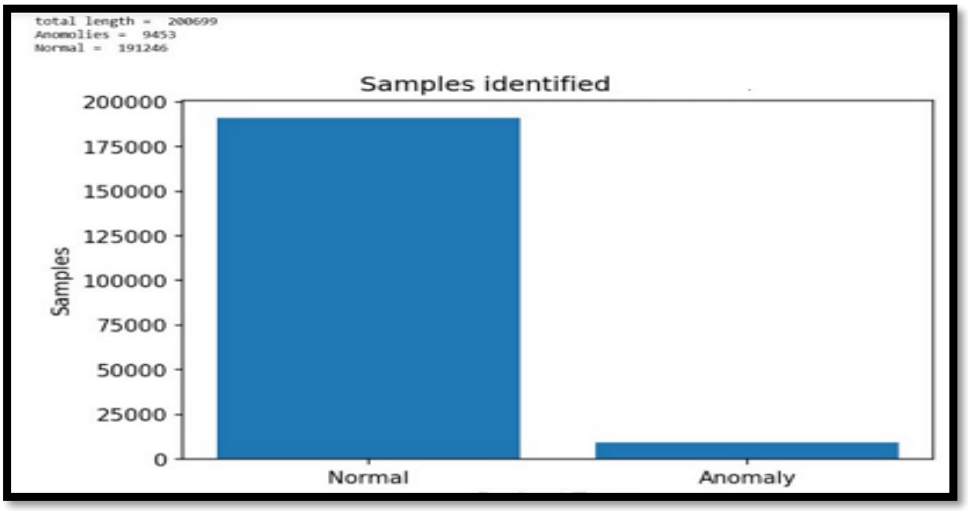


Figure-IV.8. Image visualisant les données (Normal, Anomaly) à l'aide de la bibliothèque Matplotlib

Je devrais mentionner qu'au début de notre projet, nous avions étudié plusieurs articles sur les systèmes de détection d'intrusion utilisant des techniques d'apprentissage automatique et nous avons découvert qu'aucun d'entre eux n'utilisait l'ensemble de données ISCX 2012, probablement en raison de son indisponibilité à l'époque.

Mais nous sommes l'un des premiers à utiliser le jeu de données dans un système de détection d'intrusion.

- **Environment**

J'ai créé un guide très détaillé sur la configuration de votre système pour un apprentissage en profondeur, Si, par hasard, vous utilisez **l'API ANACONDA TENSORFLOW** mais avec un moteur de **back-office Theano ou Microsoft CNTK**, vous êtes également bon. Sinon, veuillez suivre le guide mentionné ci-dessus.

- **Pré-traitement des données**

Une fois téléchargé, l'ensemble de données ISCX est illisible pour le modèle d'apprentissage approfondi lorsqu'il est dans son format de fichier.

PCAP d'origine. Pour changer cela, nous utilisons un logiciel open source appelé ISCX Flowmeter.

Débitmètre ISCX sur GitHub

Ce débitmètre prend les fichiers. PCAP et les convertit en un format de fichier .XML lisible qui empile chaque partie de données en tant que flux dans lesquels le premier paquet détermine les directions avant (source à destination) et arrière (destination vers source).

Voici un exemple de l'ensemble de données récemment converti

```
<?xml version="1.0" encoding="UTF-8"?>
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:noNamespaceSchemaLocation="TestbedMonJun14Flows.xsd" generated="2014-03-11T18:21:14">
<TestbedMonJun14Flows>
<appName>Unknown_UDP</appName>
<totalSourceBytes>16076</totalSourceBytes>
<totalDestinationBytes>0</totalDestinationBytes>
<totalDestinationPackets>0</totalDestinationPackets>
<totalSourcePackets>178</totalSourcePackets>
<sourcePayloadAsBase64></sourcePayloadAsBase64>
<destinationPayloadAsBase64></destinationPayloadAsBase64>
<destinationPayloadAsUTF></destinationPayloadAsUTF>
<direction>L2R</direction>
<sourceTCPFlagsDescription>N/A</sourceTCPFlagsDescription>
<destinationTCPFlagsDescription>N/A</destinationTCPFlagsDescription>
<source>192.168.5.122</source>
<protocolName>udp_ip</protocolName>
<sourcePort>5353</sourcePort>
<destination>224.0.0.251</destination>
<destinationPort>5353</destinationPort>
<startDateTime>2010-06-13T23:57:19</startDateTime>
<stopDateTime>2010-06-14T00:11:23</stopDateTime>
<Tag>Normal</Tag>
</TestbedMonJun14Flows>
```


Figure-IV.9. Fichier XML visualisant les données générateur ISCX Flow Meter.

ISCXFlowMeter est un générateur de flux de trafic réseau disponible à partir d'ici.

Il peut être utilisé pour générer des flux bidirectionnels, où le premier paquet détermine les directions avant (source à destination) et arrière (destination à source), de sorte que les caractéristiques temporelles statistiques peuvent être calculées séparément dans les directions avant et arrière.

Les fonctionnalités supplémentaires incluent la sélection des fonctionnalités dans la liste des fonctionnalités existantes, l'ajout de nouvelles fonctionnalités et le contrôle de la durée de temporisation du flux.

Ensuite, nous concaténons les données utiles de manière à ce que leur longueur soit de 7500.

Une fois cela terminé, nous les avons ensuite reformées en un tableau NumPy de dimensions 50x50x3.

Qui ressemble à ceci quand visualisé

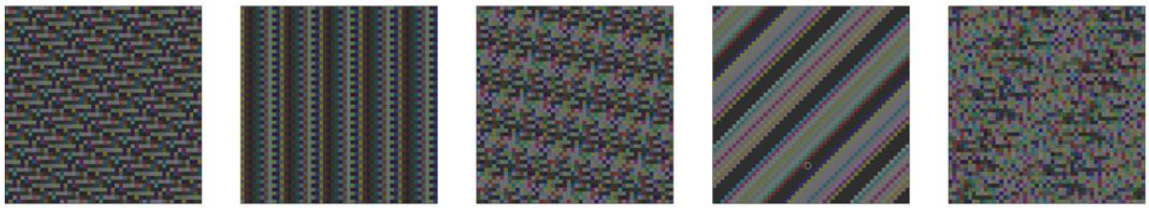


Figure-IV.10. Cinq images de données du jeu de données **ISCX**.

Nous avons ensuite empilé verticalement ces tableaux avec l'ajout d'une colonne contenant l'étiquette de chaque charge utile, c'est-à-dire Normal ou Anomaly.

Une fois que nous avons terminé, nous avons sauvegardé les données dans des fichiers.

NPY afin qu'elles puissent être utilisées comme entrée par le modèle d'apprentissage en profondeur.

- **Intrusion détection évaluation DataSet (ISCXIDS2012)**

En détection d'intrusion réseau (IDS), approches axées sur l'anomalie souffrent en particulier d'évaluation précise, de comparaison et de déploiement qui provient de la rareté des ensembles de données adéquates. Beaucoup ces ensembles de données à l'interne et ne peuvent être partagées en raison de problèmes de la vie privée, d'autres sont fortement rendues anonymes et ne reflètent pas les tendances actuelles, ou ils n'ont pas certaines caractéristiques statistiques. Ces carences sont principalement les raisons pour lesquelles un dataset parfait encore d'exister. Ainsi, les chercheurs doivent avoir recours aux ensembles de données qu'elle peut recevoir qui sont souvent sous-optimale.

À ISCX, une approche systématique pour générer les ensembles de données requis est introduite pour répondre à ce besoin. L'idée sous-jacente est basée sur le concept de profils qui contiennent des descriptions détaillées des intrusions et des modèles de distribution abstraite pour les applications, des protocoles ou des entités réseau de niveau inférieures. Traces de Real sont analysés pour créer des profils pour les agents qui génèrent des conditions de circulation réelles pour HTTP, SMTP, SSH, IMAP, POP3 et FTP. À cet égard, un ensemble de directives est établi pour décrire des ensembles de données valide, qui établit les fondements permettant de générer des profils. Ces lignes directrices sont essentielles pour l'efficacité de l'objet dataset en termes de réalisme, capacités d'évaluation, capture totale, l'exhaustivité et les activités malveillantes.

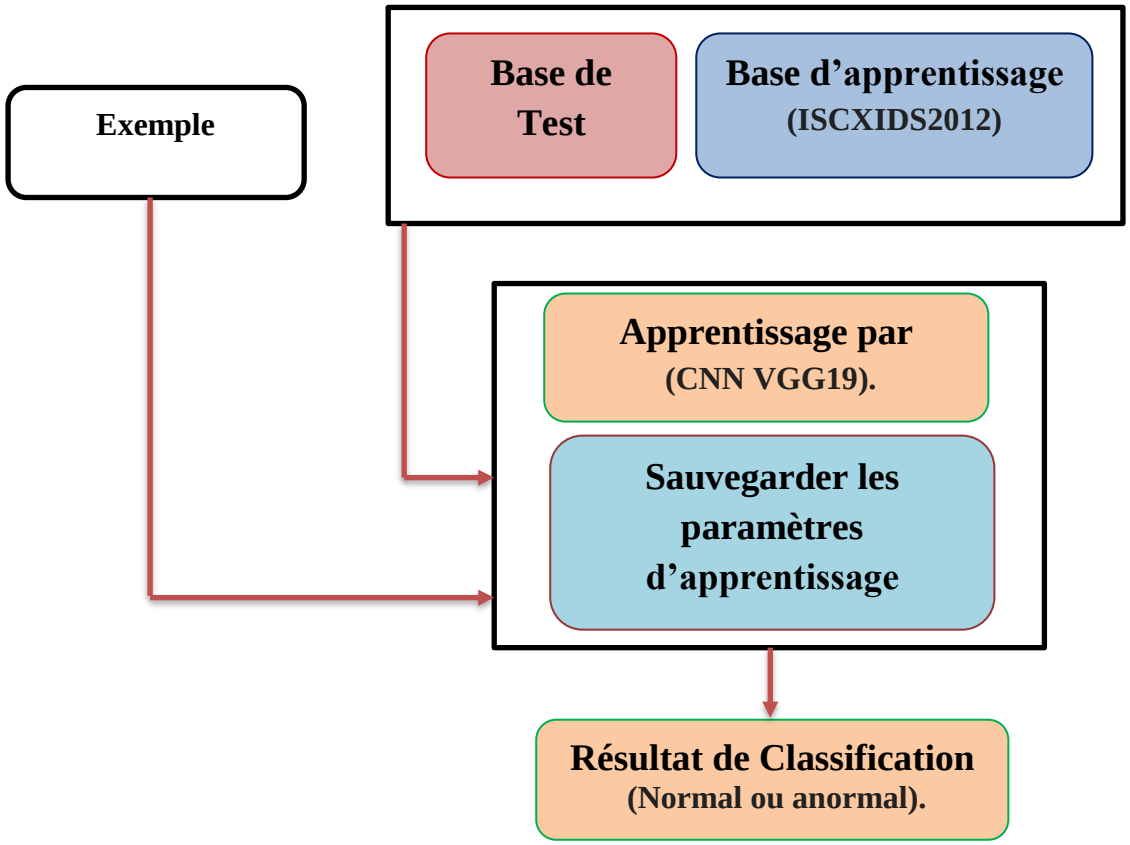


Figure-IV.11. Architecture des IDS basé sur les techniques d'apprentissage automatique.

- **Le Modèle**

Nous avons essayé diverses expériences avec différents modèles, mais celui que nous avons trouvé avec une précision satisfaisante est l'utilisation de la technique d'apprentissage par transfert sur le modèle pré-formé **Pretrained VGG-19 convolutional neural network (CNN VGG19).**

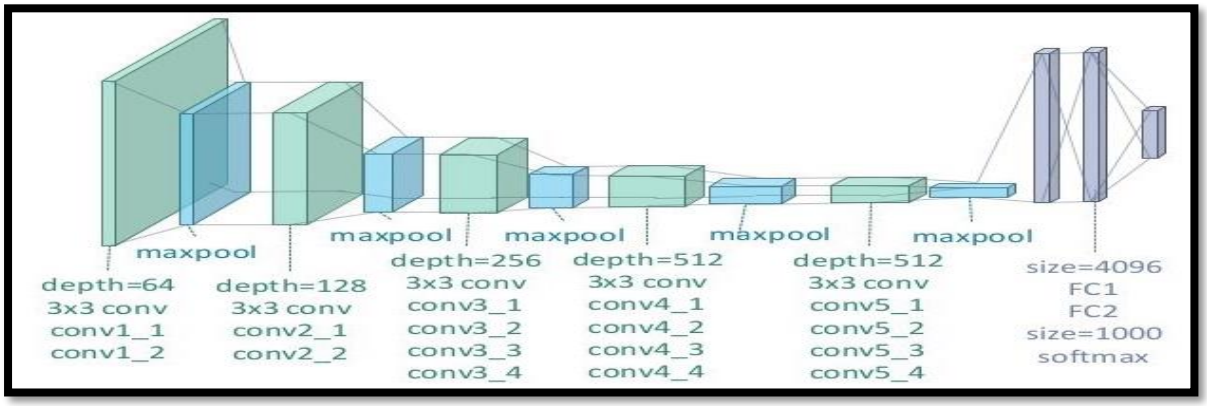


Figure-IV.12. Exemple disposition du modèle NetLayers **CNN VGG19**.

Affichez l'architecture du réseau à l'aide de la propriété Layers. Le réseau compte 47 couches. Il y a 19 couches avec des poids pouvant être appris 16 couches de convolution et 3 couches entièrement connectées.

47x1 Layer array with layers

1	'input'	Image Input	224x224x3 images with 'zerocenter' normalization
2	'conv1_1'	Convolution	64 3x3x3 convolutions with stride [1 1] and padding [1 1 1 1]
3	'relu1_1'	ReLU	ReLU
4	'conv1_2'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
5	'relu1_2'	ReLU	ReLU
6	'pool1'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
7	'conv2_1'	Convolution	128 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
8	'relu2_1'	ReLU	ReLU
9	'conv2_2'	Convolution	128 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]
10	'relu2_2'	ReLU	ReLU
11	'pool2'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
12	'conv3_1'	Convolution	256 3x3x128 convolutions with stride [1 1] and padding [1 1 1 1]
13	'relu3_1'	ReLU	ReLU
14	'conv3_2'	Convolution	256 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
15	'relu3_2'	ReLU	ReLU
16	'conv3_3'	Convolution	256 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
17	'relu3_3'	ReLU	ReLU
18	'conv3_4'	Convolution	256 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
19	'relu3_4'	ReLU	ReLU
20	'pool3'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
21	'conv4_1'	Convolution	512 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
22	'relu4_1'	ReLU	ReLU
23	'conv4_2'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding [1 1 1 1]
24	'relu4_2'	ReLU	ReLU
25	'conv4_3'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding [1 1 1 1]
26	'relu4_3'	ReLU	ReLU
27	'conv4_4'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding [1 1 1 1]
28	'relu4_4'	ReLU	ReLU
29	'pool4'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
30	'conv5_1'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding [1 1 1 1]
31	'relu5_1'	ReLU	ReLU
32	'conv5_2'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding [1 1 1 1]
33	'relu5_2'	ReLU	ReLU
34	'conv5_3'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding [1 1 1 1]
35	'relu5_3'	ReLU	ReLU
36	'conv5_4'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding [1 1 1 1]
37	'relu5_4'	ReLU	ReLU
38	'pool5'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
39	'fc6'	Fully Connected	4096 fully connected layer
40	'relu6'	ReLU	ReLU
41	'drop6'	Dropout	50% dropout
42	'fc7'	Fully Connected	4096 fully connected layer
43	'relu7'	ReLU	ReLU
44	'drop7'	Dropout	50% dropout
45	'fc8'	Fully Connected	1000 fully connected layer
46	'prob'	Softmax	softmax.
47	**'output'**	**Classification Output**	**crossentropyex with 'tench' and 999 other classes.**

Figure-IV.12. Exemple disposition du modèle NetLayers **CNN VGG19**.

Pour afficher les noms des classes apprises par le réseau, vous pouvez afficher la propriété Classes de la couche de sortie de la classification (la couche finale). Affichez les 10 premières classes en spécifiant les 10 premiers éléments.

N'avons pas utilisé les poids pré-entraînés de **VGG-19** car nos images étaient des images personnalisées.

```
x = model_vgg19_conv.output
x = Flatten()(x)
x = Dense(128, activation='relu')(x)
x = Dense(1, activation='sigmoid', name='predictions')(x)

my_model = Model(input=model_vgg19_conv.input, output=x)
my_model.summary()
```

Figure-IV.13. Code Source pré-entraînés activation (ReLU, Sigmoid) CNN **VGG19**.

La Résultat Après l'exécution ce code Source en **anaconda Jupyter**

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 50, 50, 3)	0
block1_conv1 (Conv2D)	(None, 50, 50, 64)	1792
block1_conv2 (Conv2D)	(None, 50, 50, 64)	36928
block1_pool (MaxPooling2D)	(None, 25, 25, 64)	0
block2_conv1 (Conv2D)	(None, 25, 25, 128)	73856
block2_conv2 (Conv2D)	(None, 25, 25, 128)	147584
block2_pool (MaxPooling2D)	(None, 12, 12, 128)	0
block3_conv1 (Conv2D)	(None, 12, 12, 256)	295168
block3_conv2 (Conv2D)	(None, 12, 12, 256)	590080
block3_conv3 (Conv2D)	(None, 12, 12, 256)	590080
block3_conv4 (Conv2D)	(None, 12, 12, 256)	590080
block3_pool (MaxPooling2D)	(None, 6, 6, 256)	0
block4_conv1 (Conv2D)	(None, 6, 6, 512)	1180160
block4_conv2 (Conv2D)	(None, 6, 6, 512)	2359808
block4_conv3 (Conv2D)	(None, 6, 6, 512)	2359808
block4_conv4 (Conv2D)	(None, 6, 6, 512)	2359808
block4_pool (MaxPooling2D)	(None, 3, 3, 512)	0
block5_conv1 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv2 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv3 (Conv2D)	(None, 3, 3, 512)	2359808
block5_conv4 (Conv2D)	(None, 3, 3, 512)	2359808
block5_pool (MaxPooling2D)	(None, 1, 1, 512)	0

Figure-IV.14. Résultat pré-entraînés activation (ReLU, Sigmoid) CNN **VGG19**.

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Ahmed Draia - Adrar

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

Option Systèmes intelligents.

Thème

Développement d'une Architecture Basée sur l'Apprentissage Profond (Deep Learning) pour la Détection d'Intrusion dans les Réseaux.

Préparé par

Mr. MIMOUNE Zakarya.

Encadré par

Mr. OUAHAB Abdelwhab.

Soutenu Publiquement le 01/07/2019.

Devant le jury

Président	**Mr CHERAGUI Mohamed Amine.**	**M. A. A**	**Univ. Adrar**
Promoteur	**Mr DEMRI Mohamed.**	**M. A. A**	**Univ. Adrar**
Examinateur	**Mr KABOU Salah El Dine.**	**M. C. B**	**Univ. Adrar**

Année Universitaire 2018/2019.

Nous avons donc choisi d'entraîner le modèle sur notre ensemble d'images en utilisant la technique conventionnelle de détection des anomalies, qui comprend deux phases

1. **Phase de formation** phase au cours de laquelle un profil des charges utiles normales est créé.

   Normalement, aucune donnée d'anomalie n'est nécessaire pour la formation, car l'IDS éliminera instantanément toutes les données présentant le plus petit écart par rapport aux données du profil lors de la phase suivante.

2. **Phase de test** la phase à laquelle les charges utiles entrantes sont comparées aux données stockées dans le profil.

   Ce faisant, nous avons formé notre modèle aux valeurs de données dont les balises étaient étiquetées comme étant normales, ce qui a permis d'obtenir une précision de 100% dès la toute première époque. Toutefois, avec l'introduction des valeurs de données d'anomalie, le modèle a obtenu une précision incorrecte de 100%, ce qui a permis de conclure que le modèle identifiait correctement les données normales.

   Toutefois, dans le cas d'anomalies, il les contournait simplement sans les classer.

   Nous avons donc eu recours à la fois aux données Anomalie et Normale dans nos données d'entraînement. En raison de la nature des données, il devient évident qu'il s'agit d'un problème de détection des valeurs aberrantes. Au cours de la phase de test, nous avons obtenu des résultats satisfaisants.

   Le modèle a pu identifier les données normales avec une précision de 100% et les données d'anomalies avec une précision de 85%.

   Compte tenu de notre ensemble de données, les résultats obtenus sont satisfaisants

```
training_loss= hist.history['loss']
val_loss= hist.history['val_loss']
training_acc= hist.history['acc']
val_acc= hist.history['val_acc']
xc=range(epochs)

plt.figure(1,figsize=(7,5))
plt.plot(xc,training_loss)
plt.plot(xc,val_loss)
plt.xlabel('No. of Epochs')
plt.ylabel('loss')
plt.title('Training Loss vs Validation Loss')
plt.grid(True)
plt.legend(['Train','Val'])
```

Figure-IV.15. Code Source du **Phase de Validation et Phase de Training**.

La Résultat Après l'exécution code Source ci-dessus en **anaconda Jupyter**

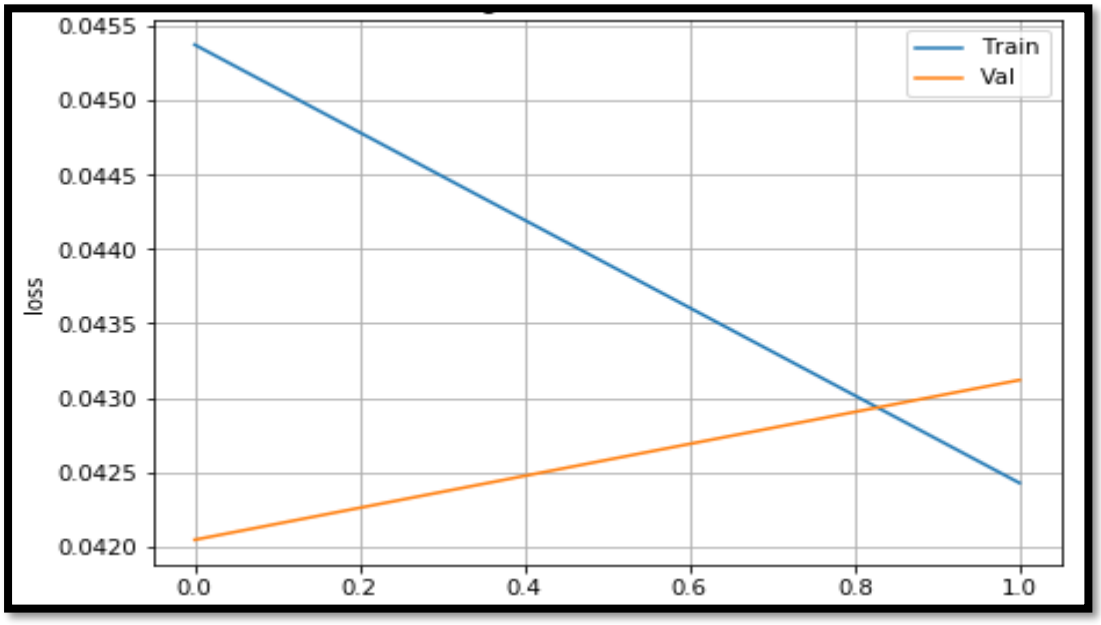


Figure-IV.16. Graphique de **précision** du modèle à 2 époques (**Training, Validation**).

```
import sklearn.metrics
target_names = ['normal','anomaly']
print(sklearn.metrics.classification_report(y_test,rounded,labels = [0,1], targe
```

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

Figure-IV.17. Résultat de Classification (Normal, Anormal).

Conclusion

En raison de la demande urgente d'un système IDS efficace en matière de sécurité du réseau, les chercheurs s'efforcent d'identifier des approches améliorées, ce travail montre à quel point le jeu de données KDD1999 est très utile pour tester différents classificateurs, Les travaux se concentrent sur la phase de prétraitement de KDD1999 afin de préparer des expériences fiables et des données de test indépendantes randomisées, Parmi les techniques de classification (**J48, MLP et Bayes Network**), le classifier J48 a atteint le taux de précision le plus élevé pour la détection et la classification de tous les types d'attaques de jeux de données KDD99 (DOS, R2L, U2R), le jeu de données KDD99 a 41 attributs et tous ont été enregistrés.

Conclusion Générale

Technique Deep Learning présente non seulement une grande capacité de modélisation pour la détection d'intrusion (IDS), mais également une grande précision dans la classification utilisant la technique Machine Learning à partir le mécanisme de Réseau de Neurone Convolution (CNN), Par rapport aux méthodes de classification traditionnelles, telles que J48, la forêt naïve bayésienne et la forêt aléatoire, la performance obtient un taux de précision et un taux de détection plus élevés avec un taux de faux faible, en particulier dans le cadre de la tâche de classification multiclassée sur le jeu de données KDD99, le modèle peut effectivement améliorer à la fois la précision de la détection d'intrusion et la capacité de reconnaître le type d'intrusion, Bien entendu, dans les recherches futures, nous veillerons toujours à réduire le temps d'entraînement grâce à l'accélération GPU, à éviter les gradients explosifs et nuls et à étudier les performances de classification du LSTM, algorithme de RNN bidirectionnel dans le domaine de la détection d'intrusion.

ANNEXE

CHAPITRE I. SECURITE des SYSTEMES INFORMATIQUE et IDS

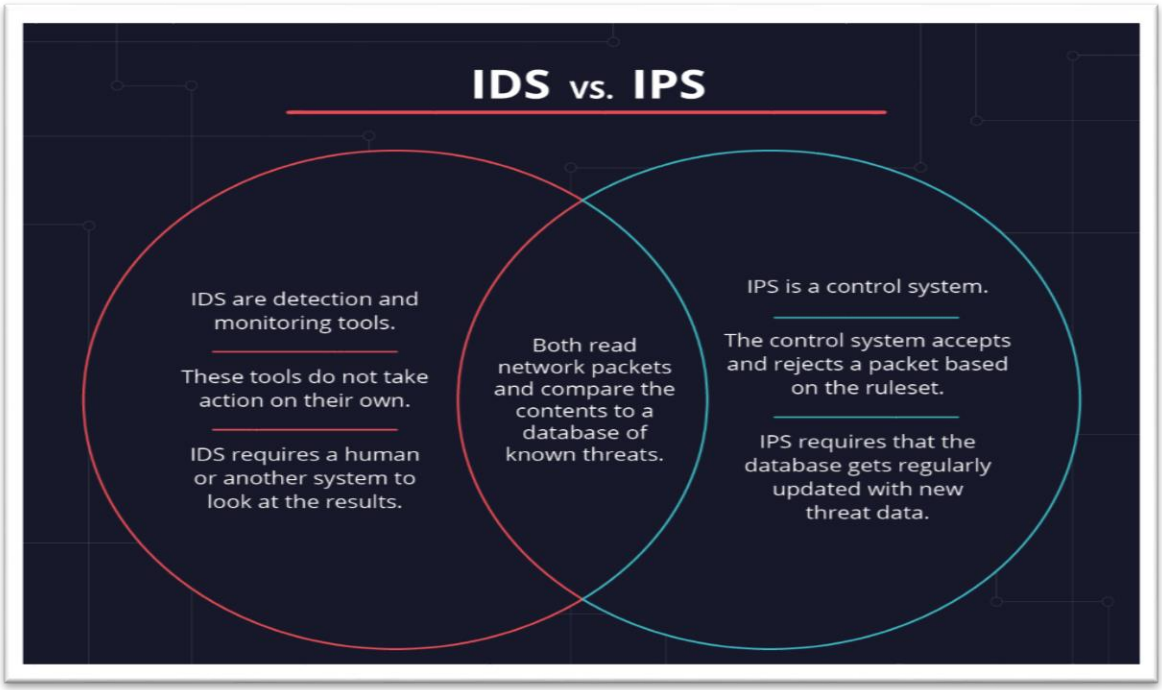


Figure.A.1. Difference between IDS to IPS.

CHAPITRE II. DEEP LEARNING (Apprentissage Profondeur) ……

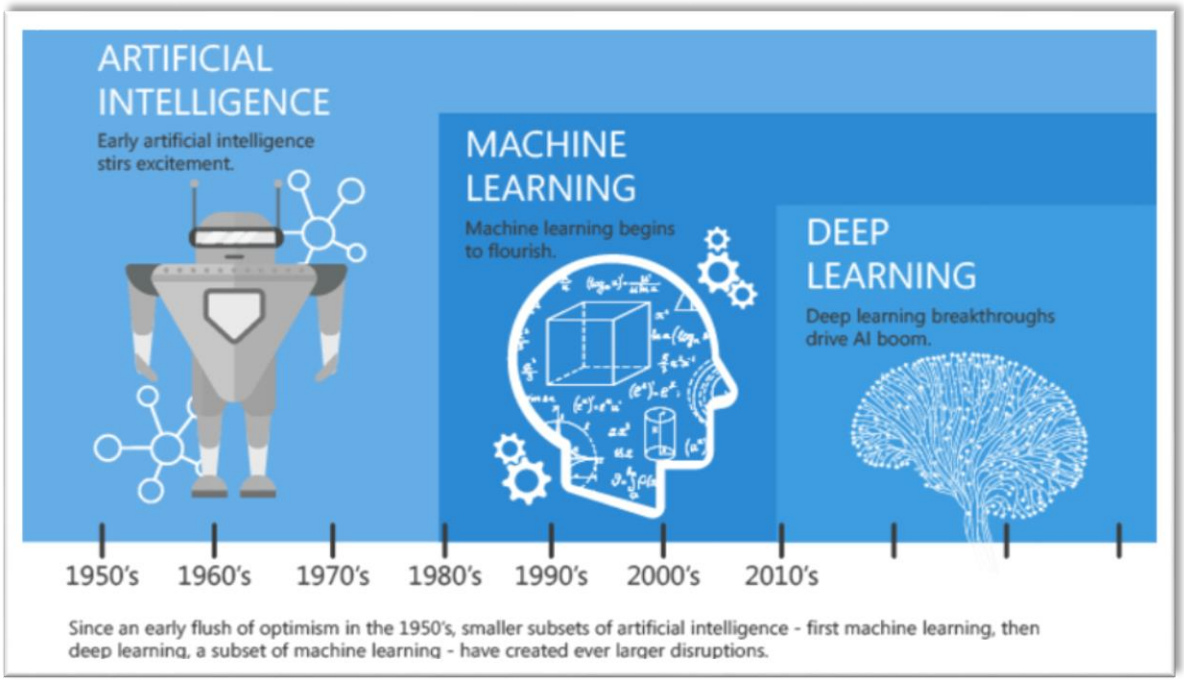


Figure.A.2. Difference between Machine Learning and Artificial Intelligence.

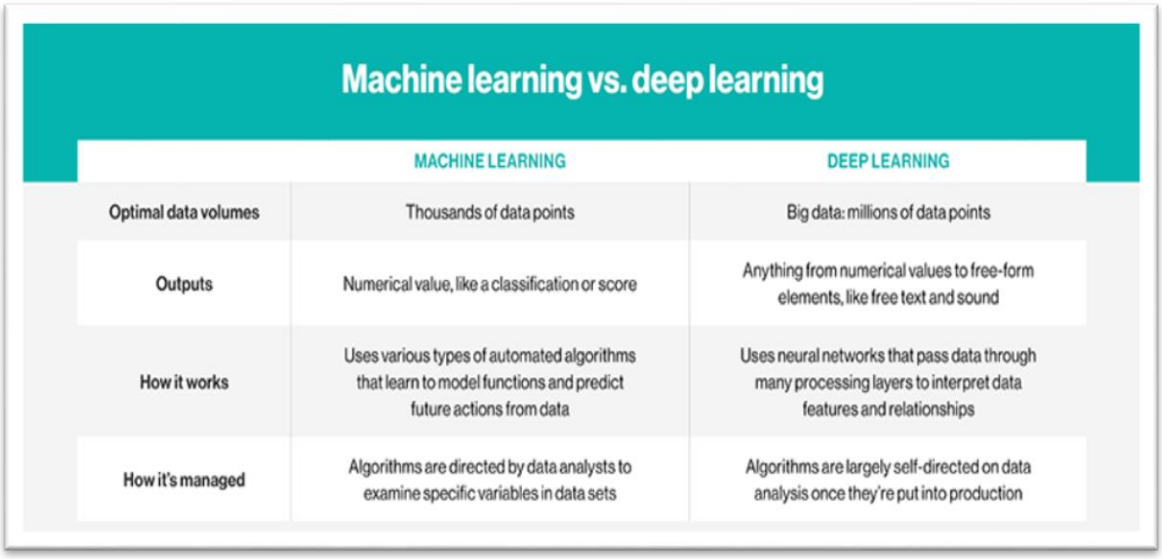


Figure.A.3. Difference between Machine Learning and Deep Learning.

CHAPITRE III. LES SYSTEMES DE DETECTION D'INTRUSION BASES SUR LA M.L

KDD is short for Data Mining and Knowledge Discovery. KDD CUP is an annual competition organized by SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining) of ACM (Association for Computing Machine). The contest homepage is here.

The following are the topics of previous KDDCUP

1. **KDD-Cup 2008**, Breast cancer
2. **KDD-Cup 2007**, Consumer recommendations
3. **KDD-Cup 2006**, pulmonary embolisms detection from image data
4. **KDD-Cup 2005**, Internet has search query categorization
5. **KDD-Cup 2004**, Particle physics; plus Protein homology prediction
6. **KDD-Cup 2003**, Network mining and usage log analysis
7. **KDD-Cup 2002**, BioMed document; plus Gene role classification
8. **KDD-Cup 2001**, Molecular bioactivity; plus Protein locale prediction.
9. **KDD-Cup 2000**, online retailer website clickstream analysis
10. **KDD-Cup 1999, Computer network intrusion detection**
11. **KDD-Cup 1998**, Direct marketing for profit optimization
12. **KDD-Cup 1997**, Direct marketing for lift curve optimization

N.B "KDD CUP 99 dataset" is the data set used by the KDD competition in 1999.

Download the KDD99 dataset from here

http//kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

This is a repository for data collected and developed through research at the ISCX. The following datasets are currently available

ISCX data set (Use this DATASET)
https//iscxdownloads.cs.unb.ca/iscxdownloads/ISCX-IDS-2012/#ISCX-IDS-2012
Botnet data set
Application level DoS data set
Android validation data set.

Liste Les Abréviations

ACL (Access Control List).

AdaGrad (AdaGrad Algorithm).

ALIDS (Active Learning Intrusion Detection System).

BPN (Back Propagation Networks).

CNNs (Convolutional Neural Network).

CPOOL (Couche Pooling).

DBN (Deep Belief Network)**.**

DL (Deep Learning).

DMZ (Demilitarized Zone).

DNN (Deep Network Neural).

FC (Couche Fully Connected).

HIDS (Host Intrusion Detection Systems).

HIPS (Host Intrusion & Prevention System).

HTTP (Hypertext Transfer Protocol).

IDS (Intrusion Detection Systems).

ICMP (Internet Control Message Protocol).

IP (Internet Protocol).

IPS (Intrusion Prevention System).

KDD cup 1999 data (Knowledge Discovery and Data Mining).

MAE (Mean Absolute Error).

MIME (Multipurpose Internet Mail Extensions).

ML (Machine Learning).

MLP (Multilayer Perceptron).

ML-DS (Machine Learning-Intrusion Detection).

MNIST (Base de Données Mixed National Institute of Standards and Technology).

MSE (Mean Squared Error).

NIDS (Network Intrusion Detection System).

NN (Neural Network).

R2L (Remote-to-Local).

RBF (Radial Basis Function).

RBMs (Restricted Boltzmann Machine).

ReLU (Unité de Rectification Linéaire).

RNN (Recurrent Neural Network).

RVB (Red Vs Blue).

SGD (Stochastic Gradient Descent).

SMOTe (Synthetic Minority Over-sampling Technique).

SNMP (Simple Network Management Protocol).

SVM (Support Vector Machines).

TPU (Tensor Processing Unit).

UDP (User Datagram Protocol).

U2R (User-to-Root).

XML (Extensible Markup Language).

Références Bibliographies

[1] DI GALLO Frédéric COURS DE RESEAUX ET SYSTEMES Cycle Probatoire CNAM BORDEAUX 1999-2000, page 189-200.

[2] DI GALLO Frédéric COURS DE RESEAUX ET SYSTEMES Cycle Probatoire CNAM BORDEAUX 1999-2000, page 191-192.

[3] Bernard Cousin Université de Rennes 1, COURS Sécurité des réseaux, Informatiques, année 2000-2001, pages 04-05.

[4] Bernard Cousin Université de Rennes 1, COURS Sécurité des réseaux, Informatiques, année 2000-2001, page 05.

[5] Daniel Guinier, Sécurité et qualité des systèmes d'information - Approche systémique, Masson, 1992

[6] Laurent Bloch et Christophe Wolfhugel, Sécurité informatique - Principes et méthode, Eyrolles, 2011

[7] Fernandez-Toro, Management de la sécurité de l'information. Implémentation ISO 27001 et audit de certification, Eyrolles, 2012

[8] Bernard Foray, La fonction RSSI (Responsable Sécurité Système d'Information) - Guide des pratiques et retours d'expérience - 2e édition, Dunod, 2011.

[9] P Biondi, Liyun Li IR5, COURS DE IDS, système de détection d'intrusions 2000-2001, page 04-16.

[10] Laurent Bloch et Christophe Wolfhugel, Sécurité informatique - Principes et méthodes à l'usage des DSI, RSSI et administrateurs, Eyrolles, 2009.

[11] (fr) OCTO Technology, ouvrage collectif, Gestion des Identités Une Politique pour le Système d'Information, OCTO Technology, 2007, (ISBN 2952589518).

[12] (fr) Vuibert Sciences, Guinier D. - Chapitre La politique de sécurité, p. 1486-1498, in l'encyclopédie de l'informatique et des systèmes d'information, 2088 pages, Vuibert Sciences, 2006, (ISBN 9782711748464).

[13] Bishop, Matt (2004). Computer security art and science, Addison-Wesley.

[14] McLean, John. (1994). Security Models. Encyclopedia of Software Engineering 2 1136–1145. New York John Wiley & Sons, Inc.

[15] (en) liste de failles connues et non encore officiellement corrigées pour chaque logiciel, par Secunia [archive].

[16,17,18,19,20] P Biondi, Liyun Li IR5, COURS DE IDS, système de détection d'intrusions 2000-2001, page 12-27.

[21] Une source pour la traduction en apprentissage profond Yoshua Bengio, « Introduction aux algorithmes d'apprentissage profonds », sur Université de Montréal.

[21] Kumar, Shantanu. "A Survey of Deep Learning Methods for Relation Extraction." arXiv preprint arXiv1705.03645 (2017)

[22] J. Zhou et O. G. Troyanskaya (2015), « Predicting effects of noncoding variants with deep learning-based sequence model », Nature Methods, 12(10), 931-934.

[23] Srivastava, Nitish, et al. "Dropout a simple way to prevent neural networks from overfitting." Journal of machine learning research (2014).

[24] Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of Machine Learning Research, Feb (2012)

[25] Kim, Y. "Convolutional Neural Networks for Sentence Classification", EMNLP (2014)

[26] Severyn, Aliaksei, and Alessandro Moschitti. "UNITN Training Deep Convolutional Neural Network for Twitter Sentiment Classification." SemEval@ NAACL-HLT (2015)

[27] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." EMNLP (2014).

[27] Ilya Sutskever et al. "Sequence to sequence learning with neural networks." NIPS (2014).

[28] Zhang, D., Wang, D. "Relation classification via recurrent NN." -arXiv preprint arXiv1508.01006 (2015).

[28] Zeng, D.et al. "Relation classification via convolutional deep neural network". COLING (2014)

[29]Bahdanau et al. "Neural machine translation by jointly learning to align and translate." ICLR (2015)

[30] Gal, Y., Islam, R., Ghahramani, Z. "Deep Bayesian Active Learning with Image Data." ICML (2017)

[31] Nair, V., Hinton, G.E. "Rectified linear units improve restricted boltzmann machines." ICML (2010)

[32] Ronan Collobert, et al. "Natural language processing (almost) from scratch." JMLR (2011)

[33] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, July 2006, pp. 1527-1554, doi10.1162/neco.2006.18.7.1527.

[34] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Bayesian event classification for intrusion detection," Proc. 19th Annual Computer Security Applications Conference, Dec. 2003, pp. 14-23, doi10.1109/CSAC.2003.1254306.

[34] C. Sinclair, L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection," Proc. 15th Annual Computer Security Applications Conference, Dec. 1999, pp. 371-377, doi10.1109/CSAC.1999.816048.

[34] J. Zhang and M. Zulkernine, "A hybrid network intrusion detection technique using random forests," Proc. First International Conferenceon Availability, Reliability and Security (ARES'06), April 2006, pp.8-16, doi10.1109/ARES.2006.7.

[35] J. Yang, J. Deng, S. Li, and Y. Hao, "Improved traffic detection with support vector machine based
on restricted Boltzmann machine," Soft Computing, vol. 19, Dec. 2015, pp. 1-12, doi10.1007/s00500-015-1994-9.

[36] C. Sinclair, L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection," IEEE Proc. 15th Annual Computer Security Applications Conf., pp. 371-377, 1999.

[37] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," Proc. Intl. Joint Conf. Neural Networks, vol. 2, pp. 1702-1707, 2002.

[38] S. H. Adil, et al., "An improved intrusion detection approach using synthetic minority over-sampling technique and deep belief network," Frontiers in Artificial Intelligence and Applications, vol. 265, pp. 94-102, doi10.3233/978-1-61499-434-3-94.

[39] J. R. Quinlan, "C4. 5 Programs for machine learning," Machine Learning, vol. 16, Sep. 1993, pp. 235-240, doi10.1007/BF00993309.

[**40**] G. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," Proc. UAI'95 Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995, pp. 338-345, http//dl.acm.org/citation.cfm?id=2074158.2074196 .

[41] C. Chang and C. Lin, "LIBSVM A library for support vector machines," 2001, Software available at http//www.csie.ntu.edu.tw/cjlin/libsvm.

[42] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, (2003). "Bayesian event classification for intrusion detection," IEEE Proc. 19th Annual Computer Security Applications Conf., pp. 14-23, 2003.

References Web

- http//igm.univ-mlv.fr/~dr/XPOSE2009/Sonde_de_securite_IDS_IPS/IDS.html**.CHI.I**
- https//fr.wikipedia.org/wiki/R%C3%A9seau_neuronal_convolutif#Pooling_stochastique **Page 39, Page 42**
- https//github.com/ISCX/ISCXFlowMeter **: page 65.**
- http//deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution. **Page 40-41.**
- http//www.natural-solutions.eu/blog/la-reconnaissance-dimage-avec-les-rseaux-de-neurones-convolutifs.**Page 40-41.**
- https//github.com/tamimmirza/Intrusion-Detection-System-using-Deep-Learning. :**Page 65**.