République Algérienne Démocratique et Populaire

Minstère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Bejaia

Mémoire de fin de cycle

En vue d'obtention du diplôme de Master en informatique

Option : Réseaux et Système Distribution

Thème :

Les systèmes de détections d'intrusion basés sur machine learning

Réalisé par :

Mr ABIZA Imad.

Soutenu le 22 septembre 2018 devant le jury composé de :

Président	M^{me} TAHAKOURT Zineb	M.A.A	U.A/Mira Béjaïa, Algérie.
Examinateur	Mr OUZEGANE Redouane	M.A.A	U.A/Mira Béjaïa, Algérie.
Examinatrice	M^{me} BENNAI Soufia	doctorante	U.A/Mira Béjaïa, Algérie.
Encadreur	Mr AMROUN Kamal	M.C.A	U.A/Mira Béjaïa, Algérie.

2017/2018

DÉDICACES

Je Dédié Ce Modeste Travail a :

Toute Ma Grande Famille

« Pour Leurs Encouragements Continu Et Leurs Soutien »

Mes Amis

« Dont La Liste Est Longue »

REMERCIEMENT

Nous remercions avant tout, dieu tout-puissant qui nous a donné la force, le courage et la volonté pour réaliser ce travail.

Un grand merci à toutes nos familles pour leurs préoccupations et le souci qu'ils se sont fait pour nous, leur Encouragement et leur suivi, avec patience, du déroulement de notre projet. Nous tenons à remercier notre Promoteur MR AMRON KAMAL De nous avoir acceptés durant cette année, pour son suivi ses orientations.

Nous considération et notre gratitude s'adressent au personnel madame KHARBACH MARIEM pour son aide et collaboration, et tous ceux qui nous ont aidés durant notre année est ces précieux, et qui a su nous faire profiter de sa grande expérience.

Nous tenons également à remercier les membres de jury d'avoir accepté de juger notre travail.

Enfin nous remercions de tout cœur tous ceux qui contribué de près ou soin à la réalisation de ce travail

Résumé

Ce travail illustre les différentes techniques utilisées dans les systèmes de détection d'intrusion. Les techniques présentées sont : datamining, machine Learning et techniques statiques. Ensuite, il présente trois méthodes de la technique machine Learning suivantes : SVM, Les réseaux de neurones et K-means, puis il établit une comparassions entre ces trois méthodes basé sur le taux de précision en utilisant la base de données NSL-KDD.

Mots-clés : les systèmes de détection d'intrusion, machine Learning, datamining, SVM, Les réseaux de neurones, K-means.

Abstract

This work illustrates the different techniques used in intrusion detection systems. The techniques presented are : datamining, machine learning and static techniques. Next, he presents three methods of the following machine learning techniques: SVM, Neural Networks, and K-means, and then establishes a comparison between these three methods based on precision rate using the NSL-KDD database.

Liste des figures

Liste des tableaux

Introduction générale

Les réseaux et les systèmes informatiques sont devenus des outils indispensables au fonctionnement des entreprises. Ils sont aujourd'hui déployés dans tous les secteurs professionnels : les universités, les banques, les assurances ou encore le domaine militaire.

L'informatique gérée par ces systèmes fait l'objet de convoitises. Elle peut être exposée à des attaques qui exploitent des éléments vulnérables du système d'information. La détection des actions malveillantes est rapidement devenue une nécessité. Les mesures de prévention se sont révélées insuffisantes et ont amené la création de systèmes de détection d'intrusions (IDS : Intrusion Détection systèmes).

Une intrusion est définit comme étant toute tentative pouvant nuire à l'intégralité, la confidentialité ou la disponibilité dans le réseau ainsi que toute tentative visant à contourner les dispositifs de sécurité mis en place sur le réseau ou une machine. Ces tentative d'intrusions peuvent être bénignes comme extrêmement dangereuses et préjudiciable pour l'entreprise.

Le domaine de la détection d'intrusion est encore jeune mais en plein développement .Nous dénombrons a l'heure actuelle environ une centaine de systèmes de détections d'intrusions (ou IDS pour Intrusion Detection System), que ce soit des produits commerciaux ou du domaine public .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.

Dans notre travail on s'intéresse sur les différentes techniques utilisées dans la détection d'intrusion, généralement ces dernières sont des techniques de datamining ou de machine Learning et technique statique

Notre mémoire est organisé comme suit :

Le premier chapitre est consacré à la présentation des différents aspects de la sécurité informatique et les systèmes de détection d'intrusion.

Le deuxième chapitre est axé à la présentation des techniques de la détection d'intrusion. Nous avons présenté : Machine Learning, Datamining et les techniques statiques.

Le troisième chapitre consiste à présenter trois méthodes de la technique machine Learning et établir une comparassions entre ces méthodes.

Chapitre1 : System de détection d'intrusion

1.1 Introduction

La sécurité des systèmes informatiques constitue un enjeu crucial car elle nous permet le bon fonctionnement des systèmes ainsi que la protection de données.

Ce chapitre est consacré pour la présentation des attaques informatiques, la politique de sécurité et les systèmes de détection d'intrusion.

1.2 Les attaques informatiques :

Une « attaque » est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables.

1.2.1 Les différentes catégories d'attaques :

Dans le domaine informatique il existe plusieurs types d'attaques qu'on peut répertorier dans quatre catégories [1].

- **L'attaque DOS (denial of service) :**

Le déni de service, connu sous le titre anglophone de « Denial Of Service » ou encore DOS, est une attaque réalisée dans le but de rendre indisponible durant une certaine période les services ou ressources d'une organisation. Généralement, ce type d'attaque a lieu contre des machines, serveurs et accès d'une entreprise afin qu'ils deviennent inaccessibles pour leurs clients. Le but d'une telle attaque n'est pas d'altérer ou de supprimer des données, ni même de voler quelconque information [2].

Il s'agit ici de nuire à la réputation de sociétés présentes sur Internet en empêchant le bon fonctionnement de leurs activités. La réalisation d'un DOS déni de service n'est pas très compliquée, mais pas moins efficace. Il est possible d'attaquer tout type d'équipements réseau tel que les serveurs, routeurs et Switch. Cela touche 99% de la planète car la plupart des dénis de service exploitent des failles liées au protocole TCP-IP. Nous pouvons diviser les impacts des attaques DOS en deux catégories :

- Les dénis de service DOS par saturation qui consistent à submerger une machine de requetés, afin qu'elle ne soit plus capable de répondre aux demandes réelles.

- Les dénis de service DOS par exploitation des vulnérabilités qui consistent à exploiter une faille du système cible afin de le rendre inutilisable.

Le principe de ces attaques DOS est d'envoyer des paquets ou des données de taille ou de constitution inhabituelle, afin de provoquer une saturation ou un état instable des équipements victimes et de les empêcher ainsi d'assurer les services réseau qu'elles sont censés offrir. Dans certains cas extrêmes, ce type d'attaque peut conduire au crash de l'équipement cible. Le déni de service DOS est donc un type d'attaque qui coute très cher puisqu'il interrompt le cours normal des transactions d'une organisation. Sachant qu'a l'heure actuelle, les sommes et les enjeux d'une entreprise sont généralement énormes, cela peut poser de graves problèmes si une telle situation se produit ne fut-ce que quelques heures. Imaginez les impacts：

- Financier d'un grand site de commerce en ligne dont sa plateforme d'hébergement serait indisponible lors des fêtes de Noël ?
- Sur l'image d'une banque qui ne pourrait plus recevoir ses mails ?
- Globaux pour vous, en tant qu'entreprise qui communique avec vos clients ?

Les contre-mesures sont compliquées à mettre en place et doivent être spécifiques à un type d'attaque DOS. Etant donné que les attaques par déni de service utilisent les services et protocoles normaux d'Internet, s'en protéger reviendrait à couper les voies de communications normales, sachant qu'il s'agit de la raison d'être principale des machines concernées (Site web, Messagerie, Extranet, …) [2].

Il faut donc essayer se protéger au mieux de certains comportements anormaux, ce qui implique notamment la vérification de l'intégrité des paquets, la surveillance du trafic, établissement de profils types et de seuils… On est donc loin de la protection absolue, mais il est tout de même possible de se protéger de façon intelligente et flexible.

Les principales attaques DOS

- **Attaque ARP Poisoning :**

Cette attaque se base sur l'envoi d'informations ARP falsifies. Ainsi, les différents équipements du LAN apprennent des mauvaises correspondances adresses IP avec MAC.

La conséquence est de rompre toutes communications entre deux équipements IP. Les cibles sont souvent les serveurs et les routeurs rendant indisponibles les services associent [3].

- **Attaque par fragmentation (fragment attack) :**

  Cette technique d'attaque est base sur la fragmentation IP. L'objectif est de planter la pile IP de la cible en modifiant les numéros de séquences. En effet, le protocole IP est prévu pour fragmenter les datagrammes de taille importante provenant de la couche 4 du modelé OSI. Le datagramme est alors fragmente en en plusieurs paquets IP possédant chacun un numéro de séquence et un numéro d'identification commun. A réception des fragments, la cible rassemble les paquets grâce aux valeurs de décalage (en anglais offset) qu'ils contiennent.

  L'astuce est de modifier les numéros de séquence afin de générer des blancs ou des recouvrements lors du réassemblage par la pile IP cible. Et certain équipement ne le supportait pas avec diffèrent conséquence tel que l'arrêt du service TCPIP.

  Cependant, revenons sur terre, aujourd'hui, comme pour le Ping de la mort, cette technique n'est plus viable du faite que les piles IP ont toutes évoluées. Le faite de détailler cette attaque permet de raconter l'histoire et se remémorer des souvenirs [3].

- **Attaque Ping de la mort (Ping of death) :**

  Cette technique d'attaque DOS est dépassée, mais elle a fait ses preuves à l'époque. Elle exploitait une faiblesse dans l'implémentation de la plus part des piles IP en envoyant un paquet ICMP d'une taille non conforme (supérieur à 64 octets). Ceci avait pour effet de planter directement la pile IP attaquée.

  Cependant, revenons sur terre, aujourd'hui, comme pour l'attaque par fragmentation, cette technique n'est plus viable du faite que les piles IP ont toutes évoluées. Nous pouvons donc discuter en datagramme ICMP de grande taille sans aucuns soucis (heureusement).

  Le faite de détailler cette attaque permet de raconter l'histoire et se remémorer des souvenirs [4].

  L'attaque DOS Ping de la mort est souvent confondue en pensant qu'elle est basée sur le faite de saturer une bande passante en ICMP. Ce n'est pas le cas, car ce principe est appliquée par l'attaque DOS Ping Flood et non pas par Ping de la mort.

- **Attaque Unreachable Host :**

  Cette attaque DOS envoie des messages ICMP de type « Host Unreachable » à une cible, provoquant la déconnexion des sessions et paralyse ainsi la victime. La simplicité

de cette attaque DOS est qu'elle demande qu'un faible débit du fait que les envois de datagramme ICMP peuvent être sur une faible cadence [3].

- **Attaque ICMP Redirect :**

  Cette attaque DOS envoi des messages ICMP de type « Redirect » a une cible pouvant être aussi bien un serveur comme un routeur. Le datagramme informera la victime qu'il faut passer par un autre chemin. Ainsi, cela provoquera une indisponibilité WAN.

- **Attaque Ping Flood (ICMP Flood) :**

  Beaucoup d'hôte Internet ou privée répondent aux paquets ICMP, il est donc facile de les inonder de l'objectif ce flux afin les rendre indisponibles. D'ailleurs, que les cibles répondent ou pas à l'ICMP, premier étant de saturer leurs bande passantes d'accès réseau, processeurs, mémoire….

  Ping Flood est la plus répandu des attaques par déni de service, car de nombreux particuliers et amateurs s'amusent simplement à pinger un host distant. Et bien sûr, ils se font plaisir en ajoutant les options permettant d'augmenter la cadence au maximum. Cependant, Microsoft a limité les options de son Ping obligeant ainsi à attendre une seconde entre chaque Ping. Ça permet d'éviter aux particuliers de s'amuser avec cette attaque, mais il faut être réaliste, Windows n'est pas le seul OS et Ping.exe n'est pas la seule application …

- **Attaque UDP Flood**

  Le concept de cette attaque DOS est identique au Ping Flood. C'est de saturer les ressources de la cible en terme de débit, processeur, mémoire à l'aide de datagramme UDP De la même manière, que la cible réponde ou pas au flux abondant émis, ne change pas le résultat.

- **Attaque * Flood**

  Le concept de cette attaque DOS, dont je viens d'inventer le nom, est de saturer une cible exactement comme le réalise Ping Flood et UDP Flood. Le concept de ces attaques se base toutes sur l'envoi massive de requête à destination de la cible. Ces requêtes ne sont pas obligatoirement basées sur ICMP ou UDP, mais elles se reposent sur TCP, IGMP, IP_raw,… D'où le nom de l'attaque * Flood indiquant que l'on peut saturer une cible avec n'importe quel flux IP.

- **Attaque Land (Land Attack)**

Cette attaque DOS consiste à démarrer une ouverture de session TCP via un SYN à destination d'un port ouvert de la machine cible. L'astuce de l'attaque est de préciser l'adresse IP source identiquement à l'IP destination ainsi que le port source identiquement au port destination. La victime recevant cette trame pense alors qu'il discuter avec lui-même ce qui généralement provoquait un crash. Cependant, revenons sur terre, aujourd'hui, comme pour le ping de la mort et, cette technique n'est plus viable du faite que les piles TCP-IP ont toutes évoluées. Le faite de détailler cette attaque DOS permet de raconter l'histoire et se remémorer des souvenirs. Du plus, les consoles IDS et les Firewall sont tous opérationnels pour bloquer de type de trame.

- **Probing**

L'attaquant de cette classe commence par un sondage de la future victime, ce que l'on appelle scan. Ce sondage va balayer chaque port IP afin de connaitre les services offerts par le système (OS, topologie du réseau, protections déployées,...). Une fois ce balayage achevé, la machine de l'intrus (celui qui réalise l'intrusion) tente alors d'identifier le système d'exploitation utilise par cette victime et d'exploiter les informations qu'elle a récoltées. Cette classe d'attaque est la plus étendue et elle requit une expertise technique minime. Les exemples de ce type d'attaque sont Ipsweep, Mscan, Nmap, Saint, Satan. [2]

Type d'attaque	Service	mécanisme	L'effet de l'attaque
Ipsweep	Icmp	Abus des propriétés	Identification des machines atives
Mscan	Plusieurs	Abus des propriétés	Recherche des vulnérabilités
Nmap	Plusieurs	Abus des propriétés	Identification des ports actifs sur une machine
Saint	Plusieurs	Abus des propriétés	Recherche des vulnérabilités
Stan	Plusieurs	Abus des propriétés	Recherche des vulnérabilités

Tableau 1.1 : *Attaques de Probe* [2]

- **U2R (User To Root)**

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

Type d'attaque	Service	Mécanisme	Effet de l'attaque
Eject	Session utilisateur	Débordement du buffer	Gagne le shell root
Ffbconfig	Session utilisateur	Débordement du buffer	Gagne le shell root
Fdformat	Session utilisateur	Débordement du buffer	Gagne le shell root
Loadmodule	Session utilisateur	Un mauvais système d'installation	Gagne le shell root
Perl	Session utilisateur	Un mauvais système d'installation	Gagne le shell root
Ps	Session utilisateur	Une mauvaise gestion des fichiers temporels	Gagne le shell root
Xterm	Session utilisateur	Débordement du buffer	Gagne le shell root

Tableau 1.2 : *Attaques de User to Root.* [2]

- **R2L (Remote To Local)**

Dans cette classe, l'attaquant (machine distante) envoie des paquets vers une machine du réseau cible, après il exploite les vulnérabilités de cette machine afin d'avoir un accès illégal comme un utilisateur. Il e a plusieurs types d'attaque de R2L, les plus connues utilisent ou exploitent les bugs ou les mauvaises configurations des applications ou des systèmes. Les exemples de cette classe d'attaque sont Dictionnaire, Ftp_write, Guest, Imap, Named, Phf, Sendmail, Xlock, Xsnoop.

Type d'attaque	Service	Mécanisme	Effet de l'attaque
Dictionary	telnet, rlogin, pop, ftp, imap	Abus des propriétés	Gagne un accèss utilisateur
ftp-write	ftp	Mauvaise configuration	Gagne un accèss utilisateur
Guest	telnet, rlogin	Mauvaise configuration	Gagne un accèss utilisateur
Imap	Imap	Bug	Gagne un accèss root
Named	dns	Bug	Gagne un accèss root
Phf	http	Bug	Exécute des commandes autant qu'utilisateur http
Sendmail	smtp	Bug	Exécute des commandes autant que root
Xlock	smtp	Mauvaise configuration	Mystifie un utilisateur pour obtenir le mot de passe
Xnsoop	smtp	Mauvaise configuration	Contrôle le stockage des clés à distance

Tableau 1.3 : *Attaques de Remote to User* [2].

1.3 Sécurité informatique

La sécurité informatique est l'utilisation de la technologie, des politiques et de l'éducation des personnes pour assurer la confidentialité, l'intégrité et l'accessibilité des données durant leur stockage, leur traitement et leur transmission. La protection des données doit dépendre du système à protéger. Ainsi, selon ce dernier, on insistera plus ou moins sur l'intégrité, la confidentialité ou la disponibilité.

Confidentialité : La confidentialité spécifie que seules les personnes autorisées à accéder à une certaine information ont la possibilité de l'atteindre. Pour cela, on utilise les contrôles d'accès et le chiffrement. La violation de la confidentialité peut se voir quand une information confidentielle est devenue publique, grâce aux logs du système ou aux changements de comportement d'une certaine personne envers l'organisation [13].

Intégrité : L'intégrité spécifie que seules les personnes autorisées peuvent modifier l'information dans le système. Sa protection est souvent la même que celle qui est garantie par la confidentialité. En effet, en contrôlant l'accès à une donnée, on assure aussi son intégrité. La détection d'une violation de cette dernière est, par exemple, la comparaison entre l'information et ses copies ou ses données de hashing. Une réponse à cela est la réparation de cette donnée [13].

Disponibilité : La disponibilité est le fait qu'une information est obtenue quand on en a besoin. Une de ces attaques connues est le déni de service. On peut le limiter en limitant les ressources consommables du système par une personne. Une réponse à cela est la réduction de la charge ou l'augmentation des capacités du système [13].

Assurer la confidentialité, l'intégrité et la disponibilité : Pour assurer tous ceux-ci, on utilise un ensemble de règles de sécurité : enlever les programmes non utilisés, utiliser des firewalls, utiliser des contrôles d'accès, configurer correctement les programmes, utiliser des anti-virus, utiliser des IDS,... Nous allons nous focaliser sur les IDS.
Les IDS surveillent le système, ce qui implique leur limitation pour rester dans la légalité du pays où le détecteur est implémenté.

1.4 Intrusions informatiques

Avec l'arrivée d'internet, de nouvelles marches ont vu le jour et de nouvelles perspectives sont apparues. La plupart des opportunités sont dans le domaine commercial, ou les entreprises peuvent exposer leurs produits et services au monde entier grâce à des sites web. Des transactions avec des sommes colossales sont effectuées chaque jour, ce qui expose les entreprises à différentes menaces.
Autrefois, on s'était beaucoup plus intéresse aux avantages de cette nouvelle technologie et rares étaient ceux qui pensaient ou mettaient quelques moyens et ressources pour assurer un minimum de sécurité [1].

Le plus grave est que plusieurs entreprises courent des risques sans le savoir, et que les administrateurs réseau ou les personnes chargées d'assurer la sécurité informatique ignorent de quoi ils devraient se protéger. On estime aujourd'hui a moins de 4 mn le temps moyen pour qu'un PC non protège connecte à Internet subisse une tentative d'intrusion ou soit contamine par un programme malicieux.
Différentes techniques ont été mises en place par des communautés des pirates informatiques. Chacune d'entre elles touche un certain aspect de l'outil informatique. Nous pouvons catégoriser les différentes techniques de piratage informatique en deux classes :

- **Attaques réseaux :** Cette classe d'attaques, regroupe l'ensemble des techniques mises au point, permettant d'exploiter les faiblesses des réseaux ou bien les attaques qui ciblent des composants réseau.

- **Attaques applicatives :** Cette deuxième catégorie contient les techniques basées sur les faiblesses et les bugs des applications, permettant ainsi d'exploiter ces dernières pour des fins malveillantes [1].

1.5 La détection d'intrusion

En sécurité informatique, la détection d'intrusion est l'acte de détecter les actions qui essaient de compromettre la confidentialité, l'intégrité ou la disponibilité d'une ressource. La détection d'intrusion peut être effectuée manuellement ou automatiquement. Dans le processus de détection d'intrusion manuelle, un analyste humain procède à l'examen de fichiers de logs a la recherche de tout signe suspect pouvant indiquer une intrusion. Un système qui effectue une détection d'intrusion automatisée est appelé système de détection d'intrusion (IDS). Lorsqu'une intrusion est découverte par un IDS, les actions topiques qu'il peut entreprendre sont par exemple d'enregistrer l'information pertinente dans un fichier ou une base de données, de générer une alerte par e-mail ou un message sur un pager ou un téléphone mobile. Déterminer quelle est réellement l'intrusion détectée et entreprendre certaines actions pour e mettre fin ou l'empêcher de se reproduire, ne font généralement pas partie du domaine de la détection d'intrusion. Cependant, quelques formes de réaction automatique peuvent être implémentées par l'interaction de l'IDS et de systèmes de contrôle d'accès tels que les pare-feu [2].

- **Que doit assurer la détection d'intrusion ?**

La détection d'intrusion permet aux organisations de protéger leurs systèmes contre les menaces qui ne cessent de croitre à cause de l'augmentation de la connectivité du réseau public (Internet), et la confiance accordée aux systèmes informatiques qui comportent des bugs. La question pour les professionnels de sécurité ne devraient pas être s'il faut utiliser la détection d'intrusion, mais quels dispositifs utiliser et quelles sont leur capacité de détection d'intrusion. Les systèmes de détection d'intrusion ont gagné l'acceptation d'être un élément nécessaire dans l'infrastructure de la sécurité informatique de chaque organisation. En effet, il e a plusieurs raisons pour acquérir et utiliser les systèmes de détection d'intrusion [3].

• Pour détecter les attaques et autres violations de sécurité qui ne sont pas empêchées par d'autres outils de sécurité.

• Pour documenter les menaces existantes dans une organisation, c'est-à-dire découvrir les vulnérabilités avant qu'elles ne soient exploitées par un attaquant.

• Pour agir en tant que contrôle de qualité pour la conception de sécurité, particulièrement dans les grandes et complexes entreprises.

• Pour fournir des informations utiles au sujet des intrusions qui ont eu lieu, et faire des diagnostics, recouvrement, et corrections des facteurs causatifs.

• Pour arrêter les intrusions afin de limiter les dégâts. Malheureusement cela n'est pas toujours possible à cause de la complexité et la diversité des intrusions, et la naissance de nouveaux tepees d'intrusions liées au développement des nouvelles technologies d'information. Les contre-mesures actives sont souvent optionnelles dans la quasi-totalité des systèmes de détection d'intrusion.

1.6 Les systèmes de détection d'intrusion (IDS)

Un système de détection d'intrusion est un mécanisme destine à repérer des activités anormales ou suspectes sur la cible analysée (réseau, hôte). Debar simplifie le système de détection d'intrusion dans un détecteur qui analyse les informations en provenance du système surveille. [3]

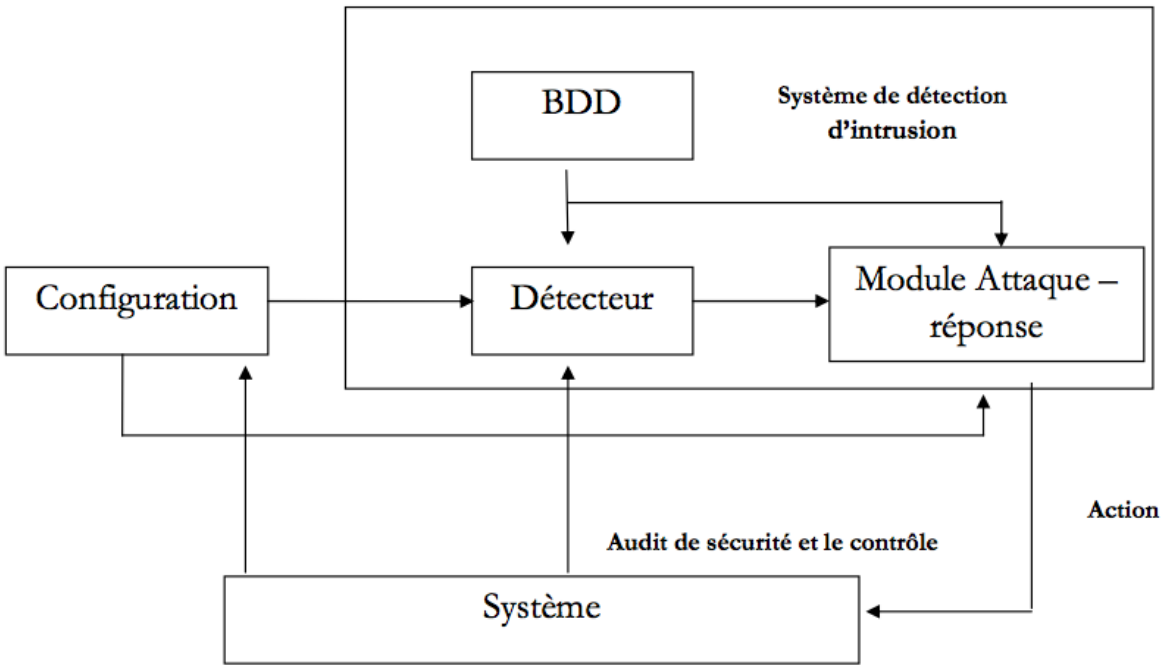

Figure 1.1 : *Modelé simplifie d'un système de détection d'intrusions* [3].

1.6.1 Evaluation des IDS

Philip et définit trois critères pour évaluer l'efficacité des systèmes de détection d'intrusion [16] :

- **L'exactitude (accuracy)** on parle de l'exactitude quand le système de détection d'intrusion déclare comme malicieux une activité légale. Ce critère correspond au faux positif.
- **La performance (performance)** la performance de système de détection d'intrusion est le taux de traitement des évènements. Si ce taux est faible, la détection en temps réel est donc impossible.
- **La complétude (completeness)** on parle de la complétude quand le système de détection d'intrusion rate la détection d'une attaque. Ce critère est le plus difficile, parce qu'il est impossible d'avoir une connaissance globale sur les attaques. Ce critère correspond au vrai négatif.

Debar et al dans **[**17] a rajouté également les deux critères suivants :

- **La tolérance aux fautes (Fault tolerance)** le système de détection d'intrusion doit lui-même résisté aux attaques, particulièrement au déni de service. Ceci est important, parce que plusieurs systèmes de détection d'intrusion s'exécutent sur des matériels ou logiciels connus comme vulnérables aux attaques.
- **La réaction à temps (Timeliness)** le système de détection d'intrusion doit s'exécuter et propager les résultats de l'analyse le plus tôt possible, pour permettre à l'officier de sécurité de réagir avant que des graves dommages n'aient lieu. Ceci implique plus qu'un calcul de performance, parce qu'il ne s'agit pas seulement de temps de traitement des évènements, mais aussi le temps nécessaire pour la propagation et la réaction à cet évènement.

1.6.2 Les différents types d'IDS :

- **IDS basé sur le hôte**

Un IDS basé sur l'hôte analyse plusieurs domaines pour déterminer le mauvais usage (activité malveillante ou abusive à l'intérieur du réseau) ou des intrusions (brèches de l'extérieur). Les IDS basés sur l'hôte consultent plusieurs types de fichiers journaux (noyau, système, serveur, réseau, pare-feu et autres) et comparent les journaux à une base de données interne de signatures courantes d'attaques connues. Les IDS UNIX et Linux basés sur l'hôte utilisent énormément la commande syslog et sa capacité à séparer les évènements enregistrés selon leur sévérité (par exemple, des messages d'imprimante mineurs contre des avertissements du noyau majeurs). La commande syslog est disponible lors de l'installation du paquetage sysklogd, inclus avec Red Hat Enterprise Linux. Ce paquetage offre une journalisation du système et la capture de messages du noyau. Les IDS basés sur l'hôte filtrent les journaux (qui,

dans le cas de journaux d'évènements de noyau ou de réseau, peuvent être très commentés), les analysent, marquent à nouveau les messages avec leur propre système d'évaluation de sévérité et les rassemblent dans leur propre journal spécialisé pour être analysé par les administrateurs.

Les IDS basés sur l'hôte peuvent également vérifier l'intégrité de données de fichiers et d'exécutables importants. Ils vérifient une base de données de fichiers confidentiels (et tout fichier ajouté par l'administrateur) et créent une *somme de contrôle* de chaque fichier avec un utilitaire d'analyse de fichiers messages comme la commande md5sum (algorithme 128-bit) ou la commande sha1sum (algorithme 160-bit). Les IDS basés sur l'hôte sauvegardent alors les sommes dans un fichier en texte clair et, de temps en temps, comparent les sommes de contrôle de fichiers avec les valeurs dans le fichier texte. Si l'une des sommes ne correspond pas, alors les IDS avertissent l'administrateur par courrier électronique ou pager.

- **Les avantages d'un IDS base hôte**

- La capacité de contrôler les activités locales des utilisateurs avec précision. Capable de déterminer si une tentative d'attaque est couronnée de succès.
- La capacité de fonctionnement dans des environnements crêpes.
- L'IDS base ĥte fonctionne sur les traces d'audit des systèmes d'exploitation ce qui lui permet de détecter certains types d'attaques (ex : Cheval de Troie).

- **Les inconvénients d'un IDS base hôte**

- La vulnérabilité aux attaques du type déni de service puisque l'IDS peut résider dans l'hôte cible par les attaques.
- La difficulté de déploiement et de gestion, surtout lorsque le nombre d'hôtes qui ont besoin de protection est large.
- Ces systèmes sont incapables de détecter des attaques contre de multiples cibles dans le réseau.

- **IDS basé sur le réseau**

Les systèmes de détection d'intrusions basés sur le réseau fonctionnent différemment des IDS basés sur l'hôte. La philosophie de conception d'un IDS basé sur le réseau est de scanner les paquets réseau au niveau de l'hôte ou du routeur, analysant les informations de paquets et

enregistrant tous les paquets suspects dans un fichier journal spécial avec des informations détaillées. Selon ces paquets suspects, un IDS basé sur le réseau peut scanner sa propre base de données de signatures d'attaques réseau connues et assigner un niveau de sévérité pour chaque paquet. Si les niveaux de sévérité sont assez élevés, un message électronique d'avertissement ou un appel de pager est envoyé aux membres de l'équipe de sécurité afin qu'ils puissent étudier plus en profondeur la nature de l'anomalie.

Les IDS basés sur le réseau sont devenus populaires avec l'internet grandissant en taille et trafic. Les IDS qui peuvent scanner les quantités volumineuses d'activités réseau et marquer avec succès les transmissions suspectes, sont accueillis dans le domaine de la sécurité. Les protocoles TCP/IP n'étant peu sûrs de nature, il est devenu impératif de développer des scanneurs, des renifleurs et d'autres outils d'analyse de réseau et de détection pour éviter les brèches de sécurité provenant d'activités réseau malveillantes comme :

- L'usurpation d'identité
- Les attaques par déni de service
- La corruption de cache arp
- La corruption de noms DNS
- Les attaques man-in-the-middle

- **Les avantages d'un IDS basé réseau**

- L'IDS base réseau est capable de contrôler un grand nombre d'hôte avec un petit cout de déploiement.
- Il n'influence pas sur les performances des entités surveillées.
- L'IDS base réseau est capable d'identifier les attaques de /a multiples hôtes.
- L'IDS base réseau assure une grande sécurité contre les attaques parce qu'il est invisible aux attaquants.

- **Les inconvénients d'un IDS basé réseau**

- L'IDS base réseau ne peut pas fonctionner dans des environnements crêpes.
- Ce type d'IDS ne permet pas d'assurer si une tentative d'attaque est couronnée de succès.
- L'évaluation et la comparaison des systèmes de détection d'intrusions est un problème en soi de par la diversité des sources de données possibles et la représentativité des

données utilisées lors des tests notamment. Une. Les systèmes de détection d'intrusions sont évalues traditionnellement suivant deux critères :

– **La fiabilité** de l'IDS : toute intrusion doit effectivement donner lieu à une alerte. Une intrusion non signalée constitue une défaillance de l'IDS, appelée faux négatif. La fiabilité d'un système de détection d'intrusions est liée à son taux de faux négatifs (c'est a-dire le pourcentage d'intrusions non-détectées), qui doit être le plus bas possible.

– **La pertinence** des alertes : toute alerte doit correspondre à une intrusion effective. Toute « fausse alerte » (appelée également faux positif) diminue la pertinence de l'IDS. Un bon IDS doit présenter un nombre de faux positifs aussi bas que possible.

Il ne suffit pas de détecter correctement les intrusions ; il faut surtout éviter de lever trop de fausses alertes. Si la pertinence des alertes est trop faible, par exemple si plus de .1% des alertes sont des faux positifs (.1% étant un chiffre prudent), l'administrateur de sécurité risque de considérer toutes les alertes comme des faux positifs , de ne pas les analyser et de ne pas prendre des mesures de protection au cas où l'alerte ne serait pas un faux positif. Dans ce cas, le système de détection d'intrusions devient inutile.

1.6.3 Efficacité des systèmes de détection d'intrusions

Philip dans définit trois critères pour évaluer l'efficacité des systèmes de détection d'intrusion :

• **L'exactitude (accu race)** on parle de l'exactitude quand le système de détection d'intrusion déclare comme malicieux une activité légale. Ce critère correspond au faux positif.

• **La performance (performance)** la performance de système de détection d'intrusion est le taux de traitement des évènements. Si ce taux est faible, la détection en temps réel est donc impossible.

La complétude (complétées) : on parle de la complétude quand le système de détection d'intrusion rate la détection d'une attaque. Ce critère est le plus difficile, parce qu'il est impossible d'avoir une connaissance globale sur les attaques. Ce critère correspond au vrai négatif [4].

Debar dans a rajoute également les deux critères suivants :

• La tolérance aux fautes (Fault tolérance) le système de détection d'intrusion doit lui-même résiste aux attaques, particulièrement au déni de service. Ceci est important, parce que plusieurs systèmes de détection d'intrusion s'exécutent sur des matériels ou logiciels connus comme vulnérables aux attaques.

• La réaction à temps (Timeliness) le système de détection d'intrusion doit s'exécuter et propager les résultats de l'analyse le plus tôt possible, pour permettre à l'officier de sécurité de réagir avant que des graves dommages n'aient lieu. Ceci implique plus qu'un calcul de performance, parce qu'il ne s'agit pas seulement de temps de traitement des évènements, mais aussi le temps nécessaire pour la propagation et la réaction à cet évènement [2].

1.7 Détection d'intrusions par signature et comportementale

Les systèmes de détection d'intrusions peuvent être classés suivant leur approche d'analyse des données. Deux grandes approches ont été proposées dans la littérature : la détection d'intrusions par signature et la détection d'intrusions comportementale. Ces deux approches s'opposent dans leur principe de détection : l'approche par signature se fonde sur la recherche de traces d'attaques ou d'intrusions alors que l'approche comportementale recherche les déviations du comportement de l'entité observée par rapport à un modèle du comportement normal de cette entité. Bien que la première approche proposée par Anderson en 1980[14], soit de type comportemental, nous allons d'abord présenter les approches par signature qui ont les faveurs des industriels de la sécurité, mais présentent des inconvénients, inhérents à l'approche, difficilement contournables.

1.7.1 Détection d'intrusions par signature

Principe

Les systèmes de détection d'intrusions par signature fondent leur détection sur la reconnaissance, dans le flux d'événements générés par une ou plusieurs sondes, de signatures d'attaques qui sont contenues dans une base de signatures. Une signature est un motif, dans le flux d'événements, de scénarios d'attaques définis au préalable. Un IDS par signature se compose :

- D'une ou plusieurs sondes, générant un flux d'événements, qui peuvent être de type réseau ou hôte.
- D'une base de signatures.

- D’un système de reconnaissance de motifs dans le flux d’événements.

La base de signatures : le taux de couverture de l’IDS dépend essentiellement de la qualité de la base de données puisque seules les attaques dont la signature est présente dans la base sont susceptibles d’être détectées. Les signatures sont décrites à l’aide de langages de description d’attaques [18]. Elles sont la plupart du temps définies par un opérateur bien que des travaux récents permettent la génération automatique des signatures [19]. La base de signatures doit également être maintenue :

- Les nouvelles attaques détectées par la communauté doivent être intégrées à la base.
- Suivant les choix de l’administrateur de sécurité, les signatures qui ne correspondent plus à une possible intrusion (parce qu’un logiciel ou un système d’exploitation a été mis à jour, remplace ou supprime par exemple.) peuvent être enlevées de la base. La maintenance de la base de signatures est une tâche importante. Sans maintenance, l’IDS ne peut détecter les nouvelles attaques. La rapidité de la propagation de certains vers comme Spammer et al dans [20] montre également une limite de cette approche, car le temps de mise à jour de la base de signatures par l’administrateur est supérieur au temps de propagation du ver.

Le système de reconnaissance de motifs est chargé d’identifier les motifs présents dans la base de signature, dans le flux d’événements. Différents systèmes de reconnaissance de motifs ont été définis dans la littérature. Cela va de systèmes simples à base de règles de Verne paxon dans [21] ou de correspondances de chaînes de caractères de Mukherjee et al dans [22] (string matching) a des systèmes bien plus complexes à base de systèmes experts de Phillip et al dans [23] ou de modélisation d’états de Steven et al [24] qui peuvent apporter suffisamment d’abstraction pour détecter des attaques inconnues mais qui font partie d’une même classe d’attaques. On pourra consulter la classification d’axels son [25] pour plus de détails sur ces systèmes.

Avantage de la détection par signature

- la prise en compte des comportements exacts des attaquants potentiels est possible.

Inconvénients de la détection par signature

- La base de règles doit être bien construite, ce qui est parfois délicat.
- Les performances du système expert sont limitées par celles de l’expert humain qui a fourni les règles. Or les connaissances des officiers de sécurité en matière de détection d’intrusion sont relativement faibles car, la plupart du temps, le volume énorme des fichiers d’audit les a découragées de toute analyse.

1.7.2 Détection d'intrusions comportementale

Principe

Une approche, proposée par J.P. ANDERSON [14] puis reprise et étendue par D.E. DENNING [26], consiste à utiliser des méthodes basées sur l'hypothèse selon laquelle l'exploitation d'une vulnérabilité du système implique un usage anormal de celui-ci. Une intrusion est donc identifiable en tant que déviation par rapport au comportement habituel d'un utilisateur. Voici quelques exemples étayant cette hypothèse :

- Un essai d'intrusion par un utilisateur non connu du système donnera lieu à un taux anormal de mots de passe erronés,
- L'attaquant par déguisement se connecte à une heure inhabituelle, il utilise abondamment de commandes lui permettant de changer de répertoire, peut-être n'utilise-t-il jamais l'utilitaire favori de l'utilisateur habituel,
- Un utilisateur connecté légitimement au système et qui essaye de contourner la politique de sécurité se connectera la nuit, exécutera des programmes qu'il n'a pas l'habitude d'utiliser, donnera lieu à un plus grand volume d'enregistrements d'audit, utilisera une imprimante sur laquelle il ne sort généralement pas de document, etc.
- Un cheval de Troie différera du programme légal dont il a pris la place en termes d'utilisation des ressources d'entrée/sortie,
- Une attaque par déni de service donnera lieu à un taux d'utilisation anormalement élève de certaines ressources du système.

Bien sûr, les phénomènes décrits dans ces exemples peuvent avoir une autre cause qu'une attaque du système, par exemple un changement de fonction de l'utilisateur au sein de l'entreprise. On s'attachera donc à trouver des méthodes possédant le plus fort taux de discrimination possible (c'est-à-dire ayant le plus fort taux de détection d'intrusion et le plus faible taux de fausses alarmes). De plus, on se référera à un seuil au-delà duquel on considérera que le comportement est intrusif.

Cette approche, dont la question de base est « le comportement actuel de l'utilisateur est-il cohérent avec son comportement passé ? », est appelée approche comportementale. Pour caractériser le comportement normal d'un utilisateur (on parle de modèle de comportement.), l'approche la plus immédiate consiste à utiliser des méthodes statistiques. Il est également possible d'envisager l'utilisation de systèmes experts ou de réseaux de neurones. Que nous allons vous présenter.

Le modèle de comportement normal est dit complet s'il modélise entièrement le comportement légitime, du point de vue de la politique de sécurité, de l'entité surveillée. Dans ce cas, toutes les alertes correspondent à des intrusions : il n'y a pas de faux positifs.

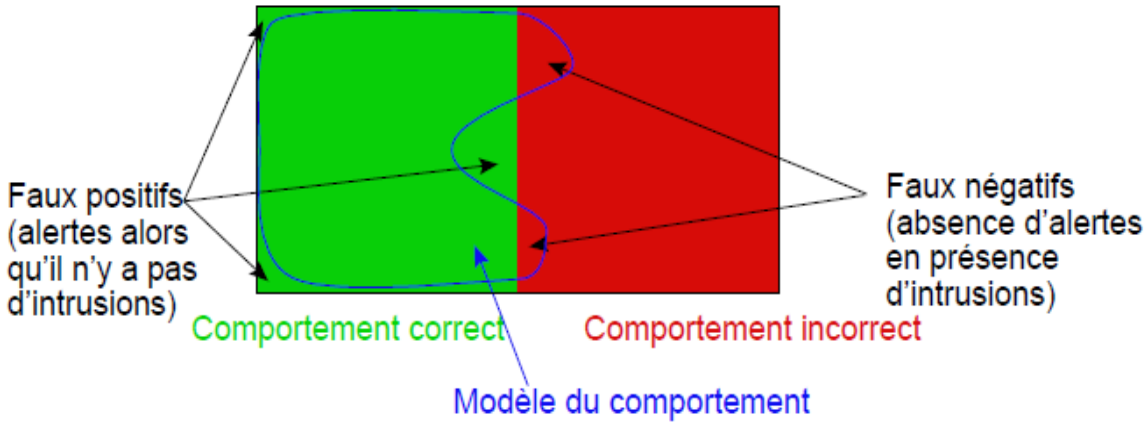

Figure 1.2 : *Caractères complet et correct du modèle de comportement normal* [26].

Les éléments du modèle de Denning

Le modèle de Denning est un modèle de comportement qui se compose de six éléments :

- **Les sujets** ce sont les initiateurs de toute activité observée sur le système, c'est-à-dire les utilisateurs ou les processus agissant pour leurs comptes.
- **Les objets** ce sont toutes les ressources du système (fichiers de données, fichiers de programme, messages, périphériques, ...). La granularité des objets sera plus ou moins fine selon le degré de sécurité recherché : dans certains cas, chaque fichier donnera lieu à la définition d'un objet, dans un autre, il faudra descendre au niveau des enregistrements, dans un troisième cas le niveau répertoire sera suffisant.
- **les enregistrements d'audit** ils sont générés par le système lors de toute action entreprise par un sujet sur un objet. Ils se composent des informations suivantes :

➢ Le sujet ayant entrepris l'action.
➢ Le type de l'action (par exemple login, lecture, écriture, ...).
➢ Les objets affectés (une action peut impliquer plusieurs objets.).
➢ Échec ou succès de la commande.

- Des éléments quantitatifs sur l'action (par exemple nombre d'octets ou d'enregistrements transférés dans une copie de fichier).
- Un horodatage de l'action.

- **Les profils** ce sont des structures qui caractérisent le comportement des sujets vis-à-vis des objets, par l'intermédiaire de valeurs statistiques résultant de l'observation des sujets. Un profil caractérise le comportement d'un utilisateur ou d'un groupe d'utilisateurs envers un objet ou un ensemble d'objets. Il donne une vue synthétique des actions des utilisateurs sur les objets.
- **Les enregistrements d'anomalie** ils sont générés quand une activité anormale est détectée.
- **Les règles d'activité** elles définissent les actions à entreprendre lorsque certaines conditions sont remplies sur les enregistrements d'audit ou les enregistrements d'anomalie. Elles ont la forme classique conditions-actions. Les actions peuvent par exemple consister à alerter l'officier de sécurité.

Un profil est constitué par un ensemble de variables représentant une quantité accumulée pendant une certaine période de temps (Minute, heure, journée, semaine, ..., ou intervalle entre deux événements audités particuliers, par exemple entre connexion et déconnexion). Voici quelques exemples de variables possibles :

- Nombre de mauvais mots de passe saisis en une minute
- Nombre, en millièmes de seconde, de quantum de temps du processeur occupés par un programme entre son lancement et sa terminaison.
- Nombre de fois qu'une commande système particulière est exécutée par un utilisateur donné, pendant le temps où il est connecté.

Les déférents modèles statistiques proposés par Denning

Le modèle statistique permet de déterminer, au vu de n observations x_1, …, x_n faites sur une variable x , si la valeur x_{n+1}de la s observation est normale ou non. DENNING [11] propose plusieurs modèles :

- **Comparaison de la nouvelle valeur de x avec une limite fixe**

Dans ce cas les n observations précédentes ne sont utiles que pour se donner une idée de la limite en question.

Exemple : on considère qu'il y a essai d'intrusion si on mesure dix mots de passe erronés en une minute.

- **Utilisation de la moyenne et de l'écart type des n observations précédentes**

L'observation $n+1$ est considérée comme anormale si x_{n+1}sort de l'intervalle de confiance défini par un écart de $(\pm D\sigma)$autour de la moyenne.

Ce modèle présente l'avantage d'être capable d'apprendre ce qui est (normal) à partir des observations passées. Pour cela, on met à jour la moyenne et l'écart type à chaque nouvelle observation en pondérant les observations de manière à ce que les plus récentes aient le plus fort poids.

- **Utilisation des covariances** ce modèle est similaire au précédent mais il permet de combiner plusieurs variables afin d'en tirer une synthèse. Il permet d'exploiter le fait que deux mesures (ou plus) représentent des manières différentes de caractériser le même aspect du comportement d'un utilisateur.
- **Utilisation des processus de Markov** ce modèle permet de définir la probabilité du passage d'un état, défini par le contenu d'un enregistrement d'audit, à un autre état, également défini par un enregistrement d'audit. Est considéré comme anormal un enregistrement qui apparaît et dont la probabilité, au regard des états précédents, est trop faible. Cette approche est intéressante pour les attaques dans lesquelles est requis un enchaînement de commandes dans un certain ordre.
- **Utilisation des séries temporelles** une nouvelle observation est anormale si sa probabilité d'apparition, au moment où elle apparait, est trop faible.

DENNING envisage la possibilité d'utiliser d'autres modèles, utilisant plus que les deux premiers moments mais moins que l'ensemble complet des enregistrements d'audit.

Avantage de la détection comportementale

- la détection d'intrusion inconnue est possible.

Inconvénients de la détection comportementale

- le choix des différents paramètres du modèle statistique est assez délicat et soumis à l'expérience de l'officier de sécurité.
- l'hypothèse d'une distribution normale des différentes mesures n'est pas prouvée.
- le choix des mesures à retenir pour un système cible donne´ est délicat,

- en cas de profonde modification de l'environnement du système cible, le modèle statistique déclenche un flot ininterrompu d'alarmes, du moins pendant une période transitoire.
- un utilisateur peut changer lentement de comportement dans le but d'habituer le système à un comportement intrusif.
- il est difficile de dire si les observations faites pour un utilisateur particulier correspondent à des activités que l'on voudrait prohiber .
- pour un utilisateur au comportement erratique, toute activité est (normale). Une attaque par déguisement sur son compte ne pourra pas être détectée.
- il n'y a pas de prise en compte des tentatives de collusion entre utilisateurs, alors même que cet aspect est très important, notamment dans le cas des réseaux.

1.7.3 Une approche hybride

Une approche hybride a été proposée par Tombini et al [27]. Cette approche consiste en la sérialisation d'un IDS comportemental suivi d'un IDS par signature. L'IDS comportemental permet de filtrer les requêtes normales et ainsi seules les requêtes anormales sont passées à l'IDS par signature. Bien que l'IDS comportemental utilisé soit simple, ceci permet de réduire le nombre de faux positifs génères globalement. La source d'entrées est le fichier d'audit du serveur web. Cet IDS est donc soumis aux mêmes problèmes que les autres utilisant cette source de données.

Il semble donc indispensable d'utiliser simultanément une approche comportementale et une approche par signature de manière à profiter des avantages de l'une et de l'autre.

1.7.4 Systèmes experts

Pour représenter l'usage « normal » qu'un utilisateur fait du système, il est possible d'utiliser un ensemble de règles au lieu d'un modèle statistique. Cela permet d'utiliser un système-expert comme outil de détection d'intrusion. Les règles d'un tel système expert peuvent être, soit entrées manuellement, soit générées automatiquement à partir des enregistrements d'audit. L'entrée manuelle sera par exemple utilisée pour exprimer une politique de sécurité. Les règles générées décrivent quant à elle des comportements. Les systèmes-experts présentent un inconvénient souvent cité : la base de règles est ni simple à créer, ni simple à maintenir.

1.8 Conclusion

Ce chapitre a été consacré pour la présentation des attaques informatique, et la politique de la sécurité informatique. En suite une définition d'une intrusion, après les systèmes de détection d'intrusion et l'utilisation des IDS. Enfin nous avons cité les approches de la détection d'intrusion. Nous allons

Chapitre 2 : Les techniques de détection d’intrusion

2.1 Introduction

Un système de détection d’intrusions basé sur la détection d’anomalies contrôle les activités du système afin de les classer comme normales ou anomalies. Il procède à construire des profils d'un comportement normal pour les activités des utilisateurs et à observer les déviations significatives de l'activité de l’utile- sauteur courante par rapport à la forme normale établie.
Pour pouvoir formaliser le comportement normal d’un système, plusieurs méthodes ont été utilisées. Ce chapitre sera consacré à une présentation générale de ces différentes approches.

2.2 Machine learning

Le domaine du machine Learning inclut la construction d'un modèle à partir de données grâce à l'utilisation d'un algorithme. Ce modèle va au mieux généraliser, en représentant ou en approximant les données. Il permet, selon les données qu'on lui donne en input, de prédire celles inconnues ainsi que de mieux comprendre celles existantes. Le domaine d'application du machine Learning est très varié : la prédiction de valeurs financières, la détection d'intrusion dans le domaine de la sécurité informatique, le moteur de recherche influençable par le pro de l'utilisateur, la détection de vols de machine, l'implémentation d'un anti-virus et la cryptanalyse. Le cycle de vie d'une implémentation du machine Learning est la suivante :

1. Réalisation du modèle
2. Obtention et nettoyage des données
3. Phase d'apprentissage
4. Phase de validation
5. Phase d'exécution

2.2.1 Obtention des données et pré-processing

La première étape à réaliser est donc l'obtention de données en suffisance, représentatives du problème à résoudre. Ceci n'est pas toujours aisé. Certaines informations sont plus coûteuses à

obtenir que d'autres. Par exemple, un header d'un paquet réseau est plus simple à obtenir qu'une information dans la partie data quand celle-ci est chiffrée. La deuxième étape est le nettoyage, appelé aussi pré-processing de la donnée récoltée, c'est-à-dire une réduction de ce qui est strictement intéressant, ainsi que leur traduction. Le but de cette étape est une meilleure précision du modèle, une optimisation de son temps d'exécution et de son apprentissage ainsi que de sa taille. Voici quelques exemples de nettoyage :

- Transposer un ensemble de nombres vers un range.
- Transposer un ensemble de réels vers un ensemble de naturels
- Ajouter des valeurs qui ont été calculées à partir des données récoltées
- Sélectionner un résumé des informations
- Enlever les informations inutiles représentant du bruit

Ce nettoyage est souvent très compliqué à mettre en œuvre et demande une bonne connaissance des données à traiter. C'est pourquoi des techniques automatiques ont fait l'objet de recherches : ce sont les techniques de feature sélection. [6]

Feature sélection :

Il existe essentiellement 3 types de méthodes de feature sélection. On retrouve d'abord le filtre méthodes, qui ne se basent que sur l'utilité d'une variable sans tenir compte de son impact dans le modèle. Ensuite on a les Wrapper methods qui tiennent compte de l'algorithme d'apprentissage pour déduire l'apport des variables. Enfin, on distingue aussi les Embedded méthodes qui sont spécifiques à un modèle et sont exécutées lors de la procédure d'apprentissage. Ces méthodes peuvent être combinées pour obtenir de meilleurs résultats. Il est conseillé de ne pas utiliser les mêmes données pour les phases de feature sélection et d'évaluation, pour éviter un biais au niveau des performances estimées du système.

Après avoir obtenu les données nettes du problème, la prochaine étape est la réalisation du modèle. [6]

2.2.2 Réalisation du modèle

Elle consiste en une recherche de la meilleure structure ainsi que l'ensemble des paramètres à initialiser dedans. La complexité, et plus précisément la qualité, du modèle aura une influence directe sur la précision de la généralisation des données. Plusieurs types de modèles vont être présentés dans la suite. Après avoir construit le modèle, il est nécessaire de le configurer selon le problème à traiter grâce à la phase d'apprentissage.

2.2.3 Phase d'apprentissage

Un sous-ensemble de l'ensemble des informations nettoyées forme les données d'entrainement, permettant d'exécuter la phase d'apprentissage du modèle. Ceci permet d'ajuster les paramètres du modèle. Il existe plusieurs sortes d'algorithmes d'apprentissage. Certains de ces algorithmes sont supervisés et d'autres non supervisés. Un algorithme supervisé est un algorithme à qui on présente l'entrée et la sortie (ou la cible) désirée en supposant qu'il y a une relation inconnue mais réelle entre les deux. Il devra minimiser l'erreur entre la sortie désirée et celle qu'il produit. Ils sont souvent utilisés pour des problèmes de reconnaissance. Un algorithme non supervisé est un algorithme à qui on présente l'entrée mais dont la sortie est inconnue. Ce type d'algorithmes est souvent utilisé pour des problèmes de partitionnement où le nombre et la nature des partitions ne sont pas connus a priori. Néanmoins, ce dernier ne donne aucun résultat si les données ne contiennent pas de partitions. Après avoir entraîné le modèle, il est important de le valider pour éviter le sur-apprentissage. [7]

2.2.4 Phase de validation

Durant cette phase, on va tester et valider le modèle et ses paramètres selon des critères se basant sur ses résultats. Il permet d'obtenir le meilleur modèle généralisant les données obtenues lors de la phase d'apprentissage. Pour cela, on a un ensemble d'exemples pour l'apprentissage et un autre pour les tests. Voici quelques méthodes pour les tests :

- **Hold-out** : On coupe aléatoirement l'ensemble des informations en deux groupes : groupe d'apprentissage et groupe de tests.
- **Leave-one-out** : Cette méthode sort de l'ensemble des informations une donnée en particulier e la laisse de côté, puis construit le modèle avec celles restantes et enfin évalue la structure avec l'exemple laissé de côté. On répète le processus pour chacune des données de l'ensemble de données. Ainsi, on peut avoir une moyenne globale de la précision du modèle.

Cross-validation : Cette méthode réalise un partitionnement des données de manière aléatoire en groupes. On utilise une partition comme un ensemble de test et le reste pour former celui d'entraînement. Comme précédemment, on applique un algorithme à l'ensemble d'entraînement et on évalue le modèle résultant sur celui de tests. On répète ce processus pour chaque partition et on regarde l'erreur moyenne.

D'autres méthodes existent telles que les boots rap. Enfin, après avoir validé le modèle, il reste à quantifier ses performances en pratique. [7]

2.2.5 Performance du modèle

Apres avoir choisi et évalué notre modèle, il est intéressant de pouvoir quantifier ses performances. Pour cela, on compare plusieurs modèles sur un même jeu de données. En effet, certains sont plus adaptés pour certains problèmes. Lors de cette phase, il est primordial de ne pas enlever de données 6 pour ne pas biaiser l'évaluation. Pour cette phase, on s'intéresse au pourcentage de vrais positifs, de vrais négatifs, de faux négatifs et enfin de faux positifs. Néanmoins, ça n'inclut pas la taille du modèle ni son temps d'apprentissage ou son temps d'exécution. La technique receveur operating characteristic (ROC) est largement utilisé et permet de tester une structure. [6]
Concrètement, un ROC est une courbe d'un modèle représentant ses vrais positifs par rapport à ses faux positif. L'aire sous cette courbe représente la performance du modèle.

2.2.6 Types de modèle

Il existe plusieurs types de modèles. Outre le fait qu'ils sont différenciables par leur côté supervisé ou non supervisé, ils le sont aussi par leur côté classification ou régression. Le premier classifie les données et le deuxième prédit des valeurs pour chaque donnée. Puisque le problème de l'IDS semble être un problème de classification, ce qui suit est une liste non exhaustive de modèles de classification qui pourraient être envisagées pour la mise en place d'un IDS.

2.3 Systèmes de détection d'intrusion basés sur la machine learning

2.3.1 Obtention des données

La première étape est l'obtention des données. Lors de la phase d'apprentissage, ces informations permettent de connaître les habitudes des utilisateurs ou les différents types d'attaques de manière générale. Lors de la phase d'exécution, ils permettent de détecter une attaque. Néanmoins, la question est de savoir s'il existe une mesure de la quantité de données nécessaire pour avoir un modèle correct, c'est-à-dire qui respecte un certain taux d'erreur accepté. Cette question est encore ouverte à ce jour. Toutefois, le choix des informations à prendre en compte dépendra du type d'apprentissage.

2.3.2 Supervisé ou non supervisé

Les apprentissages supervisés utilisent généralement des data sets représentant des attaques connues. Les apprentissages non supervisés se basent uniquement sur les comportements des utilisateurs pour détecter une variation par rapport aux comportements normaux qui représentent

une attaque. Selon l'article les algorithmes supervisés réalisent d'excellents résultats pour des intrusions connues, ils sont meilleurs que les algorithmes non supervisés. Inversement, pour des agressions inconnues, les algorithmes supervisés voient une réduction drastique de leur efficacité contrairement aux algorithmes non supervisés. Ceci peut s'expliquer par le fait que, puisque les algorithmes non supervisés ne font que partitionner des données, les attaques leur sont toujours inconnues. En effet, étant donné que ces derniers ne font que regrouper des données proches, ils ne savent pas quand elles représentent une attaque. Bien que les algorithmes supervisés et non supervisés obtiennent des résultats semblables pour des attaques inconnues, les modèles non supervisés sont préférés pour leur robustesse. En effet, ceux-ci ne changent pas drastiquement leur taux de réussite selon que l'intrusion est connue ou non. De plus, ils n'ont pas d'oracle leur disant à quelle classe appartient telle donnée. Ils réalisent solitairement les classes. Ils sont donc plus indépendants que les techniques supervisées qui ont besoin d'un oracle. Leur point fort est la classification d'attaques appartenant à un label inconnu. En effet, on ne doit pas leur présenter l'ensemble des labels. De plus, il est parfois très dur de classifier univoquement une information dans une classe. C'est pourquoi cette méthode est parfois préférée pour les IDS [8].
La thèse parle d'une implémentation du modèle SOM sur les IDS. Les résultats de cette thèse sont néanmoins peu encourageants pour ce type de structure. Par ailleurs, les algorithmes supervisés et non supervisés peuvent être combinés pour obtenir un apprentissage semi-supervisé où on utilise les avantages des deux techniques précédentes pour réaliser un IDS. Cependant, une hypothèse sur les IDS supervisés reste [8].

2.3.3 Hypothèse

Dans le cas des algorithmes supervisés, une hypothèse difficile à respecter lors de l'apprentissage est qu'il n'y a pas d'attaque contenue dans les informations modélisant le comportement normal des programmes qui peuvent pourtant avoir beaucoup d'irrégularités. En effet, dans le cas contraire, on insérerait des propriétés d'une attaque comme comportement normal. Par conséquent, cette dernière ne sera pas détectée. Inversement, un algorithme non supervisé peut lever cette hypothèse en nettoyant les données pour enlever les informations d'attaque. Pour cela, on recherche un ensemble d'enchaînement, appelé motif, de système call fortement présent dans le système et on les range suivant leur dangerosité d'attaquer le système [7].
Par après, on regarde l'ensemble des séquences des systèmes calls présentes pour les comparer aux motifs découverts précédemment, réalisant ainsi une sorte d'empreinte des différentes séquences de

système calls par rapport aux motifs qu'ils contiennent. Par après, on regarde la distance de toutes les empreintes par rapport à une autre afin d'avoir un graphique unique pour l'ensemble.
Ceci nous permet de mieux visualiser celles de même nature (qui sont donc très proches les unes des autres) par rapport à celles très différentes. Ainsi, un espacement très grand entre deux groupes d'empreintes où l'un en contient beaucoup et l'autre très peu permet de supposer que l'un en est une attaque alors que l'autre ne l'est pas. De plus, comme on suppose qu'il y a moins d'empreintes d'attaques que de non-attaques, on retrouve facilement l'ensemble de celles d'attaques. C'est pourquoi on utilise un algorithme non supervisé offline pour détecter les anomalies dans les données. Par après, un algorithme supervisé peut utiliser ces données nettoyées pour concevoir et implémenter le modèle.

2.3.4 Implémentation

Une méthode généralement utilisée, qui est basée sur l'approche comportementale, est la prise d'empreintes des utilisateurs, c'est-à-dire de leur comportement, et de regarder quand elle ne lui correspond pas sur le système. Ainsi, on peut détecter un comportement anormal et donc une attaque éventuelle. Inversement, on peut prendre l'empreinte de certains pirates connus pour les détecter. Cette dernière peut être apprise par le machine learning. Toutefois, une autre méthode, basée sur l'approche par scénario, est l'utilisation de données représentant des attaques. Pour que cela soit réaliste, il faut que l'IDS soit suffisamment rapide, efficace et flexible aux petits changements normaux des utilisateurs, sans toutefois permettre de dévier vers une situation d'attaque. Dans le cas contraire, soit elle ne sera pas détectée soit le nombre de faux positifs pourrait exponentiellement augmenter. De manière globale, on réalise un tel IDS en trois étapes :
on modélise tous les comportements normaux de chaque utilisateur ou les signatures des attaques, on le donne au machine learning pour qu'il l'apprenne et ensuite on regarde si le comportement dévie de l'habituel ou s'approche d'une situation offensive. Dans certains cas, il est intéressant de savoir le type de l'attaque et non seulement si elle a eu lieu [12].
Une dernière implémentation est la gestion d'un grand nombre d'alertes venant des IDS par du machine learning. Ainsi, cette méthode est une sorte de filtre de ces dernières permettant de se focaliser sur les alarmes les plus importantes. En effet, un IDS peut générer un nombre volumineux de fausses alertes, ce qui rend la tâche des administrateurs système impossible. Ceci est donc un complément aux IDS et non un remplacement de ceux-ci. comme vu précédemment, il existe des NIDS et des HIDS.

HIDS : Pour modéliser le comportement d'un utilisateur, on peut regarder l'ensemble des commandes qu'il a utilisées durant une période, ce qui donne un HIDS offline. Cette méthode est

justifiable puisque la plupart des personnes n'utilisent pas le système dans le même but ni de la même manière. Pour un HIDS semblable mais online, la machine apprend à reconnaître les commandes futures selon les k dernières utilisées. Néanmoins, leur ordre n'est pas révélateur pour savoir si une attaque a lieu ou pas. Il semble plus significatif de regarder l'ensemble des commandes utilisées durant une période. L'inconvénient majeur est la non-prise en compte des arguments des system calls. Ainsi, il est intéressant de prendre en compte les valeurs de retour, les statuts d'erreur et d'autres arguments pour détecter des attaques. En effet, prenons l'exemple des system calls suivants exécutés par un simple utilisateur : open, read, write. Ces trois systèmes calls peuvent sembler inoffensifs puisque c'est une simple ouverture, lecture et écriture dans un _chier. Néanmoins, la situation change si on regarde l'argument de ces system calls et que celui-ci est le fichier passwd. Une autre méthode analysée est l'apprentissage du profil des programmes et non des utilisateurs. Ainsi, la machine apprend le fonctionnement normal des logiciels sur une machine [7].

NIDS : Pour ce qui est du NIDS, il est nécessaire de bien comprendre l'ensemble des variables d'un paquet, ainsi que du protocole, pour comprendre le fonctionnement normal du système. Essayons de partitionner les paquets TCP. Pour cela, on peut d'abord voir les attributs qui ne changeront sans doute jamais entre ceux-ci : Version du protocole + les flags réservés. D'autres attributs permettent de les partitionner : adresse source/destination + protocole utilisé. Ces deux propriétés sont généralement utilisées par les,firewalls pour filtrer les paquets. Enfin, certains attributs pourraient être différents dans une même partition : taille du header, identificateur, TTL. Ce sont ceux qui sont généralement utilisés pour détecter une anomalie [8], regarder les valeurs de ces attributs pour déterminer une anomalie n'est pas une bonne manière de faire, ce qui pourrait expliquer les faibles résultats [12], détaillés précédemment. Néanmoins, regarder les changements de ces attributs au cours du temps semble être une meilleure manière de faire. Ainsi, on regarde la moyenne de certaines valeurs, le pourcentage d'événements selon la valeur d'un attribut, le pourcentage de paquets ayant telle valeur, Il faut donc s'assurer d'avoir suffisamment de paquets lors de la phase d'apprentissage pour garantir qu'aucun comportement normal non présent dans ces paquets ne soit oublié. Néanmoins, beaucoup d'attaques connues surpassent cette méthode :

- Utilisation d'un proxy qui va mapper les ports non légitimes vers des ports légitimes pour que les IDS le considèrent comme un paquet légitime.
- Modification d'un logiciel qui fonctionnera d'une telle manière pour l'attaquant et d'une autre manière pour le reste des personnes du système. Après avoir obtenu toutes les informations, il faut classifier les paquets selon le serveur à qui ils appartiennent. Pour cela, il existe deux manières de faire. La première méthode est d'utiliser un apprentissage supervisé pour

pouvoir classifier les différents services proposés grâce à des paquets connus par le système, et par la suite définir à quel service appartient tel nouveau flux de paquets.

Ceci peut être fait avec un arbre de décision. La deuxième méthode est d'utiliser la détection d'anomalies où on définit le comportement normal de chaque service. Ainsi, avec un nouveau flux de paquets, on détermine si celui-ci est conforme au comportement normal du service. Ceci est très semblable à la méthode qui regarde le comportement normal d'un utilisateur. Pour connaître le service utilisé à tel instant, on peut construire un arbre de décision qui apprendra les attributs des paquets, par exemple : le nombre de paquets avec le flag FIN selon tel service, et qui classifiera les nouveaux paquets du réseau. Pour détecter une attaque, on regarde vers où l'arbre de décision nous amène. Si par exemple celui-ci nous dit que c'est un service ftp et qu'on voit que le port utilisé est 80, alors on peut se demander si un nouveau service non autorisé est présent dans le système.

Toutefois, toutes ces études ont des problématiques en commun.

2.3.5 Problématiques des modèles

- **Non périodicité**

La non-périodicité de certains phénomènes peut avoir des répercussions problématiques. Prenons l'exemple de la rédaction d'un mémoire. Il y a une période où l'étudiant réalise beaucoup de recherches et une autre période où l'étudiant écrit son mémoire. Durant ces deux phases, c'est toujours le même étudiant qui travaille sur la même machine. Néanmoins, les demandes de ressources ne sont pas égales, ce qui pourrait faire croire au machine learning qu'il y a quelque chose d'inhabituel. Semblablement, une augmentation de fréquentation sur un site web ne signifie pas pour autant une attaque contre ce site web. Réaliser un détecteur d'intrusion qui ralentirait cette augmentation aurait une conséquence négative.

Pour contrer ce problème, il est conseillé de mélanger un grand nombre de paramètres pour réaliser le profil de l'utilisateur et de donner la possibilité à l'administrateur de considérer si c'est un cas non voulu.

- **Espacement des attaques**

Une attaque difficilement visible est une attaque réalisée en un temps très espacé. Pour les contrer.

2.3.6 Optimisation

Il y a une forte volonté de réduire le temps d'apprentissage pour pouvoir mettre en œuvre une solution commerciale. propose une manière de réduire le temps d'apprentissage ainsi que la taille de

la structure. Pour être encore plus performant, on peut définir la structure de l'ensemble des réseaux neuronaux utilisés dans l'IDS. En voici quelques-unes :

- Une première boîte contenant un filtre d'information suivie par une autre contenant le réseau neuronal.
- Une première boîte contenant un filtre d'information suivie par n boites de réseaux neuronaux suivies par une dernière qui va jouer le rôle d'arbitre en déterminant ce qui se passe sur le réseau selon les informations reçues par les n boîtes de réseaux neuronaux. Chacune d'entre elles reconnaît un type d'attaque (DOS, U2R, R2L, scan,. . .). Ceci est aussi appelé la méthode Boo Sting. [6]
- Trois premières boîtes suivies d'une dernière. L'entraînement se fait ainsi :
    - On entraîne la première boîte avec un certain nombre d'informations.
    - On prend au hasard des nouvelles informations et on entraîne une deuxième.
    - On prend des nouvelles informations et on regarde la réaction des deux premières. Si ces deux boîtes ne convergent pas vers la même idée, on prend cette information et on la met comme entraînement pour la troisième.

La dernière boîte va jouer le rôle le d'arbitre. Ceci est aussi appelé la méthode Bagging.

2.4. Datamining

2.4.1 Définition :

Le Data Mining traduit généralement en Fouille de données est un ensemble de techniques d'exploration de données permettant d'extraire d'une base de données des connaissances sous la forme de modèles de description afin de décrire le comportement actuel et/ou de prédire le comportement futur du système représenté par ces données.

Le Data Mining se situe à la croisée des bases de données, l'intelligence Artificielle, statistique et l'analyse de données :

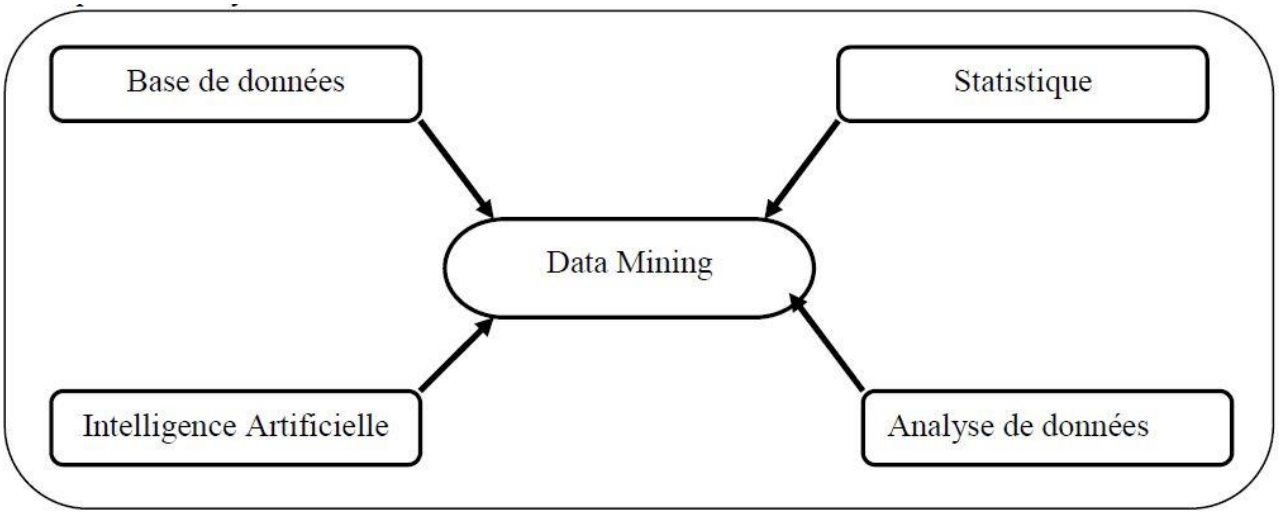

Figure 2.1 : *Techniques utilisées* [9].

- **Classification :** la classification associe chaque élément de donnée à une plusieurs catégories prédéfinies. Ces algorithmes de classification génèrent des classifieras, sous forme d'arbres de décision ou de règles. Une application dans la détection d'intrusion serait d'appliquer ces algorithmes à une quantité suffisante de données d'audit normales ou anormales pour générer un classifieur capable d'étiqueter comme appartenant à la catégorie normale ou anormale de nouvelles données d'audit.
- **Analyse de relations :** on établit des relations entre différents champs d'éléments d'audit, comme par exemple la corrélation entre la commande et l'argument dans l'historique des commandes de l'interpréteur de commandes, pour construire des profils d'usage normal. Un programmeur, par exemple pourrait avoir une forte relation entre *emacs* et des fichiers *C*. On définit ainsi des règles d'association.
- **Analyse de séquences :** Ces algorithmes tentent de découvrir quelles séquences temporelles d'événements se produisent souvent en même temps. On peut noter qu'utilise le Common Intrusion détection Framework (CIDF). Le CIDF est un effort pour développer des protocoles et des APIs pour permettre aux projets de recherche sur la détection d'intrusion de partager les informations et les ressources, et pour que les composants de détection d'intrusion puissent être réutilisés.

Un Internet Engineering Task Force (IETF) working group a été créé et nommé Intrusion Detection Working [9].

2.4.2 Stratégie du Data Mining

Les statisticiens utilisent en générale des données expérimentales pour étudier un phénomène, en fixant certaines hypothèses. Par contre, le *Data Mining* utilise les données pour trouver toutes les hypothèses valables.

Le Data Mining tente alors de réaliser un arbitrage entre ***validité*** *scientifique*, ***interopérabilité*** des *résultats* et ***facilité*** *d'utilisation*, dans un environnement professionnel où le temps d'étude joue un rôle majeur et où les analystes ne sont pas toujours des statisticiens [9].

2.4.3 Les domaines d'application Datamining

- Analyse de données et aide à la décision
- Analyse de marché
- Marketing ciblé, gestion des relations client, analyse des achats des clients, ventes croisées, segmentation du marché
- Analyse de risque
- Détection de fraudes
- Prévision des Marchés
- Autres Applications
- Text mining: news groups, emails, documents Web.
- Optimisation des requêtes, Machine Learning (Auto-Apprentissage)…etc.

2.4.4 Les étapes d'un processus de Data Mining

La création d'un modèle d'exploration de données fait partie d'un processus plus vaste qui va d'un ensemble de requêtes de consultation des données, la création d'un modèle afin d'y répondre et le déploiement du modèle obtenu dans un environnement de travail. Ce processus peut être décrit à l'aide des six étapes de base suivantes

- **Définition du problème** : cette étape consiste à définir clairement le problème et à envisager les moyens d'utilisation des données pour apporter une solution au problème.
- **Préparation des données :** l'opération consiste à consolider et à nettoyer les données identifiées à l'étape « Définition du problème ».
- **Exploration des données :** cela consiste à explorer les données préparées en se servant des algorithmes décrits.
- **Création des modèles :** on va générer le ou les modèles d'exploration de données. Vous allez utiliser les connaissances acquises à l'étape Exploration des données pour définir et créer les modèles.
- **Exploration et validation des modèles** : dans cette phase on va explorer les modèles d'exploration de données créés et à tester leur efficacité.

- **Déploiement et mise à jour des modèles** : consiste à déployer les modèles les plus efficaces dans un environnement de production [10].

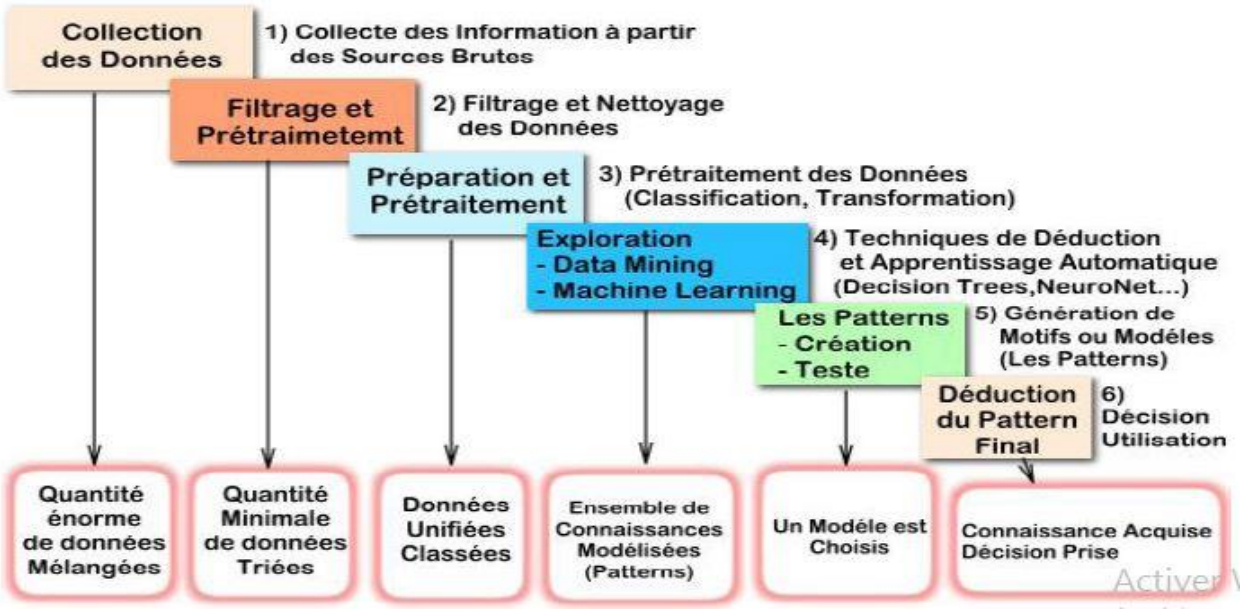

Figure 2.2 : *Les étapes d'un processus de fouille de données* [10]

2.5 Statique

2.5.1 L'Analyse en Composantes principales ACP

L'Analyse en Composantes principales "ACP" est une des méthodes descriptives multidimensionnelles appelées méthodes factorielles utilisées pour la détection de nouveauté .Cette technique statistique permet d'étudier simultanément les relations qui existent entre les variables, et de réduire la dimension d'un ensemble de données d'une taille importante afin d'analyser et traiter ces données. L'ACP peut servir à mieux connaitre les données sur lesquelles on travaille, à détecter éventuellement des valeurs suspectes, et si on parle de la classification des données, la réduction de l'espace de représentation permet de diminuer le temps de classification pour toute nouvelle

information et éventuellement de déterminer le nombre de groupes à construire, elle peut être aussi une intermédiaire de calcule en vu d'analyse ultérieurs. Cette section comprend deux parties, la première est consacrée à la modélisation par ACP linéaire et la deuxième consacrée à la détection et la localisation par le modèle obtenu. Avant de d'approfondir, il faut tout d'abord bien comprendre quelques notions de statistique liées à cette approche :

- **Notions de base**

Tableau de données :

L'ACP propose à partir d'un tableau (on dit aussi matrice) de données , une représentation géométrique permet de former ce que l'on appelle nuage de points, ou chaque point est positionne dans un repère en fonction de ses coordonnées .

$$X = \begin{array}{c} \begin{matrix} v_1 & v_2 & \cdots & v_m \end{matrix} \\ \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \end{array} \begin{matrix} \vartheta_1 \\ \vartheta_2 \\ \downarrow \\ \vartheta_n \end{matrix}$$

Figure 2.3 : *Structure d'un tableau de données.*

Modèle ACP :

L'objectif est de diminuer la dimension de l'espace de représentation des observations, une fois déterminer le nombre $\ell < m$ de composantes à retenir (la dimension réduite à choisir), la matrice X peut être approximée et pour cela, la 33 *matrice P est partitionnée sous la forme :*

$$P = \begin{pmatrix} \hat{P}_\ell & \tilde{P}_{m-\ell} \end{pmatrix}$$

- **Les Etapes pour la détermination ACP :**

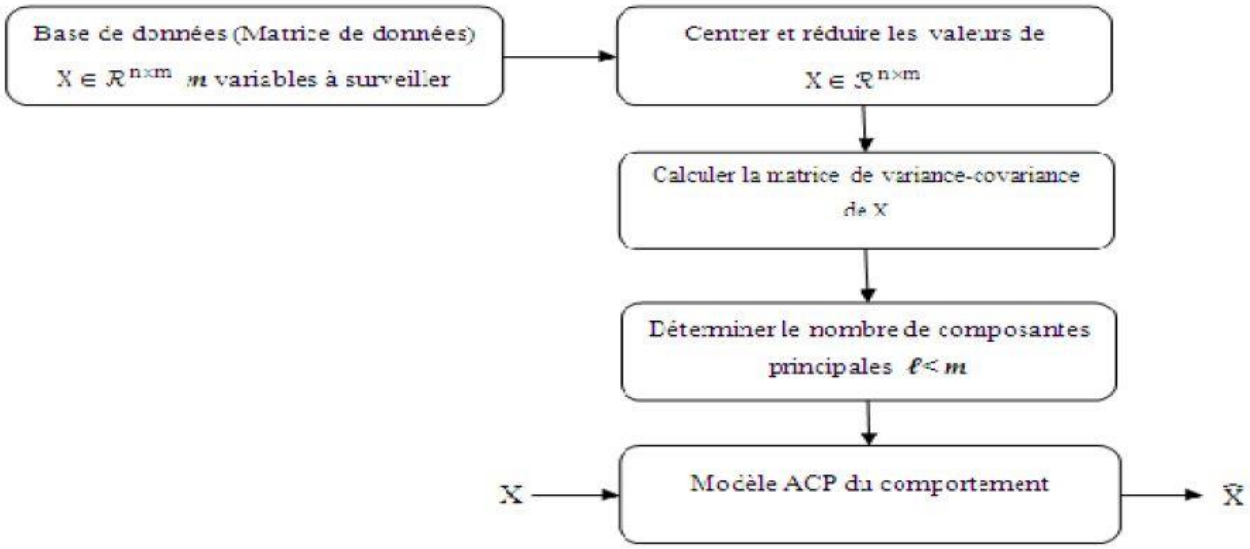

Figure 2.4 : *Etapes pour la détermination d'un modèle ACP.*

- **ACP à noyau :**

Bien que le ACP traite des données linéaires, ACP à noyau s'intéresse aux composants principaux, ou des fonctionnalités, qui sont non linéairement liées aux variables d'entrée. Pour ce faire, nous calculons les produits scalaires dans l'espace des entités à l'aide d'une fonction à noyau dans l'espace d'entrée donnée. Tout algorithme qui peut être exprimé par des produits scalaires, c'est à dire sans utilisation explicite des variables, la méthode à noyau nous permet de construire une version non linéaire de celle-ci. Le ACP tente de trouver un sous-espace linéaire à faible dimension auquel les données sont définies. cependant, parfois, les données sont associées à un sous- espace non linéaire à faible dimension où ACP à noyau aura lieu . les données sont principalement situées le long de (ou au moins proche) d'une courbe en 2-D mais le ACP ne peut pas réduire la dimensionnalité de deux à un parce que les points ne sont pas situés le long d'une ligne droite. L'ACP à noyau peut reconnaitre que ces données sont unidimensionnelles mais dans un espace dimensionnel supérieur, appelé espace caractéristique. Le ACP à noyau ne calcule pas explicitement l'espace dimensionnel supérieur, il projette plutôt les données dans cet espace caractéristique afin que nous puissions classer les données.

- **Détection par ACP à noyau**

dans cette partie nous nous intéressons uniquement à la détection de la nouveauté par la méthode à nyau de l'ACP mais avant d'appliquer ceci, nous devons formuler le ACP standard avec des produits scalaires que nous utiliserons alors lors de la dérivation du ACP _a noyau , et donc cette formulation consiste à calculer les principaux composants dans F

2.6 Conclusion

Nous avons présentés dans ce chapitre les différentes techniques utilisées pour la détecter d'intrusions, le prochain chapitre nous allons évoquer et comparer quelques méthodes utilisées pour la détection d'intrusion : SVM (Support Victor Machine), réseaux de neurones et K-means.

Chapitre 3 : La comparaison entre les méthodes de machin Learning

3.1 Introduction

Nous avons vu dans le chapitre précédant quelques généralités sur les techniques de la détection d'intrusion.

A présent dans ce chapitre nous allons évoquer et comparer quelques méthodes de machine Learning qui utilisées pour la détection d'intrusion : SVM (Support Victor Machine), réseaux de neurones et K-means.

3.2 Support Victor Machine (SVM)

3.2.1 Définitions

- **Support Victor machine (SVM)** : est une technique réalisée par Vapnik basée sur la méthode d'apprentissage automatique et sur la théorie de l'apprentissage statistique [31], et l'un des algorithmes supervisé pour résoudre des problèmes de discrimination et de régression. Le SVM est l'un des classifieur linéaires utilisés pour séparer les données par un hyperplan.
- **Hyperplans :** pour un espace vectoriel E, les hyperplans sont les analogues des plans pour l'espace usuel de dimension 3, voir exemple 3.1.
- **Victor de support :** Les points utilisés pour que la marge soit maximale.
- **Marge :** la distance entre les vecteurs de support.
- **Noyau :** Une fonction noyau est une fonction $k : x, x' \in x^2 \rightarrow$ R satisfaisant

$$k(x, x') = < \phi(x), \phi(x') >$$

Où ϕ est une fonction de x vers un espace de redescription F doté d'un produit scalaire :

$$\phi : x \rightarrow \phi(x) \in F$$

3.2.2 Principe de SVM

Le principe de base de SVM consiste en l'utilisation de la fonction dite *noyau* (kernel) qui permet une séparation optimale des données. Le SVM travaille pour trouver un hyperplan qui va séparer les données en deux classes et maximiser la distance entre elles.

- **Exemple**

Cet exemple est pris de la référence [31].

Dans le schéma on peut dessiner plusieurs hyperplans valides pour séparer le point mais la propriété remarquable des SVM est que cet hyperplan doit être optimal**.**

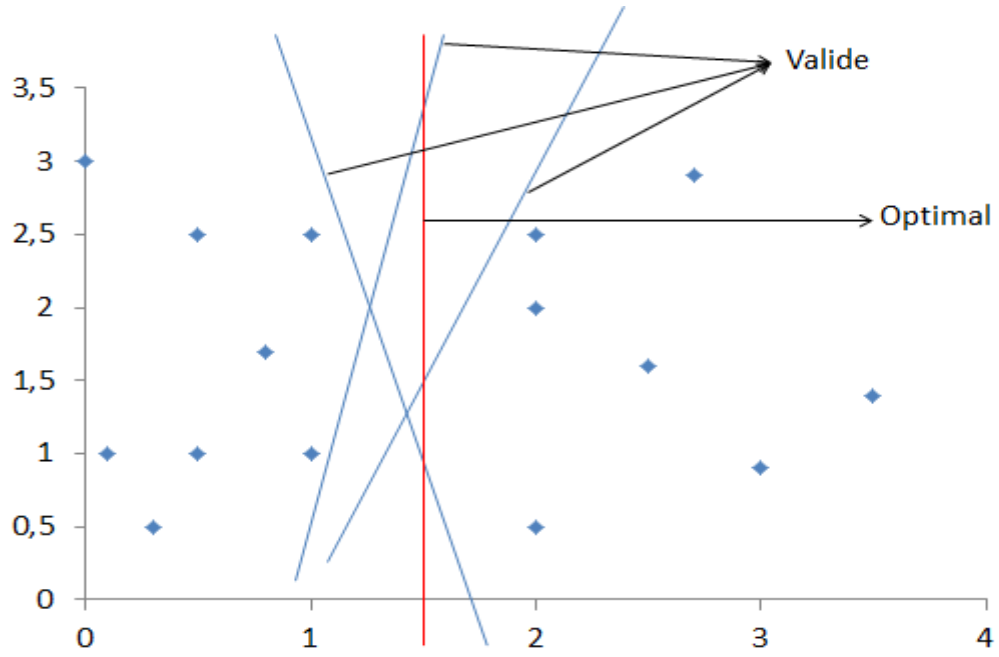

Figure 3.1: *représentation de plusieurs hyperplans sépare les données.*

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

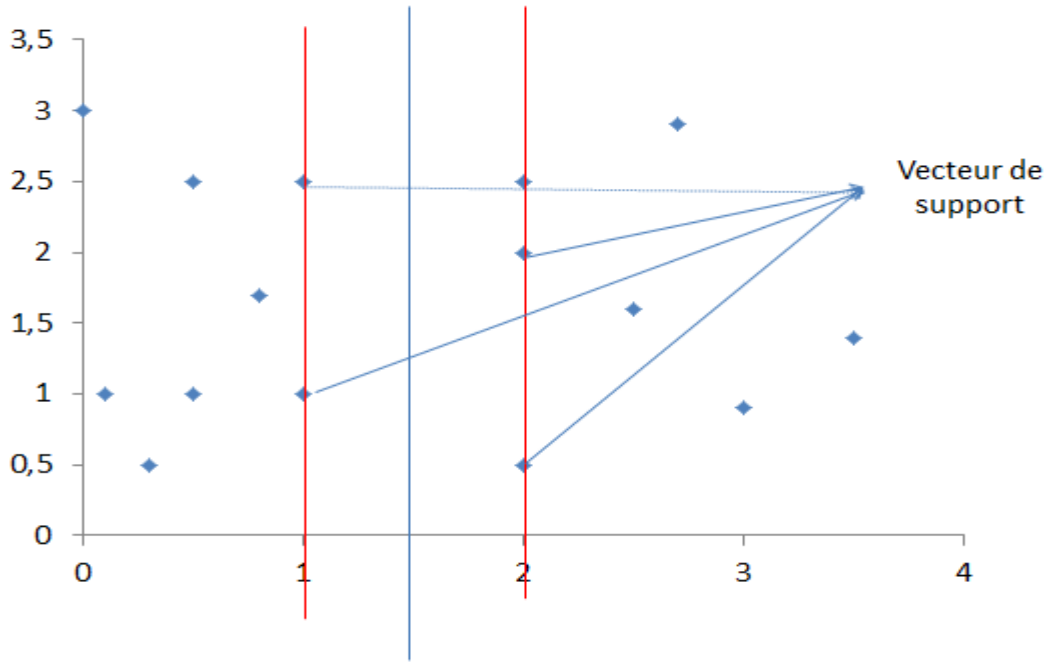

Figure 3.2: *représentation des vecteurs de support.*

Parmi les hyperplans valides, nous avons celui qui passe au milieu des points des deux classes, cela revient à chercher l'hyperplan le ≪ plus sûr ≫. En effet, supposons qu'un exemple n'est pas décrit parfaitement, une petite variation ne modifiera pas sa classification si sa distance à l'hyperplan est grande. Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance ≪ marge ≫ entre l'hyperplan et les points. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge.

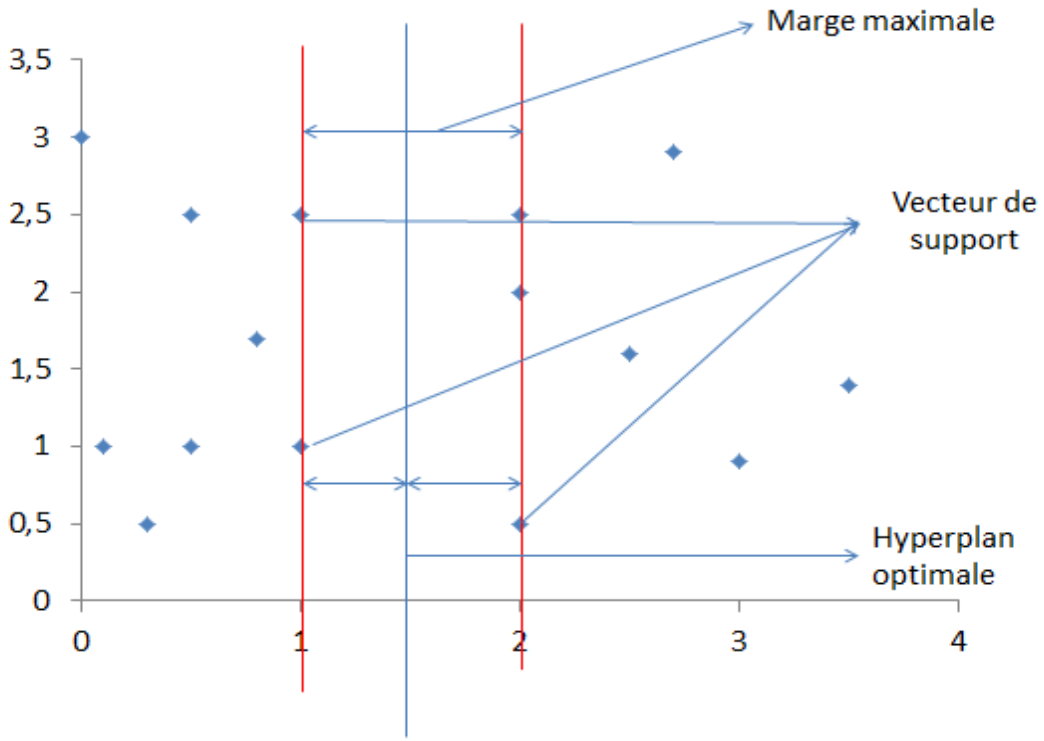

Figure 3.3: *représentation de l'hyperplan séparateur optimal.*

3.2.3 Méthode

Le but de SVM est de construire un hyperplan de séparation. Ces échantillons sont divisés en deux catégories, qui devraient satisfaire pour maximiser l'intervalle de classe et minimiser le taux d'erreur de classification.

Étant donné un échantillon défini comme [28] :

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}, x_1 \in R^n, y_i \in \{-1, +1\}, j = 1, \dots, l.$$

La fonction de décision de classification peut être obtenue en résolvant le problème d'optimisation quadratique suivant :

$$\begin{cases} \min(\frac{1}{2} < w, w > + \frac{1}{t}\sum_{i=1}^{l}(\varepsilon_i - vp \\ s.t\ \ y_i(< w, x_i > + b) \geq p - \varepsilon_i\ \varepsilon_i \geq 0, p \geq 0\ , i = 1, 2, \dots, l \end{cases} \quad (1)$$

Dans lequel, les conditions optimales de w et b est de maximiser la distance $\frac{2p}{\|w\|}$ de deux types d'échantillons à l'hyperplan, et $\varepsilon_i (i = 1, 2, \dots, l)$ est variable.

Avec l'utilisation de l'opérateur Lagrange, le problème (1) peut être converti en problème double suivant :

$$\begin{cases} \max W(a) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ s.t \sum_{i=1}^{l} a_i y_i = 0, \quad \frac{v}{l} \leq a_i \leq \frac{1}{2} \; i = 1, 2, \dots, l \end{cases} \tag{2}$$

Où $k(x_i, x_j)$ est une fonction du noyau, $k(x_i, x_j) =< \emptyset(x_i), \emptyset(x_j) >$.

Qui diffère les échantillons d'apprentissage de l'espace d'entrée vers un certain espace d'entités en appliquant la méthode de mappage non linéaire

$$\varphi: R^k \to F$$

Les échantillons dans l'espace caractéristique sont séparables ; v représente la limite supérieure de la proportion dans laquelle les échantillons ne sont pas classés correctement, ce qui signifie la limite inférieure des vecteurs de support dans l'ensemble des échantillons d'apprentissage. La fonction de prise de décision de SVM non linéaire est :

$$f(x) = sign(\sum_{x_i \in SVs} \alpha_i y_i \, K(x_i, x) + b \tag{3}$$

La fonction $sign(\,)$est une fonction mathématique qui extrait le signe d'un nombre réel, et appelée classifier.

On peut voir à partir de la fonction de décision que les vecteurs de soutien qui affectent la classification de SVM sont les échantillons qui rencontrent $a_i \neq 0$. Les vecteurs de pré-support dans la fonction de prise de décision n'ont aucun effet, donc ces vecteurs non supportés peuvent être jetés à l'avance. Et le nombre d'échantillons d'entraînement peut être réduit pour améliorer la vitesse d'entraînement.

- **Linéarité et non-linéarité**

  - **Classifier linéaire**

Un classifier est dit linéaire lorsqu'il est possible d'exprimer sa fonction de décision par une fonction linéaire en$x$. On peut exprimer une telle fonction par :

$$h(x) =< w , x > +b = \sum_{i=1}^{n} w_i x_i + b$$

Où w ($\in R^n$) est le vecteur de poids et b ($\in R^0$) le biais, alors que x est la variable du problème. x est l'espace d'entrée et qui correspond à R^n , où n est le nombre de composantes des vecteurs contenant les données. Notons que l'opérateur $< >$ désigne le produit scalaire usuel dans R^n. $w\ et\ b$ sont les paramètres à estimer de la fonction de décision $h(x)$.

Pour décider à quelle catégorie un exemple estimé x' appartient, il suffit de prendre le signe de la fonction de décision $y = sign(h(x'))$.

L'objectif de la discrimination linéaire est de trouver la bonne fonction de décision. La classe de tous les hyperplans qui en découle sera notée $h(x)$.

- **Non-linéaires**

Dans la plupart des problèmes réels, ce n'est pas toujours le cas où les données sont linéairement séparables et il est donc nécessaire de contourner ce problème (difficile de séparer n'importe quel jeu de données par un simple hyperplan). Si par exemple les données des deux classes se chevauchent sévèrement, aucun hyperplan séparateur ne sera satisfaisant.

Dans ce but, l'idée est de projeter les points d'apprentissage x_i dans un espace T de dimension q, plus élevée que n grâce à une fonction non-linéaire $\emptyset$ qu'on appelle fonction noyau, choisie a priori et d'appliquer la même méthode d'optimisation de la marge dans l'espace T . L'espace T ainsi obtenu est appelé espace des caractéristiques ou aussi espace transformé.

Tout ce qu'il nous reste à faire c'est de résoudre le problème :

$$Max_{\alpha}\, w(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j < x_i, x_j >$$

Dans la pratique on choisit un noyau qui satisfait les conditions de Mercer [32] afin de garantir la décomposition. Une famille de ces fonctions noyaux qui sont très appropriées aux besoins des SVM peut être définie, en voici les plus utilisées : [36]

- **Noyau polynomial d'ordre**

$$K(x_i, y_j) = (< x_i, x_j > +1)^d$$

La dimension de l'espace transformé induit par un noyau polynomial est de l'ordre $\frac{(p+d)!}{p!d!}$, où p est la dimension de l'espace de départ :

- **Noyau linéaire**

$$K(x_i, x_j) = x_i . x_j$$

- **Noyau gaussien de largeur de bande**

$$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma})$$

Le paramètre σ permet de régler la largeur de la gaussienne. En prenant un σ grand, la similarité d'un exemple par rapport à ceux qui l'entourent sera assez élevée, alors qu'en prenant un σ tendant vers 0, l'exemple ne sera similaire à aucun autre.

3.2.4 Avantages et inconvénients

- **Avantages**

➢ Les SVM possèdent des fondements mathématiques solides.

➢ Les exemples de test sont comparés juste avec les supports vecteurs et non pas avec tous les exemples d'apprentissage.

➢ Décision rapide, la classification d'un nouvel exemple consiste à voir le signe de la fonction de décision f(x).

- **Inconvénients**

➢ Classification binaire d'où la nécessité d'utiliser l'approche un-contre-un.

➢ Grande quantité d'exemples en entrées implique un calcul matriciel important.

➢ Temps de calcul élevé lors d'une régularisation des paramètres de la fonction noyau.

3.3 Les réseaux de neurones

3.3.1 Historique

La naissance des concepts des réseaux de neurones artificiels s'est faite en 1943 par j.McCulloch (neurophysiologiste) et W.Pitts (logicien), qui ont proposé les premières notions de neurone formel. En 1948 D.Hebb propose une règle d'apprentissage pour les réseaux de

neurones. Ce concept fut ensuite mis en réseau avec une couche d'entrée et une sortie par Rosenblatt en 1959 pour simuler le fonctionnement rétinien et tacher de reconnaître des formes [29].

3.3.2 Définition :

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau [34].

- **Propriétés des réseaux de neurones**

D'une manière générale un réseau de neurones possède les propriétés suivantes :

- **Parallélisme**

Cette notion se situe à la base de l'architecture des réseaux de neurones considérés comme ensembles d'entités élémentaires qui travaillent simultanément.

- **Capacité d'adaptation**

Celle-ci se manifeste tout d'abord dans les réseaux de neurones par la capacité d'apprentissage qui permet au réseau de tenir compte des nouvelles contraintes ou de nouvelles données du monde extérieur. De plus elle se caractérise dans certains réseaux par leur capacité d'auto-organisation qui assure leur stabilité en tant que systèmes dynamiques.

- **Mémoire distribuée**

Dans les réseaux de neurones, la mémoire d'un fait correspond à une carte d'activation des neurones. Cette carte est en quelque sortie un codage du fait mémorisé.

- **Résistance aux pannes**

A cause de l'abondance des entrées et la structure du réseau, les données bruitées ou les pannes locales dans certain nombre de ses éléments n'affectent pas ses fonctionnalités. Cette propriété résulte essentiellement du fonctionnement collectif et simultané des neurones qui le composent.

- **Généralisation**

La capacité de généralisation d'un réseau de neurones est son aptitude de donner une réponse satisfaisante à une entrée qui ne fait pas partie des exemples à partir desquels il a été adapté.

- **Structure de connexion**

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Elles sont très variées, le nombre de connexions étant énorme. Cette topologie fait apparaître une certaine régularité de l'arrangement des neurones.

3.3.3 Les neurones

❖ **Neurone biologique**

- **Structure**

On pense que le système nerveux compte plus de 1000 milliards de neurones interconnectés. Bien que les neurones ne soient pas tous identiques, leur forme et certaines caractéristiques permettent de les répartir en quelques grandes classes. En effet, il est aussi important de savoir, que les neurones n'ont pas tous un comportement similaire en fonction de leur position dans le cerveau. [37]

Les composantes d'un neurone biologique :

- Le corps cellulaire : il contient le noyau du neurone ainsi que la machine biochimique nécessaire à la synthèse des enzymes. Ce corps cellulaire contient aussi les autres molécules essentielles à la vie de la cellule. Sa taille est de quelques microns de diamètre.
- Les dendrites : ce sont de fines extensions tubulaires qui se ramifient autour du neurone et forment une sorte de vaste arborescence. Les signaux envoyés au neurone sont captés par les dendrites. Leur taille est de quelques dizaines de microns de longueur.
- L'axone : c'est à partir de l'axone que les signaux partent du neurone. Contrairement aux dendrites qui se ramifient autour du neurone, l'axone est plus long et se ramifie à son extrémité ou il se connecte aux dendrites des autres neurones. Sa taille peut varier entre quelques millimètres à plusieurs mètres.
- Synapse : une synapse est une jonction entre deux neurones, et généralement entre l'axone d'un neurone et une dendrite d'un autre neurone.

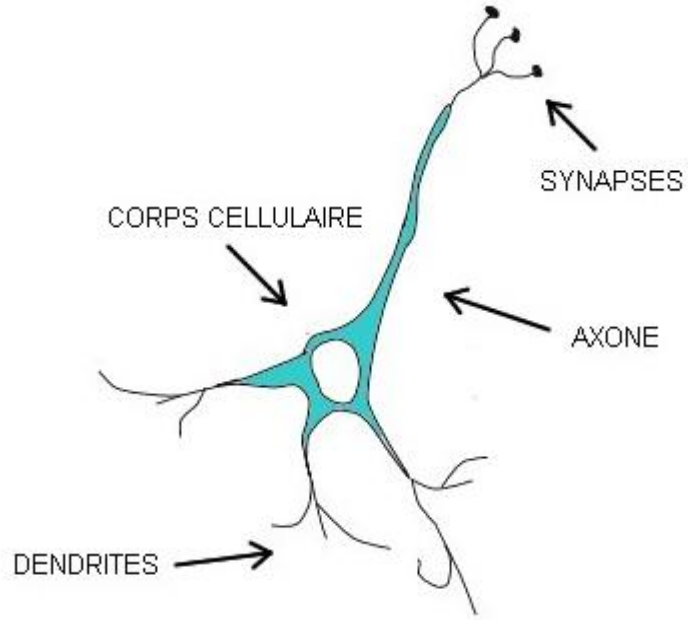

Figure 3.4 : *représentation d'un neurone biologique.*

- **Neurone formel**

  - **Définition**

Un neurone formel, ou neurone, est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées « entrées » du neurone, et la valeur de la fonction est appelée « sortie ». Un neurone est donc avant tout un opérateur mathématique [37].

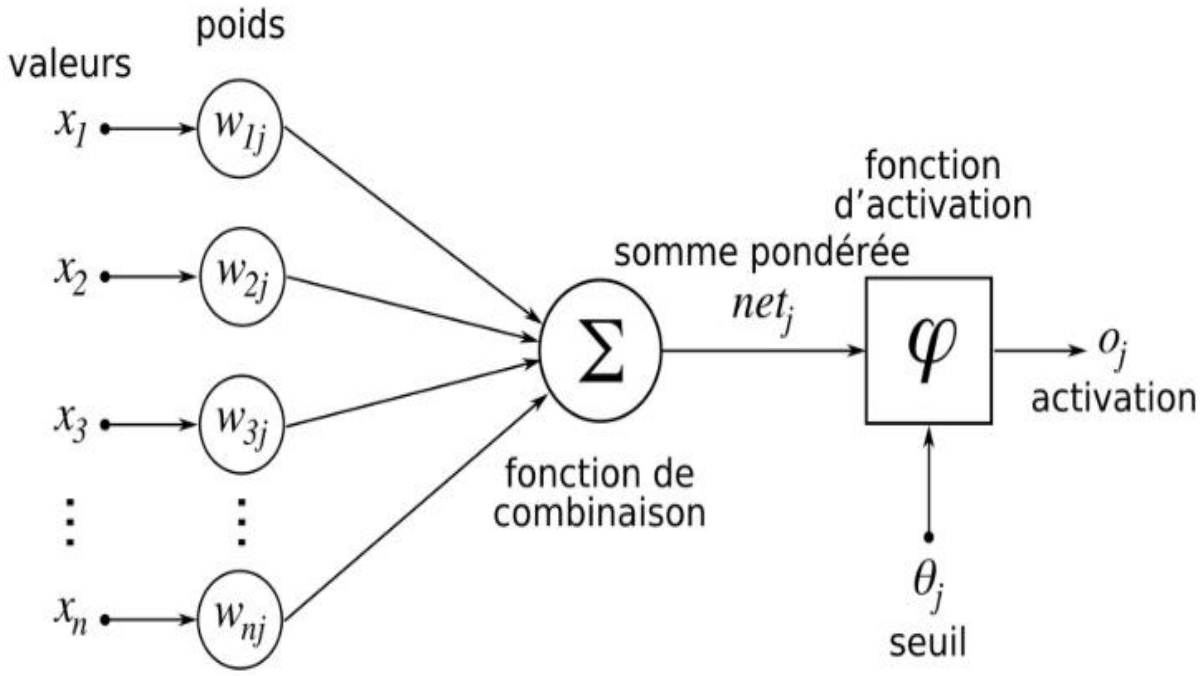

Figure 3.5 : *représentation d'un neurone formel.*

3.3.4 Les différents éléments d'un réseau de neurone

Nous allons d'abord détailler les différentes notions nécessaires à la compréhension des réseaux de neurones, pour plus de détails [33].

❖ **L'espace de représentation**

Les données d'entrée sont des vecteurs de R^d, notés comme d'habitude (en transposition) $x^T = (x_1, x_2, \dots, x_d)$.

❖ **Le neurone formel**

L'unité de traitement élémentaire dans un réseau de neurone est capable de faire seulement certaines opérations simples. Ces unités sont souvent appelées neurones formels pour leur similitude grossière avec les neurones du cerveau. Les modèles de réseaux de neurones qui nous intéressent particulièrement, les réseaux multicouches classent les unités comme des neurones d'entrée, cachés, ou de sortie.

- Les neurones d'entrée ou, simplement, une entrée, est une unité chargée de transmettre une composante du vecteur $x$des données (en particulier, les données d'apprentissage pendant la phase d'apprentissage).
- Un neurone de sortie est une unité qui fournit une hypothèse d'apprentissage, par exemple dans un problème de classification, une décision sur la classe à laquelle est attribuéx.

- Enfin, un neurone caché est un neurone qui n'est ni un neurone d'entrée, ni un neurone de sortie et il est placer entre les neurones d'entrée et les neurones de sortie.

Il existe d'autres modèles, par exemple la machine de Boltzmann pour laquelle tous les neurones formels, y compris d'entrée et de sortie, sont connectés les uns aux autres.

❖ L'état d'un neurone formel

Il est commode de décrire un réseau de neurones à un moment de son fonctionnement par un ensemble de valeursσ_i, une pour chaque neurone formel$i$. Lorsque le neurone i est un neurone d'entrée, on a: $\sigma_i = x_i$. Dans tous les autres cas, σ_i est l'état du neurone$i$, calculé par la règle de propagation.

❖ Fonctionnement d'un neurone formel

Un neurone formel est caractérisé par une fonction de sortie f qui permet de calculer pour chaque neurone i une valeur de sortie y_i en fonction de son état d'activation y_i:

$$y_i = f(\sigma_i)$$

❖ Fonctions d'activation

On peut envisager plusieurs sortes de fonctions de sortie, mais le plus souvent on utilise soit :

- la fonction seuil elle vaut 0 quand σ_i est négatif et 1 s'il est positif.
- Soit une fonction sigmoïde d'équation $y_i = f(\sigma_i) = \frac{1}{1+e^{-\lambda\sigma_i}}$ Cette fonction est paramétrée par sa pente à l'origine λ. Pour λ très grand, on retrouve la fonction seuil. Pour λ très petit, cette fonction est pratiquement linéaire dans une vaste région autour de l'origine.

Pour cette figure λ=1.

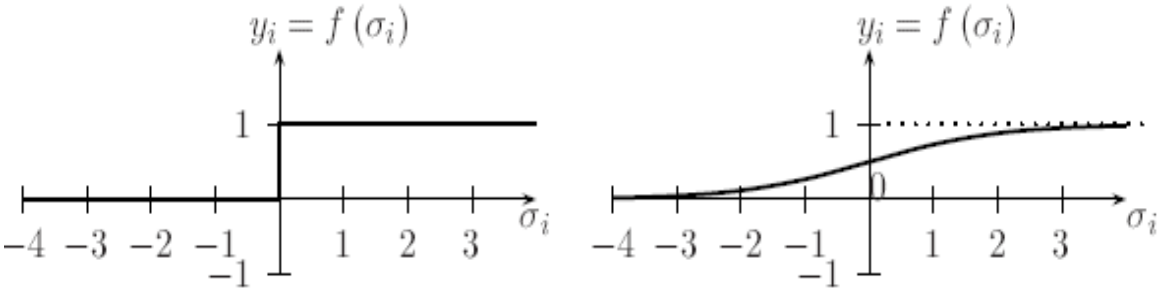

Figure 3.6 : *la fonction seuil et la fonction sigmoïde.*

Le protocole d'apprentissage :

Dans le cas des réseaux de neurones, les données d'apprentissage sont en général présentées séquentiellement ; l'apprentissage est donc incrémental. Chaque étape emploie une donnée pour modifier les poids des connexions. La suite des données utilisées peut être construite par un tirage aléatoire avec remise dans l'ensemble des exemples ou par plusieurs passages successifs de la totalité de cet ensemble. Au total, le nombre de données utilisées pour l'apprentissage est en général bien supérieur au nombre d'exemples : chacun est utilisé en moyenne ou exactement un grand nombre de fois (couramment une centaine de fois).
La caractéristique la plus intéressante d'un réseau de neurones artificiels est sa capacité d'apprendre, c'est-à-dire de modifier les poids de ses connexions en fonction des données d'apprentissage, de telle sorte qu'après un certain temps d'entrainement il ait acquis une faculté de généralisation.

3.3.5 Utilisation des réseaux de neurones pour la détection d'intrusion

On peut envisager l'application des réseaux de neurones à la détection d'intrusion de plusieurs manières :

- Pour donner une modélisation statistique du comportement des utilisateurs [28]. On est alors très proche des méthodes statistiques telles que celles présentées ci-dessus, l'avantage des réseaux de neurones étant qu'il n'est pas nécessaire de faire d'hypothèses sur les variables aléatoires.
- Pour classifier le comportement des utilisateurs [29] par un algorithme de type carte de Kohonen (classification automatique et non supervisée).
- Pour prédire le comportement des utilisateurs [27]. Le réseau apprend les séquences de commandes usuelles à chaque utilisateur. Il lui est alors possible, après chaque commande passée par l'utilisateur, de prédire la commande suivante sur la base de ce qu'il a appris. En cas de déviation entre la prévision et la réalité, une alarme est émise.
- Il faut cependant noter :
  - qu'un réseau de neurones ne fournit pas d'explication sur le raisonnement l'ayant amené à proposer un diagnostic d'intrusion.
  - que le paramétrage d'un réseau de neurones est délicat et peut influer considérablement sur les résultats fournis.

3.3.6 Les avantages et les inconvénients des réseaux de neurones

Comme chaque méthode les réseaux de neurones compte des avantages et des inconvénients.

- **Avantages**

- Capacité de représenter n'importe quelle fonction, linéaire ou pas, simple ou complexe.
- Faculté d'apprentissage à partir d'exemples représentatifs, par « rétro propagation des Erreurs ». L'apprentissage (ou construction du modèle) est automatique.
- Résistance au bruit ou au manque de fiabilité des données.
- Simple à manier, beaucoup moins de travail personnel à fournir que dans l'analyse statistique classique. Aucune compétence en matis, informatique statistique requise.
- Comportement moins mauvais en cas de faible quantité de données.
- Pour l'utilisateur novice, l'idée d'apprentissage est plus simple à comprendre.

- **Inconvénients**

- L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans la (ou les) couche(s) cachée(s).
- Le choix des valeurs initiales des poids du réseau et le réglage du pas d'apprentissage.

- Jouent un rôle important dans la vitesse de convergence. Le problème du sur-apprentissage (apprentissage au détriment de la généralisation).
- La connaissance acquise par un réseau de neurone est codée par les valeurs des poids sont inintelligibles pour l'utilisateur.

3.4 K-means

3.4.1 Historique

Cette méthode a été proposée par Stuart Lloyd en 1957 à des fins de modulation d'impulsion codée, mais elle n'a pas été publiée en dehors des Bell Labos avant 1982, Mais le terme k-means a été utilisé la première fois par MacQueen en 1967 [15].

3.4.2 Définition k-means

K-means est l'un des algorithmes d'apprentissage non supervisés les plus simples qui résolvent le problème de clustering (qui est un processus qui partitionne un ensemble de données en sous-classes (clusters) ayant du sens) [10].

- **Principe**

L'idée est de classer un ensemble de données en k nombre de grappes disjointes, où la valeur de k est fixée à l'avance. L'algorithme se compose de deux phases distinctes [10].

La première phase consiste à définir k centroïdes, un pour chaque groupe (cluster). Détaillé ces centroïdes sont choisi de manière aléatoire.

La phase suivante doit prendre chaque point appartenant à l'ensemble de données et l'associé au centriole le plus proche. Généralement on utilise la distance euclidienne pour calculer la distance entre les points de données et les centrioles. Lorsque tous les points sont inclus dans certains groupes, la première étape est terminée et un premier regroupement est effectué. À ce stade, nous devons recalculer les nouveaux centrioles, car l'inclusion de nouveaux points peut entraîner un changement dans les cent roides du cluster. Une fois que nous trouvons k nouveaux cent roides, une nouvelle liaison doit être crée entre les mêmes points de données et le nouveau centriole le plus proche, en générant une boucle. À la suite de cette boucle, les k centrioles peuvent changer leur position de manière progressive. Finalement, une situation sera atteinte là où les centrioles ne bougent plus. Cela signifie le critère de convergence pour le clustering.

3.4.3 Le pseudocode de l'algorithme de classification k-means

Entrées : $D = \{d1, d2, \dots\dots, dn\}$ensemble de n éléments de données.

k est le nombre de cluster désiré.

Sorties : un ensemble de k clusters.

Méthode :

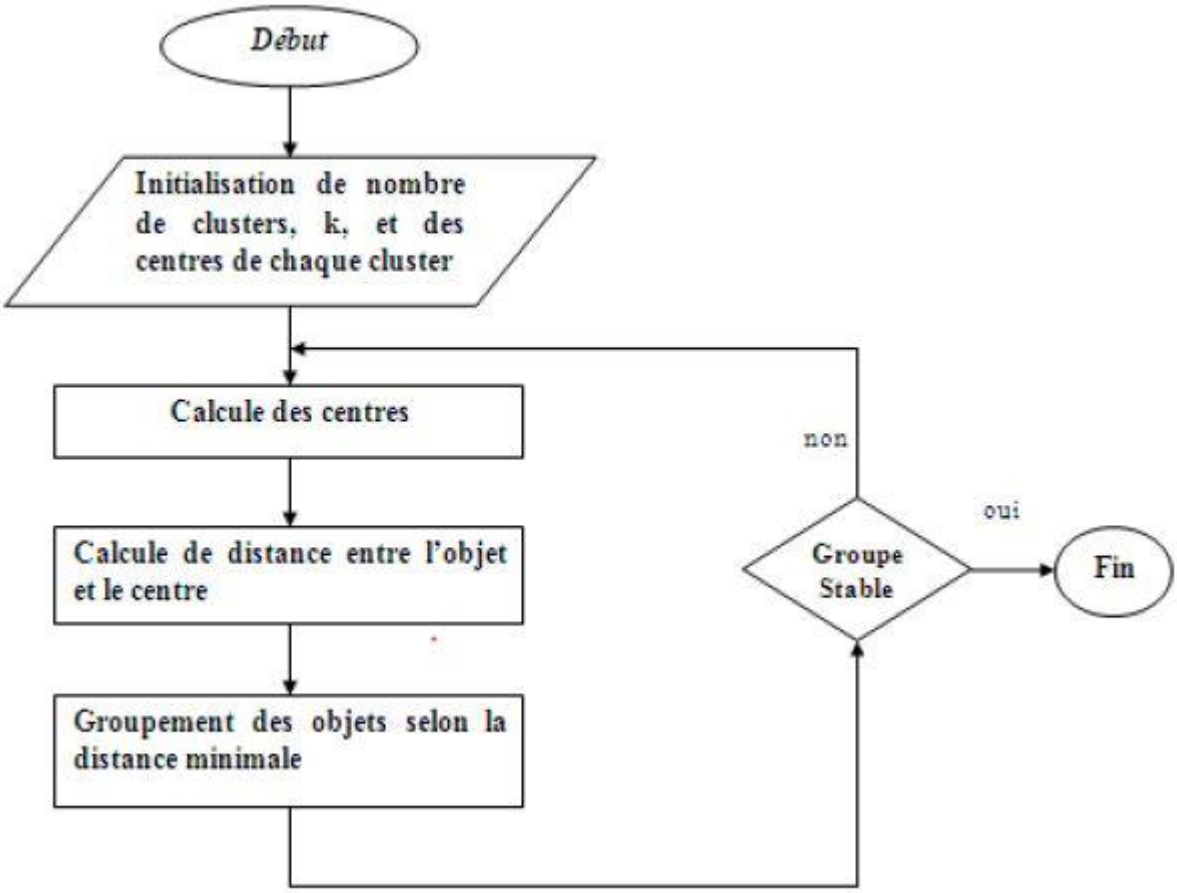

Figure 3.7 : *Organigramme de l'algorithme k-means* [10].

3.4.4 Avantages et inconvénients de k-means :

L'algorithme k-means est l'algorithme de clustering le plus étudié et il est généralement efficace pour produire de bons résultats mais cette méthode aussi présente des inconvénients [10].

- **Avantages**

- L'algorithme de k-means est très populaire du fait qu'il est très facile à comprendre et à mettre en œuvre.
- Sa simplicité conceptuelle.
- Applicable à des données de grandes tailles, et aussi à tout type de données, en choisissant une bonne notion de distance.

- **Inconvénients**

- Le nombre de classe doit être fixé au départ.
- La qualité des clusters finaux dépend fortement de la sélection des centroïdes initiaux.
- Les clusters sont construits par rapports à des objets inexistants (les milieux).
- Coûteux en termes de calcul et nécessite un temps proportionnel au produit du nombre d'éléments de données.

3.6 Description de la base de donnés NSL-KDD

La base de données NSL-KDD comporte 125973 trames, chaque trame est soit une attaque ou une trame normale. Cette base de données contient 22 types d'attaque, on peut répartir ces attaques en 4 catégories [38].

Catégorie	**Type d'attaque**	**Nombre de trame**	**Taux de trame %**
Normal	Normal	67343	53.46
Dos	neptune, smurf, teardrop , pod, back, land	45927	36.46
Prob	portsweep, ipsweep, nmap, satan	11656	9.25
R2L	guess_passwd, ftp_write, imap, phf, warezclient, multihop, warezmaster, spy	995	0.79
U2R	buffer_overflow, loadmodule, perl, rootkit	52	0.04

Tableau 3.1: *représentation de la base de données NSL-KDD.*

3.7 Les résultats des méthodes

La comparaison base sur Taux de précision.

3.7.1 Les résultats de SVM

Type d'attaque	Taux de précision.
Dos	79.25%
Normal	79.55%
Prop	79.70%
R2L	79.78%
U2R	79.87%

Tableau 3.2 : *Taux de précision de SVM*

3.7.2 Les résultats des réseaux de neurones

Type d'attaque	Taux de précision.
Dos	77.50%
Normal	79.60%
Prop	72.70%
R2L	75.00%
U2R	48.00%

Tableau 3.3 : *Taux de précision des réseaux de neurones*

3.7.3 Les résultat de K-means

type d'attaque	Taux de précision.
Dos	75.15%
Normal	69.29%
Prop	15.70%
R2L	06.76%
U2R	51.72%

Tableau 3.4 : *Taux de précision de K-means*

3.7.4 La comparaison entre les résultats

Les méthodes	**Taux de précision Max**	**Taux de précision Min**	**Taux de précision Moyen**
SVM	U2R = 79.87%	Dos = 79.25%	79.63%
Les réseaux de neurones	Normal =79.60%	U2R = 48.00%	66.56%
k-means	Dos =75.15%	R2L =06.76%	43.72%

Tableau 3.5 *: tableau comparatif des taux de précision entre le SVM, K-means et Les réseaux de neurones*

3.8 Conclusion

Ce chapitre nous a permis de découvrir et comparer les différentes techniques de machine Learning pour détection d'intrusion et leurs fonctionnements. D'après les résultats on constate que la méthode de SVM est le plus puissant en taux de précisions par apport à la méthode de k-means et les réseaux de neurones.

Conclusion générale

Dans ce travail on s'est intéressé aux techniques de détection d'intrusion, nous avons précisé sur la technique de machine Learning ainsi que on a fait la comparaison entre les méthodes de ce dernier. Cette comparaison est basée sur le taux de précisions à la base de données NSL-KDD.

D'après les résultats on constate que la méthode de SVM est le plus puissant en taux de précisions par apport à la méthode de k-means et les réseaux de neurones.

Bibliographie:

[1] Mahbod Tavallae, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani "A Detailed Analysis of the KDD CUP 99 Data Set", Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).

[2] Sapna S. Kaushik, Dr. Prof.P.R.Deshmukh," Detection of Attacks in an Intrusion Detection System", International Journal of Computer Science and Information Technologies, Vol. 2 (3), 2011, 982-986.

[3] Introduction et Initiation à la sécurité informatique. « SecuriteInfo.com ».

[4] S.Northcut, J.Novak, D.Mclachlan. (2001). « Détection des intrusions réseaux ».

[5] Tarek Abbes. (2004) « Classification du trafic et optimisation des règles de filtrage pour la détection d'intrusions ». Thèse de doctorat de l'université Henri Poincaré.Nancy1 .2004.

[6] Cedric Michel and Ludovic Me. ADeLe : an Attack Description Language for Knowledge-based Intrusion Detection. In Proceedings of the 16th IFIP International Conference on Information Security (IFIP/SEC 2001), pages 353–365, June 2001.

[7] Marcus A. Maloof (2005), _Machine Learning and Data Mining for Computer Security_ , Springer London Ltd, ISBN-10 184628029X ; ISBN-13 978-1846280290.

[8] P. Laskov & P. Düssel & C. Schäfer & K. Rieck (2005), _Learning intrusion detection: Supervised or unsupervised? _, Fraunhofer-FIRST.IDA, Berlin,Germany.

[9] N. Pasquier. *"Data Mining : algorithmes d'extraction et de réduction des règles d'association dans les bases de données"*. Doctorat d'université, Université de Clermont-Ferrand II, France, 2000.

[10] Z.Guellil et L.Zaoui, « Proposition d'une solution au problème d'initialisation cas du K-means », livre : CIIA, volume 547 of CEUR Workshop Proceedings, CEUR-WS.org, Université des Sciences et de la Technologie, Oran – Algérie, 2009.

[11] Alain B & P B "Data mining I Exploration Statistique". Version septembre 2005

[12] A. Öksüz (2007), _Unsupervised Intrusion Detection System_ , Technical University of Denmark, Informatics and Mathematical Modelling, Denmark.

[13] R. Heady, G. Luger, A. Maccabe, and M. Servilla. The architecture of a network level intrusion detection system. Technical report, Department of Computer Science, University of New Mexico, Agosto 1990.

[14] James P. Anderson. Computer security threat monitoring and surveillance. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, April 1980.

[15] Julia ALLEN, Alan CHRISTIE, William FITHEN, John MCHUGH, Jed PICKEL, Ed Stoner , State of the Practice of Intrusion Detection Technologies, Networked Systems Survivability Program , 2000.

[16] A. Phillip, Porras and Alfonso Valdes. Live traffic analysis of tcp /ip gateways.
Proc. ISOC Symposium on Network and Distributed System Security (NDSS98). (San Diego, CA, March 98), Internet Society.

[17] H. Debar, M. Dacier & A. Wespi. "A revised taxonomy for intrusion detection systems. Annales des télécommunications''. July–August 2000.

[18] Frederic Cuppens and Rodolphe Ortalo. LAMBDA: A Language to Model a Database for Detection of Attacks. In H. Debar, L. Me, and S. F. Wu, editors, Proceedings of the Third International Workshop on the Recent Advances in Intrusion Detection.

[19] Steven T. Eckmann, Giovanni Vigna, and Richard A. Kemmerer. Statl: an attack language for state-based intrusion detection. Journal of Computer Security, 10(1-2):71–103, 2002.

Cedric Michel and Ludovic Me. Adele: an attack description language for knowledge-based intrusion detection. In Proceedings of the 16th International Conference on Information Security (IFIP/SEC 2001), pages 353–365, June 2001.

David Brumley, James Newsome, Dawn Song, HaoWang, and Somesh Jha. Towards automatic generation of vulnerability-based signatures. In SP '06 : Proceedings of the 2006 IEEE Symposium on Security and Privacy, pages 2–16, Washington, DC, USA, 2006. IEEE Computer Society.

[20] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon, Stuart Staniford, and NicholasWeaver. Inside the slammer worm. Security and Privacy, 1(5):33–39, September-October 2003.

[21] Vern Paxson. Bro: A system for detecting network intruders in real-time. In *Proceedings of the 7th Usenix Security Symposium*, pages 31–51, San Antonio, TX, January 1998.

[22] Biswanath Mukherjee, L. Todd Heberlein, and Karl N. Levitt. Network intrusion detection. IEEE Network, 8(3):26–41, May- June 1994.

[23] Phillip A. Porras and Peter G. Neumann. EMERALD: Event monitoring enabling responses to anomalous live disturbances. In Proc. of the 20th National Information Systems Security Conference, pages 353–365, Baltimore, MD, October 1997.

[24] Steven T. Eckmann, Giovanni Vigna, and Richard A. Kemmerer. Statl : an attack language for state-based intrusion detection. Journal of Computer Security, 10(1-2) :71–103, 2002.

[25] Stefan Axelsson. Intrusion detection systems: A taxonomy and survey. Technical Report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, March 2000.

[26] D.E. DENNING. (An Intrusion-Detection Model). IEEE transaction on Software Engineering, 13(2):222–232, 1987.

[27] Elvis Tombini, Herve Debar, Ludovic Me, and Mireille Ducasse. A serial combination of anomaly and misuse IDSes applied to HTTP traffic. In Proceedings of ACSAC'2004, pages 428–437, Tucson, AZ, December 2004,

[28] Proc. 9th IEEE Int. Conf. on Cognitive Informatics (ICCI'10) F. Sun, Y. Wang, J. Lu, B. Zhang, W. Kinsner & L.A. Zadeh (Eds.) 978-1-4244-8040-1/10/$26.00 ©2010 IEEE.

[29] Herve´ DEBAR, Monique BECKER, et Didier SIBONI. ((A Neural Network Component for an Intrusion Detection System)). Dans *Proceedings of the IEEE Symposium of Research in Computer Security and Privacy*, pages 240–250, May 1992. Herve´ DEBAR. ((*Application des re´seaux de neurones a` la de´tection d'intrusions sur les syste`mes informatiques*)). The`se de doctorat, Universite´ de Paris 6, 1993.

[30] Elvis Tombini, Herve Debar, Ludovic Me, and Mireille Ducasse. A serial combination of anomaly and misuse IDSes applied to HTTP traffic. In Proceedings of ACSAC'2004, pages 428–437, Tucson, AZ, December 2004,

Herve Debar and Elvis Tombini. Webanalyzer : Accurate and fast detection of http attack traces in web server logs. In Proceedings of EICAR 2005, Malta, 2005.

[31] Mohamadally Hasan Fomani Boris BD Web, ISTY3 Versailles St Quentin, France.

[32] C. JUNLI, J. LICHENG, Classification mechanism of support vector machines, *Signal Processing Proceedings WCCC-ICSP, 5th IEEE International Conference on Volume 3*, Page(s): 1556–1559, 21-25 Aug 2000.

[33] live de Apprentissage-Artificiel-Concepts-et-Algorithmes-par-www.heights-book.blogspot.com.

[34] C.touzet, 1992 (LES RESEAUX DE NEURONES ARTIFICIELS, INTRODUCTION AU CONNEXIONNISME) ,127p.

[35] https://www.universalis.fr/encyclopedie/reseaux-de-neurones-formels/2-quelques definitions, (Consulté le 12 septembre 2018).

[36] http://www.mathworks.com/help/matlab/learn_matlab/product-description.html , (Consulté le 13 septembre 2018).

[37] J. B. MacQueen (1967). « Some Methods for classification and Analysis of Multivariate Observations [archive] » dans Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1: 281–297 p. Consulté le 7 September 2009.

[38] S. Revathi, Dr. A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 2 Issue 12, December - 2013].