République Algérienne Démocratique Et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université d'Ibn Khaldoun – Tiaret

Faculté des Mathématiques et de l'Informatique

Département Informatique

Thème

Apports des réseaux bayésiens dans les systèmes de détection d'intrusions

Pour l'obtention du diplôme de Master

Spécialité : Génie logiciel

Réalisé par : Sonia SAIDI.

Dirigé par : Mr. ABDELKADER Alem.

Année universitaire 2015-2016

Dédicaces

A l'aide de DIEU tout puissant, qui trace le chemin de ma vie, j'ai pu arriver à réaliser ce modeste travail que je dédie:

Amon très cher père et ma très chère mère qui n'ont pas cessé de m'encourager et de se sacrifier pour que je puisse franchir tout obstacle durant toutes mes années d'étude que Dieu me les garde en très bonne santé ; Aucune dédicace ne pourra compenser les sacrifices de mes parents;

A ma grand-mère et mon grand-père que dieu me les garde en bonne santé ;

A ma petite sœur Rania et mes frère Yassine, Karim et surtout mon petit frère Amine, que je leur souhaite une longue vie pleine de joie et de réussite.

A mes oncles, mes tantes, mes cousines, mes cousins, et à toute ma famille ;

A mes amies et mes collègues, qui m'ont aidé tout au long de ces années, je vous souhaite une vie pleine de joie, de bonheur et de réussite ;

A mes enseignants et surtout Mr Abdelkader Alem, mon encadreur;

Et à tous ceux qui m'aiment et qui me connaissent de proche ou de loin.

Remerciements

Avec un grand plaisir je remercie ***Allah*** qui m'a aidé et m'a donné la patience, le courage et la force d'achever ce travail.

Je tiens à remercier nos professeurs qui nous ont enseigné durant nos études à la faculté IBN KHALDOUN, département d'informatique Tiaret.

Je tiens à remercier sincèrement ***Mr.ALEM Abdelkader***, qui, en tant que encadreur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'orientation, la confiance, l'aide et le temps qu'il a bien voulu me consacrer et sans lui ce mémoire n'aurait jamais vu le jour.

J'exprime également ma gratitude aux membres du jury ***Mr.Meghazi Lhaj*** et ***Mr. Mestfaoui Sid Ahmed***, qui m'ont honoré en acceptant de juger ce modeste travail.

Je tiens à remercier sincèrement mon père et ma mère, de m'avoir encouragé durant toute la durée de mes études et de m'avoir apporté un soutien indispensable à ma réussite

Je souhaite d'adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire et surtout ma chére amie Slimane Ahlem.

Résumé

Les systèmes de détection d'intrusion (IDS) sont devenus très indispensable pour tout réseau informatique, il nous permet de connaitre toutes activités anormales qui peuvent présenter un danger. Dans ce travail, nous nous somme intéressés à la modélisation du problème de la détection d'intrusion à base de modèles graphiques probabilistes (Réseaux bayésien naif) et les SVMs dans l'objectif est de minimiser le nombre des fausses alarmes et augmenter les performances du système. Les approches existantes soit se basent sur des connaissances d'experts, soit utilisent des simples mesures de similarité qui ne permettent pas de détecter des attaques. Elles souffrent également d'une complexité de calcul très élevée dû par exemple à un grand nombre d'alertes. L'idée suggérée est de faire coopérer deux modules de détection, un réseau bayésien naïf sachant la décision d'un module SVM. Notre approche est illustrée en utilisant une base de données récente qui à montré de très bonne résultats.

Mots-clés : IDS, Réseau bayésien, SVM, sécurité, approche comportementale.

Abstract

Intrusion detection systems (IDS) have become very essential for any computer network, it allows us to know all abnormal activities that may present a danger. In this work, we are interested in modeling the problem of intrusion detection based on probabilistic graphical models (naive Bayesian Networks) and SVMs in the objective is to minimize the number of false alarms and increase the performance of the system. Existing approaches are based on expert knowledge, either use simple similarity measures which do not allow to detect attacks. They also suffer from a computational complexity very high due for example to a large number of alerts. The suggested idea is to cooperate two detection modules, a Bayesian network naive knowing the decision of one module SVM. Our approach is illustrated using a recent database which to shown very good results.

Keywords : IDS, Bayesian, SVM, security, behavioral approach network.

SOMMAIRE

Chapitre 02 : Les Machines à Vecteurs de Support (SVM)

Chapitre3 : Réseaux Bayésiens

Chapitre4 : Utilité des RB dans la Détection d'intrusion.

Chapitre5 : Implémentation et réalisation.

Liste des figures

Liste des tableaux

Introduction général

Actuellement, les outils de piratage et des attaques informatiques sont disponible aux experts comme aux amateurs. Une attaque réussie peut engendrer de très grave pertes, on parle aujourd'hui de plusieurs milliards de dollars de perte, des pays paralysés, des projets stratégiques sabotés, des programmes présidentiels divulgués tout ça à cause d'une attaque informatique qui varie dans le but, l'ampleur, et la dangerosité. La sécurité informatique est devenue une obligation pour toute organisation, pour faire face aux attaques afin de minimiser les risques.

Les systèmes et réseaux informatique contient diverses formes de vulnérabilité. Pour faire face à ces problème de sécurité informatique, défirent mécanisme ont été mis en place pour prévenir toute sorte d'attaque comme les pare feux, l'authentification, les proxys...etc. Malheureusement, ces mécanismes ont des limites ou certains types d'attaques peuvent les contourner pour nuire la confidentialité, l'intégrité ou la disponibilité. Pour faire face à ce problème, un nouveau concept qui s'appelle système de détection d'intrusion a été introduit comme une seconde ligne dedéfense afin de renforcer la sécurité des systèmes informatiques.

Après la publication du premier modèle de détection d'intrusion par Denning (Denning, 1987) plusieurs travaux ont été faits pour créer un système de détection d'intrusion performant et très précis. La conception des systèmes de détection d'intrusion a été basée sur les connaissances des experts de sécurité ou les méthodes statistiques et les approches de l'intelligence artificielle ont été utilisées pour créer les noyaux (moteurs) des modèles de détection d'intrusion. Face à des Problèmes tels que le grand volume du trafic réseau, la distribution des données très déséquilibrée, la difficulté de prendre une décision entre normal et anormal,L'objectif de la détection d'intrusions est d'automatiser la tâche d'audit. Il s'agit bien, théoriquement, de détecter de manière automatique les violations de politique de sécurité, qu'on appelle intrusions.

Notre travail s'articule autour de ce domaine dont il consiste à sécuriser les réseaux informatique de toute sorte de vulnérabilité à l'aide d'un système de détection d'intrusion comportementale.

L'objectif de notre travail est de créer un système de détection d'intrusion capable de gérer de grandes quantités d'alertes dont la plupart sont fausse et redondantes et diminuer le taux de ces fausses alertes. Pour atteindre notre objectif, nous avons proposé une approche basée sur deux classificateurs Réseaux Bayésien naïf et Les Machines de support vecteur (SVM).

Le premier chapitre est un chapitre descriptif pour la sécurité des réseaux, sur lequel on va définir les attaque, les intrusion, citez les défirent attaques réseaux et la politique de sécurité ainsi les principaux mécanismes de sécurité. Dans la deuxième partie on vaprésenter une architecture globale d'un IDS, la définition et le mode de fonctionnement de ce dernier. Ainsi la classification des IDS.

Le second chapitre est consacré aprésenté une généralisation des machine a support vecteur (SVM) qui est l'un des classificateur utiliser dans l'approche proposées, le troisième chapitre présente la définition et le mode de fonctionnement du deuxième classificateur, Réseaux Bayésien Naïf.

Le quatrième chapitreest divisé en deux parties,la première partie est consacré à présenterla structure, le mode de fonctionnement et l'architecture de l'approche proposée et la deuxième partie est consacré à présenter les résultats obtenu durant l'expérimentation de cette approche.

Le dernier chapitre présente l'implémentation et la réalisation de l'application qui représente note IDS comportementaux (approche proposée).

Chapitre1 :

La sécurité informatique et les IDS (Système de détection d'intrusion)

Chapitre1 : La sécurité informatique et les IDS (Système de détection d'intrusion)

Introduction :

La sécurité informatique est de nos jours devenue un problème majeur dans la gestion des réseaux d'entreprise ainsi que les violations peuvent avoir toujours lieu. Le problème n'est plus de savoir si on va se faire attaquer, mais à quel moment cela arrivera. Ainsi, des outils, de diagnostic et de détection, doivent être utilisés pour compléter la protection des systèmes d'information et permettre par la suite d'apporter les correctifs appropriés.

Dans ce chapitre introductif, nous commençons par présenter les notions de base en sécurité informatique en mets l'accent beaucoup plus sur la sécurité réseau et les attaques orientées réseau reposantes sur les faiblesses de sécurité Ensuite nous abordons une généralisation des systèmes de détection d'intrusion et enfin on termine par dresser une classification de la détection d'intrusions.

I. La sécurité informatique :

1. Définition :

Selon **[1]** « la sécurité informatique est l'ensemble des moyens mis en œuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles »

Elle consiste donc à protéger les informations contre la consultation abusive, la modification ou la destruction non autorisée et fournit les outils pour protéger l'information vitale et préserver ainsi son avantage compétitif. Elle permet ainsi la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles. Cependant, aucune technologie ne peut à elle seule sécuriser l'information à 100%, il est nécessaire ainsi de combiner plusieurs contrôles, de mettre en œuvre différents moyens de protection et de les faire évoluer en même temps que les menaces.

La sécurité du système informatique cherche à apporter une meilleure maitrise des risques qui pèsent réellement et rependre à certains enjeux qu'on peut résumer en quatre Lettres « D.I.C.A » selon les termes de Eric Léopold et Serge Lhoste **[2]** « On a l'habitude de classer les risques en quatre grandes familles : disponibilité, intégrité, confidentialité et auditabilité. ». Une seule entrave à l'un de ces principes remet toute la sécurité en cause

Figure 1 : Familles des risques.

- ✓ **Disponibilité :** garantir l'accès aux ressources, au moment voulu, aux personnes habilitées d'accéder à ces ressources.
- ✓ **Intégrité :** garantir que les données échangées sont exactes et complètes.
- ✓ **Confidentialité :** garantir que seules les personnes autorisées peuvent avoir accès aux données et aux ressources de l'entreprise.
- ✓ **Auditabilité :** garantir la traçabilité des accès et des tentatives d'accès et la conservation de ces traces comme preuves exploitables.

La sécurité informatique vise à inscrire l'évolution des systèmes dans le cadre d'un processus d'amélioration continue. La politique de sécurité du système informatique reflète donc la vision stratégique de la direction de l'organisme en matière de sécurité des systèmes d'information et de gestion de risques.

2. La politique de la sécurité informatique :

La sécurité informatique doit toutefois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, dont la mise en œuvre Se fait selon les quatre étapes suivantes : **[6]**

- ✓ Identifier les besoins en termes de sécurité, les risques informatiques pesant sur l'entreprise et leurs éventuelles conséquences.

- ✓ Elaborer des règles et des procédures à mettre en œuvre dans les différents services de l'organisation pour les risques identifiés.
- ✓ Surveiller et détecter les vulnérabilités du système d'information et se tenir informé des failles sur les applications et matériels utilisés.
- ✓ Définir les actions à entreprendre et les personnes à contacter en cas de détection d'une menace.

3. Services principaux de la sécurité informatique :

Pour remédier aux failles et pour contrer les attaques, la sécurité informatique se base sur un certain nombre de services qui permettent de mettre en place une réponse appropriée à chaque menace. Les principaux services sont :

- ✓ **La confidentialité :** la confidentialité consiste à rendre l'information inintelligible à d'autres personnes que les seuls acteurs de transaction.
- ✓ **L'intégrité de données :** Vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication (de maniéré fortuite ou intentionnelle).
- ✓ **La disponibilité :** Permettant de maintenir le bon fonctionnement du système informatique.
- ✓ **La non-répudiation :** Permettant de garantir qu'une transaction ne peut être niée.
- ✓ **L'authentification :** L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être.

4. Sécurité réseau :

4.1. Définitions :

- **Attaque :**

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 parl'exploitant du système et généralement préjudiciables. Sur internet des attaques ont lieu en permanence, à

raison de plusieurs attaques par minute sur chaque machine connectée. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées (par des virus, chevaux de Troie, vers, etc.), à l'insu de leur propriétaire. Plus rarement il s'agit de l'action de pirates informatiques. Nous distinguons deux types d'attaques :

- Les attaques passives : consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible.
- Les attaques actives : consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute.

- **Intrusion**

Une intrusion est toute utilisation d'un système informatique à des fins autres que celles prévues, généralement dues à l'acquisition de privilèges de façon illégitime.**[21].**

Une intrusion dans un système informatique est aussi définie par Heady et all **[22]** comme « N'importe Quelle ensemble d'actions essayant de compromettre l'intégrité, la confidentialité ou l'accessibilité d'une ressource ».

4.2. Les différents types d'attaques :

Les attaques réseaux les plus connues aujourd'hui sont :

4.2.1. Les attaques réseaux :

- **Spoofing IP :**

Le spoofing IP est une technique permettant à un pirate d'envoyer à une machine des paquets semblant provenir d'une adresse IP autre que celle de la machine du pirate. Le spoofing IP n'est pas pour autant un changement d'adresse IP. Plus exactement, il s'agit d'une mascarade de l'adresse IP au niveau des paquets émis, c'est-à-dire une modification des paquets envoyés afin de faire croire au destinataire qu'ils proviennent d'une autre machine.**[8]**

- **Spoofing ARP :**

Le spoofing ARP est une technique qui modifie le cache ARP. Le cache ARP contient une association entre les adresses matérielles des machines et les adresses IP, l'objectif du pirate est de conserver son adresse matérielle, mais d'utiliser l'adresse IP d'un hôte approuvé. Ces informations son simultanément envoyées vers la cible est vers le cache. A partir de cet instant, les paquets de la cible seront routés vers l'adresse matérielle du pirate.

- **Spoofing DNS:**

Le système DNS (Domaine Name system) a pour rôle de convertir un nom de domaine en son adresse IP et réciproquement, à savoir : convertir une adresse IP en un nom de domaine. Cette attaque consiste à faire parvenir de fausses réponses aux requêtes DNS émises par une victime. Il existe deux types de méthode :**[9]**

a. DNS ID spoofing : L'attaquant essaie de répondre à un client en attente d'une réponse d'un serveur DNS, avec une fausse réponse et avant que le serveur DNS ne réponde.

b. DNS Cache poisoning : L'attaquant essaie d'empoisonner le cache (table de correspondance IP-nom_machine) du serveur DNS.

4.2.2. Les attaques par déni de service :

Les attaques par déni de service (souvent abrégé DOS, en anglais Danial Of Service) consistent à paralyser temporairement (rendre inactif pendant un temps donné) des serveurs afin qu'ils ne puissent être utilisée et consultés. Le but d'une telle attaque n'est pas de récupérer ou d'altérer les données, mais de nuire à des sociétés dont l'activité repose sur un système d'information. En l'empêchant de fonctionner**. [9]**

- **La technique dite du smurf :**

La technique dite du smurf est basé sur l'utilisation de serveurs broadcast pour paralyser un réseau. Un serveur broadcast est un serveur capable de dupliquer un message et de l'envoyer à toutes les machines présentes sur le même réseau que lui. Le scénario d'une attaque est le suivant **[10]** :

La machine attaquante envoie un ping**[11]** à un ou plusieurs serveurs broadcast en falsifiant sa propre adresse IP (l'adresse à laquelle le serveur devrait théoriquement répondre

par un ping) et en fournissant l'adresse IP de la machine cible. Lorsque le serveur broadcast va dispatcher le ping sur tout le réseau, toutes les machines du réseau vont répondre par un pong, que le serveur broadcast va rediriger vers la machine cible. Ainsi lorsque la machine attaquante adresse le ping à plusieurs serveurs broadcast situés sur le réseau différents, l'ensemble des réponses de tous les ordinateurs des différents réseaux vont être reroutées sur la machine cible.

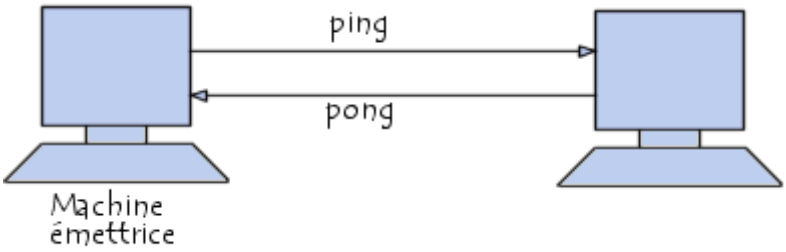

Figure 2 : ping et pong

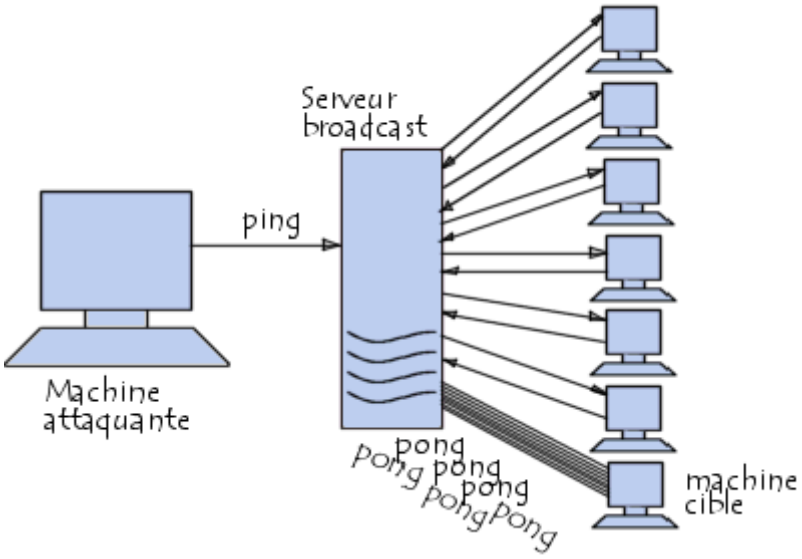

Figure 3 : Exemple d'attaque smurf.[10]

- **SYN FLOOD :**

Son objectif est de rendre indisponible un service TCP offert sur une machine. Le principe de cette attaque est de créer des connexions TCP semi-ouvertes sur la machine cible afin de remplir la file d'attente ou sont stockées les demandes d'ouverture de connexion. L'attaquant envoie un grand nombre de requête SYN à la machine cible et remplace son

adresse source avec l'adresse d'une machine indisponible ou inexistante afin que les réponses SYN/ACK ne soient jamais reçue et que donc les messages ACK ne soient jamais générés, ce qui signifie que la fille d'attente restera pleine. Les conséquences de cette attaque sont que toutes requêtes arrivant sur le port TCP cible seront ignorées et ce fait le service fourni sur ce port sera indisponible. Dans certains cas, la machine peut aussi devenir indisponible.**[3]**

- **Fragmentation :**

L'attaquant sature la connexion en envoyant des fragmentations déclenchant des exceptions (faille de la pile TCP/IP de Windows 95 et 98). **[9]**

4.2.3. Les attaques virales :

Il existe principalement quatre types de menaces distinctes **[9]** :

- ✓ **Virus :** Se reproduisent en infectant le corps de programmes hôtes.
- ✓ **Vers :** Le vers se duplique et se propage à travers le réseau, par courrier électronique par exemple.
- ✓ **Chevaux de Troie :** Exécutent des taches malignes en se cachant dans un programme sain. Il peut par exemple voler des mots de passe, copier des données, ou exécuter toute autre action nuisible.
- ✓ **Trappes (portes dérobées) :** Permet à un utilisateur externe de prendre le contrôle d'une application par des moyens détournés.

4.3. Outils de sécurité :

Nous avons constaté que les attaquants disposent de plusieurs moyens pour réussir leurs attaques. La disponibilité des outils d'attaques et la richesse des sources d'informations accentuent le risque des intrusions. Par conséquent, les administrateurs sécurisent de plus en plus leurs systèmes informatiques. Ils s'appuient sur diverse solution comme les pare feux, la cryptographie, le mot de passe, les scanners de vulnérabilités et les systèmes de détection d'intrusion. Nous détaillons dans la suite chacune de ces méthodes :

4.3.1. Cryptographie, Signature et Certificat :

- **Cryptographie :**

Le mot cryptographie est un terme générique désignant l'ensemble de techniques permettant de chiffrer des messages. Chiffrer un message consiste à le transformer au moyen d'un algorithme mathématique afin de le rendre inintelligible, sauf pour celui qui possède le moyen (une clé) de le déchiffrer. L'encryptions des informations électriques transite par le réseau, est utilisée pour assurer la confidentialité et l'authenticité des transactions. Le chiffrement se fait généralement à l'aide d'une clé de chiffrement, le déchiffrement nécessite quant à lui une clé de déchiffrement. Nous distinguons deux types de clé (Les clés symétriques et Les clés asymétriques).**[12]**

- **Signature électronique :**

Signature électrique est un code digital permet à la personne qui reçoit une information de contrôler l'authenticité de son origine, et également de vérifier que l'information en question est intacte. Aussi, les signatures électriques permettent l'authentification et le contrôle de l'intégrité et également le non répudiation.**[12]**

- **Certificat :**

Certificat est document électrique, carte d'identité émis par une autorité de certification. Il valide l'identité des interlocuteurs d'une transaction électrique, associe une identité à une clé publique d'encryptions et fournit des informations de gestion sur le certificat et le détenteur.**[13]**

4.3.2. Firewall :

Un firewall est un assemblage matériel (ordinateur) et des logiciels installés sur celui-ci dont l'objectif principal est de protéger le réseau interne contre les accès et actions non autorisés en provenance de l'extérieur, en contrôlant le trafic entrant et sortant.**[14]**

Il existe plusieurs types de techniques de firewall **[15]** :

- ✓ La technique de filtrage des paquets : chaque paquet d'information entrant ou sortant est accepté ou rejeté selon des règles établies par l'utilisateur.

- ✓ La technique des serveurs proxy : qui empêche l'extérieur de connaitre les adresses internes du réseau.
- ✓ La technique des passerelles : qui fournissent des systèmes de sécurité pour établir des connexions TCP/IP entre l'extérieur et l'intérieur ou pour certains services comme FTP et TELNET.

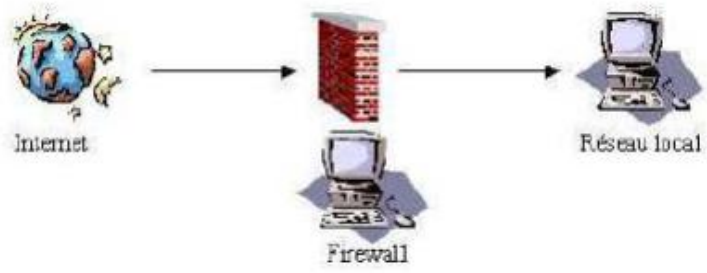

Figure 4 : Placement d'un firewall

4.3.3. Scanners de vulnérabilités :

Les scanners de vulnérabilités automatisent la découverte des failles de sécurité. Ils sont utilisés par les attaquants pour localiser les faiblesses du réseau cible. De plus, les administrateurs peuvent en tirer profit pour corriger les vulnérabilités de leurs systèmes informatiques. Nous citons à titre d'exemple Nessus **[18]** et Saint **[17].**

Cependant les scanners présentent quelques limites qui peuvent être résumées en trois points : l'exhaustivité, la mise à jour et l'exactitude. En effet, malgré le grand nombre de vulnérabilités détectées, les scanners d'aujourd'hui sont inaptes à déterminer toutes les faiblesses possibles.**[16]**

4.3.4. Fichiers historiques :

La consultation régulière des fichiers historiques constituée doit notamment permettre de vérifier les anomalies dans le trafic des trafics des transactions. Par exemple, les messages répétitifs en provenance d'une même adresse extérieure et rejetés par le Firewall peuvent être un signe d'essai d'intrusion. Cependant, ces fichiers peuvent être consultés par les hackers afin d'effacer leurs traces. Pour cela, les administrateurs sauvegardent des copies de ces fichiers (back up) dans des endroits sécurisés et sur des machines différentes.**[3]**

4.3.5. Pot de miel :

Les pots de miel sont des systèmes qui simulent plusieurs services réseaux pour leurrer des intrus en exposant des vulnérabilités connues délibérément.

Un attaquant pense que ces services vulnérables sont actifs et qu'il peut les utiliser pour s'introduire dans le réseau. Il s'y colle pendant un certain temps. Pendant ce temps l'administrateur enregistre les activités de l'intrus pour découvrir ses actions et ses techniques.

Une fois ces techniques sont connues. L'administrateur emploie ces informations plus tard pour durcir la sécurité sur les serveurs réels.**[16]**

4.3.6. Systèmes de détection d'intrusions :

- **Contexte**

Le concept de système de détection d'intrusion a été introduit en 1980 par Anderson **[19].** Mais le sujet n'a pas eu beaucoup de succès. Il a fallu attendre la publication d'un modèle de détection d'intrusion par Denning**[20]** en 1987 pour marquer réellement le départ du domaine.

La recherche dans le domaine s'est ensuite développée, Le gouvernement des Etats-Unis a investi des millions de dollars dans ce type de recherches dans le but d'accroitre la sécurité de ses machines **[20].**

Actuellement les IDS sont très populaire à cause de **[23]** :

- ✓ L'évolution continue des attaques.
- ✓ L'apparition de nouvelles attaques.
- ✓ La nécessite de détecter et réagir le plus vite possible aux attaques survenant dans le réseau.

Snort**[24]** est un exemple de IDS Open source disponible au grand public basé sur l'approche par connaissance (nous détaillons cette approche dans la 2eme partie de ce chapitre).

II. Système de détection d'intrusion :

1. Définition :

Un IDS (Intrusion Détection System) est un outil logiciel ou matériel qui permet d'écouter le trafic réseau de façon furtive dans le but de détecter des activités anormales qui pourraient être assimilées à des intrusions définies comme des tentatives pour compromettre la confidentialité, l'intégrité et la disponibilité d'une ressource ou éviter des mécanismes de sécurité de l'ordinateur ou du réseau.

Les IDS traditionnellement suivent deux critères :

- **Fiabilité :** toute intrusion doit effectivement donner lieu à une alerte. Une intrusion non signalée constitue une défaillance de l'IDS, appelée faux négatif. (voir Figure 5)
- **Pertinence des alertes :** toute alerte doit correspondre à une intrusion effective. Toute « fausse alerte » (appelée également faux positif) diminue la pertinence de L'IDS. (voir Figure 5)

Un IDS est parfaitement fiable en absence de faux négatif ; il est parfaitement pertinent en l'absence de faux positif**[26]** .

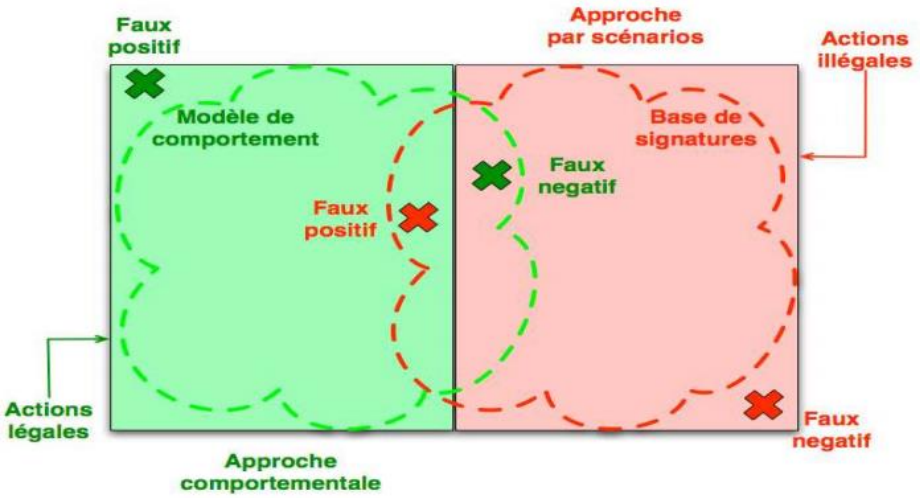

Figure 5: Problèmes des IDS

Les IDS proposent les fonctions suivantes:

- ✓ Détection d'attaques (actives ou passives).
- ✓ Génération des rapports.

- ✓ Outils de corrélation avec d'autres éléments de l'architecture de sécurité.
- ✓ Réaction aux attaques par le blocage de route ou la fermeture de connexion.
- ✓ Transfert d'activités.

2. Description du système de détection d'intrusion :

Comme l'illustre la figure 6, un système de détection d'intrusions est simplifié par un détecteur qui analyse les informations en provenance du système surveillé.**[27]**

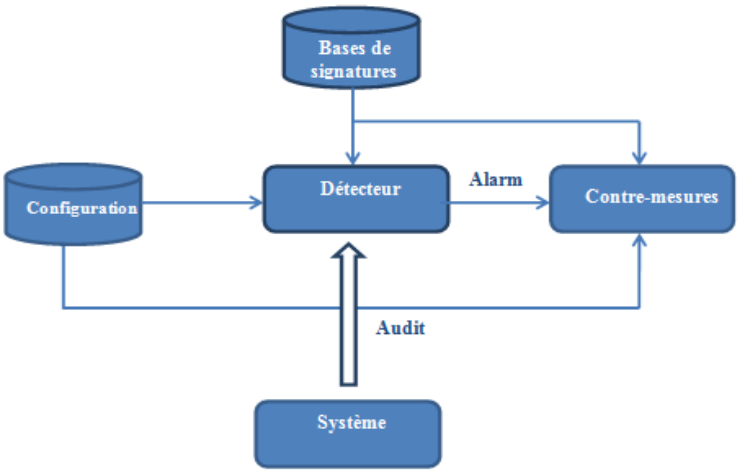

Figure 6 : Modèle simplifié d'un système de détection d'intrusions.

Le détecteur analyse trois types d'informations : les informations de long terme relatives aux techniques utilisées dans la détection (Base de données des signatures), les informations de configuration qui déterminent l'état courant du système, et les informations d'audit qui décrivent les évènements survenant dans le système. **[27]**

Philip dans **[28]** définit trois critères pour évaluer l'efficacité des systèmes de détection d'intrusion :

- **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 du 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 dans **[27]** a rajouté également les deux critères suivants :

- **La tolérance aux fautes (Faulttolerance) :** 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 de 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 de temps nécessaire pour la propagation et la réaction à cet évènement.

3. Architecture d'un IDS :

Nous décrivons dans cette section les trois composants qui constituent classiquement un système de détection d'intrusions **[25].** La Figure 7 illustre les interactions entre ces trois composants.

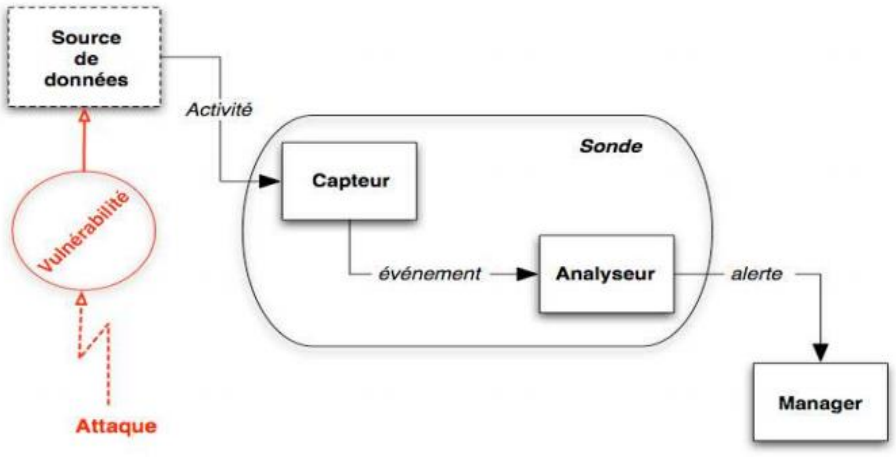

Figure 7: Architecture classique d'un IDS

3.1. Capteur :

Le capteur observe l'activité du système par le biais d'une source de données et fournit à l'analyseur une séquence d'événements qui renseignent de l'évolution de l'état

du système. Le capteur peut se contenter de transmettre directement ces données brutes, mais en général un prétraitement est effectué.

On distingue classiquement trois types de capteurs en fonction des sources de données utilisées pour observer l'activité du système : les capteurs système, les capteurs réseau et les capteurs applicatifs.

3.2. Analyseur :

L'objectif de l'analyseur est de déterminer si le flux d'événements fourni par le capteur contient des éléments caractéristiques d'une activité malveillante.

3.3. Manager :

Le manager collecte les alertes produites par le capteur, les met en forme et les présente à l'opérateur. Éventuellement, le manager est chargé de la réaction à adopter qui peut être :

- ✓ Confinement de l'attaque, qui a pour but de limiter les effets de l'attaque ;
- ✓ Eradication de l'attaque, qui tente d'arrêter l'attaque ;
- ✓ Recouvrement, qui est l'étape de restauration du système dans un état sain ;
- ✓ Diagnostic, qui est la phase d'identification du problème.

Du fait du manque de fiabilité des systèmes de détection d'intrusions actuels, les réactions sont rarement automatisées, car elles peuvent se traduire par un déni de service en cas de faux positif.

4. Classification des systèmes de détection d'intrusion :

Depuis les travaux d'Anderson et all **[19],** le domaine de la détection d'intrusions est en plein développement. On trouve à l'heure actuelle plusieurs systèmes de détection d'intrusions opérationnels, que ce soit des produits commerciaux ou du domaine public. Il est donc très utile d'utiliser des critères pour classifier ces systèmes de détection d'intrusions qui seront présentés dans cette section.

Les différents systèmes de détection d'intrusions disponibles peuvent être classés selon plusieurs critères (ils seront détaillés ultérieurement)(Figure 8) :**[27]**

a. la méthode de détection (la méthode d'analyse).

b. Le comportement de la détection.
c. Emplacement de la source d'audit.
d. La fréquence d'utilisation.

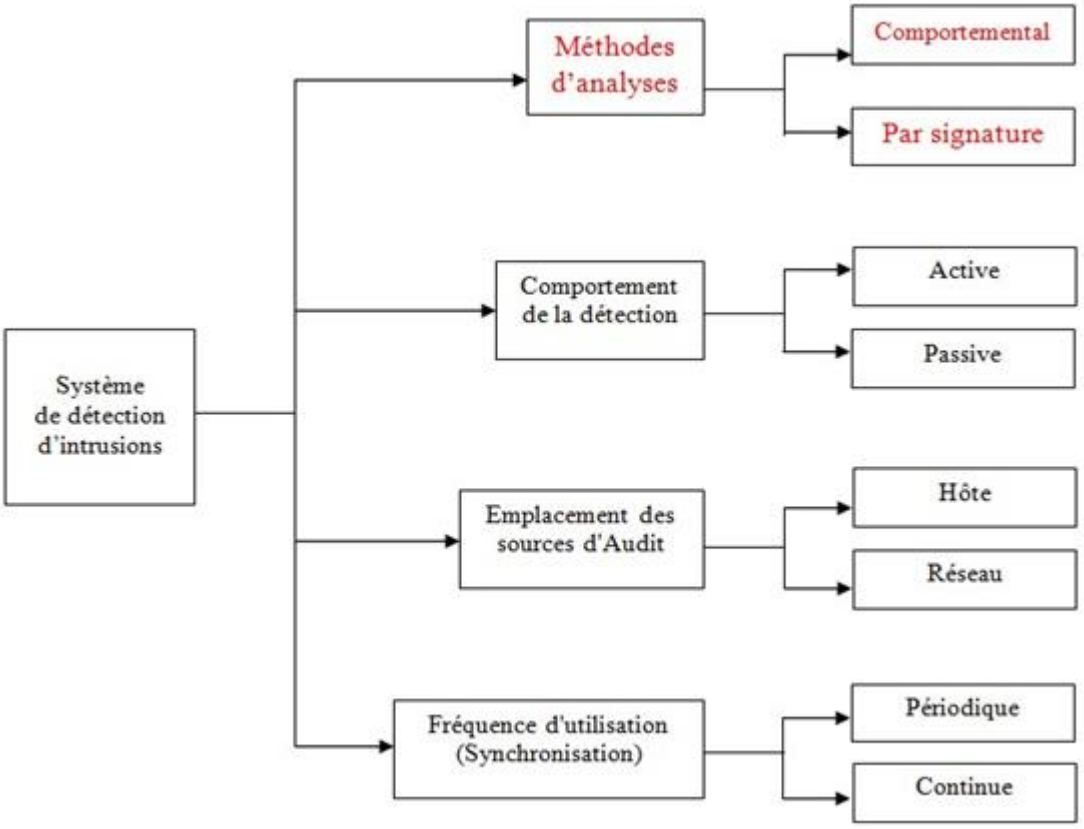

Figure 8 : Taxonomie des systèmes de détection d'intrusion.

4.1. Méthodes de détection :

Il existe deux grandes catégories de méthodes de détection explorées dans la littérature : L'approche comportementale modélise le comportement normal des utilisateurs, du système informatique et de l'activité réseau. Ensuite, toute déviation par rapport à la normal constitue un événement suspect. La deuxième approche, appelée détection par connaissances, recherche explicitement les signatures des attaques connues dans les fichiers de sécurité et le trafic réseau.

4.1.1. L'approche par scénarios :

L'approche par signature est sans doute la méthode la plus utilisée par les IDS aujourd'hui. Elle consiste, comme son nom l'indique, à vérifier si les données reçue

correspondent à une des signatures, correspondant à un comportement anormal, du système de détection d'intrusions.

Ces signatures peuvent être formulées de plusieurs manières, afin de couvrir le panel le plus important de cas détectables.

Si la source des données est le réseau, les signatures pourront par exemple être constituées des numéros de ports source et/ou destination, ou des adresses IP émettrice et/ou destinataire des paquets. Dans le cadre du monitoring des accès à un fichier, la signature pourra être le nom de l'utilisateur qui accède au fichier.

Le système de détection va donc remonter des alertes si les conditions dénies dans la signature sont remplies par les données reçues. La détection par signature est une approche qui autorise par défaut tout le trac, et lance des alertes sur le trac juge « anormal ».

4.1.2. L'approche comportementale :

L'approche de détection comportementale est fondamentalement différente de celle par signatures, bien que l'idée générale soit la même.

En lieu et place des signatures des attaques qui sont la base de la détection par signatures, l'approche comportementale va posséder des modelés de comportements « légitimes ».

Les données reçues par les IDS sont analysées de la même façon que lors de la précédente approche, à la différence que le système va chercher à détecter si les actions sont autorisées. Si elles ne le sont pas, elles lèveront des alertes.

L'apprentissage du modelé des comportements légaux peut se faire de différentes manières.

La première est de partir d'une base vierge, et de rajouter à ce modèle des cas légitimes qui généreraient des évènements inutiles. La seconde technique est d'obtenir des statistiques des actions réalisées régulièrement et qui peuvent donc être considérées comme autorisées.

Contrairement à la première approche, nous pouvons observer que par défaut toutes les actions sont jugées comme anormales, et que des actions des administrateurs sécurité sont nécessaires pour que les activités « normales » ne soient plus considérées comme illégitimes.

4.2. Le comportement de la détection (réponse) :

Le comportement de la détection décrit la réponse du système de détection d'intrusions à une attaque, elle est qualifiée d'active, si le détecteur réagit activement par des actions correctives, ou proactives (changer les règles de filtrage de Firewall des connexions TCP, ou encore attaquer l'attaquant, etc.). Si le système de détection d'intrusions génère simplement des alarmes (afficher un message sur l'écran, générer un son spécifique, envoi d'un email, archivage dans un fichier ou dans une base de donnée, etc.), la réponse est qualifiée de passive. **[25]**

4.2.1. Les réponses actives :

Les réponses actives des systèmes de détection d'intrusions sont des actions automatisées prises quand certains types d'intrusions sont détectés.

Il y a trois catégories de réponses actives : **[25]**

- **Rassembler des informations additionnelles**

Il est très important de rassembler des informations additionnelles sur une attaque afin de l'identifier avec précision. Dans le cas des systèmes de détection d'intrusions, cela se traduira par l'exigence d'analyse des informations additionnelles, faire de corrélations, ou bien communiquer avec d'autres types de systèmes de détection d'intrusions installés sur le réseau.

- **Changer l'environnement**

Une autre réponse active doit stopper une attaque en progression ensuite bloquer l'accès de l'attaquant. Typiquement les systèmes de détection d'intrusions n'ont pas les capacités de bloquer l'accès d'une personne spécifique, mais ils peuvent uniquement rompre des connexions ou bloquer certains paquets spécifiques en s'appuyant sur les mécanismes des protocoles Internet.

- **Agir contre l'intrus**

La première option dans la réponse active est d'agir contre l'intrus.

En effet, la forme la plus agressive de cette réponse implique le lancement des contres attaques ou d'essayer d'obtenir activement les informations sur l'hôte ou l'emplacement de l'attaquant.

4.2.2. Les réponses passives :

Les réponses passives des systèmes de détection d'intrusions fournissent l'information nécessaire aux administrateurs réseau et aux responsables de la sécurité pour les aider à prendre des mesures basées sur cette information. Beaucoup de systèmes de détection d'intrusions se fondent seulement sur des réponses passives dont les principales sont : **[25]**

- **L'alarme :** Les alarmes sont produites par les systèmes de détection d'intrusions pour informer les administrateurs réseau quand des attaques sont détectées. La forme la plus connue est d'afficher un message d'alerte concernant des informations détaillées de l'intrusion détectée sur la console du responsable de la sécurité réseau.

- **SNMP Trap :** Certains systèmes de détection d'intrusions sont conçus pour produire des alertes et envoyer les rapports aux systèmes de gestion du réseau (network management system). Ils utilisent le protocole SNMP (Sample Network Management Protocol), qui est un protocole dédié à la gestion du réseau.

- **L'archivage :** L'archivage permet aux analystes de faire des analyses approfondies, et de faire des corrélations avec l'historique dont ils disposent concernant les événements qui se sont produits auparavant.

4.3. L'emplacement des sources d'audit :

Les IDS s'appuient généralement sur des sources de données différentes. Certains IDS dit IDS réseau au NIDS (Network IDS) examinent les paquets transportés par le réseau, et se déploient généralement dans des endroits précis du réseau par exemple, dans la zone démilitarisée ou juste avant ou / et après un Firewall. D'autre dit IDS Système ou HIDS (Host IDS) examinent les données des journaux de sécurité établis par le système d'exploitation et les applications qui tournent sur ces machines. Ces IDS sont déployés directement sur les hôtes du réseau.

4.3.1. NIDS (Network Based Intrusion Detection System) :

Le rôle essentiel d'un NIDS est l'analyse et l'interprétation des paquets circulant sur un réseau. L'implantation d'un NIDS sur le réseau se fait de la façon suivante (Figure 9) : des captures sont placées aux endroits stratégiques des réseaux et qui captent les paquets circulant. Ces paquets sont envoyés à un analyseur sécurisé, qui les analyses et les traite éventuellement en utilisant la base de signatures et il génère l'alerte. Cet analyseur est généralement situe sur un réseau isolé, qui relie uniquement les capteurs et l'analyseur. **[29]**

Les capteurs placés sur le réseau sont placé en mode furtif, de façon à être invisibles aux autres machines. Pour cela, leur carte réseau est configurée en mode ou aucune adresse IP n'est configurée. **[29]**

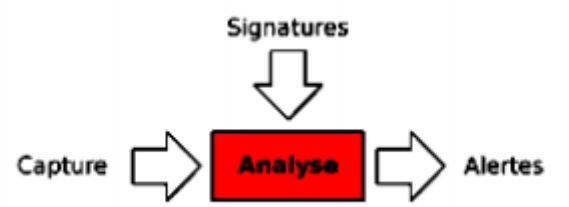

Figure 9 : Composant d'un NIDS.

Les Avantages des NIDS sont les suivants :

- ✓ Ils peuvent être complètement cachés sur un réseau, donc un attaquant ne saura pas qu'il est contrôlé.
- ✓ Un système NIDS unique peut être employé pour contrôler le trafic d'un grand nombre de système cible potentiels.
- ✓ Il peut capturer le contenu de tous les paquets envoyés à un système cible.
- ✓ Une seule tache à effectuer : regarder le trafic et le traiter.
- ✓ Déployer un NIDS à un faible impact sur un réseau existant.
- ✓ Les NIDS sont des systèmes à temps réel.

Les inconvénients des NIDS :

- ✓ Le taux élevé de faux positifs qu'ils génèrent.
- ✓ Ils ne peuvent donner d'alarmes que si le trafic correspond aux règles ou aux signatures pré configurées.
- ✓ Ils peuvent manquer le trafic intéressant si le trafic est important sur la bande passante ou si des routes altérées sont utilisées.
- ✓ Il ne peut pas déterminer si une attaque a réussi.
- ✓ Il ne peut pas examiner le trafic chiffré.
- ✓ Il faut des configurations spéciales sur les réseaux commutés pour que le NIDS puisse voir tout le trafic.

4.3.2. HIDS (Host Based Intrusion Detection System) :

Les HIDS analysent le fonctionnement, ou l'état des machines sur lesquelles ils sont installés, afin de détecter les attaques. Pour cela, ils auront pour mission d'analyser les journaux systèmes, de contrôler l'accès aux appels systèmes, de vérifier l'intégrité des systèmes de fichiers, etc. Ils sont en général, placés sur des machines sensibles, susceptibles de subir des attaques et possédantes des données sensibles pour l'entreprise. Les serveurs web et applicatifs peuvent notamment être protégés par un HIDS. **[12]**

Les avantages des HIDS sont les suivants :

- ✓ Il est possible de constater immédiatement l'impact d'une attaque de mieux réagir.
- ✓ Il est possible d'observer les activités se déroulant sur l'hôte avec précision et d'optimiser le système en fonction des activités observées.
- ✓ Ils permettent de détectera plus facilement les attaques de type Cheval de Troie, alors que ce type d'attaque est difficilement détectable par un NIDS.
- ✓ Les HIDS peuvent souvent fonctionner dans des environnements avec un trafic réseau chiffré.
- ✓ Ils permettent également de détecter les attaques impossibles à détecter avec un NIDS, car elles font partie du trafic crypté.

- ✓ Ils génèrent peu de faux positifs, permettant d'avoir des alertes pertinentes.

Les inconvénients des HIDS :

- ✓ Ils peuvent être identifiés et mis hors service par un attaquant.
- ✓ Ils ne peuvent donner l'alerte que si les entrées des journaux d'événement ou les appels au système correspondent à des signature ou règles pré configurées.
- ✓ Sensibles aux attaques de type Déni de Service.
- ✓ Ils sont assez gourmands en CPU et peuvent parfois altérer les performances de la machine hôte.

4.3.3. IDS hybrides (NIDS+HIDS) :

Les systèmes de détection d'intrusion hybrides, rassemblent les caractéristiques de plusieurs systèmes de détection d'intrusion différents. En pratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil de surveiller le réseau et l'hôte. Les sondes sont placées dans les points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes les sondes remontent alors les alertes à une machine qui va centraliser, agréger, et lier les informations d'origines multiples. **[12] [3]**

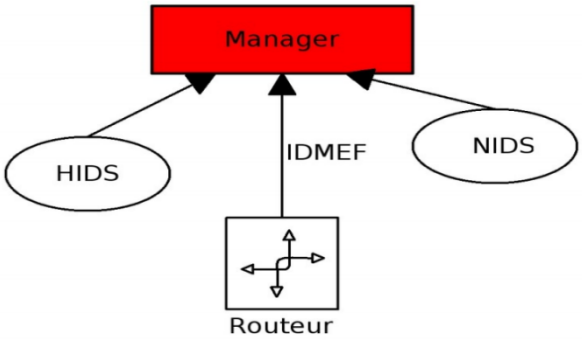

Figure 10: Composant d'un IDS HYBRIDE.

Les avantages des IDS hybrides sont multiples :

- ✓ Moins de faux positifs
- ✓ Meilleure corrélation
- ✓ Possibilité de réaction sur les analyseurs

4.4. La fréquence d'utilisation :

Une autre caractéristique des systèmes de détection d'intrusion, c'est leur fréquence d'utilisation :**[14]**

- **Périodique :**

Certains systèmes de détection d'intrusion, analysent périodiquement les fichiers d'audit à la recherche d'une éventuelle intrusion ou anomalie passée. Cela peut être suffisant dans des contextes peu sensibles, par exemple du fait qu'un HIDS analyse des fichiers traces transmis seulement toutes les heures.

- **Continue :**

La plupart des systèmes de détection d'intrusions récents, effectuent leur analyse des fichiers d'audit ou des paquets réseau de manière continue, afin de proposer une détection en quasi temps réel. Cela est nécessaire dans des contextes sensibles (confidentialité). C'est toutefois couteux en temps de calcul car il faut analyser à la volée tout ce qui se passe sur le système.

Conclusion :

Aucun système d'information n'est sûr à 100% ,pour une machine connectée à un réseau, le problème aujourd'hui n'est plus de savoir si elle va se faire attaquer, mais quand cela va arriver ; une solution possible est alors d'essayer de repousser les risques dans le temps par la mise en œuvre de divers moyens destinés à augmenter le niveau de sécurité.

La plupart des IDS sont fiables, ce qui explique qu'ils sont souvent intégrés dans les solutions de sécurité. Les avantages qu'ils présentent face aux autres outils de sécurités les favorisent, mais d'un autre côté cela n'empêche pas que les meilleurs IDS présentent aussi des lacunes et quelques inconvénients. Nous comprenons donc bien qu'ils sont nécessaires mais ne peuvent pas se passer de l'utilisation d'autres outils de sécurité visant à combler leurs défauts.

Dans le chapitre suivant nous présentons les principes de fonctionnements des machines à vecteurs de support (SVM).

Chapitre 02 :

Les Machines à Vecteurs de Support (SVM)

Chapitre 02 : Les Machines à Vecteurs de Support (SVM)

Introduction :

Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik, les Machines à Vecteurs de Support(SVM) constituent la forme la plus connue. SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente pour la classification. Elle repose sur l'existence d'un classificateur linéaire dans un espace approprié.

Puisque Notre problème est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonctions dites noyau (kernel) qui permettent une séparation optimale des données. Dans la présentation des principes de fonctionnements, nous schématiserons les données par des « points » dans un plan.

1. Apprentissage statistique et SVM :

La notion d'apprentissage étant importante, nous allons commencer par effectuer un rappel. L'apprentissage par induction permet d'arriver à des conclusions par l'examen d'exemples particuliers. Il se divise en apprentissage supervisé et non supervisé. Le cas qui concerne les SVM est l'apprentissage supervisé. Les exemples particuliers sont représentés par un ensemble de couples d'entrée/sortie. Le but est d'apprendre une fonction qui correspond aux exemples vus et qui prédit les sorties pour les entrées qui n'ont pas encore été vues. Les entrées peuvent être des descriptions d'objets et les sorties la classe des objets donnés en entrée. **[32]**

1.1. Objectif de l'apprentissage statistique :

Effectuer une classification consiste à déterminer une règle de décision capable, à partir d'observations externes, d'assigner un objet à une classe parmi plusieurs. Le cas le plus simple consiste à discriminer deux classes. D'une manière plus formelle, la classification bi-classe revient à estimer une fonction f : x $\rightarrow\{+1,-1\}$ à partir d'un ensemble d'apprentissage constitué de couples (xi, yi), qu'on suppose indépendants et identiquement distribués suivant une distribution de probabilité P(x, y) inconnue, tels que

$$(xi, yi) \in X \times Y \text{ où } i = 1,\ldots, Nx \text{ et } Y = \{+1,-1\},$$

De sorte à ce que f classifie correctement des exemples inconnus (xt, yt). Par exemple, on peut assigner xt à la classe (+1) si f (xt) ≥0, et à la classe (-1) sinon. Les exemples inconnus sont supposés suivre la même distribution de probabilité P(x, y) que ceux de l'ensemble d'apprentissage. La meilleure fonction f est celle obtenue en minimisant le risque :

$$R[f] = \int L\,[f(x), y]\, d\,P(x, y). \qquad (4,1)$$

Où L désigne une fonction de coût, comme par exemple :

$$L\,[f(x),y] = (f(x)-y)^2$$

Malheureusement, le risque (4.1) ne peut être directement minimisé dans la mesure où la distribution de probabilité sous-jacente P(x, y)est inconnue. Aussi, on va chercher une fonction de décision proche de celle optimale àpartir de dont on dispose, c'est-àdire l'ensemble d'apprentissage et la classe de fonctions F est à laquelle la solution f appartient. Pour ce faire, on approxime le minimum du risque théorique par le minimum du risque empirique qui s'écrit :

$$R_{emp}[f] = \frac{1}{Nx} \sum_{i=1}^{Nx} L\,[f(x_i),y_i]\,. \qquad (4,2)$$

Il est possible de donner des conditions au classifieur pour qu'asymptotiquement (si Nx→ ∞), le risque empirique (4.2) converge vers le risque (4.1). Cependant, si on dispose de peu d'exemples pour faire l'apprentissage (exemple Nx petit), on s'expose au risque de sur-apprentissage. Pour éviter le sur-apprentissage, on peut restreindre la complexité de la classe F à laquelle appartient f. Intuitivement, une fonction de décision simple (la classe la plus simple se constituant des fonctions linéaires) capable de discriminer correctement les données est préférable à une fonction complexe. Pour cela, on introduit un terme de régularisation pour limiter la complexité des fonctions de F.

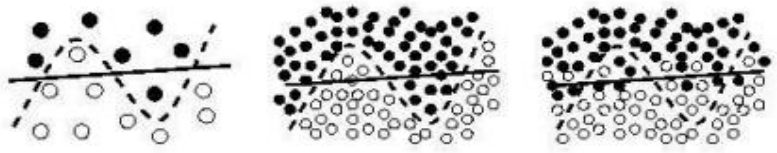

Figure 11: Exemple du problème de sur apprentissage.

Etant donné un petit ensemble d'apprentissage (schéma de gauche), deux frontières de discrimination (représentées par les lignes continue et discontinue) sont possibles. La ligne discontinue est plus complexe mais minimise davantage le risque empirique. Seul un ensemble d'exemples plus grand permet de déterminer la meilleure des deux frontières de décision. S'il s'agit de la ligne discontinue, alors la ligne continue n'est pas suffisamment discriminante (schéma du milieu, s'il s'agit de la ligne continue, alors la ligne discontinue ne convient pas et caractérise un sur apprentissage (schéma de droite).**[31]**

1.2. Théorie de Vapnik-Chervonenkis :

Une manière de contrôler la complexité d'une classe de fonctions est donnée par la théorie de Vapnik-Chervonenkis (VC) et le principe de minimisation du risque structurel. Ici, le concept de complexité de la fonction de décision f s'exprime par la dimension de VC (notée h) de la classe de fonctions F à laquelle appartient f. Grossièrement, la dimension de VC mesure combien d'échantillons de l'ensemble d'apprentissage peuvent être séparés par toutes les classifications possibles issues des fonctions de la classe.**[31]**

Considérons une famille imbriquée de classes de fonctions :

$$F1 \subset F2 \subset \ldots \subset Fk ;$$

Avec une dimension de VC non-décroissante, et f1…fk les fonctions minimisant le risque empirique dans chacune de ces classes.

La minimisation du risque structurel consiste à choisir la classe Fi (et la fonction fi) de sorte à ce qu'une borne supérieure de l'erreur de généralisation puisse être minimisée (grâce, par exemple, au théorème suivant). **[31]**

Théorème 1 : Soient h la dimension de VC de la classe de fonctions F, Remp[f] le risque empirique défini par (4.2) avec la fonction perte 0/1 (i.e. L [f(xi),yi] =H(-yf(x)) Où H désigne la fonction de Heaviside). Pour tout δ>0 et f∈ F, l'inégalité bornant le risque

$$R[f] = R_{emp}[f] + \sqrt{\frac{h(\ln\frac{2Nx}{h}+1) - \ln(\frac{\delta}{4})}{Nx}} \qquad (4,3)$$

Est vraie avec une probabilité de moins (1- δ) pour Nx>h **[32]**.

Cette borne n'est qu'un exemple et des formulations du même type ont été démontrées pour d'autres fonctions perte et d'autre mesures de complexité. Le but recherché ici est de minimiser l'erreur de généralisation R[f] en obtenant un faible risque empirique Remp [f] tout en gardant la plus petite classe de fonctions possible.

L'inégalité (4.3) fait apparaître deux cas extrêmes :

- ✓ une très petite classe de fonctions (par exemple F1) fait décroître rapidement le terme de complexité (celui en racine carrée), mais le risque empirique demeure grand,
- ✓ une très grande classe de fonctions (par exemple Fk) implique un risque empirique petit, mais le terme de complexité explose.

La meilleure classe de fonctions est généralement intermédiaire entre la plus petite et la plus grande, puisque nous cherchons d'avoir une fonction qui explique au mieux les données tout en préservant un faible risque empirique.**[31]**

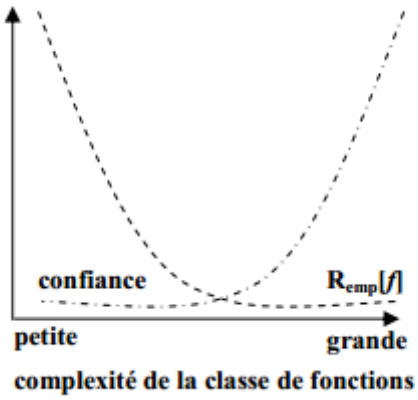

Figure 12 : Illustration de l'inégalité (4.3) **[31]**.

La courbe croissante, appelée confiance, correspond à la borne supérieure du terme de complexité. Les comportements du terme de complexité et de l'erreur empirique sont clairement opposés. On recherche donc le meilleur compromis entre complexité et erreur empirique.**[31]**

1.3. Marge et dimension de Vapnik-Chervonenkis :

Supposons pour l'instant que les échantillons de l'ensemble d'apprentissage sont séparables par un hyperplan, on choisit des fonctions de décision de la forme :

$$f(x)=\langle w, x\rangle + b. \qquad (4.4)$$

La marge est la distance minimale entre les échantillons de l'ensemble d'apprentissage et la frontière de décision.

Il a été montré que pour la classe des hyperplans, la dimension de VC peut être bornée en fonction de la marge. La marge peut à son tour être mesurée grâce au vecteur poids w : puisque nous supposons que les échantillons sont séparables, on peut redéfinir w et b de sorte à ce que les échantillons x les plus proches de l'hyperplan satisfassent $|\langle w, x\rangle + b| = 1$.

Considérons maintenant deux échantillons x1 et x2 de classes différentes telles qu'on ait $\langle w, x1\rangle + b = +1$ et $\langle w, x2\rangle + b = -1$. La marge γ correspond alors à la distance entre x1 et x2 mesurée perpendiculairement à l'hyperplan

$$\gamma = \langle w / \|w\|, x_1 - x_2 \rangle = 2/\|w\| ;$$

Les résultats liant la dimension de VC de la classe des hyperplans de séparation à la marge et à la longueur du vecteur poids w sont respectivement donnés par les inégalités suivantes:

Où R est le rayon de la plus petite boule englobant les données. Ainsi, en bornant la marge de la classe de fonction, on peut contrôler sa dimension de VC.**[31]**

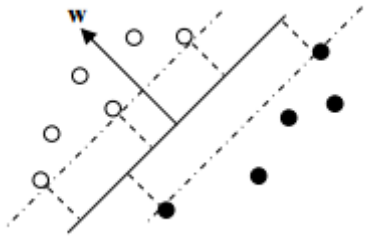

Figure 13 : Classifieur linéaire et marge **[31].**

Un classifieur linéaire est défini par un vecteur normal à l'hyperplan w et un biais b:la frontière de décision est $\{x| \langle w, x\rangle + b = 0\}$(ligne continue).Chacun des deux sous-espaces séparés par l'hyperplan correspond à une classe, i.e. $f(x) = signe(\langle w, x\rangle + b)$. La marge du classifieur linéaire est la distance minimale entre les échantillons de l'ensemble d'apprentissage et la frontière de décision. Sur le schéma, il s'agit de la distance entre la ligne continue et les lignes discontinues.**[31]**

2. Principe de fonctionnement général du SVM :

2.1. Notions de base : Hyperplan, marge et support vecteur :

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan. **[32]**

Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points

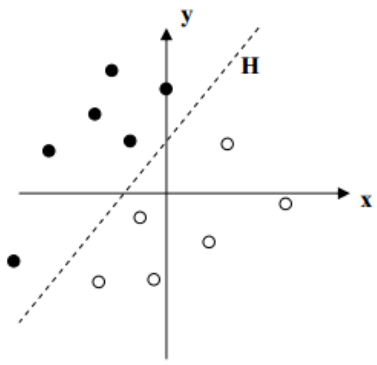

Figure 14 : Exemple d'un hyperplan séparateur **[32].**

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

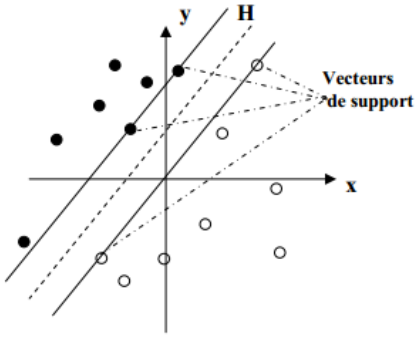

Figure 15 : Exemple de vecteurs de support **[32]**.

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « plus sûr ».**[33]**

En effet, supposons qu'un exemple n'ait pas été 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 **[33].**

On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge **[33].**

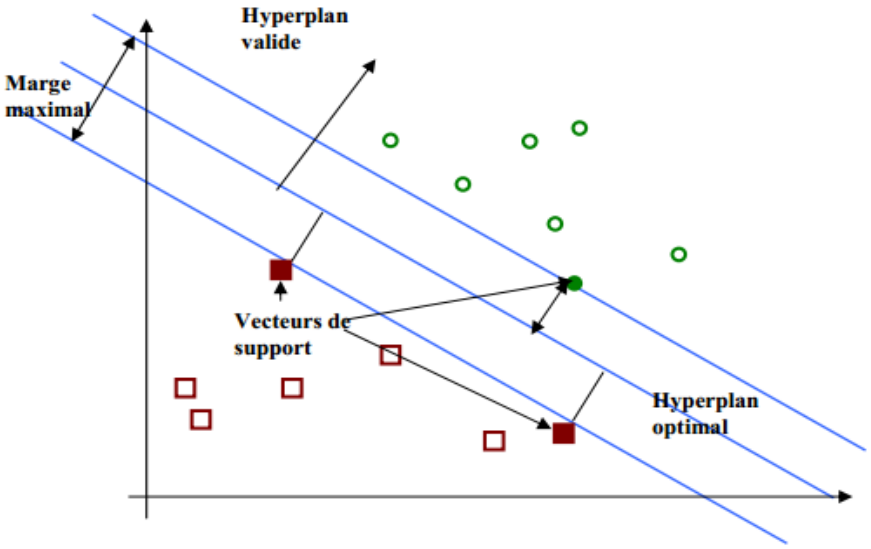

Figure 16 : Exemple de marge maximal (hyperplan valide).

- **Hyperplan optimal :** est un Hyperplan qui classé correctement les données (lorsque c'est possible) et qui se trouve le plus loin possible de tous les exemples, on peut dire aussi que cet hyperplan maximise la marge.
- **Vecteurs de support :** ce sont Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan.
- **La marge :** est la distance entre l'hyperplan et les exemples. La marge est calculée à partir du produit scalaire entre les vecteurs situés la frontière de chaque classe et le vecteur unitaire normal de l'hyperplan séparateur **[34].**

2.2. Le but de maximiser la marge :

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples. Dans le schéma qui suit, la partie droite nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe

dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé.**[32]**

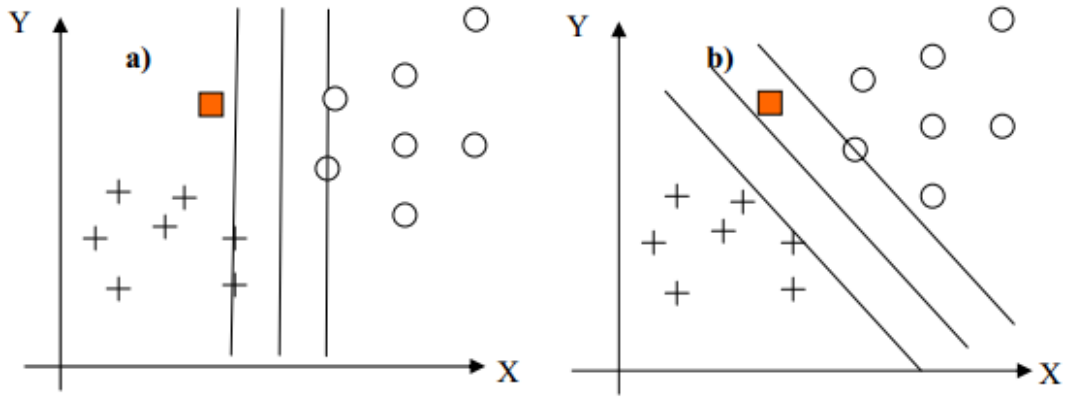

Figure 17 : a)Hyperplan avec faible marge, b) Meilleur hyperplan séparateur **[32].**

En général, la classification d'un nouvel exemple inconnu est donnée par sa position par rapport à l'hyperplan optimal. Dans le schéma suivant, le nouvel élément sera classé dans la catégorie des « +».

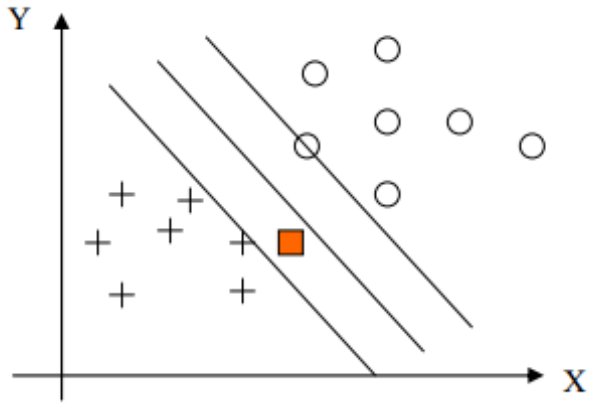

Figure 18 : Exemple de classification d'un nouvel élément.

2.3. Linéarité et non-linéarité :

Parmi les modèles des SVM, on constate les cas linéairement séparable et les cas non linéairement séparable. Les premiers sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables. **[32]**

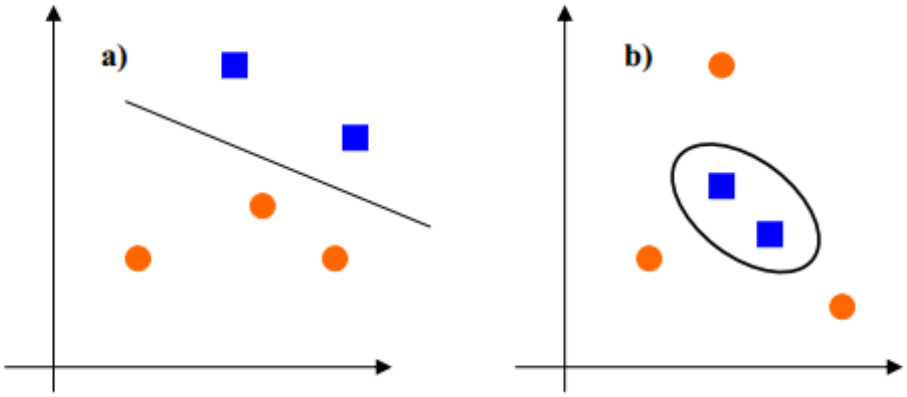

Figure 19 :a)Cas linéairement séparable, b) Cas non linéairement séparable **[32].**

3. Fondement mathématique des SVMs :

Le fondement mathématique des séparateurs à Vaste Marge est expliqué dans plusieurs ouvrages comme **[35],[36], [37].**

3.1. Principe général :

Les SVMs peuvent être utilisés pour résoudre des problèmes de discrimination binaire, c'est-à-dire, décidé à quelle classe appartient un échantillon. La résolution de ce problème passe la construction d'une fonction f qui, à un vecteur d'entrée x ∈ X fait correspondre une sortie f(x) : Il est alors décidé que x est de classe +1 si f(x)>0 et de classe -1 si f(x)<0. C'est un classifieur linéaire. La frontière de décision f(x)=0 est un hyperplan séparateur.

Soit H un hyperplan, w son vecteur normal et b, son décalage par rapport à l'origine (voir figure 20). L'hyperplan H est alors donné par :

$$f(x) = w^T x + b = 0$$

Le but de l'algorithme d'apprentissage d'un SVM est de trouver les paramètres w

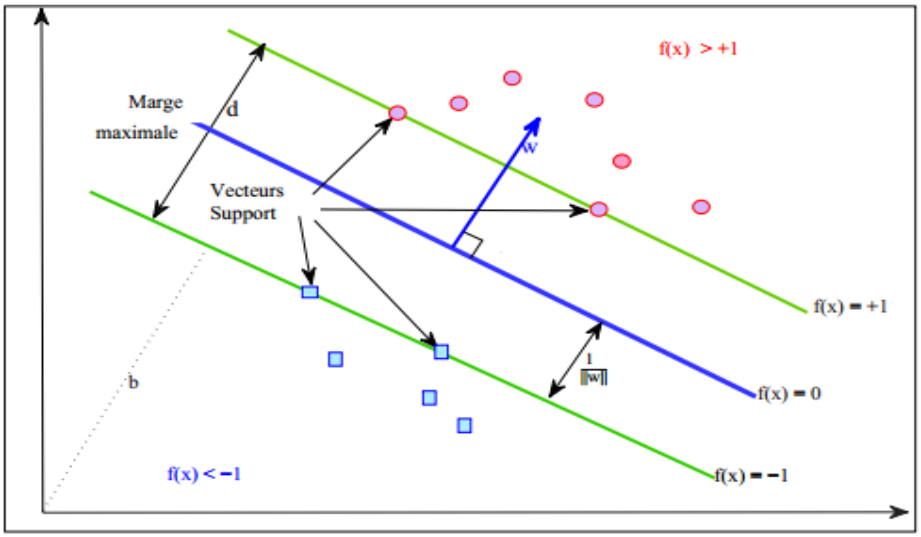

Figure 20 : Illustration de la marge et du vecteur support.

Et b du meilleur hyperplan par le biais d'un ensemble d'apprentissage.

$$\mathcal{X} \times \mathcal{Y} = \{(x_1, y_1), .., (x_i, y_i)\} \in \mathbb{R}^N \times \{+1, -1\}$$

Où les y_i sont les labels respectifs des x_i, N la taille de l'ensemble d'apprentissage.

3.2. Cas linéairement séparable :

On se place ici dans le cas où le problème est linéairement séparable. Même dans ce cas simple, le choix de l'hyperplan séparateur n'est pas évident car il existe en effet une infinité d'hyperplans séparateurs. Pour résoudre ce problème, il a été montré qu'il existe un unique hyperplan optimal, défini comme étant celui qui maximise la marge entre les échantillons et l'hyperplan séparateur. **[36]**

3.2.1 Formulation du problème d'optimisation primal :

La marge est la distance entre deux points, les plus proches de l'hyperplan mais appartenant à des classes différentes (voir figure 20). Ces derniers sont appelés :

Vecteurs Support (VS) : Il s'agit alors de trouver le couple (w, b) qui maximise la marge afin de déterminer l'équation de l'hyperplan optimal H. ce couple est défini par :

$$\arg\max_{w,b} \min_i \|x - x_i\| : \ x \in \mathcal{X}, \ (w^T x + b) = 0, \ i = 1, \ldots, N$$

Soit x+ et x- deux points de classes différentes situés respectivement sur les frontière positive et négative délimitant la marge maximale. Pour simplifier le problème

d'optimisation, on considère x+ et x- sont situés sur les hyperplans canoniques tels que f(x+)=+1 et f(x-)= -1, c'est-à-dire $w^T x^+ + b = +1 \quad et \quad w^T x^- + b = -1.$

On sait que la distance d'un point quelconque x à H est définie par :

$$d_{x,H} = \frac{|w^T x + b|}{\|w\|}$$

La distance entre chacun des deux points x+ et x- et H est alors 1/||w||. Dans ce cas, la marge est :

$$d = \frac{w^T}{\|w\|}(x^+ - x^-) = \frac{2}{\|w\|}$$

Nous déduisons à partir de là que maximiser la marge revient à minimiser ||w|| sous contraintes que $y_i(w^T x_i + b) \geq 1.$ Cette contrainte signifie que le SVM tient compte non seulement de la position des exemples par rapport à l'hyperplan (signe(f(x)), mais aussi de leurs distances par rapport à cet hyperplan. Le problème d'optimisation est alors posé comme suit :

$$\begin{cases} \min\limits_{w,b} & \frac{1}{2}\|w\|^2 \ , \\ \text{S.c} & y_i(w.x_i + b) \geq 1, \quad i = 1, \ldots, N \ . \end{cases} \qquad (5,7)$$

Notons qu'il est plus aisé de minimiser $\|w\|^2$ plutôt que $\|w\|$.

3.3. Cas non linéairement séparable :

Les données d'apprentissage peuvent être bruitées et non séparables, même dans l'espace F. Il faut alors trouver un bon compromis entre risque empirique et complexité (2 ,2). En 1995, Corinna Cortes et Vladimir Vapnik proposèrent une technique dite de marge souple en introduisant des variables ressortε_i (slack variables) pour relâcher sensiblement les contraintes sur la marge. Ces dernières deviennent alors :

$$y_i f(x_i) \geq 1 - \xi_i$$

3.3.1. Formulation du problème primal :

Les variables de relaxation autorisent quelques erreurs de classification lors de l'apprentissage. Le problème d'optimisation (5, 7) devient alors :

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} |1 - y_i f(x_i)|_+ \qquad (5, 12)$$

Avec ou $|\cdot|_+ = \max(\cdot, 0)$ Le problème (5, 12) est souvent exprimé en fonction des variable d'écart ξ_i comme suit :

$$\begin{cases} \min\limits_{w,b,\xi} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \ , \\ \text{S.c} & y_i(w^T x_i + b) \geq 1 - \xi_i \ , \\ & \xi_i \geq 0, \quad i = 1, \ldots, N \ . \end{cases} \qquad (5, 13)$$

La constante C est un paramètre déterminant la tolérance de SVM aux exemples mal classées. Elle permet de controler le compromis entre nombre d'erreurs de classement et la largeur de la marge (figure 20). Plus C est grand, plus on pénalise les mauvaises classification et le compléxité de le classe de fonction de décision devient grand. Le choix automatique de ce paramètre de régularisation est un probleme statistique majeur.

A travers le problème (5, 12), on cherche à maximiser le marge et à minimiser la fonction de pertes (fonction de cout) définie par :

$$\ell(y_i,\ f(x_i)) = C\sum_{i=1}^{N} |1 - y_i f(x_i)|_+ = C\sum_{i=1}^{N} \xi_i \qquad (5, 14)$$

Cette fonction couramment appelée « hinge loss » est une fonction convexe. Elle garantit une solution unique au problème **[36]**

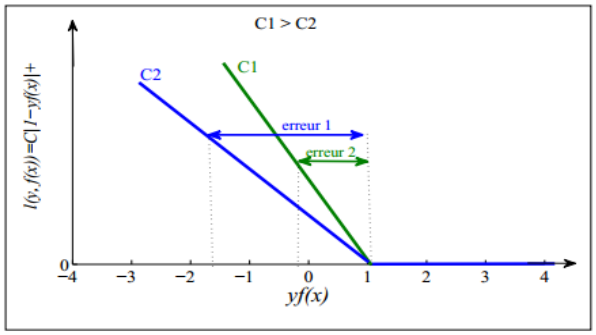

Figure 21 : Représentation du compromis entre la largeur de la marge et le cout d'erreur.

3.3.2. Architecture générale d'une machine à vecteurs supports :

Une machine à vecteur support, recherche à l'aide d'une méthode d'optimisation, dans un ensemble d'exemples d'entrainement, des exemples, appelés vecteurs support, qui caractérisent la fonction de séparation. La machine calcule également des multiplicateurs associés à ces vecteurs.

Les vecteurs supports et leurs multiplicateurs sont utilisés pour calculer la fonction de décision pour un nouvel exemple. Le schéma de la figure 22 résume l'architecture générale.

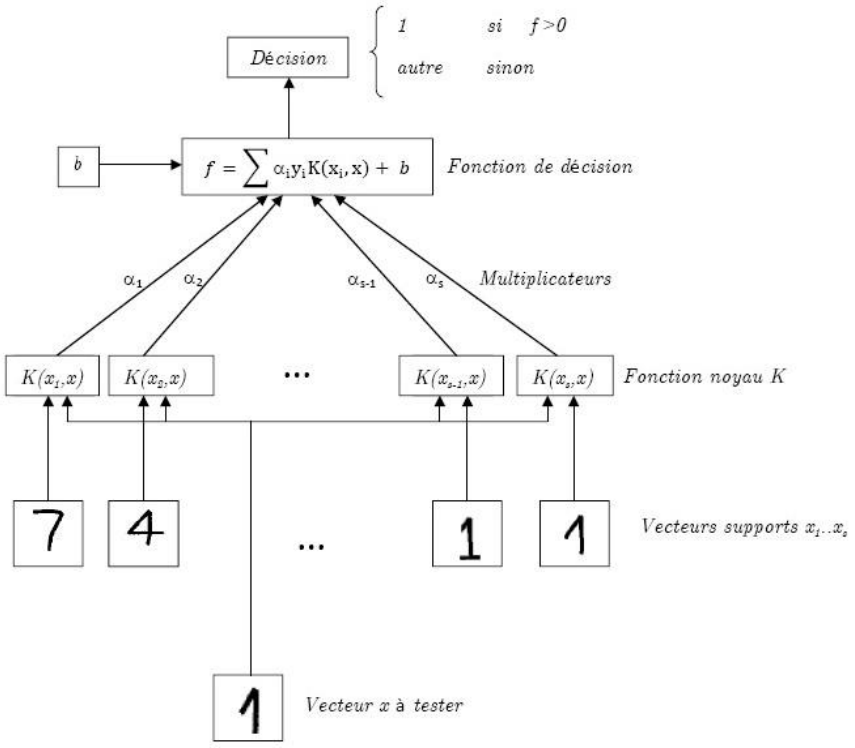

Figure 22 : Architecture d'une machine à vecteur support

La fonction noyau *K* est utilisée pour calculer la distance entre le vecteur à tester *x* et chaque vecteur support dans l'espace de caractéristique. Les résultats sont ensuite linéairement combinés en utilisant les multiplicateurs de Lagrange a_i et ajoutés au biais *b*. Le résultat final *f* permet de décider à propos du nouveau vecteur : si *f(x)* est positive, il s'agit du chiffre "1", sinon, il s'agit d'un autre chiffre.

4. Outils et Applications des méthodes SVM :

De nombreux outils sont disponibles pour l'implémentation ou l'utilisation d'un algorithme SVM :

a) **SVMTorch :** assez facile à utiliser : **[63].**

b) **rainbow :** c'est un logiciel pour la classification de textes ; il comprend notamment un module MVS. Il est disponible à l'url **[5].**

c) **libsvm :** bibliothèque de fonctions **[4]** ; nécessite de Programmer ;

d) **SVMlight :** bibliothèque de fonctions **[65]** ; nécessite de programmer.

5. Conclusion :

Dans ce chapitre, nous avons présenté de manière simple et complète la méthode d'apprentissage introduite par Vladimir Vapnik, les « Support Vector Machines ».Nous avons donné une vision générale et le fondement mathématique des SVM.

Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des classes de données. Nous avons exposé les cas linéairement séparable et les cas non linéairement séparables qui nécessitent l'utilisation de fonction noyaux (Kernel) pour changer d'espace. Cette méthode est applicable pour des taches de classification a deux classes (comme dans notre cas la classe normal et anormal).Nous verrons dans le chapitre suivant une autre approche de classification : les réseaux bayésien.

Chapitre3 :

Réseaux Bayésiens

Chapitre3 : Réseaux Bayésiens

Introduction :

Un réseau bayésien est en informatique et en statistique un modèle graphique probabiliste représentant des variables aléatoires sous la forme d'un graphe orienté acyclique. Intuitivement, ils sont à la fois : des modèles de représentation des connaissances. Des «machines à calculer » les probabilités conditionnelles. Pour un domaine donné, on décrit les relations causales entre variables d'intérêt par un graphe. Dans ce graphe, les relations de cause à effet entre les variables ne sont pas déterministes, mais probabilisées. Ainsi, l'observation d'une cause ou de plusieurs causes n'entraîne pas systématiquement l'effet ou les effets qui en dépendent, mais modifie seulement la probabilité de les observer. L'intérêt particulier des réseaux bayésiens est de tenir compte simultanément de connaissances a priori d'experts (dans le graphe) et de l'expérience contenue dans les données. Dans ce chapitre en va présenter une généralisation sur les réseaux bayésien et on met l'accent beaucoup plus sur les réseaux bayésien naïf.

1. Définition :

Les RB sont des modèles qui permettent de représenter des situations de raisonnement probabilistes basé sur le théorème de Bayés exprimer par la formule (1,1), et ce à partir de connaissances incertaines.

$$P(A/B) = \frac{P(B)P(B/A)}{P(A)} \qquad (1,1)$$

Ainsi, les RB associent une partie qualitative que sont les graphes et une partie quantitative représentant les probabilités conditionnelles associées à chaque nœud du graphe relativement au parent **[40].** La partie qualitative exprime des indépendances conditionnelles entre variable et des liens de causalités et ce grâce à un graphe orienté acyclique dont les nœuds correspondent à des variables aléatoires. La partie quantitative est constituée de tables de probabilités dans le cas discret ou distribution gaussiennes dans le cas continu. Un réseau bayésien B = {G, P} est donc défini par un graphe dirigé, un espace probabiliste et un ensemble de variable aléatoires. Le graphe est sans circuit G = (X,E) ou X est l'ensemble des nœuds (ou sommets) et E, l'ensemble des arcs. L'espace probabiliste est tel que (Ω, P) ou Ω

est l'univers des probabilités et P l'ensemble de variables aléatoires X={X1,….,Xn} associées aux nœuds de graphe et tel que :

$$P(X_1,...,X_n)=\prod_{i=1}^{n} P(X_i / Pa(X_i)) \qquad (1,2)$$

Dans cette expression, Pa(Xi) est l'ensemble des parents du nœud Xi dans G.

2. Modèles graphiques :

L'aspect fondamental des modèles graphiques est la modularité dans le sens où un système complexe est construit par la combinaison de parties plus simples. Deux composantes sont généralement introduites lorsqu'un problème donné est modélisé par un modèle :

- **composante graphique** : qui permet de représenter le problème traité sous forme d'un graphe qui met en évidence les variables du domaine et les relations d'influences qu'elles entretiennent entre elles.
- **composante numérique** : dépendante du problème modélisé, elle consiste à assurer la cohérence de la structure du graphe construit. En effet, cette composante peut concerner une quantification des paramètres avec des probabilités, des possibilités ou autres, comme elle peut concerner d'autres contraintes telles que les préférences (tables de préférences conditionnelles), contraintes temporelles, mesures de désirabilité (utilité espérée), etc.

2.1. Différents modèles graphiques des réseaux bayésiens :

Il existe plusieurs variantes des RB telles que **[41]** : les RB multi agents, les RB de niveaux deux, les RB orientés objets, les diagrammes d'influence, Les RB dynamiques **[42]** (temporels), les RB multi entités, les filtres bayésiens **[43]** : qui sont des RB dynamiques particuliers et RB adaptés à la classification tels que ; les RB naïf, les RB naïf augmenté, etc.

En classification, particulièrement, les RB sont largement. Dans ce cas, le nœud parent est considéré comme une variable non observée à quelle classe appartient chaque objet alors que les nœuds enfants des variables observées correspondant aux différents attributs caractérisant cet objet.

Plusieurs modèles sont conçus dans ce but. Parmi ces réseaux, on peut citer le RB naïf qui est le plus simple, le réseau bayésien augmenté par n'importe qu'elle structure ou par une structure arborescente et autre. Les RB ont une structure simple et unique qui comprend deux niveaux. Le premier niveau contient un seul nœud parent et le second plusieurs enfants avec

la forte hypothèse naïve d'indépendance conditionnelle des enfants (X) conditionnellement au parent. Ils sont largement utilisés pour résoudre des problèmes de classification **[44].** La figure 23 rappelle le principe de fonctionnement et de classification par RBN.

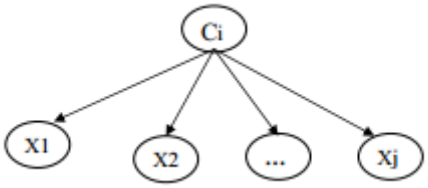

Figure 23 : Exemple de structure d'un réseau naïf augmenté.

Ou :

Ci est le nœud classe et i est la ieme classe. Xj sont les nœuds des attributs et j est j ieme attribut ou paramétré.

Ce classifieur est connu pour ses performances malgré sa simplicité et il dépasse des techniques beaucoup plus sophistiquée même lorsque l'hypothèse d'indépendance est violée. L'hypothèse d'indépendance des variables permet d'écrire la probabilité a posteriori de chaque classe comme l'indique l'équation suivante :

$$P(C_i / x) = P(C_i)\prod_{j=1}^{p} P(x_j / C_i) \qquad (1,3)$$

Par conséquent, en présence d'un ensemble d'apprentissage, la seul opération à faire est de calculer les probabilités conditionnelles, en appliquant la règle de décision « d » de Bayes comme suit :

$$\begin{aligned} d(X) &= argmax_{classe}\ P(classe / X) \\ &= argmax_{classe}\ P(X_j / classe)\ P(classe) \\ &= argmax_{classe}\ P(C_i)\prod_{j=1}^{p} P(X_j / C_i) \end{aligned} \qquad (1,4)$$

2.2. Structure d'un modèle graphique :

La structure d'un modèle graphique dépend de la nature du problème modélisé, particulièrement en ce qui concerne le type des relations qu'entretiennent les variables du graphe entre elles. Étant donné un ensemble fini de variables aléatoires X = {X1,X2, … , Xn}, E un sous-ensemble du produit cartésien X × X. *D***X***i* est un domaine associé à la variable X*i* (valeurs possibles que cette variable peut prendre). G = (X, E) est un graphe

représentant la structure du modèle graphique utilisé pour modéliser un problème donné (G peut être un réseau bayésien, un diagramme d'influence, un graphe de préférences conditionnelles, etc.). X constitue l'ensemble des nœuds du graphe et E est l'ensemble des liens entre les nœuds dans X. Selon la nature des liens entre les nœuds, il existe deux familles de graphes :

- **Graphes orientés :** un graphe orienté est constitué d'arcs, c'est-à-dire que l'ensemble des liens entre les nœuds du graphe sont orientés. Dans ce type de graphes, il existe la notion de parents et fils : s'il existe un lien orienté entre Xi et Xj (Xi → Xj), Xi est appelé *parent* de Xj et Xj est le fils de Xi. Un graphe orienté ne possédant pas de cycles (acyclique) est appelé DAG (Directed Acyclic Graph). En particulier, un réseau bayésien est un graphe orienté sans cycles.
- **Graphes non orientés :** ces graphes sont également appelés graphes de markov. Dans un graphe non orienté, les liens entre les nœuds ne sont pas orientés, c'est-à-dire que les notions de fils et parents n'existent pas. L'avantage de ce type de graphes est la possibilité de former des cycles, ce qui n'est pas le cas pour les graphes orientés.

2.3. Utilisation des modèles graphiques :

Les modèles graphiques peuvent être utilisés dans de nombreux domaines d'applications. Le choix d'un modèle graphique dépend de la nature des informations du problème à traiter : recherche d'un plus court chemin, satisfiabilité d'une contrainte, construction de règles ou de modèles, classification d'un événement dans une classe prédéfinie, recherche de la meilleure décision. En particulier,

a. **La représentation et extraction des connaissances :** en présence des quantités importantes d'informations variées et complexes, les modèles graphiques apportent certainement beaucoup de facilités pour la représentation et l'extraction de ces informations. Dans le domaine médical par exemple, ce type de représentation est efficace car les informations médicales sont souvent disponibles en grande quantité. Les connaissances contenues dans ces données peuvent être extraites et modélisées par un réseau Bayésien par exemple basé les variables aléatoires du problème (Age, maladie, poids, etc.).
b. **La représentation des préférences :** la notion de préférences est introduite lorsqu'un agent se retrouve face à un choix parmi un ensemble d'alternatives, c'est-à-dire qu'il doit réagir sur la base des informations disponibles plus ou moins connues. L'utilisation d'un

modèle graphique pour représenter certains types de préférences telles que les préférences conditionnelles **[45]** est possible et c'est avérée intéressante vu le nombre d'alternatives qui peut croître exponentiellement avec la taille du problème. Si nous considérons les graphes de préférences conditionnelles (CP-nets) **[46]**, le graphe (orienté) est construit comme suit : considérer les variables sur lesquelles portent les préférences d'un agent comme les nœuds du graphe et les relations de dépendances préférentielles entre les variables sont représentées par des arcs orientés. Chaque nœud est associé d'une table de préférences conditionnelles. L'objectif de cette représentation est de déterminer la meilleure alternative en parcourant le graphe du haut (nœud racine) en bas (nœuds feuilles).

c. **L'aide à la décision :** pour les problèmes de décisions, les informations sont souvent incertaines. C'est pourquoi, les modèles graphiques apportent beaucoup de facilités au niveau des calculs de l'incertitude mais également au niveau de l'expression du raisonnement humain (préférences, croyances). A titre d'exemple, certains modèles graphiques tels que les diagrammes d'influences **[48]** ne permettent pas uniquement d'évaluer les états du système par des probabilités, mais ils donnent également la possibilité d'associer une valeur de désirabilité (utilité) à chaque décision possible dans le but d'évaluer la qualité de toutes les décisions. D'autres mesures peuvent également être considérées telles que l'aspect temporel dans les processus de décisions markoviens **[49]**, etc.

   En détection d'intrusions par exemple, il est question de décider dans quelle catégorie faut-il classer un événement ayant lieu dans le réseau. C'est pourquoi les réseaux bayésiens sont des modèles graphiques qui répondent parfaitement à ce type de problèmes essentiellement par leur structure graphique et leur aspect algorithmique comme l'apprentissage automatique, l'inférence et la classification.

d. **L'apprentissage automatique :** pour construire un modèle graphique permettant de modéliser un problème particulier, il est nécessaire de formaliser et représenter des connaissances qu'on a sur ce problème. Ces connaissances sont fournies soit par un expert, soit automatiquement à l'aide d'outils d'apprentissage automatique. L'apprentissage automatique consiste à construire à partir des données disponibles qui sont souvent en grandes quantités des représentations compactes, et compréhensibles, sous forme de règles, modèles graphiques, etc.

3. Inférences dans les réseaux bayésiens :

L'inférence est le calcul de la probabilité de n'importe qu'elle variable d'un modèle probabiliste à partir de l'observation d'une ou plusieurs autres variables. Il consiste à propager une ou plusieurs informations au sein de ce réseau, pour en déduire comment sont modifiées les croyances concernant les autres nœuds. La structure du graphe joue un rôle important dans la complexité de ces calculs ainsi dans le choix de la méthode d'inférence. On peut distinguer deux catégories d'algorithmes d'inférence**[47]** : l'inférence exacte et l'inférence approché. Plusieurs méthodes ou algorithmes conçues spécialement pour les problèmes d'inférence exacte pour les réseaux bayésiens.

Parmi les méthodes d'inférence exactes citons la méthode "Messages locaux", "Ensemble de coupes", "Arbre de jonction", "Inversion d'arcs» et "élimination de variables".

Lorsque la dimension des réseaux bayésiens augmente, le temps de calcul est de plus en plus important. Lorsque les tables de probabilités conditionnelles sont issues de données (par apprentissage), ces tables ne sont pas exactes. Dans ce cas il est intéressant de ne pas perdre du temps en faisant de l'inférence exacte sur des probabilités non exactes, donc le recours aux méthodes d'inférence approchée.

Parmi les méthodes d'inférence approchées existent les méthodes basées sur la "simulation stochastique par Chaîne de Monte-Carlo", "Loopybelief propagation**[50]**"**,** les méthodes variationnelles, les méthodes de recherche de masse, d'autres sont basées sur la simplification du réseau.

4. Apprentissage des réseaux bayésien :

La construction d'un réseau bayésien consiste à trouver une structure ou un graphe et estimer les paramètres (probabilité conditionnelles). Cependant devant une très grande base de données, personne ne peut extraire seule la structure adaptée à une telle quantité de données. C'est ici qu'intervient l'apprentissage artificiel.

Deux sortes d'apprentissages peuvent être envisagées :

- L'apprentissage des paramètres, où nous supposons que la structure du réseau a été fixée, et où il faudra estimer les probabilités conditionnelles de chaque nœud du réseau.

- L'apprentissage de structure, où le but est de trouver le meilleur graphe représentant la tâche à résoudre.

4.1. Apprentissage des paramètres :

L'apprentissage de paramètres consiste à supposer que la structure du réseau et fixe et donc à déterminer les probabilités conditionnelles de chaque variable qui se trouve dans le réseau. Les données peuvent être complètes ou incomplètes, discrètes ou continues. Pour chaque cas, l'algorithme d'apprentissage des paramètres diffère. Dans le cas où toutes les variable sont observés et discrètes, la méthode la plus simple et la plus utilisée pour estimer les paramètres est l'estimation statique de la probabilité d'un évènement par la fréquence d'apparition de l'évènement dans la base de donnée. Cette méthode est appelée maximum de vraisemblance **[44],** données par l'expression suivante :

$$P(X_i = x_k / \Pi_i = \pi_{ij}) = \hat{\theta}_{ijk}^{MV} = \frac{N_{ijk}}{\sum_{k=1}^{ri} N_{ijk}} \qquad (1,6)$$

N ijk = Le nombre d'occurrences simultanées dans la base de Xi =xk etΠi = πij

Avec k ∈ 1…ri et j ∈ 1….qi

L'inconvénient de cette méthode est que, on peut avoir une probabilité nulle à cause de non apparition d'un évènement dans la base de données, ce qui est faux en réalité. Pour remédier à ce problème on fait recours à d'autres approches dites méthodes bayésiennes. Ces méthodes sont connues sous le nom de maximum à posteriori (MAP) et espérance à posteriori (EIA) **[51].**Supposons que les paramètres θi admettent une densité de probabilité exponentielle de Dirichlet alors on peut écrire l'expression ci-dessous :

$$P(\theta_i / \alpha_1, \ldots \alpha_n) = \frac{\Gamma(\sum_{i=1}^{ri} \alpha_i)}{\prod_{i=1}^{ri} \Gamma \alpha_i} \times \prod_{i=1}^{r} \theta^{\alpha_i - 1} \qquad (1,7)$$

Avec : Γ : la fonction gamma d'Euler.

Γ(x+1) = x Γ (x) et Γ (1)=1 dans R.

Le maximum a posteriori est donné par :

$$P(X_i = x_k / \Pi_i = \pi_{ij}) = \hat{\theta}_{ijk}^{MAP} = \frac{N_{ijk} + \alpha_{ijk} - 1}{\sum_{k=1}^{ri} N_{ijk} + \alpha_{ijk} - 1} \qquad (1,8)$$

L'espérance a posteriori est donné par :

$$P(X_i = x_k / \Pi_i = \pi_{ij}) = \hat{\theta}_{ijk}^{MAP} = \frac{N_{ijk} + \alpha_{ijk} - 1}{\sum_{k=1}^{ri} N_{ijk} + \alpha_{ijk} - 1} \quad (1,9)$$

Dans le cas de données incomplets, Il existe plusieurs algorithmes pour estimer les données manquantes à partir des données connues, au lieu de les ignorer. Le plus utilisé est l'algorithme EM(Expectation maximisation ou Espérance maximisation) **[52]**.

4.2. Apprentissage de structure :

L'apprentissage de structure ayant pour but de trouver le meilleur réseau permettant de représenter les données le mieux possible, Cependant, la recherche dépend de nombre de variables, d'arcs et de valeurs. Le nombre de structure généré à partir de n nœuds est très grand, il est donné par la relation suivante :

Soit : r(n) le nombre de graphe possible, et n le nombre de nœuds existants.

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} C_n^i \, 2^{i(n-i)} r(n-i) = n^{2^{O(n)}} \quad (1,10)$$

Ainsi pour n=4 on a r(4)=543 et pour n=7 on a r(7)=1,4.10^9

Beaucoup de travaux se sont intéressés aux problèmes de l'apprentissage de structure **[53].**Là aussi comme pour l'apprentissage de paramètres, on a deux cas, selon que les données sont totalement ou partiellement observables. Pour le premier cas deux familles d'approches ont été proposées,c'est-à-dire le cas au toutes les mesures sont complètes (critère : Apprentissage de structure : données complètes).

La première famille (algorithmes IC, PCetc.)**[61]**consiste à déterminer dans un premier temps un graphe non orienté en tenant compte des différentes indépendances conditionnelles qui existent entre les variables de ce graphe, puis à orienter ce graphe pour obtenir un réseau bayésien. Ces algorithmes sont peu efficaces dans le cas de problèmes de grande taille puisque la détermination de ces indépendances est exponentielle en fonction du nombre des variables.

La deuxième approche consiste à parcourir tous les graphes possibles, associer un score à chaque graphe, puis choisir le graphe ayant le score le plus élevé. Toutefois, cette méthode n'est pas simple, principalement à cause de la taille super-exponentielle de l'espace de recherche en fonction du nombre de variables

Des idées ont été proposées pour résoudre ce problème. Une première consiste à remplacer l'espace de recherche (espace des réseaux bayésiens) par un espace plus petit, l'espace des arbres (MWST ou Maximal Weight Spanning Tree). Une deuxième idée consiste à ordonner les nœuds pour limiter la recherche des parents possibles pour chaque nœud (algorithme K2)**[60].** Une troisième consiste à faire une recherche gloutonne dans l'espace des réseaux bayésiens (algorithme GS)**[62]**.

5. Classification et réseaux bayésiens :

La classification est une tâche basique en analyse de données et en apprentissage. Elle consiste à attribuer une classe à un ensemble d'attributs caractérisant un objet. Cependant, construire un classifieur à partir d'un ensemble de données pré-classées (étiquetées) est un problème central en apprentissage. Plusieurs méthodes ont été proposées, tels que les arbres de décision, les réseaux de Neurones, les règles d'association, etc.

Un classifieur Bayésien est un réseau Bayésien utilisé pour la classification, qui est un cas particulier d'inférence. Dans un réseau Bayésien utilisé pour la classification, on cherche à inférer la valeur la plus plausible d'une seule variable non observée, appelé Classe, les autres variables, sont observables et constituent généralement les attributs des objets de la classe.

5.1. Classifieurs bayésien naïf :

Un classifieur Bayésien naïf **[54]** représente la forme la plus simple des réseaux bayésiens. Il se compose d'un graphe avec un seul parent et plusieurs nœuds feuilles, avec une forte hypothèse d'indépendance entre les feuilles dans le contexte de leur parent (Figure 24).

Dans le réseau bayésien naïf de la Figure 24, la variable C a deux instances c1 et c2 alors que les trois autres variables A1, A2 et A3 peuvent prendre les valeurs Vrai ou Faux. La composante quantitative de ce réseau est constituée de quatre distributions de probabilités locales : la distribution a priori de la variable C et les trois distributions conditionnelles de A1, A2 et A3 dans le contexte de la variable C.

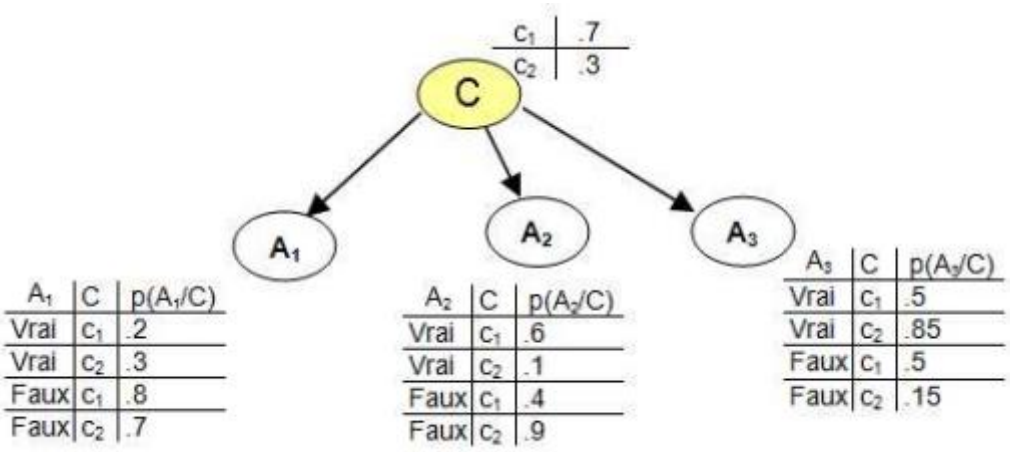

Figure 24 : Exemple de réseau Bayésien naïf

La classification est assurée dans les réseaux Bayésiens naïfs en considérant le nœud parent (racine) comme une variable non observée qui représente la classe d'un objet et les nœuds feuilles comme étant des variables observées correspondant aux différents attributs spécifiant cet objet. Par conséquent, en présence de données d'apprentissage, la seule tâche à faire est d'estimer les probabilités conditionnelles puisque la structure du réseau est unique et connue a priori. Une fois le réseau Bayésien paramétré, il est possible de classifier tout nouvel objet, sachant les valeurs de ses attributs. La classification Bayésienne consiste à déterminer l'instance de la variable C la plus pour une instance du vecteur d'attributs observés a1a2..an. L'expression classe plus probable exprime l'instance ck de la variable classe C qui a la plus grande probabilité a posteriori sachant l'observation a1a2..an. Formellement,

$$classe = argmax_{c_k \in D_C}(P(c_k / a_1 a_2 .. a_n)) \qquad (1,11)$$

où ck représente une instance de la variable C et P(ck/a1a2.. an) représente la probabilité a posteriori de ck sachant l'observation a1a2.. an (une instance du vecteur d'attributs A1A2.. An). La règle de Bayes permet de calculer cette probabilité a posteriori comme suit

$$P(c_i|A) = \frac{P(A|c_i) * P(c_i)}{P(A)} \qquad (1.12)$$

Sous l'hypothèse que les attributs sont indépendants dans le contexte du nœud parent C, la probabilité P(ci|A) peut être développée comme suit :

$$P(c_i|A) = \frac{P(a_1|c_i) * P(a_2|c_i) * .. * P(a_n|c_i) * P(c_i)}{P(A)} \quad (1.13)$$

Notons qu'il n'est pas nécessaire de calculer explicitement le dénominateur P(A) car il est déterminé par la condition de normalisation des distributions de probabilités. Par conséquent, il suffit de calculer pour chaque ci le numérateur de l'Équation 1.13 pour classifier toute nouvelle instance a1a2..an.

5.2. Apprentissage des classifieurs Bayésiens naïfs :

En général, dans un classifieur Bayésien naïf (CBN) chaque variable possède un seul parent qui représente la variable de la classe. Cela signifie que la structure est fixe, et la seule tâche de l'apprentissage est d'estimer les paramètres. Les paramètres d'un CBN sont facilement déterminés. Si les observations sont complètes, nous pouvons construire le modèle de maximum de vraisemblance par un simple comptage de fréquence et si les observations contiennent des valeurs manquantes, l'algorithme EM peut être utilisé **[55].**

Cependant, toutes les méthodes d'apprentissage des classificateurs à partir de données ont un problème avec les cas très rares, qui sont peu représentés dans les données. Supposons, par exemple, que certaines valeurs d'un attribut ne se produisentpas avec une valeur donnée de la classe, l'estimation du P(A|C) produit une valeur nulle et rend difficile l'étape d'inférence. Pour éviter les valeurs nulles dans les paramètres, nous pouvons introduire des cas virtuels. Un moyen facile de gérer ceci est de donner à tous les paramètres une petite valeur positive.

Les CBN sont faciles à apprendre et faciles à utiliser, et comme ils sont très flexibles en ce qui concerne les valeurs manquantes, ils sont très répandus. Malgré l'hypothèse forte d'indépendance des attributs, les classifieurs bayésiens naïfs réalisent de très bonnes performances **[57].** En effet, ils sont, de nos jours, très compétitifs et plusieurs études comparatives empiriques ont montré qu'ils dépassent souvent les autres techniques de classification ayant fait jusque-là autorité sur de nombreuses bases de données. Une raison à cela est que, quand on fait la classification, on est intéressé par la classe de la probabilité maximale et non pas par la distribution de probabilités exacte sur les classes **[55].** Naturellement, la performance des réseaux Bayésiens naïfs atteint son optimum lorsque les attributs sont effectivement indépendants **[58].** Le classifieur naïf de Bayes est robuste et peu

sensible aux attributs non pertinents dans le sens que le processus de classification utilise tous les attributs **[59].**

Conclusion :

Nous avons présenté dans ce chapitre une brève description des modèles graphiques et leur utilité dans différents domaines. Nous nous sommes intéressés particulièrement aux réseaux bayésiens qui sont considérés parmi les modèles graphiques les plus utilisés. Nous avons également présenté l'inférence dans les réseaux bayésien, l'apprentissage et la classification de ces réseaux.

Par ailleurs, le choix des modèles graphiques est le fait qu'ils sont des outils très puissants de représentation des connaissances qui permettent de manipuler des données complexes, incomplètes et incertaines. De plus, les modèles graphiques sont capables de modéliser tout système simple ou complexe en le simplifiant en un ensemble de variables pertinentes et une distribution de probabilités jointe qui couvre tous les cas possibles du système.

Enfin, les modèles graphiques peuvent traiter des données incomplètes et surtout incertaines, ce qui est souvent le cas en détection d'intrusions. En effet, les outils de la sécurité informatique ne sont pas totalement fiables, des attaques peuvent passer inaperçues et d'autres sont annoncées à tort. Finalement, les modèles graphiques représentent une panoplie d'outils relativement simples à mettre en œuvre mais surtout très puissants et offrent des mécanismes d'inférence très précis et très efficaces.

Dans le chapitre suivant en va présenter l'architecture et les différents composants de l'approche proposée.

Chapitre4 :

Utilité des RB dans la Détection d'intrusion

Chapitre4 : utilité des RB dans la Détection d'intrusion.

Introduction :

Ce chapitre aborde l'approche proposée pour la détection d'intrusions réseau et la corrélation d'alertes des IDS. Récemment, quelques travaux ont été proposés pour tenter de résoudre ce problème, voir par exemple **[38] [39].**

L'approche proposée est de développer un **IDS** hybride, hiérarchique qui comprend deux niveaux. Le premier niveau contient le classificateur **SVM** les Machines à Vecteur Support, ce classificateur est utilisé pour son taux élevés de correcte classification. La prédiction du premier niveau est sélectionnée et utilisées comme une entrée du second niveau qui contient le réseau bayésien naïf comme classificateur final.

L'étude comparative a montré que notre approche donne un fiable taux de fausse alarme et le taux de détection le plus élevé. De plus notre système est plus efficace que certains classificateurs bien que SVM, Réseau Bayésien Naïf ou il donne un taux d'exactitude égal à 99.62% dans le cadre du projet PLACID.

1. Approche proposée :

La plupart des travaux de détection d'intrusion utilisent les classificateurs du même niveau de façon isolée. Dans notre cas, nous proposons une nouvelle approche qui combine deux différents classificateurs dans chaque niveau, ou les itérations nécessaires pour construire notre modèle représentent le nombre de niveaux.

1.1. La structure de notre modèle :

Dans cette sous-section, nous présentons les différents composants de notre modèle et ses utilités. Comme le montre la figure 25, notre système contient deux niveaux :

- Le premier niveau : Contient le classificateur **SVM (**les Machines à Vecteur Support), Ce classificateur est sélectionnés pour sa hautes performances dans la classification d'une ou de plusieurs classes de connexion, la prédiction est relative aux deux comportements normal et anormal. Cette prédiction est fusionnée avec le résumé de l'ensemble de données Test est utilisée comme une entrées pour le second niveau.
- Le deuxième niveau : Contient aussi un seul classificateur utilisé pour sa haute performance en tant que classificateur final. Il analyse l'ensemble de données test obtenu du premier niveau et prend la décision final, cette décision peut être soit une attaque (anormal) soit un comportement normal.

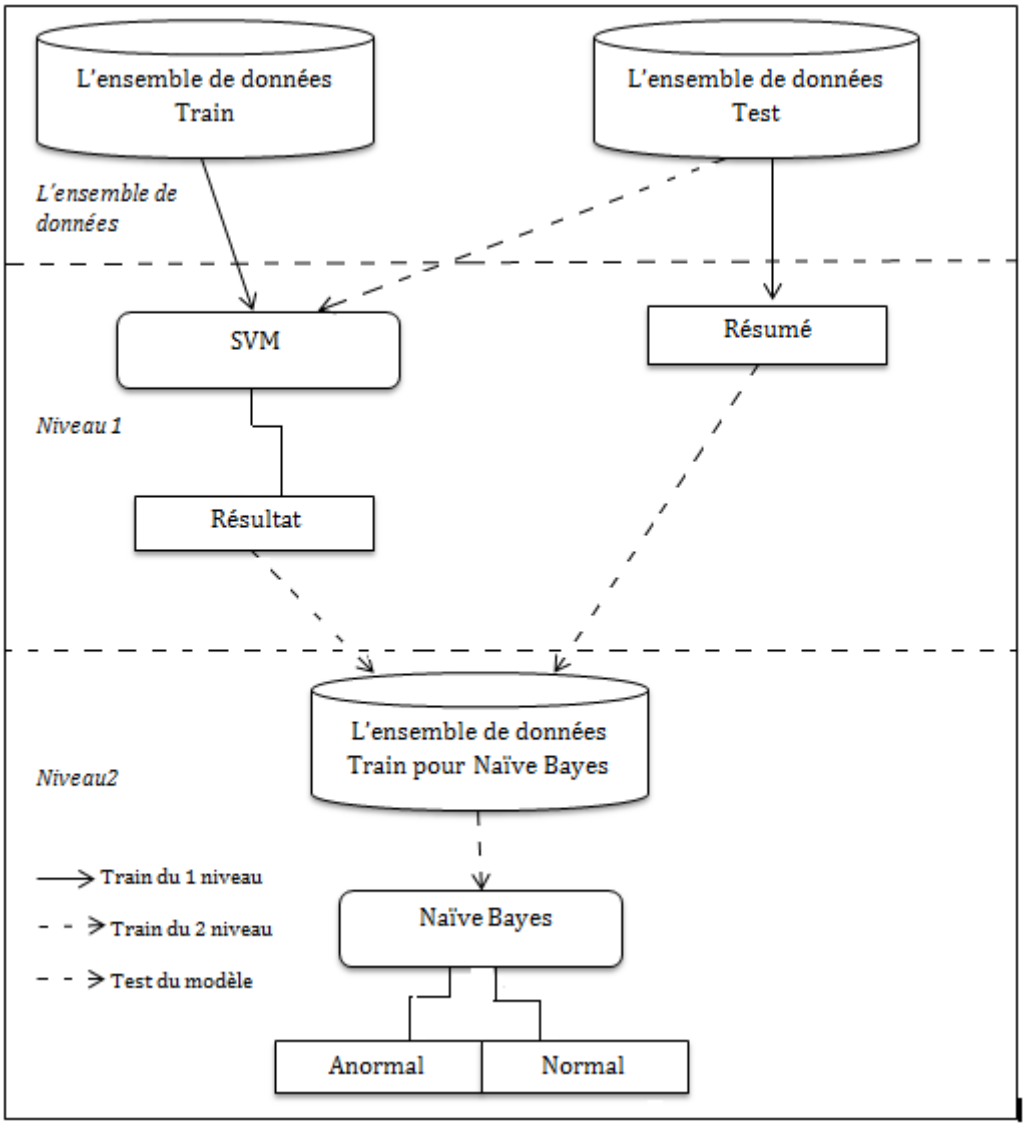

Figure 25 : La structure pratique de notre modèle hiérarchique.

1.2. Le mode de fonctionnement de notre modèle :

Le mode de fonctionnement de notre modèle hiérarchique se compose de deux phases : la phase d'apprentissage et la phase de test.

1.2.1. La phase d'apprentissage :

Dans cette phase, nous effectuons l'apprentissage de notre modèle dans le but de le préparer pour la phase de test. Cette étape est composée de deux étapes :

- Former le premier niveau : nous formons le classificateur du premier niveau avec l'ensemble de données d'apprentissage, ou chaque élément de l'ensemble de données d'apprentissage représente une entrée pour le classificateur.

- Former le second niveau : un nouvel ensemble de données est créé à partir des prédictions de classificateur du premier niveau. Pour générer ce nouvel ensemble de données d'apprentissage, nous associons les prédictions sélectionnées avec l'étiquette correcte comme le montre le **tableau 1.** le nouvel ensemble de données d'apprentissage est utilisé pour former le classificateur sélectionné pour le deuxième niveau.

Prédiction d'Anormal	Prédiction du Normal	**Étiquette**
7615	22	**Anormal**
584	60794	**Normal**

Tableau 1 : Les nouvelles données d'apprentissage pour le deuxième niveau.

1.2.2. La phase de test

Dans cette phase, nous testons la performance de notre approche après l'achèvement de la phase d'apprentissage, ou nous utilisons l'ensemble de données de test. Nous traitons chaque enregistrement de l'ensemble de données de test par le classificateur du premier niveau. Ensuite, nous utilisons les sorties des prédictions sélectionnées de classificateur du premier niveau comme des entrées pour le classificateur du deuxième niveau.

2. Expérimentation :

Cette section est divisée en trois parties. Dans la première partie, nous détaillons l'ensemble de données d'apprentissage et de test. La deuxième partie présente l'architecture de notre approche, dans le but de présenter les entrés et les sorties de classificateur pour le premier et le deuxième niveau. La troisième partie représente une étude comparative entre notre nouvelle approche hiérarchique et d'autre classificateur bien connus.

Nous avons effectué une série d'expérimentation avec les données qui sont collectées dans le cadre du projet PLACID, qui représente l'ensemble de données de détection d'intrusion le plus utilisé dans la dernière décennie. Weka Data Minin et SVM_Light sont utilisés pour la mise en œuvre des classificateurs. Les résultats sont obtenus sur un PC Windows avec Core i5, 2.50GHz et 4 Go de RAM.

2.1. Les données d'apprentissage et de test :

Les données que nous avons utilisées dans l'expérimentation sont collectées dans le cadre du projet ANR PLACID. Dans ce qui suit nous présentons une brève description du projet PLACID, ainsi que les données utilisées dans notre approche.

2.1.1. Présentation du projet PLACID :

Le projet PLACID (Probabilistic graphical models and Logics for Alarm Correlation in Intrusion Detection). Il a pour objectif d'offrir une solution globale pour la gestion des alertes, en fournissant un cadre unifié et formel pour la représentation des alertes et des informations contextuelles. Cette solution globale inclut aussi une approche Bayésienne basée sur la représentation de l'incertitude et la détection d'attaques coordonnées. En outre, le projet prend en compte également l'opérateur de sécurité par la modélisation de ses préférences.

Les objectifs du projet PLACID comprennent la réalisation :

- D'une représentation formelle pour les informations en détection d'intrusions, appelé IDDL (Intrusion Detection Description Logic), basée sur les logiques de description. IDDL fournit aux outils de sécurité un cadre formel pour caractériser leurs observations, partager leurs connaissances avec des outils tiers et de raisonner sur leurs complémentarités.
- D'une approche Bayésienne pour la corrélation d'alertes. Le but est de modéliser l'incertitude associée aux alertes, pour représenter les actions malveillantes, et de modéliser les relations de corrélation entre les alertes. L'utilisation des réseaux Bayésiens a plusieurs avantages tels que l'évaluation du succès des attaques, en réduisant l'ensemble des scénarios d'attaque possibles, l'apprentissage des relations de corrélation, ou de trouver les causes premières des alertes.
- De composants logiciels pour la corrélation d'alertes. Le projet comprend le développement de logiciels d'application de corrélation basés sur une approche Bayésienne et des outils de raisonnement IDDL, intégré dans une solution globale pour le traitement d'alertes.

Ce projet combine l'expertise en intelligence artificielle et la sécurité informatique afin d'une part de développer un modèle formel pour la représentation des alertes hétérogènes et d'autre part d'exploiter la puissance expressive de réseaux Bayésiens pour faire face à l'incertitude et de corréler des alertes.

2.1.2. Présentation et répartition des données utilisée dans notre approche :

On trouve dans le projet PLACID deux types de données : données d'apprentissage et données de test. La totalité de la base d'apprentissage PLACID comporte exactement 98993 lignes, ou chaque ligne du corpus est une connexion caractérisée par 25 attributs, le **tableau 2** représente les 25 attributs :

Les Attributs	
A1	http_req_length
A2	http_uri_length
A3	num_safe_req
A4	num_unsafe_req
A5	uri_ressource_type
A6	num_param
A7	num_arg
A8	response_code
A9	response_time
A10	script_type
A11	writing_script
A12	is_html_response
A13	directory_traversal
A14	shell_cmds
A15	sensitive_files
A16	default_login_passwd
A17	num_nonprintcar
A18	sql_cmds_tricks
A19	css_scripts
A20	num_req_same_host
A21	num_req_same_URI
A22	inter_req_time_interval
A23	req_same_host_diff_URI_rate
A24	http_error_rate
A25	class

Tableau 2 : liste des attributs de la connexion utilisée dans notre approche

Dans le projet PLACID, une connexion est classifiée soit comme connexion normale (avec l'étiquette Normal) soit comme une connexion faisant partie d'une attaque, auquel cas, elle porte le nom de cette attaque. Les connexions appartiennent à l'une des catégories suivantes : Normal, Dos, R2L, U2R et Porbe. Dans notre approche tous les types d'attaques (R2L, bo, value_mis, iv_R2L, il_R2L, flooding, URL_R2L, poor_DoS, DoS, vul_scan, XSS, shell_cmds, sqli,sqli_auth et new) sont regroupés dans une seule catégorie (avec l'étiquette Anormal). Le **tableau 3** montre comment les connexions sont regroupées dans la base PLACID utilisée dans notre l'approche.

	Données d'apprentissage		Données de test	
	Effectif	%	Effectif	%
Normale	55342	55.90%	61378	88.93%
Anormal	43651	44.10%	7637	11.07%
Total	98993	100%	69015	100%

Tableau 3 : Répartitions des données PLACID utilisées par l'approche.

2.2. L'architecture de l'approche proposée :

2.2.1. Le premier niveau :

Dans ce niveau, nous effectuons l'apprentissage de notre modèle dans le but de préparer l'entrée du deuxième niveau. Cette étape est composée de deux étapes :

La première étape : consiste le classificateur **SVM** (les Machines à Vecteur Support), Nous avons choisi **SVMLight** comme outil de classification, Il existe trois phases dans la construction des systèmes de détection d'intrusion SVM :

- La première phase est la phase de *pré-traitement*, qui traite les données TCP / IP de vidage premières sélectionnées au hasard à l'aide automatisé des analyseurs et le convertit en forme lisible par machine.
- La deuxième phase est la phase de *la formation* dans lequel les SVM sont formés sur les différents types d'attaques et de données normales. Les données dispose d'un total de 25 caractéristiques d'entrée et peuvent être classés en deux catégories : normal (1) et d'attaque (-1). Le SVM sera formés à la fois avec le type de données : normal ainsi que des données intrusives.
- La dernière phase est la phase de *test*. Cette phase d'apprentissage consiste à mesurer les performances des données qui sont testés.

Théoriquement, l' SVM's sont les machines d'apprentissage qui intrigue tous les vecteurs de formation dans l'espace caractéristique dimensionnelle et tous les vecteurs sont étiquetés en fonction de leur classe. En SVM les données sont classés en fonction du vecteurs de support qui sont des éléments de l'ensemble d'entrée d'apprentissage dans un hyperplan décrivant la fonctionnalité espace. Le processus de classement des données en 2 classes

consiste à diviser les données en normal et attaque. Le principal objectif des SVM est de séparer la normal (1) et intrusive des données (-1). Ainsi, les SVM sont formés avec des motifs à la fois normaux et intrusifs.

Quatre mesures adaptées de récupération de l'information sont utilisées pour évaluer la performance d'un modèle SVM:

Précision Rappel (recall) taux de faux négatifs taux de faux positifs	= D + A / (A+B+C+D), = A / A + C, = C / C + D, = B / B + A,

Ou : **A :** sont les connexions anormales détectés comme anormales.

B : sont les connexions normales détectées comme anormales.

C : sont les connexions anormales détectées comme normales.

D : sont les connexions normales détectées comme normales.

Un faux négatif se produit quand une action d'intrusion a eu lieu mais le système considère comme un comportement non intrusif. Un faux positif survient lorsque le système classifie une action comme une intrusion alors qu'il s'agit d'une action légitime.

2.2.1.1. Le prétraitement :

Le corpus PLACID est un ensemble de données mixte, c'est-à-dire qu'il contient à la fois des attributs numériques et non-numériques. Or, la présence d'attributs non-numériques constitue un problème pour les benchmark de classification. Par exemple, il est impossible de leur appliquer les formules de la distance ou de la moyenne, qui ne sont applicables que sur des données numériques. Pour remédier à ce problème, il faut convertir les attributs non-numériques en valeurs numériques. Il sera alors possible d'appliquer l'algorithme de classification sur les valeurs converties.

La méthode consiste à remplacer chaque attribut non-numérique par une valeur numérique. La description suivant illustre cette codification sur un exemple :
La ligne suivante représente un enregistrement de la Base d'apprentissage train.txt :

161,40,1,0,.htm,0,0,404,0.000326,none,none,0,0,0,0,0,0,0,0,429,0,0.05,1,0.91,vul_scan

On a Presque tous les attributs avec des valeurs numériques sauf : **htm**, **none**, **none** et **vul_scan**.

Supposons que le code de chaque attribut est **: htm** : 1**, none** : 2,**none** : 2,

vul_scan : représente la classe soit **anormal :-1 ou normal : 1**

Après la réécriture on obtient la ligne suivante :

-1**1**:0 **2**:1 **3**:2 **4**:3 **5**:1**6**:0 **7**:0 **8**:0 **9**:0 **10**:2**11**:2**12**:0 **13**:0 **14**:0 **15**:0 **16**:0 **17**:0 **18**:0 **19**:0 **20**:0 **21**:0 **22**:0 **23**:229 **24**:10 **25**:0.00

(-1) : représente la classe.

Tous ce qui est en gras est le numéro de l'attribut,

Le tableau 4 suivant présente toutes les codifications de tous les attributs de la base:

Attribut 05 : uri_ressource_type		**Attribut 10 : script_type**		**Attribut 11 : writing_script**		**Attribut 25 : class**	
Les Valeurs	Code	Les Valeurs	Code	Les Valeurs	Code	Les Valeurs	Code
.swf	1	None	1	None	1	R2L	-1
.exe	2	Js	2	Cookie_set	2	Bo	-1
Other	3	vb	3	Doc_loc	3	Value_mis	-1
None	4			Cookie_rd	4	Iv_R2L	-1
.htm	5					Il_R2L	-1
Cgi-bin	6					Flooding	-1
.asp	7					URL_R2L	-1
.pl	8					Poor_Dos	-1
.dat	9					Dos	-1
.gif	10					Vul_scan	-1
.ps	11					XSS	-1
.pdf	12					Shell_cmds	-1
.css	13					Sqli	-1
.jpg	14					Sqli_auth	-1
.php	15					Normal	1
.tar	16					new	-1
.txt	17						
.ico	18						
.ppt	19						
.js	20						
.zip	21						
.cgi	22						
.asm	23						
.sh	24						
.mov	25						
.com	26						
.ini	27						

Tableau 4 : codifications des attributs non numérique de la base PLACID.

2.2.1.2. La formation :

En réalité, l'algorithme ***SVMLight*** est formé par un seule couple de fichier : ***svm_learn*** et ***svm_classify***. Les différentes options et paramètres d'entrée pour ces deux programmes (fichiers) sont décrits sur le site web de **Joachims[41]**. On doit leur fournir les données d'entrée pour l'apprentissage et la classification dans des fichiers selon une syntaxe

spéciale expliquée aussi sur le même site. Le SVM résultant d'un processus d'apprentissage, est enregistrée sur un fichier (modèle) et peut ainsi être utilisé ultérieurement pour des tâches de test et classification. La **figure 26** représente le résultat d'apprentissage obtenue après l'exécution de la commande suivante :**[system prompt]>svm_learn.exetrain.txt model.txt**

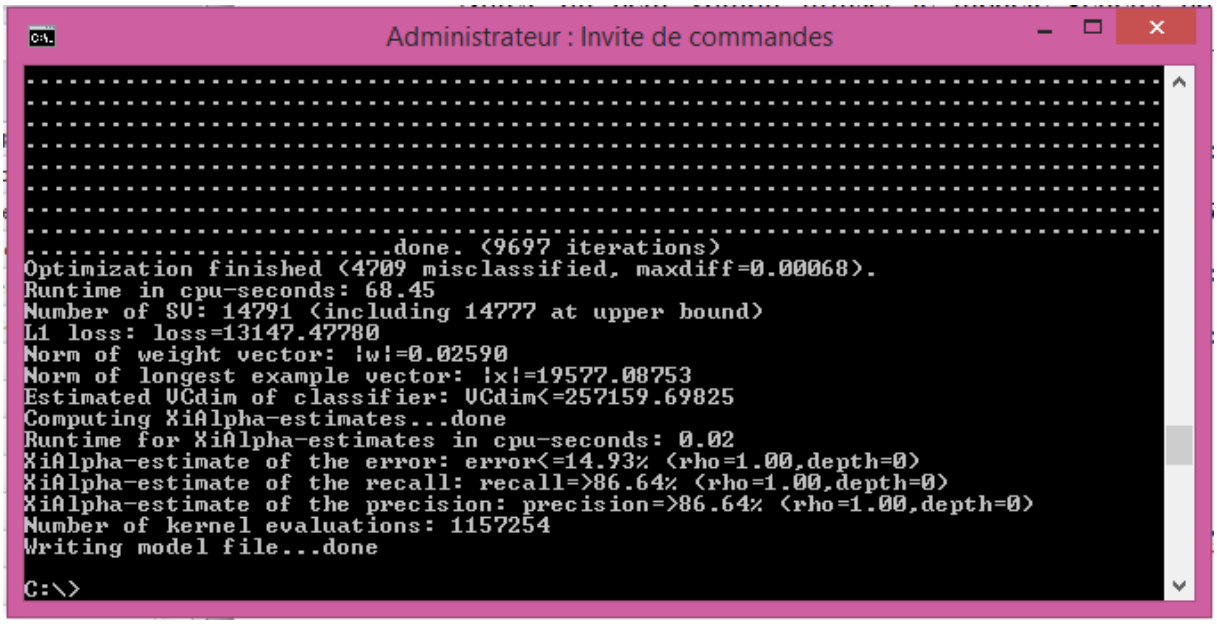

Figure 26 : Capture d'écran du résultat d'apprentissage.

Un modèle (**model.txt**) sera générer pour tester les performances de généralisation sur une base de test contenue.

2.2.1.3. Le test :

Dans cette phase on peut utiliser le modèle générer pour tester les performances de généralisation sur une base de test contenue, par exemple, dans le fichier «***exemple_test.txt***» avec les 4 lignes suivant :

1 1:0 2:1 3:53 4:10 5:3714 6:393 12:1 13:1 14:1 15:1.00 16:12 17:86 18:0.75 19:0.17 23:0.08 24:0.05

1 1:11610 2:1 3:59 4:10 5:100 6:1459 12:1 15:1 16:1 17:1.00 18:13 19:6 20:0.31 21:0.15 22:0.08 24:0.33

1 1:0 2:1 3:53 4:10 5:1044 6:403 12:1 13:1 14:1 15:1.00 16:14 18:87 19:0.71 20:0.14 22:0.07 24:0.05

1 1:0 2:1 3:53 4:10 5:2871 6:401 7:1 12:1 13:1 14:1.00 18:15 19:88 20:0.73 21:0.13 22:0.07 24:0.05

Il suffira, donc, d'écrire la commande suivante : **[System prompt]>svm_classify.exe exemple_test.txt model.txt prediction.txt**

Le retours de cette commande est représentée dans La **figure 27** ci-dessous:

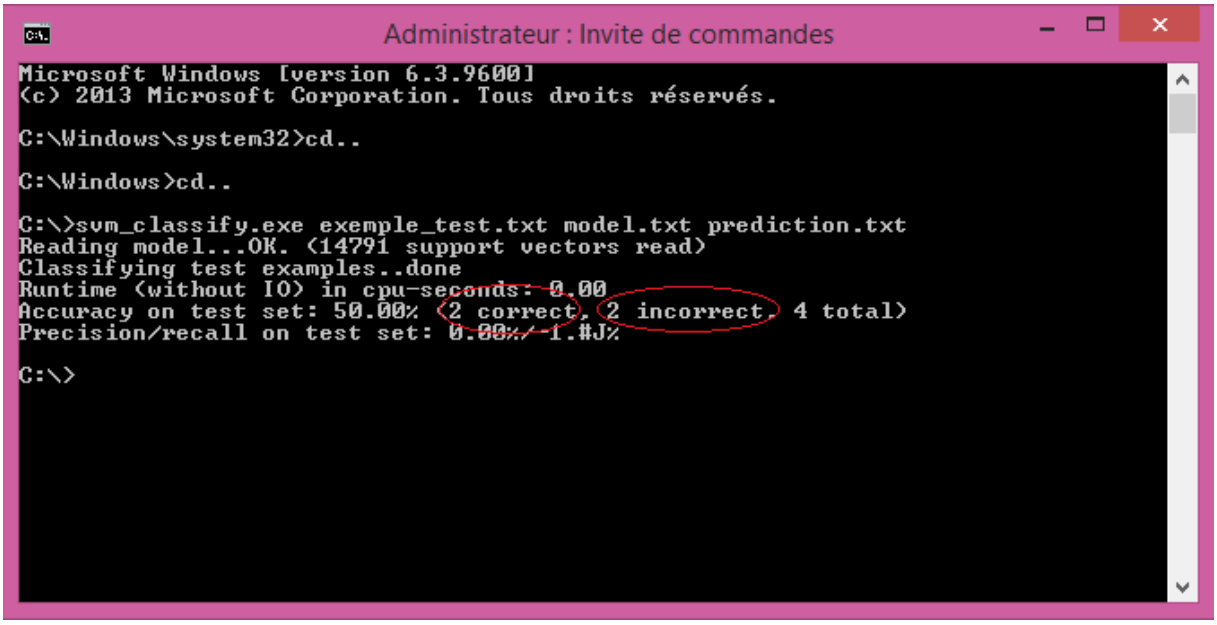

Figure 27 : Capture d'écran du résultat de classification

Et un fichier «***prediction.txt*** » avec le contenue suivant :

1.0706077

1.0874746

-10.911971

-10.921697

Qui corresponds à : 1, 1, -1, -1 (Normale, Normale, Anormal, Anormal).

La deuxième étape : dans cette étape on va présenter les intervalles utilisé pour résumé l'ensemble de données dans le but de réduire le nombre des valeurs de chaque attribut de ce ensemble, **le tableau 5** représente les codifications des valeurs de chaque attribut dans l'ensemble.

Attribut 01 : http_req_length	**Les intervalles**	**Attribut 02 : http_uri_length**	**Les intervalles**
C1_1 C1_2 C1_3	0 - 500 501 - 1000 1001 - ∞	C2_1 C2_2 C2_3	0 - 40 41 - 90 91 - ∞
Attribut 05 : uri_ressource_type	**Les intervalles**	**Attribut 08 : response_code**	**Les intervalles**
Application Autre Doc Media web	.exe, .sh, .pl Other,none, .asm, .ini, .ico .ps, .pdf, .txt, .ppt, .dat, .tar, .zip .swf, .mov, .gif, .jpg .htm, cgi, cgi_bin, .asp, .css, .php, .js, .com	C3_1 C3_2 C3_3	0 - 300 301 - 400 401 - ∞

Attribut 09 : response_time	Les intervalles	Attribut 20 : num_req_same_host	Les intervalles
C4_1 C4_2	0 – 0.5 0.51 - 1	C5_1 C5_2 C5_3	0 - 2500 2501 - 7500 7501 - ∞
Attribut 21 : num_req_same_URI	**Les intervalles**	**Attribut 22 : inter_req_time_interval**	**Les intervalles**
C6_1 C6_2 C6_3	0 - 2500 2501 - 7500 7501 - ∞	C7_1 C7_2 C7_3	?< 0 0 - 50 51 - 100
Attribut 23: req_same_host_diff_URI _rate	**Les intervalles**	**Attribut 24 : http_error_rate**	**Les intervalles**
C8_1 C8_2	0.00 -0. 50 0.51 – 1.00	C9_1 C9_2	0.00 -0. 50 0.51 – 1.00

Tableau 5 : les codifications des valeurs de chaque attribut dans l'ensemble de données

La ligne suivante représente un enregistrement de l'ensemble de test (test.txt) après le résumé:

C1_1,C2_1,1,0,autre,0,0,C3_3,C4_1,none,none,1,0,0,0,0,0,0,0,C5_1,C6_1,C7_1,C8_1,C9_2, anormal

2.2.2. Le deuxième niveau :

Dans cette section, nous expliquons l'entrée et la sortie du deuxième niveau qui est basé sur le classificateur naïve Bayes, notre approche comprend trois étapes principales :

- Prétraitement des données d'observations : cette étape concerne le prétraitement de l'historique des observations. Le résultat de cette étape est un ensemble de données formatées.
- Construction du RB naïf : dans cette étape, nous estimons la distribution des probabilités de chaque variable du RB naïf.
- Prédiction des objectifs d'intrusion : dans cette étape nous prédisons les objectifs d'intrusion par l'application des mécanismes d'inférence des RB.

Nous allons maintenant décrire les trois étapes de notre approche

2.2.2.1. Prétraitement des données d'observations:

Pour construire le RB naïf, nous allons effectuer un certain prétraitement sur les données d'observations. Les données contiennent un ensemble d'alertes qui rapportent les actions exécutées, la figure suivante présente le prétraitement des données.

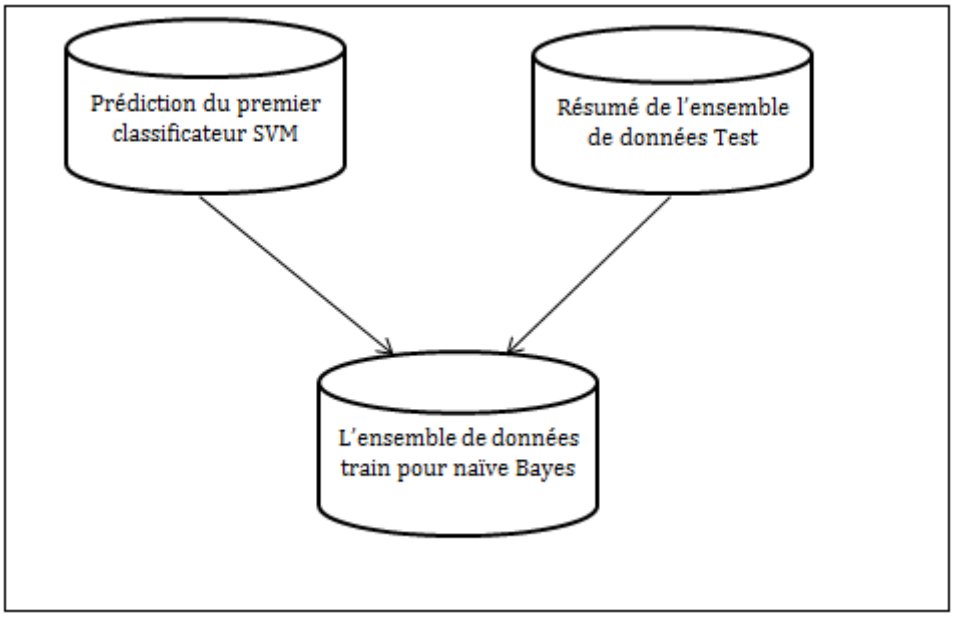

Figure 28 : préparation des données.

La méthode consiste à fusionné les attributs de l'ensemble de données test avec les prédictions du premier classificateur SVM. La description suivant illustre cette fusion sur un exemple :

La ligne suivante représente un enregistrement de l'ensemble test résumé (test.txt)

C1_1,C2_1,0,0,autre,0,0,C3_1,C4_1,none,none,0,0,0,0,0,0,0,0,C5_1,C6_1,C7_1,C8_2,C9_1 , normal

La prédiction SVM de cette connexion est représentée dans la ligne suivante :

anormal

Après la réécriture on obtient la ligne suivante :

C1_1,C2_1,0,0,autre,0,0,C3_1,C4_1,none,none,0,0,0,0,0,0,0,0,C5_1,C6_1,C7_1,C8_2,C9_1 , anormal, normal

2.2.2.2. Construction du réseau bayésien naïve :

Nous construisons un RB naïf pour chaque objectif d'intrusion. La raison pour laquelle nous considérons un RB par objectif d'intrusion au lieu d'un seul RB avec une variable classe, La **figure 29** montre la structure du RB naïf.

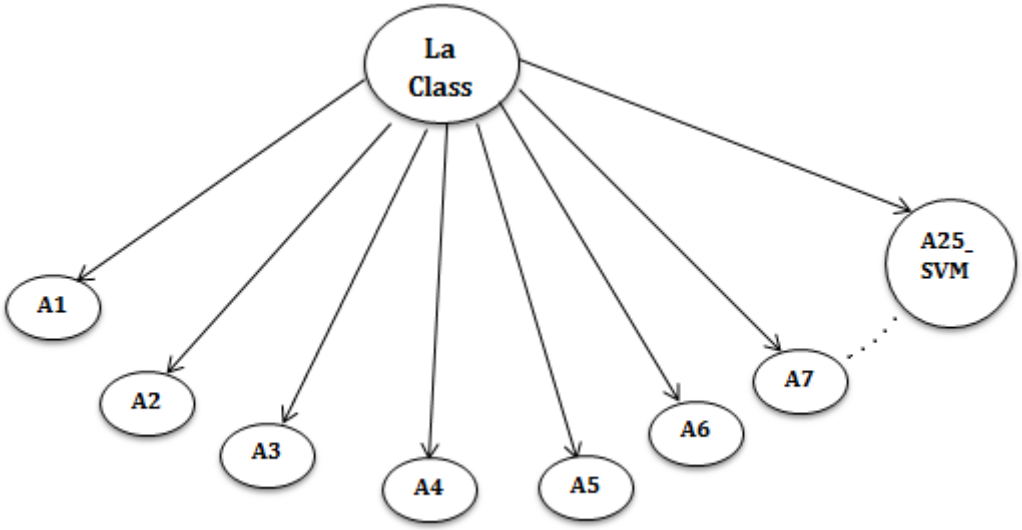

Figure 29 : La structure du model proposée.

2.2.2.3. Prédiction des objectifs d'intrusion :

Notre objectif est de montrer comment inférer (prédire) les objectifs d'intrusion étant donné que certaines actions sont récemment observées. Le but de l'inférence est d'estimer les valeurs des nœuds non observés, étant donné les valeurs des nœuds observés. Dans les RB naïfs, nous sommes intéressés à déterminé :

		Attribut	Prob_Normal	Prob_Anormal
A1	**http_req_length**	C1_1	0,850337254	1
		C1_2	0,149304311	0
		C1_3	0,000358435	0
A2	**http_uri_length**	C2_1	0,6575646	0,416393872
		C2_2	0,341506729	0,424774126
		C2_3	0,000928672	0,158832002
A3	**num_safe_req**	0	0,090765421	0,000130941
		1	0,907491284	0,999869059
		2	0,001401154	0
		3	0,000146632	0
		5	0,00013034	0
		6	6,51699E-05	0
A4	**num_unsafe_req**	0	0,99770276	0,999738117
		1	0,002280948	0,000261883
		2	1,62925E-05	0
A5	**uri_ressource_type**	web	0,210107856	0,000261883
		media	0,465443644	0
		doc	0,119847502	0
		application	0,000977549	0
		autre	0,203623448	0,999738117

Tableau 6 : Distribution des probabilités des actions.

A6	num_param	0 1 2 3 4	0,999364593 0,00013034 0,000407312 0 9,77549E-05	0,414036925 0,292523242 0,292916067 0,000261883 0,000261883
A7	num_arg	0 1 2 3 4	0,999266838 0,000211802 0,000423605 0 9,77549E-05	0,414036925 0,292523242 0,292916067 0,000261883 0,000261883
A8	response_code	C3_1 C3_2 C3_3	0,798918179 0,099351559 0,101730262	0,586094016 0,000261883 0,413644101
A9	response_time	C4_1 C4_2	0,998875819 0,001124181	1 0
A10	script_type	js none vb	0,040991886 0,959008114 0	0 1 0
A11	writing_script	cookie_rd cookie_set doc_loc none	0,004561895 0 0,000765747 0,994672358	0 0,000261883 0 0,999738117
A12	is_html_response	0 1	0,720974942 0,279025058	0,000392824 0,999607176

A13	directory_traversal	0 1	0,999967415 3,2585E-05	0 1
A14	shell_cmds	0 1	1 0	0,999476234 0,000523766
A15	sensitive_files	0 1	1 0	0,998952468 0,001047532
A16	default_login_passwd	0 1	1 0	0,999738117 0,000261883
A17	num_nonprintcar	0	1	1
A18	sql_cmds_tricks	0 1	1 0	0,998166819 0,001833181
A19	css_scripts	0 1	1 0	0,80568286 0,19431714
A20	num_req_same_host	C5_1 C5_2	1 0	0,346471127 0,653528873
A21	num_req_same_URI	C6_1 C6_2	1 0	0,786172581 0,213827419
A22	inter_req_time_interval	C7_1 C7_2	0,996627456 0,003372544	1 0
A23	req_same_host_diff_URI_rate	C8_1 C8_2	0,082048943 0,917951057	0,416524813 0,583475187
A24	http_error_rate	C9_1 C9_2	0,958388999 0,041611001	0,586355899 0,413644101
A25	prédiction_svm	Normal Anormal	0,939343087 0,060656913	0,413775043 0,586224957

Les valeurs de la classe, étant donnés les valeurs de certaines variables observées, cela peut se faire par la formule de bayes :

$$P(classe = x|y) = \frac{P(classe = x).P(y|classe = x)}{P(y)},$$

Où classe est la variable non observée (dans notre cas, l'objectif d'intrusion) et y est l'évidence observée (dans notre cas, les actions observées). Quand les observations concernent plus d'une variable, cette formule peut être écrite comme suit :

$$P(classe = x|y_1, ..., y_n) = \frac{P(classe = x, y_1, ..., y_n)}{P(y_1, ..., y_n)}$$
$$= \frac{1}{\propto}.P(y_1, ..., y_n|classe = x).P(classe = x)$$

Notons que $\propto$ est une constante qui peut être obtenue par normalisation. Maintenant,

$$P(classe = x, y_1, ..., y_n) = P(y_1, y_2, ..., y_n|classe = x).P(classe = x)$$
$$= P(y_1|y_2, ..., y_n, class = x).$$
$$P(y_2, ..., y_n|classe = x).P(classe = x)$$

Rappelons qu'en RB naïf, par définition *y1*, dans le contexte de la class, est indépendant de *y2, …, yn*. D'où :

$$P(classe = x, y_1, ..., y_n) = P(y_1|classe = x)\cdot$$
$$P(y_2, ..., y_n|classe = x).P(classe = x)$$

L'exemple suivant illustrele calcule de prédiction de chaque connexion de l'ensemble d'apprentissage en utilisent la formule suivante :

$$P(anormal/A1, A2, …., A25) = \frac{P(A1/anormal).\ P(A2/anormal)…..P(A25/anormal).P(anormal)}{P(A1,A2,……..,A25)}$$

La ligne suivante est un enregistrement de l'ensemble d'apprentissage du deuxième niveau :

C1_1,C2_1,1,0,autre,3,3,C3_1,C4_1,none,none,1,0,1,1,0,0,0,0,C5_1,C6_1,C7_1,C8_1,C9_1,normal,anormal

$$P(anormal/C1\_1, C2\_1,\ldots, normal) = \frac{P(C1\_1/anormal).\ P(C2\_1/anormal)\ldots..P(normal/anormal).P(anormal)}{P(C1\_1,C2\_1,\ldots\ldots,normal)}$$

$$P(anormal/C1\_1, C2\_1,\ldots, normal) = \frac{1.0,416393872.0,999869059\ldots\ldots..0,413775043.0,1106571}{P(C1\_1,C2\_1,\ldots\ldots,normal)}$$

$$P(anormal/C1\_1, C2\_1,\ldots, normal) = \frac{0,00000000000248516}{P(C1\_1,C2\_1,\ldots\ldots,normal)}$$

$$P(normal/C1\_1, C2\_1,\ldots, normal) = 1 - P(anormal/C1\_1, C2\_1,\ldots, normal) \approx 1$$

➔P(C1_1/normal).P(C2_1/normal)…..P(normal/normal).P(normal)>P(C1_1/anormal).P(C2_1/anormal)…..P(normal/anormal).P(anormal)

Alors la prédiction est : Normale.

Pour évaluer notre travail an a utilisées weka explorer pour la classification du deuxième niveau (naïve bayes) les figure 30 et 31représente le prétraitement et la classification des données.

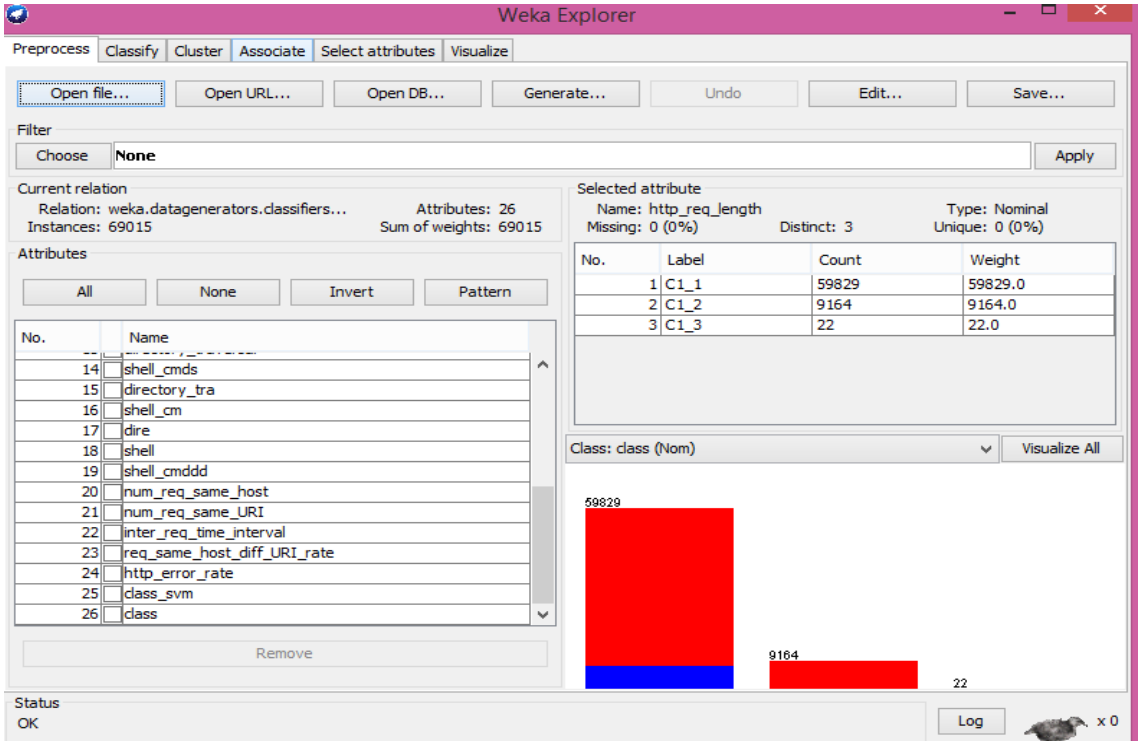

Figure 30 : prétraitement des données par weka explorer

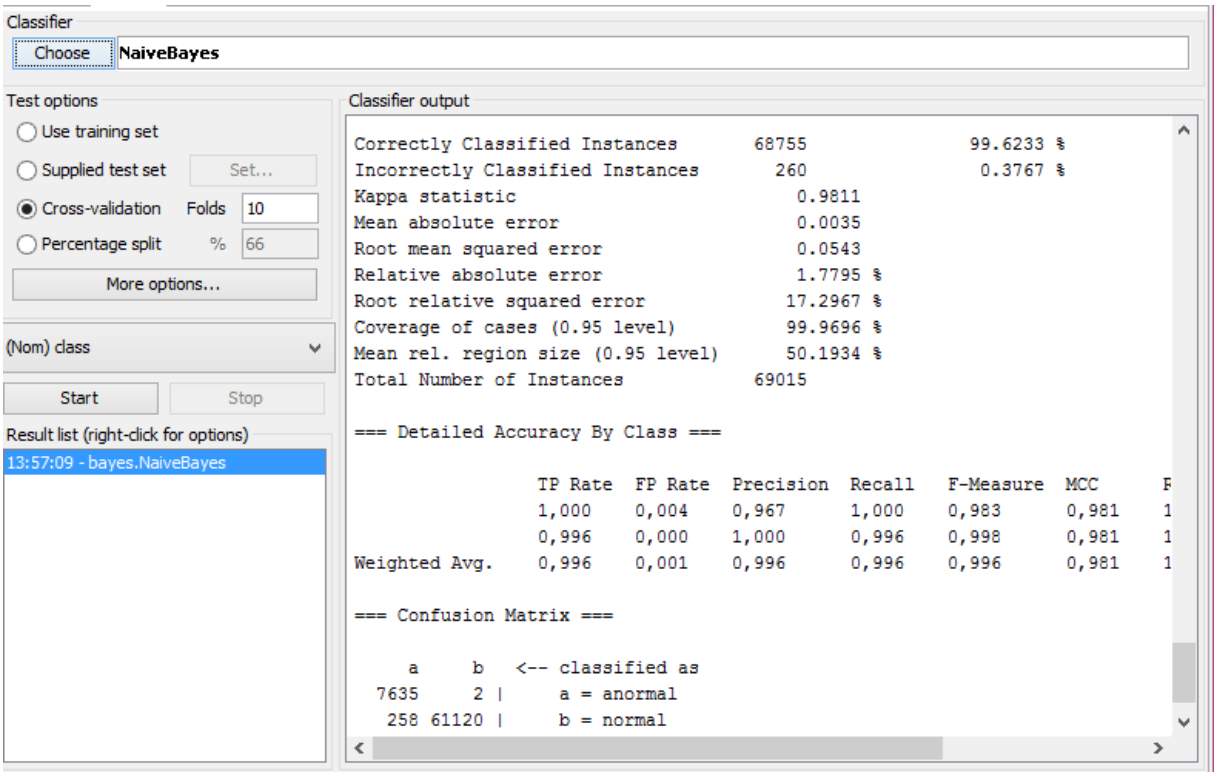

Figure 31 : classification des données par weka explorer

2.3. Etude comparative entre les classificateurs et l'approche proposée:

Pour comparer les différents classificateurs utilisés dans le premier et le deuxième niveau par rapport à l'approche, nous avons effectué une série d'expérimentation, pour chaque classificateur en a calculé les mesure suivant (Exactitude, taux de faux positif et taux de faux négative) le tableau représente les mesure obtenu :

	Exactitude	Faux positif	Faux négative
SVM	99.12%	7.12%	0.0361%
Naive Bayes	97.81%	16.33%	0.0367%
Approche	99.62%	3.26%	0.0032%

Tableau 7 : Etude comparaison entre les classificateurs.

Comme le montre la figure 32, l'approche proposé à des meilleurs résultats par rapport aux deux classifications utiliser dans l'approche, on a 99.62% pour l'Exactitude, 3.26% pour le taux des faux positif et 0.003% pour le taux de faux négative.

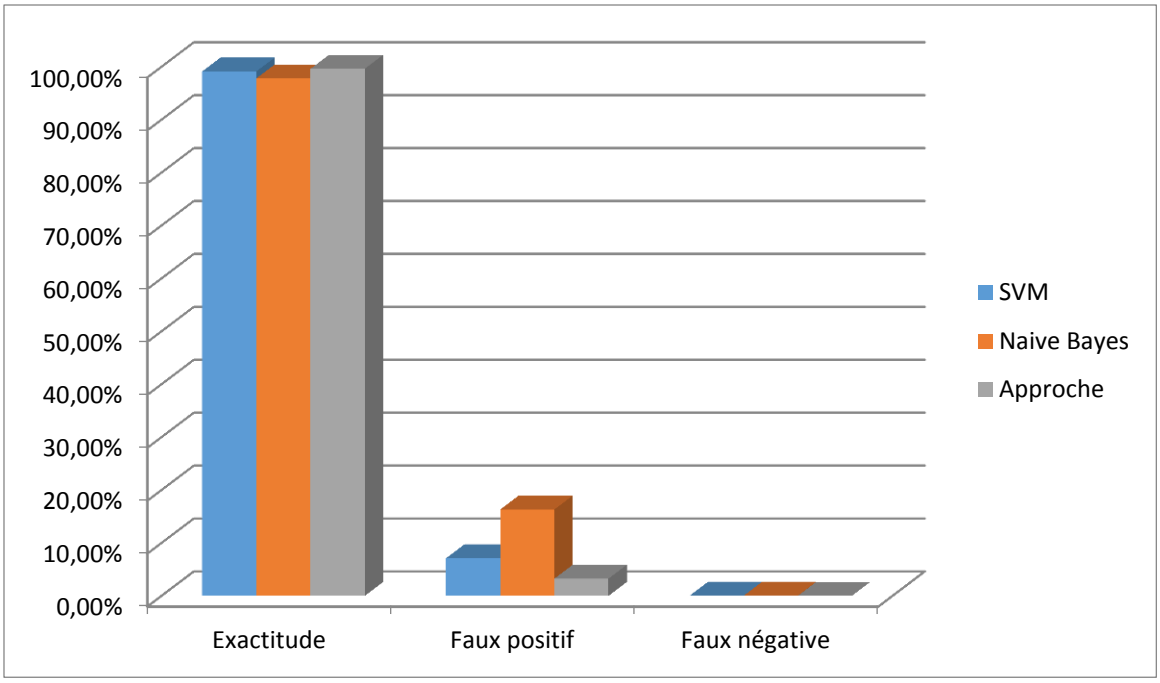

Figure 32 : Etude comparaison entre les classificateurs.

Conclusion :

Dans ce chapitre, nous avons proposé une nouvelle approche de détection d'intrusion basé sur la combinaison de différents classificateurs qui possédé une très grande capacité de généralisation. Notre approche proposé répond aux exigences suivantes : mieux détecter les attaques non fréquentes et nouvelles, donne un taux de vrai positif élevé pour les attaques fréquentes et donne un taux de fausse alarma faible. Notre approche comprend deux niveaux.

Le premier contient un seul classificateur SVMLight, la prédiction du premier niveau est sélectionnées et utilisées comme entrées du second niveau qui contient le réseau bayésien naïf comme classificateur final. Les expérimentations ont montré que notre approche donne un taux de détection et d'exactitude le plus élevé et un taux de fausse alarme faible par rapport à certains travaux bien connu. En outre, notre approche a montré sa capacité à mieux détecter les attaques en minimisant le taux de faux positif (fausse alerte).

Chapitre5 :

Implémentation et réalisation.

Chapitre5 : Implémentation et réalisation.

Introduction :

Un système de détection d'intrusion est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions.

Ce chapitre consiste à présenter l'environnement de programmation utilisée dans l'implémentation et à décrire la réalisation en détail de notre application qui est pour but de faciliter l'utilisation de l'approche proposée.

1. L'environnement de la programmation :

1.1. Présentation du langage java :

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'Unix Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java. Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java, dont le nom technique est JRE (Environnement d'exécution Java). Cette dernière est constituée d'une JVM Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java.

1.2. Présentation de NetBeans IDE :

NetBeans est un environnement de développement intégré (EDI), placé en open source, permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris, Mac. Le projet Netbeans comprend un IDE et une plateforme d'applications qui permettent aux développeurs de créer rapidement des applications entreprise, bureau, web et mobiles.

Le projet NetBeans est supporté par une communauté de développeurs dynamique et offre une documentation complète et des ressources de formation, en plus d'une sélection variée de plugins tiers.

1.3. Présentation de Weka :

Weka (Waikato Environment for Knowledge Analysis) est un ensemble d'outils permettant de manipuler et d'analyser des fichiers de données, implémentant la plupart des algorithmes d'intelligence artificielle, entre aux, les arbres de décision et les réseaux Bayésien. Il est écrit en java, disponible sur le web **[64]**, et s'appuie sur le Livre : « Data Mining, practical machine learning tools and techniques with Java implementation », Il se compose principalement :

- De classes Java permettant de charger et de manipuler les données.
- De classes pour les principaux algorithmes de classification supervisée ou non supervisée.
- D'outils de sélection d'attributs, de statistiques sur ces attributs.
- De classes permettant de visualiser les résultats.

On peut l'utiliser à trois niveaux :

- Via l'interface graphique, pour charger un fichier de données, lui appliquer un algorithme, vérifier son efficacité.
- Invoquer un algorithme sur la ligne de commande.
- Utiliser les classes définies dans ses propres programmes pour créer d'autres méthodes, implémenter d'autres algorithmes, comparer ou combiner plusieurs méthodes.

2. Les étapes de la réalisation du projet :

Dans notre travail, nous avons procédé à la mise en place d'une application qui représente notre approche proposée, montrer les résultats obtenu et la déférence entre la prédiction de cette approche et celle des classificateurs utilisé dans les déférents niveaux de l'approche.

On a créé cinq classes pour la réaliser ce projet :

- ✓ Accueil (main)
- ✓ SVM_Light_Classifieur
- ✓ Naive_Nayes_Classifieur
- ✓ Approche_classifieur
- ✓ Comparaison_classifieur

- **Accueil**

Interface accueil permet à l'utilisateur de consulté les classificateurs existant dans notre approche et de choisir un de ces classifieurs, sachant que l'approche proposée basé sur les deux classifieurs existant.

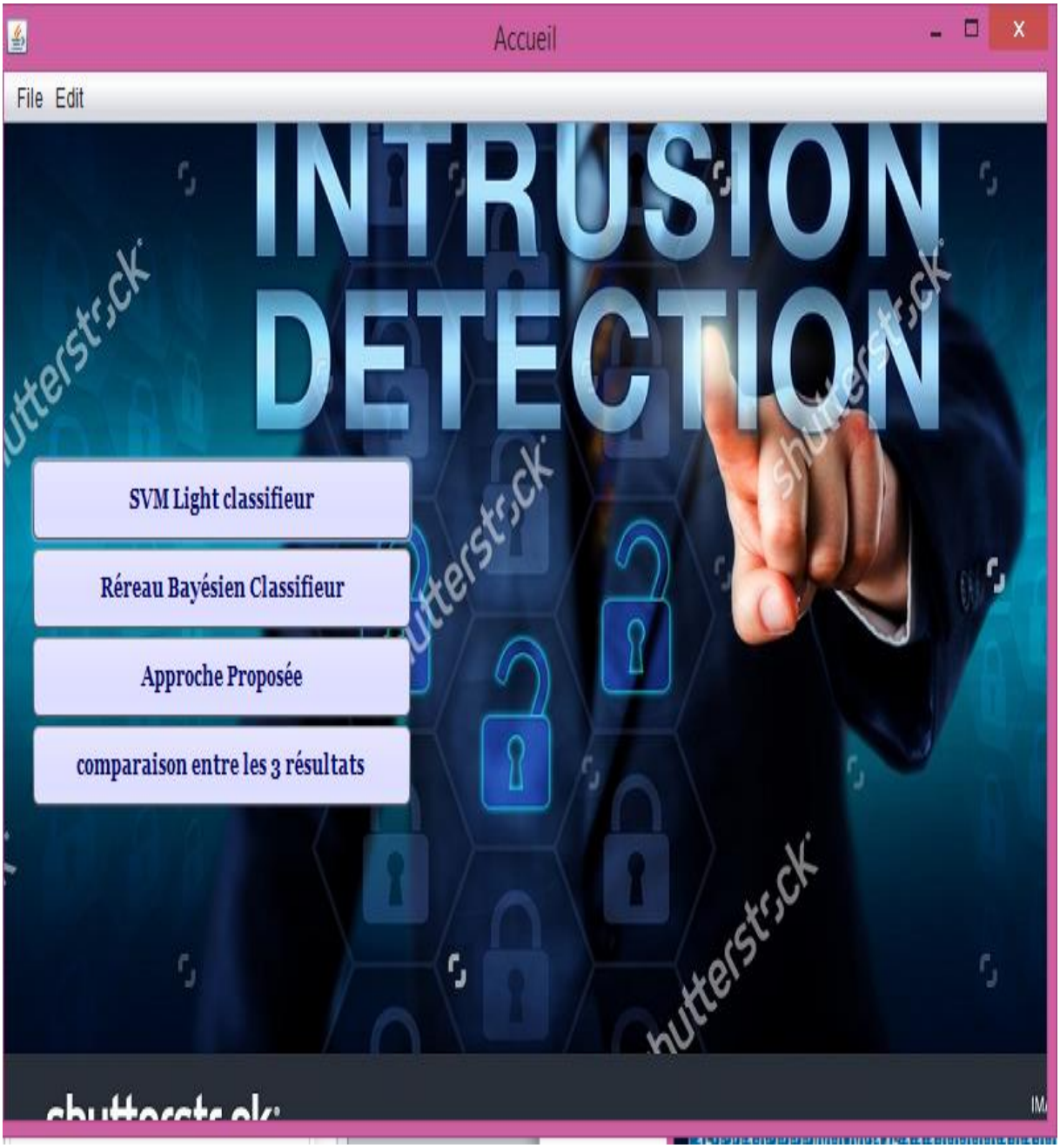

Figure 33 : page d'accueil

- **SVM_Light_Classifieur**

Interface permet de classifier les connexions de la base PLACID à l'aide de SVMLight, est-ce là par l'exécution de l'invité de commande CMD.

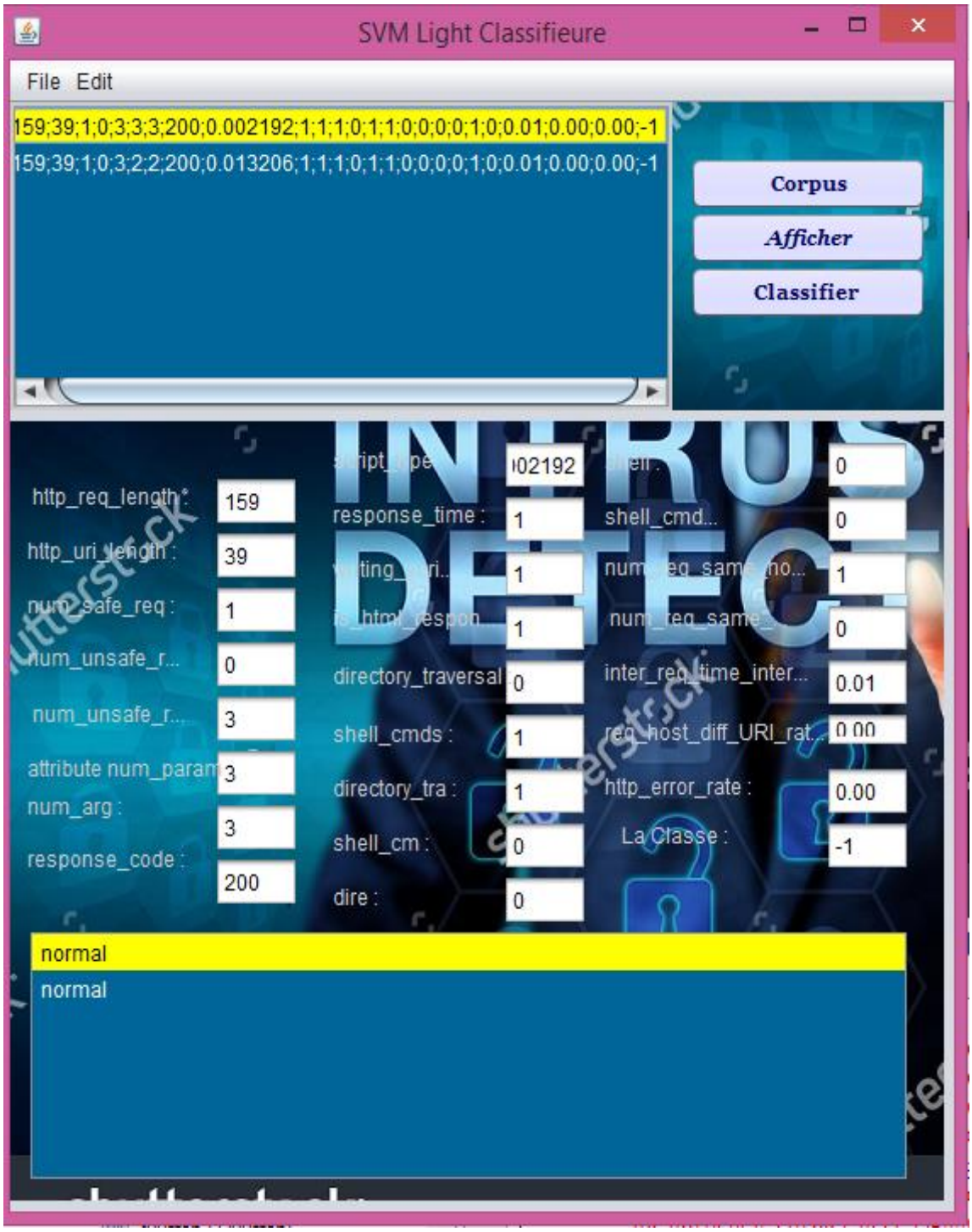

Figure 34 : Interface Svm_light_classifieur

Le bouton corpus affiche les données de de la base PLACID comme le montre ma figure 35.

Figure 35 : bouton corpus

- **Naive_Bayes_Classifieur**

Interface permet de sélectionner et de classifier les connexions de la base PLACID par un classificateur Naïve Bayes à l'aide de Weka Explorer.

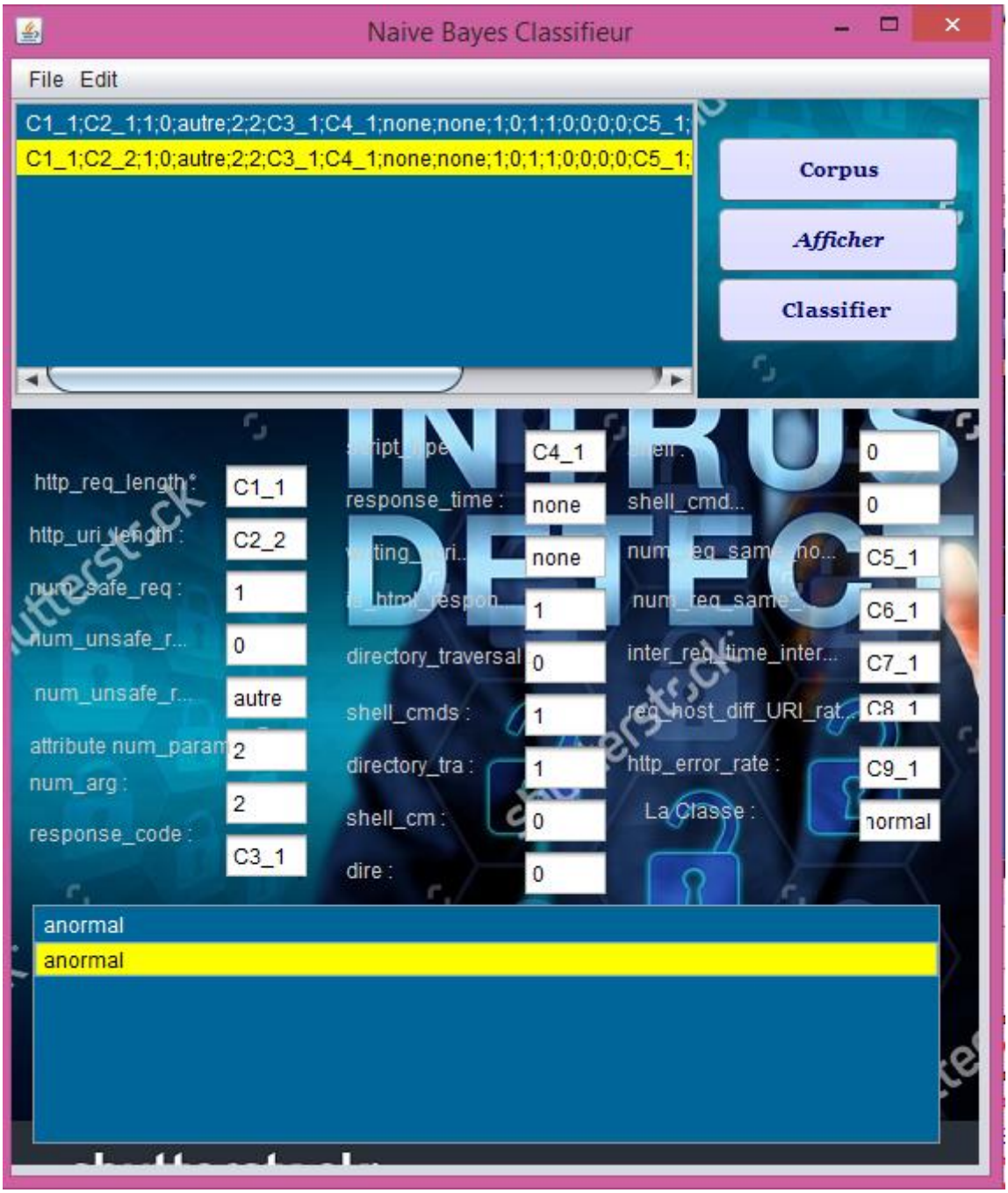

Figure 36 : Interface du classificateur Naïve Bayes.

- **Approche_Classifieur**

Interface permet de récupérer et de fusionner les connexions sélectionnées dans classificateur_Naive_Bayes avec la prédiction SVMLight.

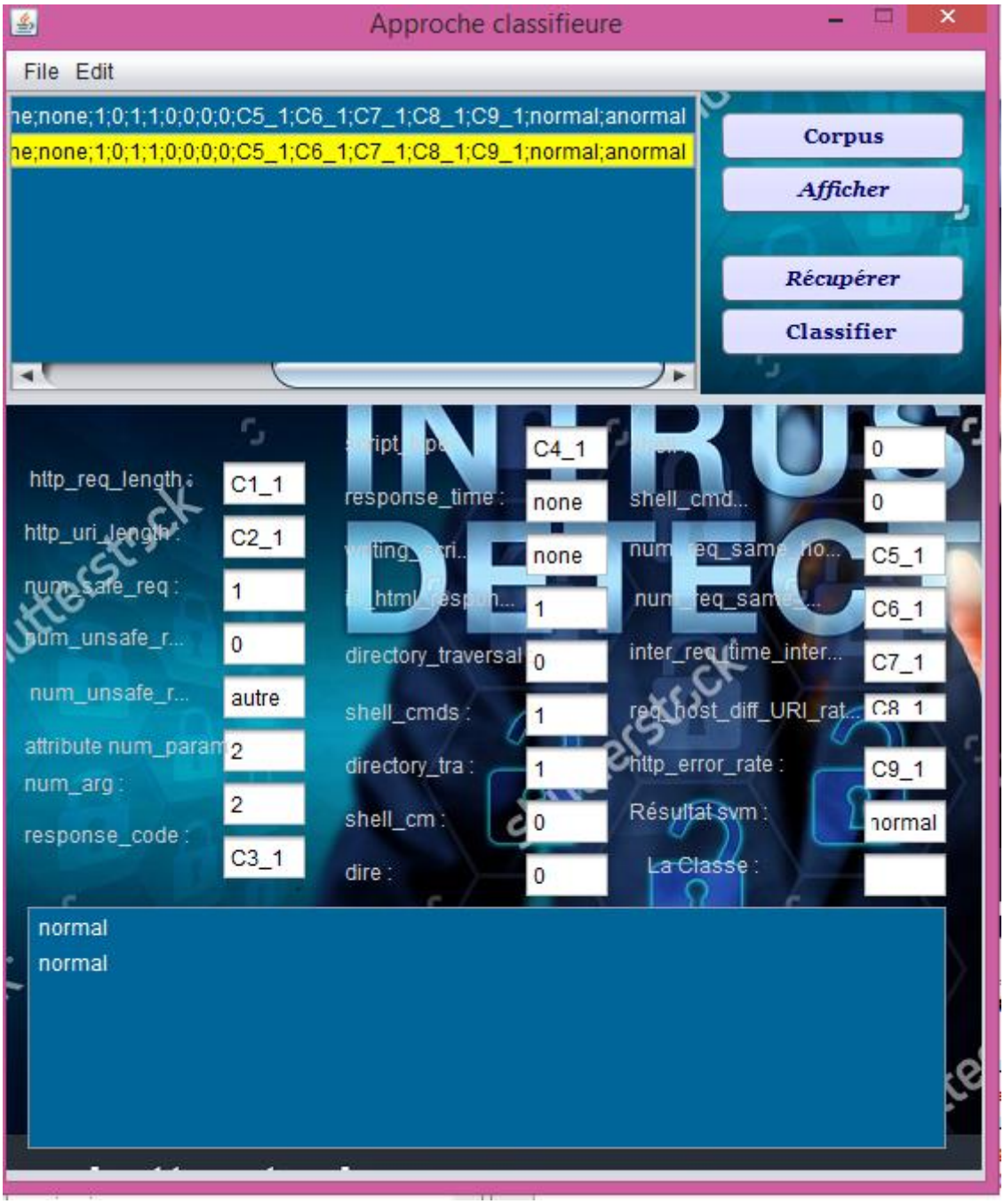

Figure 37 : Interface approche_Classifieur

- **Comparaison_classifieur :**

Cette interface présente la comparaison entre les résultats des deux classificateurs et le résultat de l'approche.

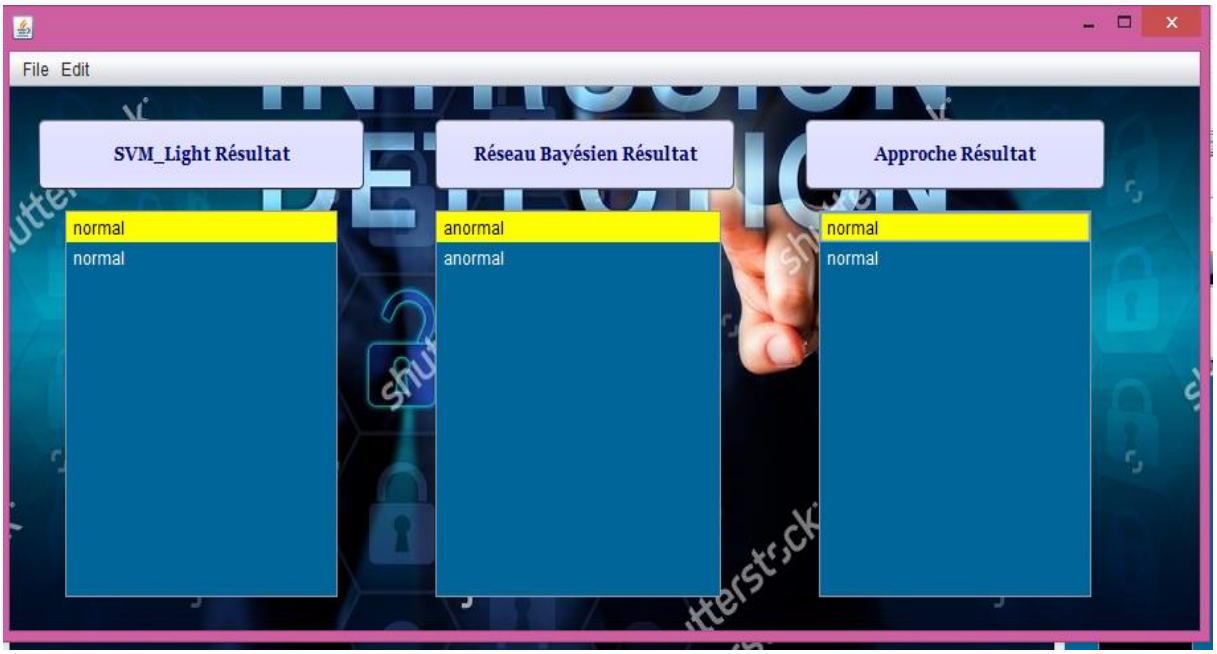

Figure 38 : Interface Comparaison_classifieur

Conclusion :

Au cours de ce dernier chapitre, on a réalisé une petite application (représentation d'approche) et on a implémenté un système de détection d'intrusion IDS dont le but est de montrer la performance de l'approche proposée avec un taux d'exactitude 99.62%, un taux de faux positif 3.26% et un taux de faux négative 0.003%.

Conclusion général

La sécurité réseau est un des problèmes les plus sérieux que connaissent les entreprises dotées d'un réseau informatique, Il ne sera jamais possible de sécuriser totalement un système d'information, car il y'aura toujours des hackers pour découvrir des nouvelles failles dans le système, mais en peut toujours rendre une intrusion plus difficile en appliquant de nouvelle approche, de ce fait nous avons proposé dans ce mémoire, une nouvelle solution pour la détection d'intrusion basé sur deux classificateur les Machine a support vecteur (SVM) et Les réseau Bayésien Naïf.

Le système de détection d'intrusion (IDS) est devenu très indispensable pour tout réseau informatique, il nous permet de connaitre toutes activités anormales qui peuvent présenter un danger pour notre réseau.

L'approche proposer vise à répondre aux objectifs de la sécurité réseau, et surtout à diminuer le taux de fausses alertes et gérer de grandes quantités de ce dernier. Cette approche nécessite une phase d'apprentissage, pour cela, en a utilisé les données du projet PLACID afin de traiter et tester l'approche, cette base.

Ce travail nous a permis d'avoir une idée claire sur les applications du domaine de sécurité informatique. On a également découvert les IDS et leurs approches, plus précisément L'approche comportementale. Cette approche qu'on a élaborée présente des avantages comme la détection rapide des anomalies ainsi qu'un taux de fausses alertes limité.

En outre, il est important de noter que le risque nul d'être piraté n'existe pas et il faut s'avoir s'appuyer au mieux sur les outils (nouvellement) disponibles afin de tendre vers cet idéal. Plusieurs perspectives s'imposent, nous citons quelques une qui nous semblent les plus importantes :

- Réaliser et tester d'autre IDS en utilisant d'autre classificateur pour diminuer plus le taux de faux positif (faux alerte).
- Améliorer l'approche pour mieux détecter les attaques de faible fréquence sans diminuer le taux de détection des attaques fréquentes.

Bibliographie

[1] **:** KALONJI BILOLO, La Sécurité Informatique au Congo, Editions Universitaires européennes, SARBRUCK, 64 pages.

[2] **:** Eric léopold et Serge Lhoste, La sécurité informatique, P.U.F « Que sais-je ? »,2007, 128 pages.

[3] **:** Karima Boudaoud, Détection d'intrusion : Une nouvelle approche par système multiagents. Thèse de doctorat de l'Université de Genève.2002.

[7] : CARPENTIER, Jean-François, La sécurité informatique dans la petite entreprise : état de l'art et bonnes pratiques, Editions ENI, Paris, 2009, 277 pages

[8] : David Burgermeister, Jonathan Krier, les systémes de détection d'intrusions.

[9] : Karim Tamine, Sécurité dans les réseaux, Cours Master2 (recherche)-Informatique, Décembre 2004.

[12] : G.Zémor, Cours de cryptographie, 2000.

[13] : Douglas Stinson, Cryptographie, théorie et pratique, Présentation claire des mathématiques de la cryptographies, 2003.

[15] : David Burgermeister et Jonathan Krier, les systèmes de détection d'intrusions.

[16] : Tarek Abbes, Classification du trafic et optimisation des règles de filtrage pour la détection d'intrusion, Thèse de doctorat de l'université Henri Poincaré, Nancy1, 2004.

[18] : J. Justen, Network and System Professional Association Inc., Nessus 2.0.8. Technical report, Novembre 2003.

[19] : J. Anderson, Computer Security threat monitoring and surveillance, 1980.

[20] : Dorothy E. Denning , An intrusion detection model, IEEE Transaction on software engeneering, SE-13 : 222-232, 1987.

[21] : Philipe Biondi, Architecture expérimentale pour la détection d'intrusion dans un système informatique, Avril-Septembre 2001.

[22] : R. Heady, G. Luger, A. Maccabe et M. Sevilla, The architecture of a network level intrusion detection system, Aout 1990.

[23] : Benjamin Morin, Corrélation d'alertes issues d'outils de détection d'intrusion avec prisent compte d'informations sur le système surveillé, Thèse de doctorat de l'Université de rennes, 2004.

[25] **:** sécurité de l'Information et Cryptologie, Rapport de Stage de fin d'étude Master 2, Conix Security 34, rue Guynemer ,92130 Issy Les Moulineaux, Université de Limoges

[26] **:** Sécurité d'une application Web à l'aide d'un système de détection d'intrusions comportementale, Mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique**,** Université Abou Bakr Belkaid– Tlemcen**,**

[27]: H. Debar, M. Dacier ET A. Wespi, towards a taxonomy of intrusion detection systems, Computer Networks, Elseiver, 1999.

[28]: A. Phillip, Porras ET A. Valdes, Live traffic analysis of tcp/ip gateways. Proc. ISOC Symposium on Network and Distributed System Security (NDSS98).San Diego, Mars 1998

[30]: Yacine Bouzida, Application de l'analyse en composante principale pour la détection d'intrusion et détection de nouvelle attaque par apprentissage supervisé, Thèse de doctodat de l'Université de rennes, 2006.

[31] :P. Mahé, Noyaux pour graphes et Support Vector Machines pour le criblage virtuel de molécules, Rapport de stage, DEA MVA 2002/2003, Septembre 2003

[32] : Mohamadally Hasan, Fomani Boris, SVM machine à vecteurs de support ou séparateur à vaste marge, BD Web, ISTY3, Versailles St Quentin, France,janvier 2006.

[33] : A. Cornuéjols , Une nouvelle méthode d'apprentissage : Les SVM. Séparateurs à vaste marge, Université de Paris-Sud, Orsay, France, Juin2002.

[34] : Gueddouh Soumia , L'apprentissage des SVMs incrémentales ,Université Mohamed Khider Biskra, Alger 2011.

[35] : J. Christopher et C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and knowledge Discovery, pages 121-167, 2005.

[36] : V.N. Vapnik. The Nature of Statistical Learning Theory. Springer Series in Statistics, 1995.

[37] : G. Loosli, S. Canu, S. Vishwanathan et M. Chattopadhay. Boite à outils SVM simple et rapide. RIA – Revue d'Intelligence Artificielle, vol. 19, no. 4, pages 741-767, 2005.

[38] : M. Moorthy, S. Sathiyabama ,Hybrid Fuzzy Based Intrusion Detection System for Wireless Local Area Networks.

[39] : Salem Benferhat, Tayeb Kenaza , Vers une évaluation globale des classifieurs Bayésiens pour la détection d'intrusion.

[40] : Laurence Grammont, cours de probabilités 2eme années d'économie et de gestion, semestre2.2004

[41] : L.Smail : Algorithmes pour les réseaux bayésiens et leurs extensions, Thèse doctorat de l'université de Poly Tech Nantes, Années 2004.

[42] : Eduardo Sanchez Soto : Réseaux bayésiens dynamique pour vérification du locuteur. Thèse doctorat 2005.

[43] : Pascal Cheung- Mon- Chan : Réseau bayésiens et filtres particuliers pour l'égalisation adaptative et le décodage conjoints thése doctorat de l'ecole normale superieure de Cachan.spécialité mathématique, Années 2006.

[44] : Tom. Mitchell : Generative and discriminative classifier, Naïve bayes and logistic regression. Machine Learning, Draft 2010.

[45]: C. Boutilier, *Toward a Logic for Qualitative Decision Theory*. Dans KR'94, 1994.

[46]: C. Boutilier, R. I. Brafman, C. Domshlak, H.H. Hoos, et D. Poole. *A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements*. Journal of Artificial Intelligence Research (JAIR), 2004.

[47] : Camille Séka kotchi, Véronque Delacroix et sylvain piechowiak : Etude de la performance des algorithmes d'inférence dans les réseaux Bayésiens.

[48]: F .V. Jensen. *Bayesian Networks and Decision Graphs*. Information Science and Statistics. Springer, Juillet 2001.

[49]: M.L. Puterman. *Markov Decision Processes : Discrete Stochastic Dynamic Programming.* Wiley-Interscience, 2005.

[50] : K.Tanaka, J. I. Imoune et D.M.Titterington : loopy belief propagation and probabibilistic image processing.1999

[51] : Sabine Barrat, Modèle graphiques probabilistes pour la reconnaissance de formes. Thèse doctorat de l'université Nancy 2. Spécialité informatique. Année 2009.

[52] : C. Aaron, Algorithme EM et classification non supervisés. Thèse de doctorat d'Université Paris I .années 2001

[53] : O.Francois, p.leray, etude comparative d'algorithme d'apprentissage de structure dans les réseaux bayésiens.Proceedings of the IEEE,vol. 60,no. 4, page 586-704 ,2004

[54]: R. D. Shachter and M. A. Peot. Decision making using probabilisticinference methods. In Uncertainty in Artificial Intelligence, pages 276-283, 1992.

[55]: F .V . Jensen and T .D. Nielsen, Bayesian Networks and Decision Graphs ,Information Science and Statistics, Springer , 2007.

[57]: N. Friedman and M. Goldszmidt. Building classiers using bayesian networks. In 13th National Conference on Articial Intelligence AAAI'96, pages 1277-1284, 1996.

[58]: P. Langley , W. Iba, and K. Thompson. An analysis of bayesian classiers. In 10. th. National Conference on Articial Intelligence AAAI'92, pages 223-228, San Jose, CA, 1992. AAAI Press.

[59]: R. Kohavi. Scaling up the accuracy of naive-bayes classiers : a decision tree hybrid. In the 2nd International Conference on K nowledge Discovery and Data Mining, pages 202-207, 1996

[60]: Gregory F. Cooper et Edward Herskovits, A Bayesian Method for the Induction of Probabilistic Networks from Data, Mach. Learn, 9(4) : 309-347,1992.

[61]: Chris Meek, PC Algorithm for Nonparanormal Graphical Models, journal of Machine learning research 14 (2013) 3365-3383.

[62]: Jimmy Vandel, Apprentissage de la structure de réseau bayésien, Application aux données de génétique-génomique, Thèse Doctorat de l'université de Toulouse, 7 Décembre 2012.

Webographie

[4] : https://www.csie.ntu.edu.tw/cjlin/libsvm, consulté le 14/01/2016.

[5] : http://www.cs.cmu.edu/~mccallum/bow; consulté le 14/01/2016.

[6] : http://www.commentcamarche.net/contents/1033-introduction-a-la-securite-informatique, [Introduction à la sécurité informatique], consulté le 02/05/2016.

[10] : http://www.commentcamarche/, [Réseaux et protocoles issu de comment ça marche], consulté le 10/04/2016.

[11] : http://www.commentcamarche.net/contents/713-ping, [Ping], consulté le 02/04/2016

[14] : http://www.commentcamarche/, [Prévention et détection d'intrusion issu de comment ça marche.], consulté le 02/04/2016.

[17] : http://www.saintcorporation.com/, [Saint Corporation, Saint documentation contents], consulté le 03/04/2016.

[24] : https://www.snort.org/ , consulté le 03/02/2016.

[29]: https://www.securiteinfo.com/ , [Introduction et Initiation a la sécurité informatique], consulté le 03/02/2016

[63] : http://www.idiap.ch/scientific-research/themes, consulté le 16/01/2016.

[64] : https://sourceforge.net/projects/weka/files/weka-3-7-windows-x64/, consulté le 07/03/2016

[65] : http://svmlight.joachims.org/, consulté le 02/02/2016.

Lexique

ARP : Address Resolution Protocol

CPU : Central processing unit,

CBN : Classifieur Bayésien Naïf

CP : préférences conditionnelles

DNS : Domain Name System

DOS : Danial of Service

DAG : Directed Acyclic Graph

FTP : File Transfer Protocol

HIDS : Host Based Intrusion Detection System

IDDL : Intrusion Detection Description Logic,

IDS : Intrusion Détection System

IP : Internet Protocol

MWST : Maximal Weight Spanning Tree

NIDS : Network Based Intrusion Detection System

SYN : synchronize

SNMP : Sample Network Management Protocol

SVM : Machines à Vecteurs de Support

TCP : Transmission Control Protocol

RB : Réseaux Bayésien

TELNET : Terminal Network ou Telecommunication Network, ou encore teletype network

PLACID : Probabilistic graphical models and Logics for Alarm Correlation in Intrusion Detection

VC : Vapnik-Chervonenkis