

République Algérienne Démocratique 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 fonctions de croyance dans les
Systèmes de Détection d'Intrusion**

Pour l'obtention du diplôme de Master

Spécialité : Génie Informatique

Option : Système d'information

Réalisé : - Imene SAFA.
- Anissa ZENATI.

Dirigé par : Mr. Abdelkader ALEM.

Année Universitaire : 2015-2016

Dédicace

Je dédie ce modeste travail très spécialement à mes chers parents pour tout ce que vous avez fait pour moi. Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir.

*À mon frère et mes sœurs et toute ma famille,
À mon binôme SAFA Imene en souvenir de tous les bons moments qu'on a vécus ensemble.*

A mes très chers amis, J'espère de tout mon cœur que notre amitié durera éternellement.

A tous mes professeurs et les collègues d'études.

Anissa.

Dédicace :

Je dédie ce modeste travail ;

A mes très chers parents ;

A mes frères Said, Khaled, Kada ;

A ma chère et unique sœur Fatima ;

A mes belles sœurs Fatiha, Souad ;

Pour la gentillesse, la générosité, la joie de vivre, la patience et la volonté dont vous m'avez toujours entourée et que vous m'avez transmise.

Je vous remercie également du fond du cœur pour m'avoir encouragée et conseillé durant mes études.

A mes chers neveux Adem, Abd ELIlah ;

A mon binôme Mlle Anissa ZENATI pour les beaux moments que je les passés avec elle et toute sa famille ;

Je n'oublierai pas d'adresser ma gratitude à tous mes amies et à tous mes camarades pour leur soutien et encouragements.

Merci !!

Imene

Remerciements

*Avant toute chose je tiens à remercier le grand « **Dieu** » de nous avoir donné le courage et la volonté qui nous ont permis de réaliser ce modeste travail.*

*Nos remerciement particulier à **Mr Alem Abdelkader** pour ses remarques pertinentes et son optimisme qui nous fait parfois défaut. En outre ses qualités d'encadreur, nous ont toujours permis d'avancer à un rythme régulier dans notre travail, nous encourageant à persévérer.*

*Nous exprimons toute notre gratitude à nos examinateurs Monsieur **Bekar Khaled** et **Aid Lahcen** pour avoir accepté d'examiner ce travail et leurs participations au jury.*

Merci également à l'ensemble des enseignants du département Informatique Université Tiaret.

Enfin, nous ne saurions terminer ces remerciements sans y associer toute personne qui, de près ou de loin, nous a apporté son aide ou sa sympathie.

Sommaire

| | |
|--|-----------|
| Introduction générale | 1 |
| <i>Chapitre I : Sécurité Informatique et système de détection d'intrusion(IDS)</i> | |
| Introduction..... | 3 |
| I. Sécurité informatique..... | 3 |
| 1. Définition..... | 3 |
| 2. Les objectifs de la sécurité Informatique..... | 5 |
| 3. Les champs d'application de la sécurité Informatique..... | 5 |
| 4. Terminologie de la sécurité Informatique..... | 6 |
| 4.1 Vulnérabilité | 7 |
| 4.2. Menace | 7 |
| 4.3Attaque..... | 9 |
| 5. Les possibilités en matière de sécurité réseaux | 18 |
| 5.1. Les Firewalls | 19 |
| 5.2. Les filtres de paquets | 19 |
| 5.3. Les systèmes de détection d'intrusions..... | 19 |
| II. Les Systèmes de détection d'intrusions | 19 |
| 1. Définition | 20 |
| 1.1. Intrusion..... | 20 |
| 1.2. Système de détection d'intrusion..... | 20 |

| | |
|---|-----------|
| 2. Les familles de Système de détection d'intrusion | 21 |
| 2.1. Système de détection d'intrusion réseau NIDS | 21 |
| 2.2. La détection d'Intrusion basée sur l'hôte HIDS | 22 |
| 2.3. Système de Détection d'Intrusion de Nœud Réseau (NNIDS)..... | 23 |
| 2.4. Système de Détection d'Intrusion Hybride | 23 |
| 3 .Concepts de base | 24 |
| 4. Les critères pour évaluer l'efficacité des systèmes de Détection d'intrusions | 24 |
| 5. Classification des systèmes de détection d'intrusion..... | 25 |
| 5.1. La méthode d'analyse..... | 26 |
| 5.2. Le comportement de la détection..... | 28 |
| 5.3. L'emplacement des sources d'audits..... | 29 |
| 5.4. La fréquence d'utilisation..... | 29 |
| Conclusion | 30 |
| Chapitre II : Machine à Support Vecteur et Réseaux Bayésiens. | |
| Introduction..... | 31 |
| I. Classification | 31 |
| 1. Apprentissage non supervisé | 31 |
| 2. Apprentissage supervisé | 32 |
| II. Machine à Support Vecteur | 32 |
| 1. Notion de base..... | 33 |
| 2. Propriété fondamentale..... | 34 |

| | |
|---|-----------|
| 3. Classification à marge maximal..... | 36 |
| 4. SVM binaire..... | 37 |
| II. Les Réseau bayésien..... | 40 |
| 1. Définition..... | 41 |
| 2. Exemple d'un Réseau Bayésien | 42 |
| 3. La construction d'un réseau Bayésien | 43 |
| 4. Intérêts des Réseau Bayésien | 44 |
| 5. D-Séparation..... | 44 |
| 6. Apprentissage des Réseaux Bayésiens | 45 |
| 7. Inférence dans les Réseaux Bayésiens..... | 47 |
| 8. Classification dans les Réseaux Bayésiens | 47 |
| 9. Avantages et Limites des Réseaux Bayésiens | 47 |
| Conclusion..... | 48 |

Chapitre III : Apport des fonctions de croyance dans la détection d'intrusion.

| | |
|--|-----------|
| Introduction..... | 50 |
| I. Théorie des fonctions de Croyance..... | 51 |
| 1. Modélisation des fonctions de Croyance..... | 51 |
| 2. Fonction de masse | 51 |
| 3. Fonction de croyance | 52 |
| 3.1. Mesure de conflit | 53 |
| 3.2. Fonction de non croyance..... | 54 |

| | |
|---|----|
| 3.3. Fonction de communalité..... | 55 |
| 4. Combinaison..... | 55 |
| 4.1. Combinaison conjonctive..... | 55 |
| 4.2. Combinaison disjonctive | 56 |
| 4.3. Combinaison mixte..... | 56 |
| II. Approche proposée | 56 |
| 1. La description du modèle | 57 |
| 2. La base théorique de notre approche..... | 57 |
| 3. La structure générale de notre modèle..... | 58 |
| 3.1. Module de détection SVM..... | 59 |
| 3.2. Module de détection RB..... | 60 |
| 4. Exemple illustratif..... | 62 |
| 5. Expérimentations | 63 |
| Conclusion..... | 65 |

Chapitre IV : Implémentation et réalisation.

| | |
|---|----|
| Introduction..... | 66 |
| I. Outil de construction d'application..... | 66 |
| 1. les langages utilisés dans le développement..... | 66 |
| 2. Environnement de développement intégré | 66 |
| 3. Le Weka | 67 |
| 4. SVMLight | 69 |

| | |
|---------------------------------------|-----------|
| II. Scénario d'exécution | 72 |
| Conclusion..... | 74 |
| Conclusion générale | 76 |

Liste de figures :

| | |
|---|----|
| <i>Figure I.1</i> : Organisation de la sécurité informatique..... | 4 |
| <i>Figure I. 2</i> : Typologie des menaces..... | 8 |
| <i>Figure I. 3</i> : Attaque directe | 11 |
| <i>Figure I.4</i> : Attaque indirecte par rebond..... | 11 |
| <i>Figure I. 5</i> : Attaque indirecte par réponse..... | 12 |
| <i>Figure I. 6</i> : Typologie des faiblesses de sécurité..... | 13 |
| <i>Figure I.7</i> : Les différents types de scanning..... | 14 |
| <i>Figure I.8</i> : Attaque par déni de service distribué..... | 16 |
| <i>Figure I.9</i> : Architecture d'un système de détection d'intrusions..... | 21 |
| <i>Figure I.10</i> : système de détection d'intrusion réseau NIDS..... | 22 |
| <i>Figure I.11</i> : Système de détection d'Intrusion HIDS..... | 23 |
| <i>Figure I.12</i> : Taxonomie des systèmes de détection d'intrusion..... | 26 |
| <i>Figure II.1</i> : Le Principe générale de Classification par SVM..... | 32 |
| <i>Figure II.2</i> : L'hyperplan H qui sépare les deux ensembles de points..... | 33 |
| <i>Figure II.3</i> : L'hyperplan H optimal, vecteurs supports et marge maximale...34 | |
| <i>Figure II.4</i> : meilleur hyperplan séparateur..... | 34 |
| <i>Figure II.5</i> : cas linéairement séparable et non linéairement séparable..... | 35 |
| <i>Figure II.6</i> : Transformation des données dans un espace de grande dimension | 36 |

| | |
|--|----|
| Figure II. 7 : SVM binaire..... | 37 |
| Figure II. 8 : SVM binaire à marge souple..... | 39 |
| Figure II. 9 : Étapes de construction d'un réseau bayésien..... | 43 |
| Figure III.1 : L'architecture générale de notre approche hybride..... | 58 |
| Figure III. 2 : La phase de fuzzification..... | 59 |
| Figure III. 3 : La structure de RB naïf..... | 61 |
| Figure III.3 : histogramme comparative des résultats de SVM, NaiveBayes, et l'Approche | 65 |
| Figure VI.1 : La fenêtre intitulée Weka GUI Chooser..... | 67 |
| Figure IV.2 : Le classifieur NaiveBayes..... | 68 |
| Figure IV.3 : Les résultats de classification par NaiveBayes..... | 69 |
| Figure IV.4 : La fenêtre du svm_learn..... | 70 |
| Figure IV.5 : l'optimisation et la création du modèle..... | 70 |
| Figure IV.6 : La fenêtre du svm_classify..... | 71 |
| Figure IV.7 : Les résultats de classification en fonction de précision, rappel et exactitude..... | 71 |
| Figure IV.8 : interface principale..... | 72 |
| Figure IV.9 : Interface SVM..... | 73 |
| Figure IV.10 : Interface NaiveBayes..... | 73 |
| Figure IV.11 : Interface Hybride..... | 74 |

Liste des tableaux :

| | |
|---|----|
| <i>Tableau III.1</i> : le prétraitement des données en entrées a le SVM..... | 60 |
| <i>Tableau III.2</i> : Les attributs de corpus appliquer sur NaiveBayes..... | 62 |
| <i>Tableau III.3</i> : Exemple d'échantillon du corpus..... | 63 |
| <i>Tableau III.4</i> : Résultats de test sur les différentes mesures..... | 64 |
| <i>Tableau IV.1</i> : La matrice de confusion pour NaiveBayes..... | 69 |
| <i>Tableau IV.2</i> : La matrice de confusion pour SVMLight..... | 72 |
| <i>Tableau IV.3</i> : La matrice de confusion pour le modèle hybride..... | 74 |

Résumé :

Afin d'assurer la mise en œuvre de la politique de sécurité, différents outils ont été développés, parmi ces outils on trouve les systèmes de détection d'intrusion (IDS). Un IDS représente tout outil, méthode et ressource qui nous aident à prévoir ou à identifier toute activité non autorisée dans un réseau.

L'évolution des systèmes de détection d'intrusion est passée par deux générations, la première génération ad hoc, cette génération à montrer beaucoup de limites par rapport au grand volume du trafic réseau. La deuxième génération a été proposée afin de traiter les problèmes de la première génération, ou les techniques de data mining ont été utilisées. Elle nous offre beaucoup d'avantage comme la capacité d'analyser un large volume de données, malgré la puissance et l'efficacité des techniques de data mining ; les systèmes de détection d'intrusion de la deuxième génération souffrent de certaines limites comme la nécessité de faire une mise à jour régulière, la nécessité de préparer les données d'apprentissage, la difficulté de détecter les nouvelles formes d'attaques... etc. Les systèmes de détection d'intrusion adaptatifs sont proposés afin de traiter ces limites.

Dans ce mémoire nous avons proposé une approche hybride d'un système de détection d'intrusion par la fusion de deux classificateurs, le premier les SVM et le deuxième les réseaux bayésiens naïfs, cette combinaison est basée sur la théorie des fonctions de croyance et la logique floue pour initialiser les masses de ses fonctions. Notre approche a présenté des meilleures performances par rapport aux deux classificateurs. Our approach has shown better performance compared to the two classifiers.

Mots clés : Système de détection d'intrusions, Approche comportementale, SVM, Réseaux Bayésiens, Fonction de croyance.

Abstract

To ensure the implementation of the security policy different tools have been developed, from these various tools of computer security, we found the intrusion detection systems (IDS). An IDS is any tool, method and resources that help us to predict or identify any unauthorized network activity.

The evolution of intrusion detection systems has passed through two generations, the first ad hoc generation, and this generation to show many limitations with the large volume of network traffic. The second generation has been proposed to deal with the problems of the first generation, or data mining technique is used. It offers us a lot of advantage as the ability to analyze large volumes of data, despite the power and effectiveness of data mining techniques; the second generation of intrusion detection systems suffer from certain limitations such as the need for regular updating, the need to prepare the training data, the difficulty of detecting new forms of attack ... etc. Adaptive Intrusion Detection systems are available to address these limitations.

In this paper, we proposed a hybrid approach of an intrusion detection system by the merger of two classifiers, the first SVM and the second the NaiveBayes networks, this combination is based on the theory of Dempster-Shafer. Our approach has shown better performance compared to the two classifiers.

Keywords: Intrusion Detection System, Behavioral Approach, SVM, NaiveBayes Networks, theory of Dempster-Shafer.

Chapitre I : Sécurité Informatique et Systèmes de Détection d'Intrusion (IDS)

Introduction:

Aujourd'hui, la sécurité est un enjeu majeur pour les entreprises ainsi que pour l'ensemble des acteurs qui l'entourent. Elle n'est plus confinée uniquement au rôle de l'informaticien. Sa finalité sur le long terme est de maintenir la confiance des utilisateurs et des clients. La finalité sur le moyen terme est la cohérence de l'ensemble du système d'information. Sur le court terme, l'objectif est que chacun ait accès aux informations dont il a besoin.

I. Sécurité informatique :

1. Définition:

La sécurité des systèmes d'information (**SSI**) est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité du système d'information [1].

La (**SSI**) repose sur trois propriétés fondamentales : *la confidentialité, l'intégrité et la disponibilité*. L'interprétation de ces trois aspects varie suivant le contexte dans lequel elles sont utilisées :

a. Confidentialité : La confidentialité est ainsi définie par (ISO) [3] comme : *“le fait de s'assurer que l'information est seulement accessible qu'aux entités dont l'accès est autorisé”*. Cette définition implique que l'information ne doit pas être accessible par certaines entités, mais doit être accessible par d'autres.

b. Intégrité : Les données doivent être celles que l'on attend, et ne doivent pas être altérées de façon fortuite, illicite ou malveillante. En clair, les éléments considérés doivent être exacts et complets.

c. Disponibilité : Le système doit fonctionner sans faille durant les plages d'utilisation prévues et garantir l'accès aux services et ressources installées avec le temps de réponse attendu.

D'autres aspects peuvent aussi être considérés comme des objectifs de la sécurité des systèmes d'information, tels que :

- *La traçabilité* (ou « **Preuve** ») : garantie que les accès et tentatives d'accès aux éléments considérés sont tracés et que ces traces sont conservées et exploitables.
- *L'authentification* : L'identification des utilisateurs est fondamentale pour gérer les accès aux espaces de travail pertinents et maintenir la confiance dans les relations d'échange.
- *La non-répudiation et l'imputation* : Aucun utilisateur ne doit pouvoir contester les opérations qu'il a réalisées dans le cadre de ses actions autorisées, et aucun tiers ne doit pouvoir s'attribuer les actions d'un autre utilisateur.

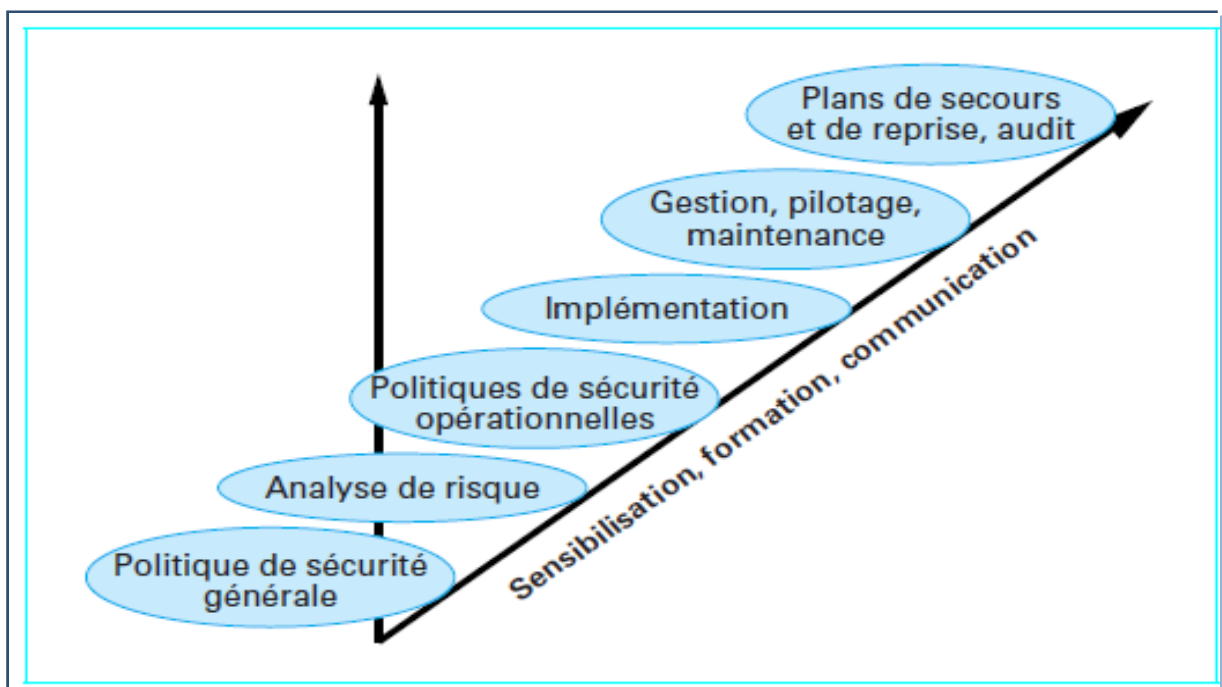


Figure I.1 : Organisation de la sécurité informatique.

La *figure 1* représente les différentes phases de l'organisation de la sécurité informatique :

a. Sensibilisation : cette phase est plus ou moins indépendante de la sécurité informatique, elle base sur la politique générale de la sécurité et l'analyse des risques.

b. Formation : dans cette phase on traite le côté informatique de la sécurité tel que les politiques de sécurité opérationnelles et l'implémentation de ces politiques.

c. Communication : dans la dernière phase, on assure la maintenance, la gestion, et les plans de secours pour notre (SSI).

2. Les objectifs de la sécurité informatique :

Les objectifs de la sécurité informatique sont liés aux types de menaces ainsi qu'aux types de ressources, etc... Néanmoins, les principaux points sont les suivants :

- d'empêcher des personnes non autorisées d'agir sur le système de façon malveillante.
- d'empêcher les utilisateurs d'effectuer des opérations involontaires capables de nuire au système.
- de sécuriser les données en prévoyant les pannes.
- de garantir la non-interruption d'un service.

3. Les champs d'application de la sécurité informatique :

Ces objectifs s'appliquent dans différents domaines ou champs d'applications, chacun faisant appel à des techniques différentes pour atteindre le ou les mêmes objectifs; d'un point de vue organisationnel, nous pouvons distinguer les sous-domaines de la sécurité informatique suivants [3] :

- **Sécurité logicielle :** gère la sécurité au niveau logiciel du système d'information (par exemple : l'intégration des protections logicielles comme l'antivirus).
- **Sécurité du personnel :** comprend la formation et la sensibilisation des personnes utilisant ou travaillant avec le système d'information.
- **Sécurité physique :** regroupe la politique d'accès aux bâtiments, la politique d'accès aux matériels informatiques, et les règles de sécurité pour la protection des équipements réseaux.
- **Sécurité procédurale :** définit les procédures et les règles d'utilisation du système d'information.

- **Sécurité réseau** : Elle est d'un niveau de que l'ensemble des machines du réseau fonctionnent de façon optimale et que les utilisateurs desdites machines possèdent uniquement les droits qui leur ont été octroyés.

Elle s'occupe de l'architecture physique et logique du réseau, la politique d'accès aux différents services, la gestion des flux d'informations sur les réseaux, et surtout les points de contrôle et de surveillance du réseau [5].

- **Veille technologique** : s'occupe du suivi des dernières mises à jour et failles des systèmes d'exploitation et des applications à partir des bulletins de sécurité des éditeurs, des forums, et des organisations de la sécurité informatique [6].

Elle permet d'évaluer la sécurité au cours du temps afin de maintenir un niveau suffisant de protection des systèmes d'information.

4. Terminologie de la sécurité informatique :

La sécurité informatique utilise un vocabulaire bien défini que nous utilisons dans notre chapitre. De manière à bien le comprendre, il est nécessaire de définir certains termes :

- *Les vulnérabilités* : ce sont les failles de sécurité dans un ou plusieurs systèmes. Tout système vu dans sa globalité présente des vulnérabilités, qui peuvent être exploitables ou non.
- *Les attaques (exploits)*: elles représentent les moyens d'exploiter une vulnérabilité. Il peut y avoir plusieurs attaques pour une même vulnérabilité mais toutes les vulnérabilités ne sont pas exploitables.
- *Les contre-mesures* : ce sont les procédures ou techniques permettant de résoudre une vulnérabilité ou de contrer une attaque spécifique (auquel cas il peut exister d'autres attaques sur la même vulnérabilité).
- *Les menaces* : ce sont des adversaires déterminés capables de monter une attaque exploitant une vulnérabilité.
- *Risque* : Couple (menace, vulnérabilité).

Dans ce chapitre, on va détailler ensemble les termes suivants : « vulnérabilités », « menaces » et « attaques ».

4.1. Vulnérabilité :

C'est une faiblesse, une faille dans les mesures de protection ou encore dans l'absence de mesures de protection [4] et de contrôles (physique ou autres) qui peuvent être exploitées par une menace. Elles sont souvent interprétées par l'absence de mesure de protection [7].

Une vulnérabilité est difficile à détecter, même pour les spécialistes. Il existe des organismes spécialisés pour l'identification des vulnérabilités [8]. Des listes de vulnérabilités classées par domaine (organisationnel, matériel,...etc.) existent, tels que, les listes des vulnérabilités des systèmes d'exploitation proposées par les CERT [9], le CSI [10] et bien d'autres.

Les vulnérabilités existent dans le matériel et dans le logiciel, dans les règles et dans les procédures et aussi parmi le personnel. Tout ce qui peut être exploité pour obtenir un avantage non accordé est une vulnérabilité. Nous proposons les quelques une des causes de vulnérabilités les plus courantes [8] [11] :

- Vulnérabilité de matériel : disque dur, périphériques,...etc.
- Vulnérabilité de logiciel : les systèmes d'exploitation et les applications.
- Vulnérabilité d'infrastructure : réseau de communication hors service.
- Vulnérabilité des processus de contrôle : les règles de sécurité sont mal interprétées ou mal implémentées.

4.2. Menace :

Une menace est une source de danger pour le système et se traduit par la présence d'une violation potentielle de la sécurité [4] [7]. Cela peut être une personne, une chose, un événement ou une idée qui constitue un danger à un patrimoine en termes de confidentialité, d'intégrité, de disponibilité et d'utilisation approuvée du système.

Les menaces peuvent être classées en deux catégories : « accidentelles » et « intentionnelles » [8] [4] comme montre la *figure2* :

a. Les menaces accidentelles (ou non intentionnelles) : qui peuvent être réalisées par une exposition ou une modification des informations. Par exemple :

- L'erreur humaine, ce type d'erreur est de loin la menace la plus répandue contre les ressources d'un système d'information, ce sont les utilisateurs autorisés commettant des erreurs susceptibles de causer des pertes.
- Panne du système informatique, le système informatique comprend du matériel, du logiciel et une infrastructure, et ces composants sont toutes susceptibles de pannes à des degrés divers...etc.

b. Menaces intentionnelles : qui correspond aux attaques dont le but est de violer la sécurité du système. Par exemple :

- Actes de malveillance : les actes de malveillances sont le fait d'individus ou de groupe d'individus qui visent tel ou tel système particulier, exemple les hackers, des espions industriels,...etc.
- Logiciels malveillants : nous désignons sous ce nom des logiciels créant ou exploitant une vulnérabilité. Ce sont des outils pouvant être utilisés de façon constructive ou destructive. Par exemple dépassement de capacité de buffer, les bombes logiques, parasite, sniffer, spoof, cheval de Troie, virus, vers, ... etc.

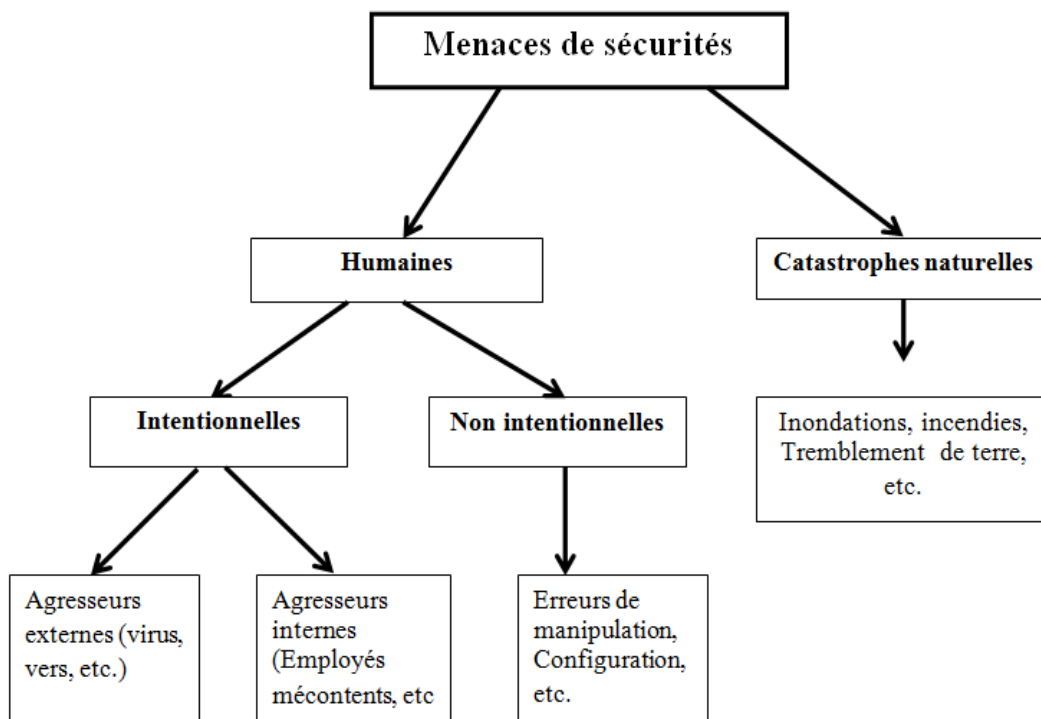


Figure I. 2 : Typologie des menaces.

4.3. Attaque :

C'est l'action entreprise par un objet (individu ou programme) pour modifier l'état d'un système [12] [7] [4]. Une attaque peut aboutir en exploitant les vulnérabilités du système, elle concrétise une menace. Elle peut être directe auquel cas elle s'adresse à la ressource ciblée ou indirecte où elle obtient des informations d'une autre ressource sans attaquer la ressource ciblée directement [4].

- **Classes d'attaques :**

- ***Attaque passive:*** elle consiste à observer le système pendant son exploitation et collecter des informations, par exemple l'analyse et la surveillance des communications non protégée et la capture d'information d'authentification (telle que des mots de passe). Ce type d'attaque est difficile à détecter.
- ***Attaques sur la confidentialité :*** obtention d'informations sur un système, sur un utilisateur ou un projet.

Méthodes possibles :

- Ecoute
- Injection de code
- Usurpation d'identité
- Intrusion
- Abus de droits

- ***Attaque active :*** elle change le comportement du système on modifie les sources ciblées par l'attaque (atteinte aux critères d'intégrité, de disponibilité, de confidentialité).
- ***Attaques sur l'intégrité:*** modification ou destruction de données ou de configurations.

Méthodes possibles :

- Injection de code
- Action physique
- Intrusion

- ***Attaques sur l'authentification :*** utilisation des ressources de façon clandestine sur un système.

Méthodes possibles :

- Abus de droits.
- Intrusion.

- **Attaques sur la disponibilité** : perturbation d'un échange par le réseau, d'un service ou d'un accès à un service.

Méthodes possibles :

- Abus de droits.
 - Action physique.
 - Intrusion.
- **Structure d'une attaque :**

Une attaque est souvent décrite à l'aide des 5 "p" [13]:

➤ **Probe (Analyser) :**

C'est la collecte d'informations sur le système cible, elle peut s'effectuer de plusieurs manières. Comme par exemple un scan des ports grâce au programme *Nmap* pour déterminer la version des logiciels utilisés, et des outils comme *firewalk*, *hping* ou *SNMP Walk* permettent quant à eux de découvrir la nature d'un réseau.

➤ **Penetrate (Pénétrer) :**

C'est l'utilisation des informations récoltées pour pénétrer un réseau. Des techniques comme le brute force ou les attaques par dictionnaires peuvent être utilisées pour outrepasser les protections par mot de passe.

➤ **Persist (Peréniser) :**

C'est la création d'un compte avec les droits de super utilisateur pour pouvoir se ré-infiltrer ultérieurement. Une autre technique consiste à installer une application de contrôle à distance capable de résister à un reboot.

➤ **Propagate (Propager) :**

Cette étape consiste à observer ce qui est accessible et disponible sur le réseau local.

- **Paralyse (Paralyser) :** Cette étape peut consister en plusieurs actions. Le pirate peut utiliser le serveur pour mener une attaque sur une autre machine, détruire des données ou encore endommager le système d'exploitation dans le but de planter le serveur.

Types d'attaques [14] [2] :

Actuellement, il existe plusieurs types d'attaques, le pirate peut lancer ensemble d'attaques afin de récolter des informations, mais aussi de pénétrer un réseau.

Ces attaques peuvent être regroupées en trois familles différentes :

➤ Les attaques directes :

C'est le plus simple type d'attaques, le pirate attaque directement sa victime à partir de son ordinateur comme illustre la *figure3* :

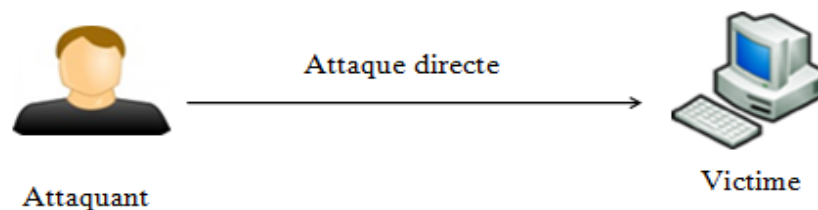


Figure I. 3 : Attaque directe.

L'inconvénient de ce type d'attaque qu'il est facile de remonter jusqu'à l'attaquant.

➤ Les attaques indirectes par rebond :

Ce type d'attaque est très prisé par les attaquants, son principe en lui-même est très simple : les paquets d'attaqués sont envoyés à l'ordinateur intermédiaire, qui répercute l'attaque vers la victime, d'où le terme *rebond*.

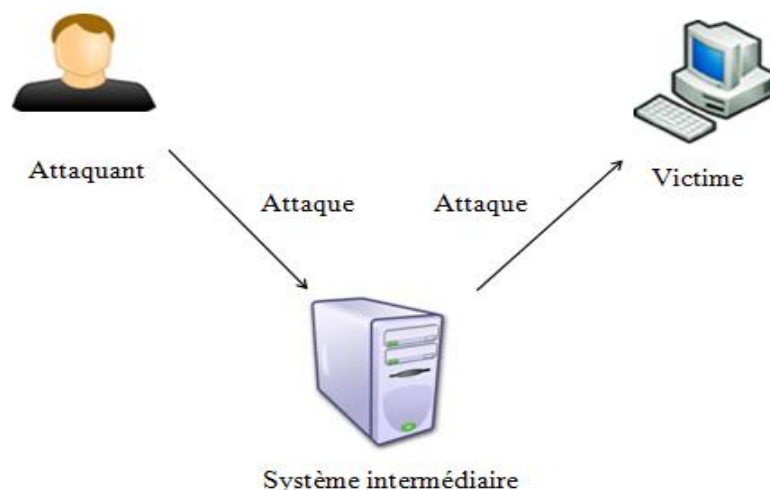


Figure I.4 : Attaque indirecte par rebond.

En effet, le rebond a des avantages :

- Masquer l'identité (adresse IP) de l'attaquant ;
- Utiliser les ressources de l'ordinateur intermédiaire à cause de son puissance (CPU, Bande passante,...) pour attaquer.
- Il est difficile de remonter jusqu'à l'attaquant.

➤ **Les attaques indirectes par réponse :**

Certaines attaques, dites indirectes par réponse, offrent au pirate les mêmes avantages que les attaques par rebond. Au lieu d'envoyer l'attaque au système intermédiaire pour qu'il la répercute, l'attaquant lui envoie une requête, et c'est la réponse à cette requête qui est envoyée au système cible, comme l'illustre la *figure5* :

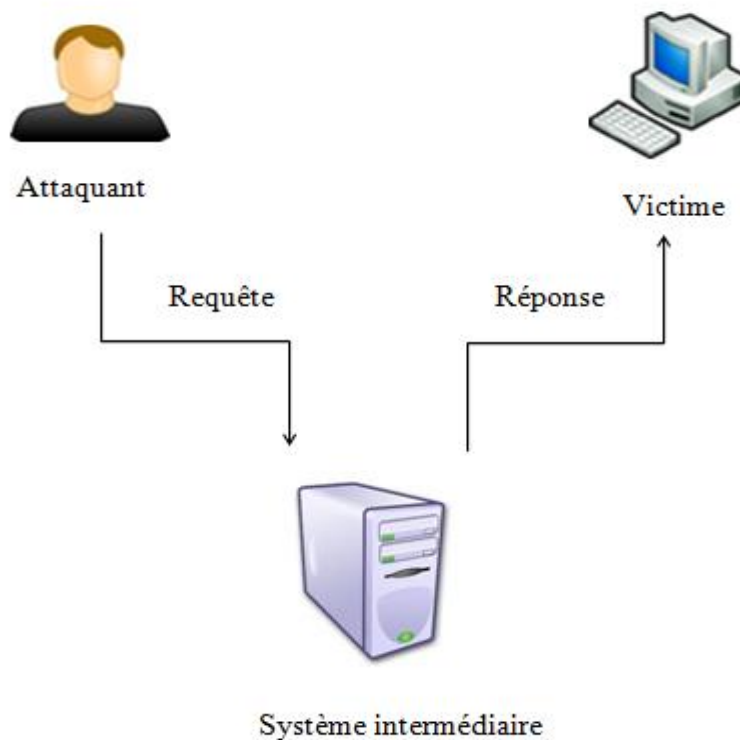


Figure I. 5 : Attaque indirecte par réponse.

Là aussi après l'attaque, il est difficile de remonter jusqu'à l'attaquant. Les attaques réseau sont aujourd'hui si nombreuses qu'il serait illusoire de prétendre les décrire toutes. Il est cependant possible de dresser une typologie des faiblesses de sécurité afin de mieux appréhender ces attaques, qui ont pour point commun d'exploiter des faiblesses de sécurité [2].

Comme tout effet a une cause, les attaques réseau s'appuient sur divers types de faiblesses, que l'on peut classer par catégorie, comme illustré à la *figure6* [15].

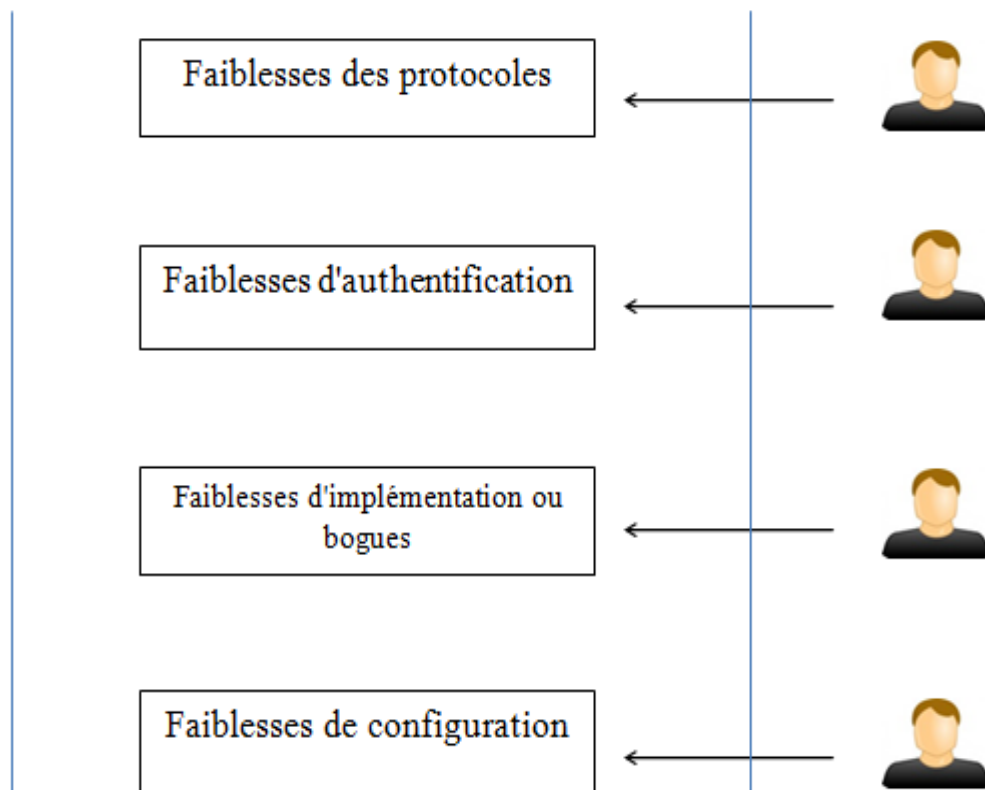


Figure I. 6: Typologie des faiblesses de sécurité.

4.3.1. Les attaques sur les protocoles réseau :

1. Attaques permettant d'établir la cartographie du réseau :

Les attaques visant à établir la cartographie d'un réseau ont pour but de dresser les artères de communication des futurs systèmes cibles. Elles ont recours pour cela à des outils de diagnostic tel que « Traceroute », qui permet de visualiser le chemin suivi par un paquet IP d'un hôte à un autre [2].

2. Attaques permettant d'identifier les systèmes réseau (scanning) :

Certaines attaques visent à identifier un système dans le but de dresser les futurs moyens de pénétration de ce système [2].

En dehors des outils classiques de découverte de services, les outils de scanning permettent de réaliser des prises d'empreinte des systèmes cibles.

Il existe différentes techniques de balayage des systèmes comme illustré à la *figure 7* :

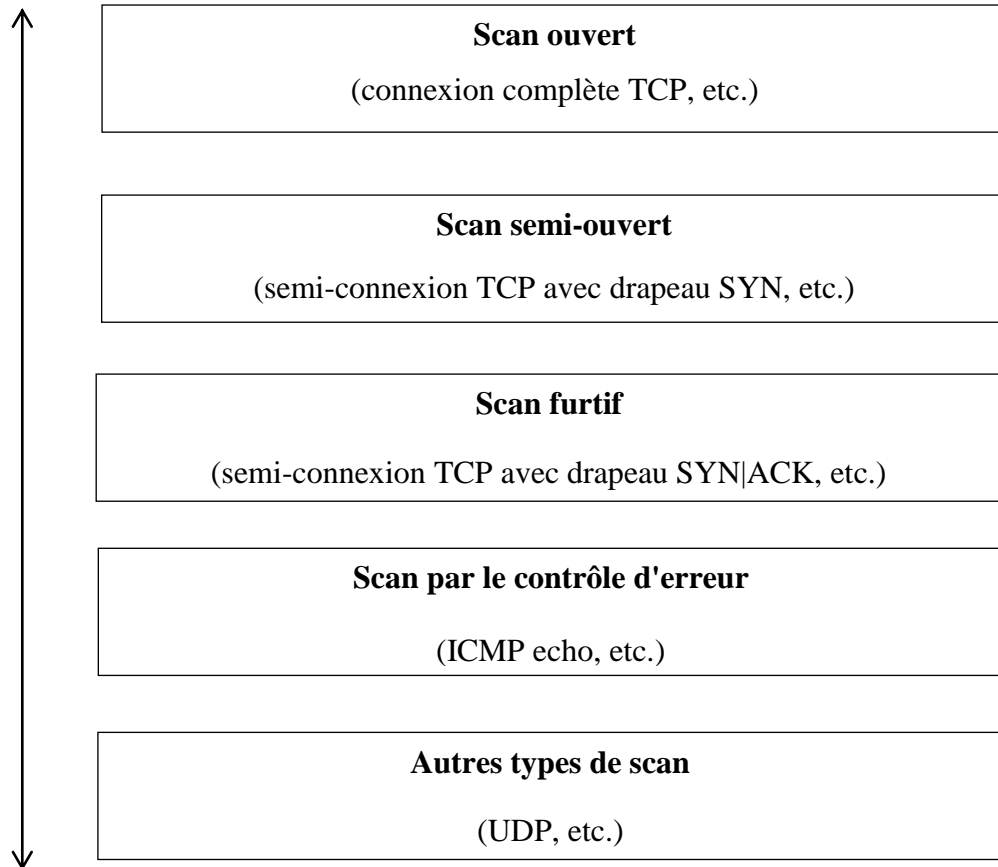


Figure I. 7 : Les différents types de scanning.

3. Attaques permettant d'écouter le trafic réseau (sniffing) :

L'attaque par sniffing est généralement utilisée par les pirates pour capturer les mots de passe. Lorsqu'on se connecte à un réseau qui utilise le mode broadcast, toutes les données en transit arrivent à toutes les cartes réseau connectées à ce réseau [16]. En temps normal, seules les trames destinées à la machine sont lues, les autres étant ignorées.

Grâce à un sniffer, il est possible d'intercepter les trames reçues par la carte réseau d'un système pirate et qui ne lui sont pas destinées. Le système pirate se situe donc sur le réseau local et capture tous les paquets réseau transitant sur ce réseau afin d'obtenir des mots de passe.

4. Attaques sur la fragmentation des paquets IP :

Les attaques par fragmentation ont été les premières attaques à passer au travers des éléments de filtrage IP réalisés par les pare-feu [2].

- **L'attaque par Tiny Fragments :** consiste à fragmenter sur deux paquets IP une demande de connexion TCP ou d'autres demandes sur une machine cible tout en traversant et en déjouant, par le mécanisme de fragmentation, un filtrage IP [14].
- **L'attaque par Fragment Overlapping :** consiste à fragmenter deux paquets IP au moyen de l'option Overlapping pour faire une demande de connexion TCP ou une autre demande sur une machine cible tout en traversant un filtrage IP [17].

5. Attaques par déni de service et par inondation (DoS)

Le déni de service est une attaque qui vise à rendre indisponible un service, un système ou un réseau. Ces attaques se basent généralement soit sur une faiblesse d'implémentation ou bogue, soit sur une faiblesse d'un protocole [2].

Les premières attaques par déni de service sont apparues entre 1998 et l'an 2000 et visaient de grands sites Internet (Yahoo, Ebay, eTrade, etc.). Concernant le site Yahoo, ce site a été attaqué en février 2000 et a été "noyé" (flood) sous un gigabyte de données en quelques secondes pendant plus de 3 heures d'au moins 50 points réseau différents [18].

- **L'inondation :** généralement l'inondation est la méthode la plus classique pour empêcher un réseau d'assurer sa mission. Son principe de fonctionnement est simple, une ou plusieurs machines inondent le réseau avec des paquets réseau afin de saturer la bande passante de celui-ci. Une fois que toute la bande est occupée, les autres machines ne peuvent plus travailler, ce qui génère une situation de refus de service. [2] [16].
- **L'attaque par smurf-and-fraggle :** est une variante de la précédente qui s'appuie sur une faiblesse de configuration des routeurs. Cette technique consiste à inonder le réseau avec des ping qui n'utilisent que des adresses de broadcast [14].
- **L'attaque DDoS (Distributed Denial Of Services) :** est un dérivé de la précédente sous une forme distribuée comme illustré à la figure suivante :

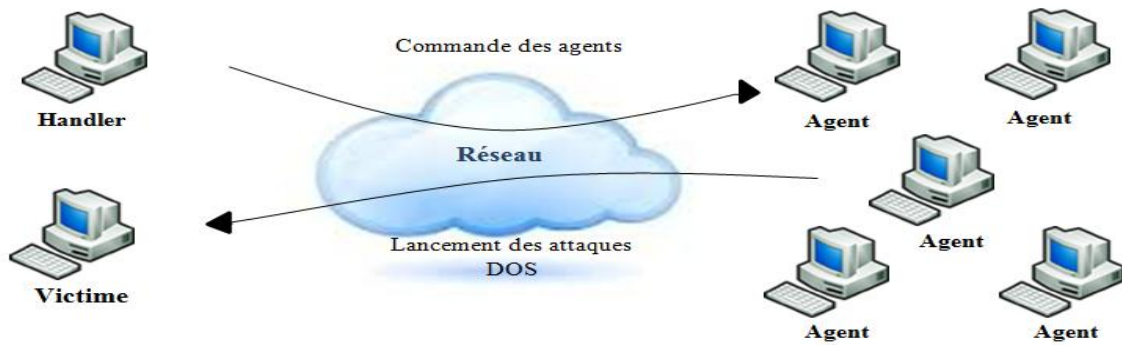


Figure I. 8 : Attaque par déni de service distribué.

- La première étape consiste à pénétrer par diverses méthodes des systèmes dits handlers, ou maîtres (masters), et agents, ou esclaves (slaves).
- Deuxièmement le pirate contrôle ensuite directement un ensemble de systèmes handlers, qui contrôlent eux-mêmes un ensemble de systèmes agents.
- La dernière étape consiste pour le pirate à déclencher son attaque vers un ou plusieurs systèmes cibles donnés. Cet ordre d'attaque aura été donné par les systèmes handlers, qui auront eux-mêmes reçu cet ordre du pirate.

4.3.2. Les attaques s'appuyant sur les faiblesses d'authentification :

1. L'attaque IP spoofing :

L'attaque IP spoofing consiste à se faire passer pour un autre système en falsifiant son adresse IP [18]. Le pirate commence par choisir le système qu'il veut attaquer. Après avoir obtenu le maximum de détails sur ce système cible, il détermine les systèmes ou adresses IP autorisés à se connecter au système cible.

2. L'attaque man-in-the-middle:

L'attaque man-in-the-middle consiste à faire passer les échanges réseau entre deux systèmes par le biais d'un troisième, sous le contrôle du pirate [18] [2]. Ce dernier peut transformer à sa guise les données à la volée, tout en masquant parfaitement à chaque acteur de l'échange la réalité de son interlocuteur.

3. Attaques de déchiffrement et de pénétration des systèmes par mots de passe :

La plupart des protocoles et services réseau associés utilisent une procédure d'authentification fondée sur un couple (compte, mot de passe). Des attaques itératives de pénétration, dites Brute Force Attack [16], par le biais de séquences de tentatives d'authentification sur des comptes et des mots de passe différents peuvent se révéler redoutables pour peu qu'elles s'étalent dans le temps afin de laisser le moins de traces possibles.

4.3.3. Les attaques s'appuyant sur les faiblesses d'implémentation :

1. L'attaque TCP SYN :

La technique d'inondation SYN, ou SYN flooding, n'est pas une inondation simple [2]. Elle s'appuie sur une demande de connexion qui n'aboutit pas et qui sature les ressources du système visé.

Pour gérer les états de connexion entre deux parties, le protocole TCP recourt aux drapeaux URG, ACK, PUSH, RST, SYN et FIN présents dans l'en-tête TCP [16].

2. Les faiblesses du code :

Les piles IP/TCP développées par différents constructeurs ou fournisseurs de services manifestent des différences de comportement malgré les définitions des RFC et contiennent de multiples faiblesses, qui peuvent être exploitées par des attaques bien ciblées. [16]

Comme il est théoriquement impossible de vérifier l'absence de bogues dans un programme conçu avec les langages de programmation modernes, il existe une forte probabilité que des bogues permettent à des pirates de gagner des privilèges.

Les principales attaques qui s'appuient sur les erreurs de programmation associées aux piles TCP/IP sont :

- **L'attaque "ping de la mort"**
- **L'attaque "baiser de la mort"**
- **L'attaque "winnuke"**

- **L'attaque "land"**
- **L'attaque "teardrop"**

3. Attaques sur les bogues des systèmes d'exploitation :

Les systèmes d'exploitation et les produits ou services additionnels qui y sont greffés contiennent de multiples faiblesses susceptibles d'être exploitées par des attaques ciblées [2].

- **le buffer overflow** (ou débordement de tampon) :

C'est la principale de ces attaques. Elle consiste à copier plus de données dans un tampon que celui-ci ne peut en contenir. Si les contrôles ne sont pas suffisants, le débordement du tampon permet d'écrire dans la pile d'exécution du programme.

4.3.4. Les faiblesses de configuration :

Les faiblesses de configuration des équipements réseau, pare-feu, etc., sont également souvent utilisées pour mener à bien des attaques. Ces dernières peuvent provenir de :

- L'exploitation d'erreurs de configuration du système.
- Des configurations des équipements réseau, qui doivent suivre des règles strictes afin d'éviter que le réseau ne joue, un rôle de rebond dans des attaques éventuelles, notamment l'attaque smurf.
- D'une politique d'accès ou de mots de passe trop laxiste.
- De comptes utilisateurs génériques, standardisés avec des mots de passe triviaux et associés à des droits d'accès permissifs. Dès lors, un pirate peut commencer son intrusion non pas par la recherche de failles exploitables, mais simplement par des tentatives itératives de pénétration.

Celles-ci peuvent commencer par les comptes classiques comme oracle, admin, toor, sybase, solaris, linux, etc. et avec des mots de passe identiques aux noms des comptes.

5. Les possibilités en matière de sécurité réseaux :

Il existe plusieurs possibilités de protection d'un réseau informatique ou d'un système d'information.

Chacune de ces techniques se base sur des principes fondamentalement différents, mais celles-ci ont un but commun : permettre une connexion entre Internet (réseau non sécurisé) et le réseau de l'entreprise concernée, en assurant la sécurité des équipements et des informations disponibles sur ce réseau, tout en tenant compte des contraintes de plus en plus présentes, telles que les interconnexions de réseaux, les besoins de « contacts électroniques » pour le personnel (mails, transferts de fichiers, accès Web, etc.), les systèmes d'informations complexes, et autres [15] :

5.1. Les Firewalls :(Pare-feu) est un matériel ou logiciel permettant de réaliser l'isolement, le masquage des ressources le filtrage de données, le contrôle de flux, contribuant à la protection des événements informatiques privés dans un réseau ou sur une machine.

Les pare-feu sont généralement placés aux interfaces, entre le réseau et l'extérieur.

5.2. Les filtres de paquets : Un filtre de paquets permet de filtrer les paquets circulant sur un réseau. Plus précisément, on peut même dire que le filtrage s'effectue sur les paquets traversant une interface réseau. Celui-ci fonctionne en analysant le contenu de ces paquets, après qu'on fait établir une série de règles de filtrage qui reflète la politique de sécurité de l'entreprise. Les paquets ne satisfaisant pas aux règles de filtrage seront alors bloqués (supprimés).

5.3. Les systèmes de détection d'intrusions : sont des mécanismes destinés à réparer des activités anormales ou suspectes sur un réseau ou une machine.

Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées d'instruction. Dans la deuxième partie de ce chapitre on va détailler ensemble les **IDS**.

II. Les Systèmes de détection d'intrusions :

Les systèmes de détection d'intrusions (**Intrusion Detection System**) ont pour objectif de révéler, généralement via des alertes, toute activité pouvant être considérée comme intrusive, depuis ou vers un système d'information, par analyse de données. Les sources de ces données correspondent à des événements générés par différents services ou utilisateurs. Les premiers travaux en détection d'intrusions ont débuté avec Anderson [19] en 1980 et Denning [21] en 1987. Aujourd'hui, il existe plus de 140 systèmes de détection d'intrusions [22].

. 1. Définitions :

1.1. Intrusion : Faute opérationnelle, externe, intentionnellement nuisible, résultant de l'exploitation d'une vulnérabilité dans le système.

Les intrusions sont provoquées par : l'accès d'attaquants externes aux systèmes via des réseaux ouverts comme Internet, des utilisateurs autorisés qui essayent de gagner des privilèges additionnels pour lesquels ils ne sont pas autorisés, ou des utilisateurs autorisés qui abusent de leurs privilèges [19].

Une intrusion peut causer des dommages graves à la victime :

- Dommages matériels
- Dommages financiers
- Dommages psychologiques

1.2. Système de détection d'intrusion :

Les systèmes de détection d'intrusions (SDI) sont les systèmes logiciels ou matériels qui automatisent la tâche de surveillance et d'analyse [20].

Ces utilitaires permettent de détecter une attaque et de vous en informer. Un IDS analyse tout ce qui se passe sur une station. Il détecte les débordements de droits (obtention du compte root d'une manière suspecte) et d'autres types d'attaques, il contient une base de données sur différentes vulnérabilités.

La *figure9* illustre, de manière simplifiée, les différents modules composant un système de détection d'intrusions selon la normalisation proposée par Intrusion Detection Working Group [24]

Cette architecture est composée de 3 modules communs à la majorité des IDS.

L'Activité du système d'information fournit une source de données à des **Capteurs**. Ces Capteurs ont alors pour rôle d'extraire et de transformer certaines informations afin de les transmettre sous forme d'événements à un **Analyseur**. Le module d'analyse utilise alors ces événements afin de déceler une possible intrusion et génère en conséquence des alertes. Ces alertes sont finalement envoyées à un gestionnaire d'alertes (**Manager**). Ce dernier est chargé de traiter les

alertes émanant des différents analyseurs et de notifier toute activité suspecte sur le système d'information à l'opérateur de sécurité.

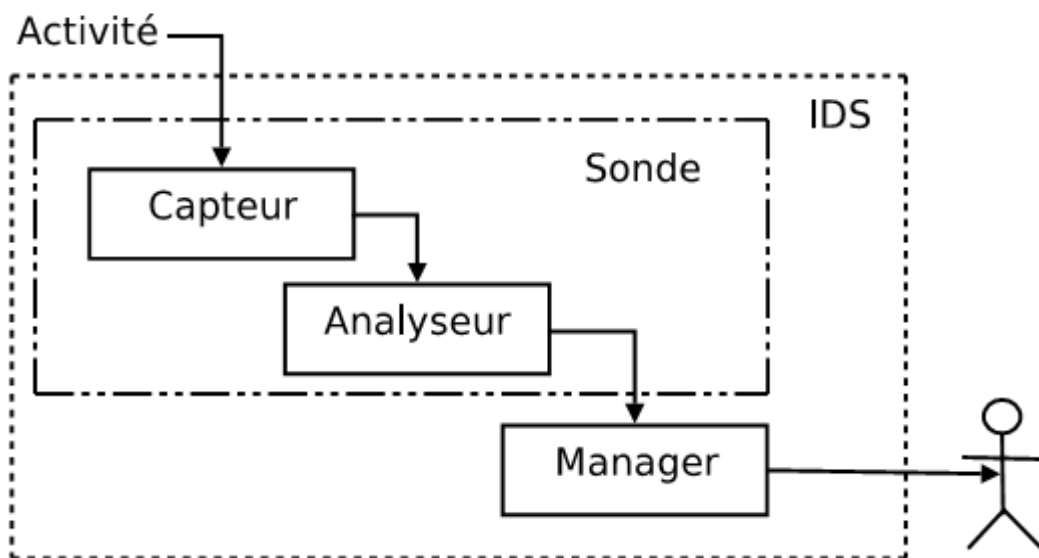


Figure I. 9 : Architecture d'un système de détection d'intrusions.

2. Les familles de Système de détection d'intrusion :

Les systèmes de détection d'intrusions sont généralement classés en deux grandes catégories suivant le type de données à analyser [23] : les systèmes de détection d'intrusions système (**HIDS** pour **H**ost-based **I**DS) et les systèmes de détection d'intrusions réseau (**NIDS** pour **N**etwork-based **I**DS).

2.1. Système de détection d'intrusion réseau NIDS :

Un NIDS (Network Intrusion Detection System) travaille de la même manière qu'un IDS, mais sur les données transitant sur le réseau. Il peut détecter en temps réel une attaque s'effectuant sur l'une des machines du réseau. Il contient une base de données avec tous les codes malicieux et peut détecter leurs envois sur une des machines [25]. Les NIDS place une ou plusieurs cartes d'interfaces réseau du système dédié en mode promiscuité « *promiscuous mode* », elles sont alors en mode « furtif » afin qu'elles n'aient pas d'adresse IP.

Les NIDS n'est pas visible et n'affecte pas les performances du réseau mais ils sont un unique de défaillance et très faible devant les attaques de dénis de services

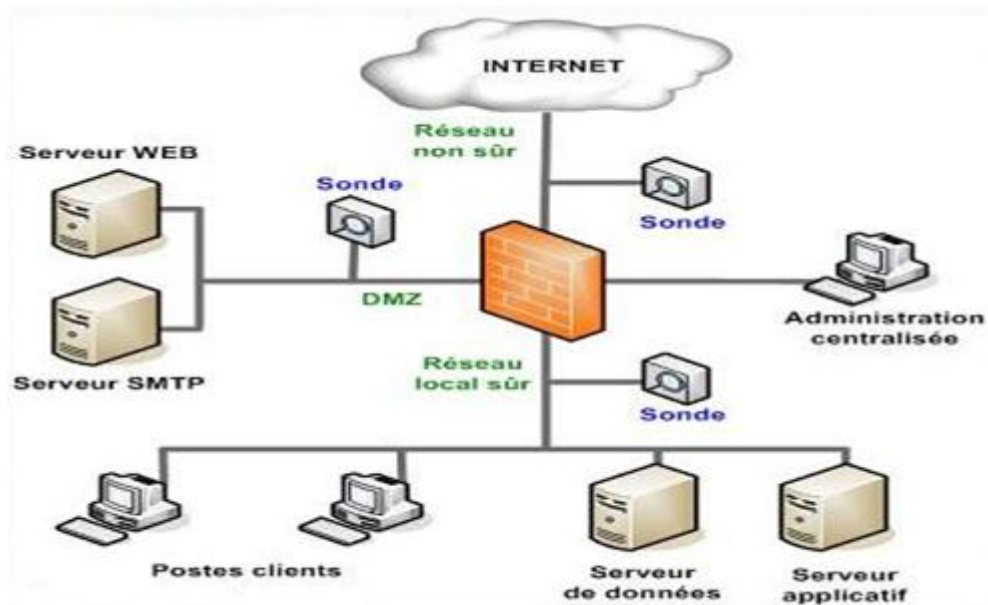


Figure I. 10 : système de détection d'intrusion réseau NIDS.

2.2. La détection d'Intrusion basée sur l'hôte HIDS :

L'IDS Systèmes ou Host Based IDS (HIDS) surveille le trafic sur une seule machine.

Il analyse les journaux systèmes, les appels systèmes et enfin vérifie l'intégrité des systèmes de fichiers. Les HIDS sont de part leur principe de fonctionnement dépendant du système sur lequel ils sont installés. Ce système peut s'appuyer ou non sur le système propre au système d'exploitation pour en vérifier l'intégrité et générer des alertes [29].

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

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

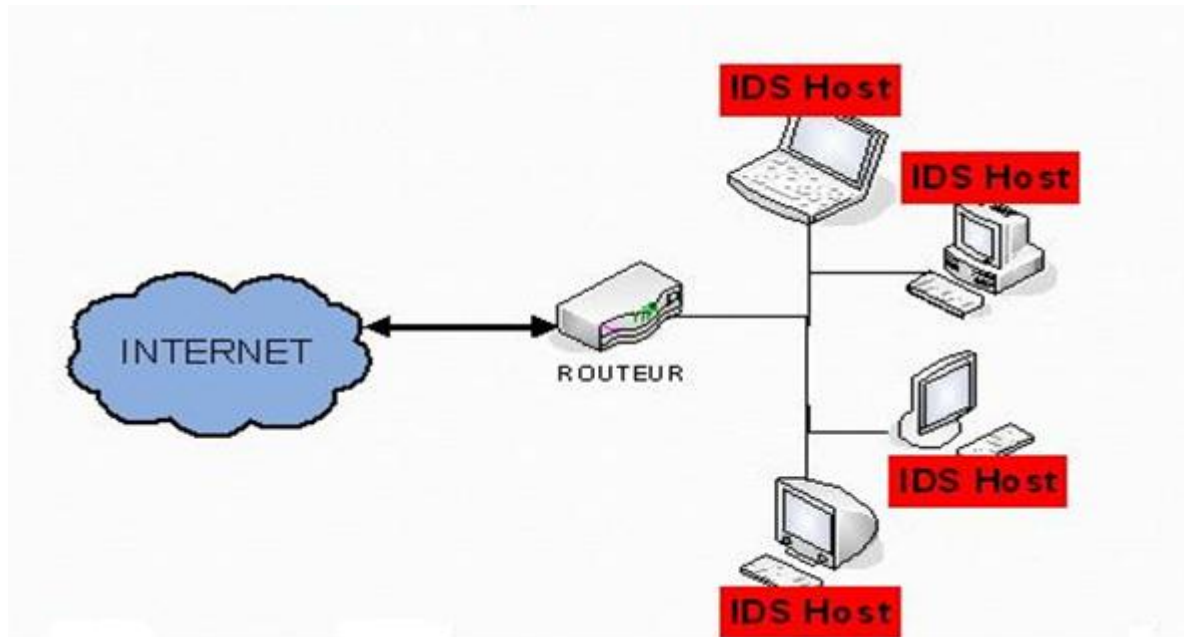


Figure I. 11 : Système de détection d'Intrusion HIDS.

Les HIDS présente comme avantages que vous constatez l'impact d'une attaque et pouvez donc mieux réagir, vous pouvez détecter des attaques impossibles à détecter avec des IDS réseau puisque le trafic y est souvent crypté, et vous pouvez observer les activités sur l'hôte avec précision mais ils ont des inconvénients car ils sont plus vulnérables aux attaques de type DoS, l'analyse des traces d'audit du système est très contraignante en raison de la taille de ces dernières et ils consomment beaucoup de ressources CPU.

L'utilisation de ces deux types d'IDS seulement peut montrer inefficace, d'où vient d'autre famille d'IDS tel que l'**IDS de Nœud Réseau** et **IDS Hybride** :

2.3. Système de Détection d'Intrusion de Noeud Réseau (NNIDS) :

Ce nouveau type (NNIDS) fonctionne comme les NIDS classiques, c'est-à-dire on analyse les paquets du trafic réseau. Mais ceci ne concerne que les paquets destinés à un noeud du réseau (d'où le nom). Une autre différence entre NNIDS et NIDS vient de ce que le NIDS fonctionne en mode "promiscuous", ce qui n'est pas le cas du NNIDS. Puisque tous les paquets ne sont pas analysés, les performances de l'ensemble sont améliorées. [26]

2.4. Système de Détection d'Intrusion Hybride :

D'où son nom Hybride, l'IDS Hybride rassemble les caractéristiques des NIDS et HIDS.

Il permet de surveiller le réseau et les terminaux à la fois dans un seul outil. Les sondes sont placées en des points stratégiques et agissent comme un NIDS ou/et HIDS suivant leur emplacement. Toutes les sondes remontent alors des alertes à une machine qui va centraliser les données, et agréger ou lier des informations d'origines multiples.

Les avantages d'IDS Hybride sont multiples car ils ont une meilleure corrélation (la corrélation permet de générer de nouvelles alertes à partir de celles existantes) et il présente moins de faux positifs et ils offrent une possibilité de réaction aux analystes.

3. Concepts de base :

Quelques définitions à savoir avant de continuer :

-Alarme : c'est la réponse générée par le système de détection d'intrusions lors de la détection d'une intrusion. Cependant les erreurs de détection peuvent être classées selon deux types :

- *Le faux positif :* signifie qu'un système de détection d'intrusions signale une intrusion là où aucune intrusion réelle n'a été commise.
- *Le faux négatif :* À l'inverse de « positif faux », « négatif faux » signifie que le système de détection d'intrusions n'a pas détecté une intrusion ayant réussi.

- Sonde : est un composant de l'architecture IDS qui collecte les informations brutes (capteur et analyseur).

- Système : dénote un système d'information contrôlé par un système de détection d'intrusions. Cela peut être un poste de travail, un élément du réseau, une unité centrale, un pare-feu, un serveur Web, un réseau d'entreprise, ...etc.

4. Les critères pour évaluer l'efficacité des systèmes de détection d'intrusions :

Philip dans [27] 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 malicieuse une activité légitime. 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 [28] a rajouté également les deux critères suivants :

- **La tolérance aux fautes (Fault tolerance) :** le système de détection d'intrusion doit lui-même résisté aux attaques, particulièrement au déni de service. Ceci est important, parce que plusieurs systèmes de détection d'intrusion s'exécutent sur des matériels ou logiciels connus comme vulnérables aux attaques.
- **La réaction à temps (Timeliness) :** le système de détection d'intrusion doit s'exécuter et propager les résultats de l'analyse le plus tôt possible, pour permettre à l'officier de sécurité de réagir avant que 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.

5. Classification des systèmes de détection d'intrusions :

Actuellement, on trouve plusieurs systèmes de détection d'intrusions opérationnels.

Il est donc très utile d'utiliser des critères pour classer ces systèmes de détection d'intrusions qui seront présentés dans cette section. Les différents systèmes de détection d'intrusions disponibles peuvent être classés [27] selon plusieurs critères qui sont :

- La méthode d'analyse.
- Le comportement de la détection.
- Emplacement de la source d'audit.
- La fréquence d'utilisation.

La *figure 11* présente la classification des IDS selon leurs critères :

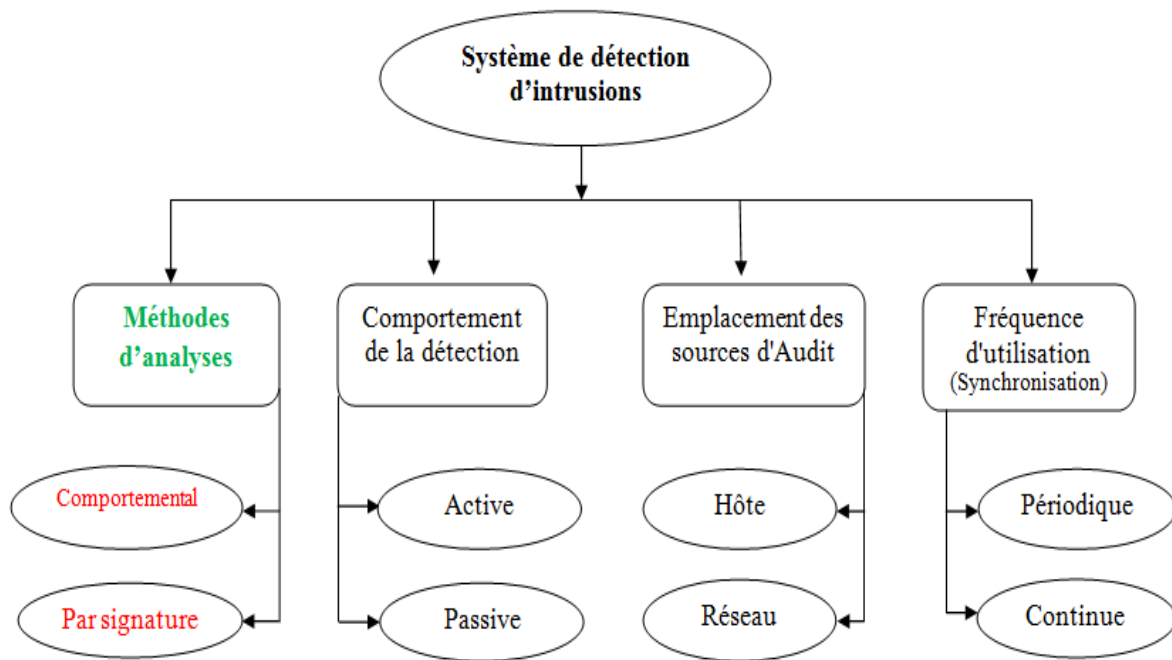


Figure I. 12 : Taxonomie des systèmes de détection d'intrusion.

5.1. La méthode d'analyse

Les systèmes de détection d'intrusions (IDS) sont classés en deux grandes catégories suivant leur approche d'analyse des données. Deux grandes approches ont été proposées dans la littérature [29]:

- la détection d'intrusions par signature.
- la détection d'intrusions comportementale.

- **Approche par scénario :**

Les systèmes à base de signatures qui consistent à rechercher dans l'activité de l'élément surveillé les signatures (empreintes) d'attaques répertoriées et donc connues.

Ce principe de détection d'intrusion est réactif et pose plusieurs contraintes, en effet il ne détecte que les attaques répertoriées dont il possède l'empreinte. De ce fait il nécessite des mises à jour fréquentes. Ce principe de détection implique aussi que les pirates peuvent contourner celui-ci en maquillant leurs attaques, il modifie en fait la signature connue par les IDS et de ce fait l'attaque devient invisible par l'IDS.

Il existe différentes méthodes pour repérer les attaques :

- *Analyse de motif*: la plus simple et la plus couramment utilisée pour détecter une intrusion. Une base de connaissance contient toutes les chaînes alphanumériques caractéristiques d'une intrusion.
- *Recherches génétiques* : Adaptée pour les virus. On regarde dans le code exécutable les commandes qui sont potentiellement dangereuses. Par exemple, une commande DOS non référencée est détectée, des émissions de mails, des instructions liées à des attaques connues.
- *Contrôle d'intégrité* : Effectue une photo de tous les fichiers d'un système et génère une alerte en cas d'altération de l'un des fichiers. Aujourd'hui l'exemple le plus connu utilisant cette approche est l'IDS SNORT.

Cette approche fournit un diagnostic clair, il est donc possible de réagir et de contre attaquer, mais ils ne peuvent détecter que les attaques contenues dans la base de connaissances. Il faut en permanence maintenir à jour cette base. Il est possible de rendre inactif un IDS utilisant cette approche par une attaque en déni de service.

- **Approche comportementale :**

Les systèmes à approche comportementale consistent à détecter les différentes anomalies sur le réseau. C'est l'administrateur qui définira le fonctionnement "normal" des éléments surveillés. Par la suite l'IDS sera en mesure de signaler à l'administrateur toute situation qui divergera du niveau de fonctionnement de référence qui peut être élaboré par différentes analyses statistiques de l'élément à surveiller.

Ce système de détection présente un avantage par rapport au précédent : il détecte les nouveaux types d'attaques. Cependant il faudra faire parfois des ajustements afin que le fonctionnement de référence corresponde au mieux à l'activité normale des utilisateurs et ainsi réduire les fausses alertes qui en découleraient.

Il existe différentes techniques pour repérer les attaques :

- *Approche probabiliste* : On prévoit quelle est la probabilité d'avoir un évènement après un autre.
- *Approche statistique* : Effectue des tests sur d'autres éléments concernant l'utilisateur :

- Le taux d'occupation de la mémoire
- Le taux d'occupation de la mémoire
- L'utilisation des processeurs
- La valeur de la charge réseau
- Le nombre d'accès à l'Intranet par jour.

Cette approche a l'avantage de ne pas avoir besoin d'une base de signature. Elle permet donc, en théorie, de détecter des attaques inconnues.

5.2. Le comportement de la détection (la réponse) :

Le comportement d'un IDS après la détection d'une intrusion est l'ensemble des actions prises par le système lorsqu'il détecte une attaque. Ces réponses peuvent être *passives* ou bien *actives*.

- **Réponse Passive :**

- **Alarme :** Les alarmes sont produites par les IDSs pour informer les administrateurs réseau lorsque des attaques sont détectées. La forme la plus commune est d'afficher un message d'alerte contenant des informations détaillées de l'intrusion détectée sur la console du responsable de la sécurité.
- **L'archivage :** L'archivage (logging) 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.
- **SNMP Trap :** Certains IDSs sont conçus pour produire des alertes et envoyer les rapports au système de gestion de réseau (network management system). Ils utilisent le protocole *SNMP (Simple Network Management Protocol)*, qui est un protocole dédié à la gestion du réseau [20].

- **Réponse Active :** 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 :

- **Rassembler des informations additionnelles :**

Dans le cas des systèmes de détection d'intrusions, rassembler des informations additionnelles sur une attaque est très important, 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 afin d'identifier l'attaque avec précision.

➤ **Changer l'environnement :**

Dans ce cas il faut stopper une attaque en progression et puis bloquer l'accès de l'attaquant, mais les *IDSs* 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.

Parmi ces actions on trouve :

- La configuration des routeurs et des Firewalls pour bloquer les paquets provenant de l'adresse *IP* de l'attaquant ou bien selon le numéro de port, le protocole, ou le service utilisé par l'attaquant.
- L'envoi des paquets *TCP* de type Reset ou des paquets *ICMP* au système de l'attaquant pour arrêter la connexion.

5.3. L'emplacement des sources d'audits :

L'emplacement des sources d'audits est un critère généralement utilisé pour classer les *IDSs*, ainsi, il existe deux grandes familles distinctes d'*IDS* : *NIDS* (Network-Based *IDS*) et *HIDS* (Host Based *IDS*) comme on a détaillé dans la section **II.2**.

5.4. La fréquence d'utilisation (la synchronisation) [20] :

La synchronisation se rapporte au temps écoulé entre les événements qui sont surveillés et l'analyse de ces événements. Elle est réalisée en temps « réel » ou « différé ».

- **En temps différé (Périodique) :** Ce type de système de détection d'intrusions, analyse périodiquement les différentes sources de données, à la recherche d'une éventuelle intrusion ou une anomalie passée. Cette approche est employée surtout, dans les *Host-IDSs*, qui scrutent les logs du système d'exploitation dans des intervalles de temps réguliers.

- **En temps réel (Continu) :**

Les *IDSs* en temps réel, traitent des flux continus d'informations à partir des différentes sources d'informations. C'est la technique prédominante de synchronisation pour les *IDSs* réseau,

qui recueillent l'information du trafic réseau .Par conséquent Les *IDSs* peuvent prendre des actions pour affecter la progression d'une attaque détectée.

Conclusion :

Dans ce chapitre, nous avons présenté le concept général de politique de sécurité et les systèmes de détection d'intrusion.

Dans la première section, on a vue les principes fondamentaux, les domaines d'application de la sécurité Informatique et des réseaux ainsi que la possibilité en matière de sécurité notamment la possibilité de sécurité par les Systèmes de Détection des Intrusions qui était bien détaillé dans la deuxième section. Nous avons fait une étude générale des systèmes de détection d'intrusions en procédant notamment à une étude descriptive, une étude sur les critères d'évaluation de l'efficacité des IDS et une étude sur la taxonomie des IDS en se basant sur un certain nombre de critère.

Dans le chapitre suivant on va présenter deux techniques de data mining pour la classification dans la détection d'intrusion tel que les *Machine à Support Vecteur* et les *Réseaux Bayésiens*.

Chapitre II : Machine à Support Vecteur et Réseaux Bayésiens.

Introduction :

La classification automatique de documents est un problème connu en informatique, il s'agit d'assigner un document à une ou plusieurs catégories ou classes. Le problème est différent selon la nature des documents en question, en effet la classification de textes diffère de la classification de documents images, vidéo ou encore son. On peut aussi imaginer des classifications selon des paramètres associés aux documents tels que par exemple l'auteur, la date de parution... Dans cette partie nous nous baserons sur la classification de connexion réseau selon leur nature attaque ou normal [56]. Et pour cela on va voir dans ce chapitre deux techniques de classification tel que les *Machine à Support Vecteur* et les *Réseaux Bayésiens*.

I. Classification :

C'est à cette étape que se fait l'assignation du document à la classe à laquelle il appartient. La détermination de la classe se fait grâce à des algorithmes de classification qui exploitent les données extraites et qui donnent en sortie la classe correspondante. Ces algorithmes se basent sur leur expérience passée où ils ont appris comment classer les textes, c'est de là que vient leur nom « Algorithmes d'apprentissage ». Il existe deux types de ces algorithmes : Les algorithmes *d'apprentissage supervisé* et les algorithmes *d'apprentissage non supervisé*.

1. Apprentissage non supervisé :

L'apprentissage non supervisé consiste à apprendre à classer sans supervision. Au début de processus nous ne disposons ni de la définition des classes, ni de leurs nombres. C'est l'algorithme de classification qui va déterminer ces informations. Nous ne disposons pas non plus de données en entrée qui sont déjà classées, c'est aussi à l'algorithme de découvrir par lui-même la structure plus ou moins cachée des données et de former des groupes d'individus dont les caractéristiques sont communes. [31]

L'apprentissage non supervisé est utilisé dans plusieurs domaines tels que : Médecine, Le traitement de la parole, traitement d'images.....etc.

2. Apprentissage supervisé :

Contrairement à l'apprentissage non supervisé, nous commençons ici par un ensemble de classes connues et définies à l'avance. Nous disposons aussi d'une sélection initiale de données dont la classification est connue. Ces données sont supposées indépendantes et identiquement distribuées.

Elles nous servent pour l'apprentissage de l'algorithme. La classification se fait par l'algorithme selon le modèle qu'il a appris. [31][32].

II. Machine à Support Vecteur :

Les « Supports Vectors Machines » appelés aussi « *Maximum Margin Classifier* » sont des techniques d'apprentissage supervisé basées sur la théorie de l'apprentissage statistique ou automatique. Les SVM sont relativement nouveaux, ils sont apparus en 1995 suite aux travaux de Vapnik. SVM traite d'un problème de classification bi classes. [33][34]

Le principe général de la classification par SVM peut être expliqué par la figure suivante:

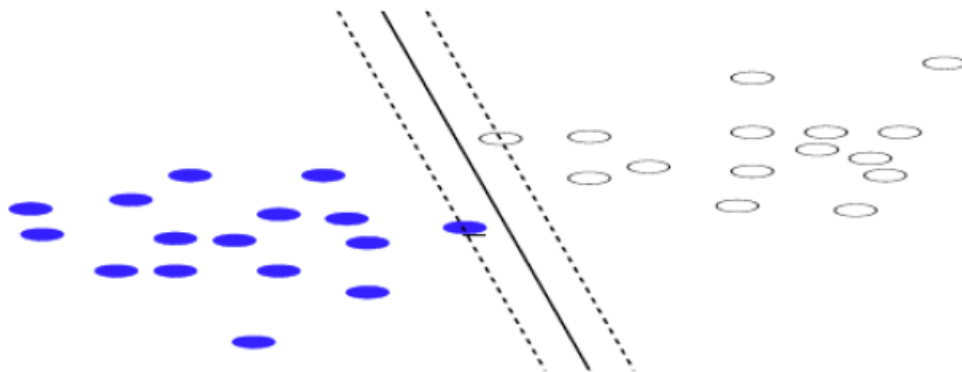


Figure II.1 : Le Principe générale de Classification par SVM.

Le but de SVM est de déterminer si un élément appartient à une classe ou pas. Nous disposons d'un ensemble de données et nous cherchons à séparer ces données en deux groupes. Le premier est l'ensemble de données appartenant à une classe, ces données sont

étiquetées par (+) et un autre ensemble qui contient les éléments qui n'appartiennent pas à la classe donc étiquetées (-).

L'algorithme SVM permet de trouver un hyperplan séparateur entre ces deux groupes. Pour optimiser la séparation, SVM cherche l'hyperplan pour lequel la distance entre la frontière des deux groupes et les points les plus proches est maximale, c'est le principe de maximisation de la marge.

1. Notions de base :

- **Hyperplan :**

Plaçons-nous dans le cas d'une classification binaire (i.e. les exemples à classifier réparties en 2 classes). On appelle hyperplan séparateur un hyperplan qui sépare les deux classes **Figure I.2**, en particulier il sépare leurs points d'apprentissage. Comme il n'est en général pas possible de trouver un, on se contentera donc de chercher un hyperplan discriminant qui est une approximation au sens d'un critère à fixer (maximiser la distance entre ces deux classes) [35].

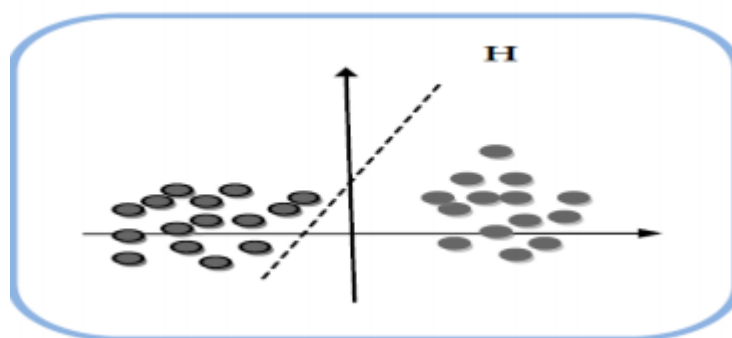


Figure II.2 : L'hyperplan H qui sépare les deux ensembles de points.

- **Vecteurs supports :**

Pour une tâche de détermination de l'hyperplan séparable des SVM est d'utiliser seulement les points les plus proches (i.e. les points de la frontière entre les deux classes des données) parmi l'ensemble total d'apprentissage, ces points sont appelés *vecteurs supports* **Figure I.3** [35].

- **Marge :**

Il existe une infinité d'hyperplans capable de séparer parfaitement les deux classes d'exemples. Le principe des SVM est de choisir celui qui va maximiser la distance minimale

entre l'hyperplan et les exemples d'apprentissage (i.e. la distance entre l'hyperplan et les vecteurs supports), cette distance est appelée la marge. *Figure I.3*

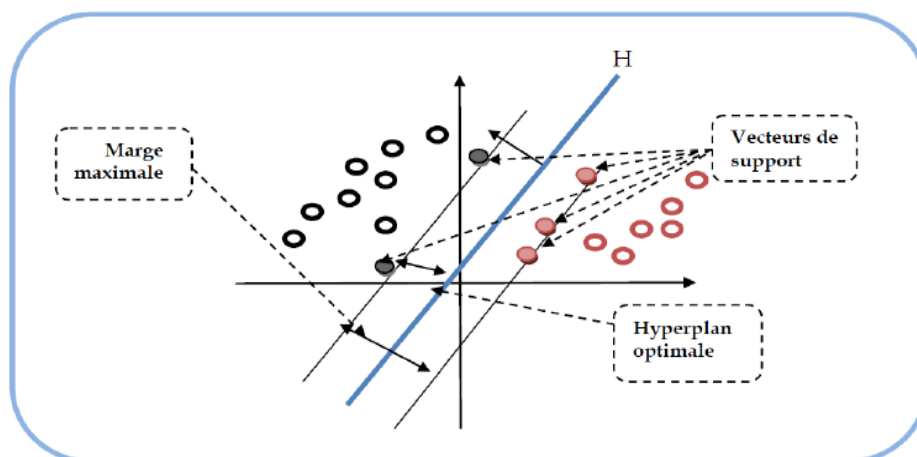


Figure II.3 : L'hyperplan H optimal, vecteurs supports et marge maximale.

2. Propriétés fondamentales :

- **Maximiser la marge :**

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsqu'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 *Figure I.4*, 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. [35]

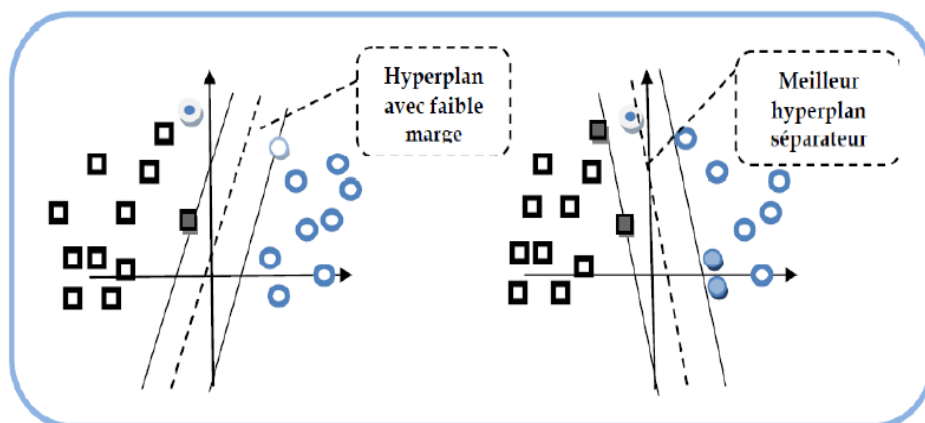


Figure II.4: meilleur hyperplan séparateur.

- **Linéarité et non-linéarité [35] :**

Parmi les modèles des SVM, on constate : les cas linéairement séparables et les cas non linéairement séparables.

1. cas linéairement séparables :

Les premiers sont les plus simples des 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.

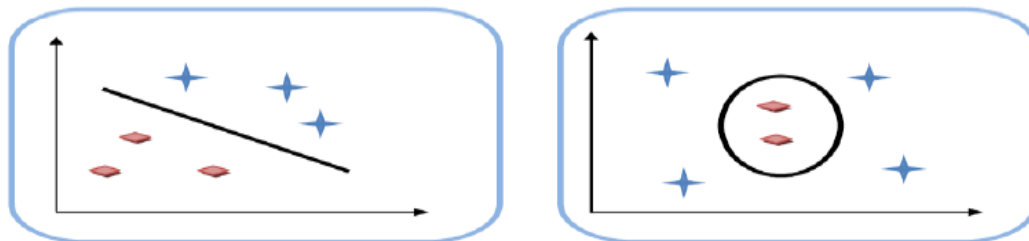


Figure II.5: à gauche cas linéairement séparable, à droite non linéairement

2. Cas non linéairement séparables :

Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace.

On va donc avoir un changement de dimension. Ce nouvel espace est appelé « espace de re-description ». En effet, intuitivement, plus la dimension de l'espace de re-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée.

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de re-description de plus grande dimension. Cette transformation non linéaire est réalisée via une fonction noyau.

En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des tests pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomial, gaussien, sigmoïde et laplacien (voir *Figure I.6*). [35]

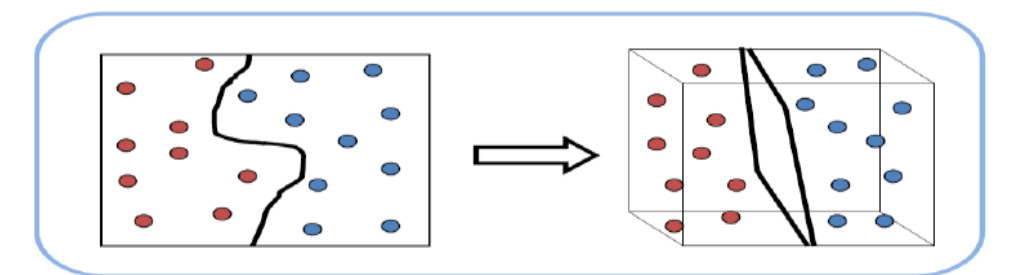


Figure II.6 : Transformation des données dans un espace de grande dimension.

3. Classification à marge maximale :

Maintenant que nous avons défini les notions de marges et d'hyperplans, nous pouvons formuler un problème d'optimisation mathématique tel que sa solution nous fournisse l'hyperplan optimal (maximisant la marge) :

$$\text{QP1 Minimiser } W(w, b) = \frac{1}{2} \|w\|^2$$

$$\text{Tel que } y_i (\langle w, x_i \rangle + b) \geq 1$$

Il s'agit d'un problème quadratique dont la fonction objective est à minimiser. Dans cette formulation, les variables à fixer sont les composantes w_i et b . Le vecteur w possède un nombre de composantes égal à la dimension de l'espace d'entrée. En gardant cette formulation telle quelle Pour éviter cela, il est nécessaire d'introduire une formulation dite duale du problème. Un problème dual est un problème fournissant la même solution que le primal mais dont la formulation est différente.

Pour dualiser QP1, nous devons former ce que l'on appelle le Lagrangien. Il s'agit de faire rentrer les contraintes dans la fonction objective et de pondérer chacune d'entre elles par une variable duale :

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i [y_i (\langle w, x_i \rangle + b) - 1]$$

$$\text{QP2 Maximiser } W(\alpha) = \sum_{i=0}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle$$

$$\text{Tel que } \begin{cases} \sum_{i=0}^n \alpha_i y_i = 0 \\ \alpha_i \geq 0 \forall i = 1..n \end{cases}$$

La résolution du dual permet donc de calculer le vecteur w à moindre coût, cependant cette formulation ne fait à aucun moment apparaître le terme b . Pour calculer ce dernier nous devons utiliser les variables primales :

$$b = - \max y_i = -1 \langle w, x_i \rangle + \max y_i = +1 \langle w, x_i \rangle / 2.$$

4. SVMs binaire :

Le cas le plus simple est celui où les données d'entraînement viennent uniquement de deux classes (+1 ou -1), on parle alors de classification binaire. L'idée des **SVMs** est de rechercher un hyperplan (droite dans le cas de deux dimensions) qui sépare le mieux ces deux classes (*Figure I.7*).

Si un tel hyperplan existe, c'est-à-dire si les données sont linéairement séparables, on parle d'une machine à vecteur support à marge dure (Hard margin). [36]

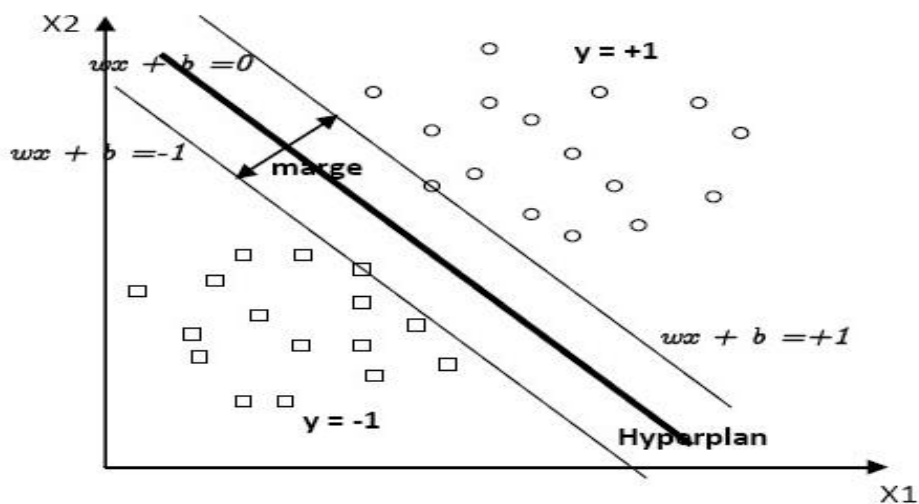


Figure II.7 : SVM binaire

- **SVM à marge dure :**

L'hyperplan séparateur est représenté par l'équation (1.1) suivante :

$$H(x) = w^T x + b \quad (1.1)$$

Où w est un vecteur de m dimensions et b est un terme.

Chapitre II: Machine à Support Vecteur et Réseaux Bayésiens

La fonction de décision, pour un exemple x , peut être exprimée comme suit :

$$\begin{cases} \text{classe} = 1 & \text{si } H(x) > 0 \\ \text{classe} = -1 & \text{si } H(x) < 0 \end{cases} \quad (1.2)$$

Puisque les deux classes sont linéairement séparables, il n'existe aucun exemple qui se situe sur l'hyperplan, c-à-d qui satisfait $H(x) = 0$. Il convient alors d'utiliser la fonction de décisions suivante :

$$\begin{cases} \text{classe} = 1 & \text{si } H(x) > 1 \\ \text{classe} = -1 & \text{si } H(x) < -1 \end{cases} \quad (1.3)$$

Les valeurs +1 et -1 à droite des inégalités peuvent être des constantes quelconques +a et -a, mais en divisant les deux parties des inégalités par a, on trouve les inégalités précédentes qui sont équivalentes à l'équation (1.4) :

$$y_i(w^T x_i + b) \geq 1, i = 1..n \quad (1.4)$$

L'hyperplan $w^T x + b = 0$ représente un hyperplan séparateur des deux classes, et la distance entre cet hyperplan et l'exemple le plus proche s'appelle la marge.

La région qui se trouve entre les deux hyperplans $w^T x + b = -1$ et $w^T x + b = +1$ est appelée la région de généralisation de la machine d'apprentissage. Plus cette région est importante, plus est la capacité de généralisation de la machine.

La maximisation de cette région est l'objectif de la phase d'entraînement qui consiste, pour la méthode SVM, à rechercher l'hyperplan qui maximise la région de généralisation c-à-d la marge. Un tel hyperplan est appelé "hyperplan de séparation optimale". En supposant que les données d'apprentissage ne contiennent pas des données bruitées (mal-étiquetées) et que les données de test suivent la même probabilité que celle des données d'entraînement, l'hyperplan de marge maximale va certainement maximiser la capacité de généralisation de la machine d'apprentissage.

La fonction de décision H peut être calculée, donc, pour chaque nouvel exemple x par l'équation $H(x) = \sum_s a_i y_i x^T x_i + b$ et la décision peut être prise comme suit :

$$\begin{cases} x \in \text{classe} + 1 & \text{si } H(x) > 0 \\ x \in \text{classe} - 1 & \text{si } H(x) < 0 \\ x \text{ inclassifiable} & \text{si } H(x) = 0 \end{cases} \quad (1.5)$$

La zone $-1 < H(x) < 1$ est appelée la zone de généralisation.

Si on prend un exemple x_k de l'ensemble d'entraînement appartenant à la classe x_k et on calcule sa fonction de décision $H(x_k)$ on peut se trouver dans l'un des cas suivants :

1. $y_k * H(x_k) > 1$: dans ce cas l'exemple est bien classé et ne se situe pas dans la zone de la marge. Il ne représente pas un vecteur support.
2. $y_k * H(x_k) = 1$: dans ce cas l'exemple est bien classé et se situe aux frontières de la zone de la marge. Il représente un vecteur support.
3. $0 < y_k * H(x_k) < 1$: dans ce cas l'exemple est bien classé et se situe dans de la zone de la marge. Il ne représente pas un vecteur support.
4. $y_k * H(x_k) < 0$: dans ce cas l'exemple se situe dans le mauvais coté, il est mal classé et ne représente pas un vecteur support.

- **SVM à marge souple :**

En réalité, un hyperplan séparateur n'existe pas toujours, et même s'il existe, il ne représente pas généralement la meilleure solution pour la classification.

En plus une erreur d'étiquetage dans les données d'entraînement (un exemple étiqueté +1 au lieu de -1 par exemple) affectera crucialement l'hyperplan. Dans le cas où les données ne sont pas linéairement séparables, ou contiennent du bruit (outliers : données mal étiquetées) les contraintes de l'équation (1.4) ne peuvent être vérifiées, et il y a nécessité de les relaxer un peu.

Ceci peut être fait en admettant une certaine erreur de classification des données (**Figure I.7**) ce qui est appelé "SVM à marge souple (Soft Margin)" [36]

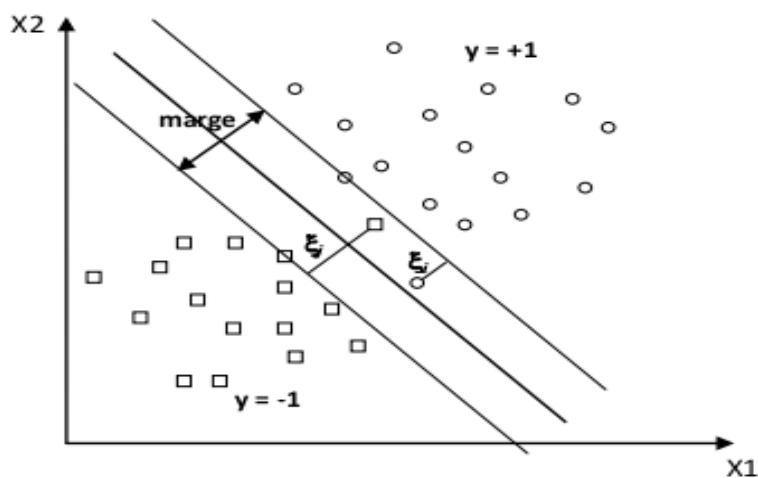


Figure II.8 : SVM binaire à marge souple.

Chapitre II: Machine à Support Vecteur et Réseaux Bayésiens

Dans cette première partie, nous avons introduit les machines à vecteur support, leur fondement théorique et leurs différentes variantes sans être très exhaustifs.

Les SVMs sont en avance dans plusieurs points que d'autres méthodes d'apprentissage tel que les réseaux de neurones et les arbres de décision :

- Les SVMs permettent de traiter plusieurs problèmes de datamining : classification, régression, clustering, détection des outliers,...etc.
- Les SVMs permettent de traiter les données numériques et symboliques, ce qui les a favorisées dans plusieurs applications complexes tel que le textmining, la reconnaissance des images, la reconnaissance vocale, les séquences biologiques,...etc.
- Les capacités de généralisation et la simplicité d'entraînement des SVMs sont bien au-delà des autres méthodes.
- Les SVMs sont très efficaces sur les données à nombre élevé d'attributs, même avec peu d'exemples. Elles n'imposent aucune limite sur le nombre d'attributs sauf les limites imposées par le hardware.

II. Les Réseaux Bayésien :

Les Réseaux Bayésiens font partie de la famille des modèles graphiques. Ils regroupent au sein d'un même formalisme la théorie des graphes et celle des probabilités afin de fournir des outils efficaces autant qu'intuitifs pour représenter une distribution de probabilités jointe sur un ensemble de variables aléatoires.

Ce formalisme très puissant permet une représentation intuitive de la connaissance, sur un domaine d'application donné et facilite la mise en place de modèles performants et clairs. La représentation de la connaissance se base sur la description, par des graphes, des relations de causalité existant entre des variables décrivant le domaine d'étude. A chaque variable est associée une distribution de probabilités locale quantifiant la relation causale.

Ces modèles de raisonnement probabiliste se caractérisent par deux aspects :

- un aspect graphique ou qualitatif permettant de représenter d'une manière très simple la connaissance sous forme d'un graphe orienté sans cycles,
- un aspect probabiliste ou quantitatif offrant un moyen de quantifier l'incertitude des relations d'influences entre les variables du domaine étudié.

Plus précisément, un réseau bayésien est défini comme un graphe orienté acyclique (DAG).

Le rôle des graphes dans les modèles probabilistes et statistiques est triple:

1. fournir un moyen efficace d'exprimer des hypothèses,
2. donner une représentation économique des fonctions de probabilité jointe,
3. faciliter l'inférence à partir d'observations.

Il est muni d'un ensemble de tables de probabilités conditionnelles (CPT) pour quantifier l'incertitude relative aux relations d'influences.

1. Définition :

Formellement, étant donné un ensemble de variables aléatoires $X = \{X_1, X_2, \dots, X_n\}$, $\beta = \langle G; \Theta \rangle$ est un réseau bayésien où $G = (X, E)$ est un graphe orienté acyclique représentant la structure graphique de β . X est l'ensemble des nœuds où chacun représente une variable aléatoire X_i . A chaque X_i , on associe une table de probabilités locales Θ_i qui représente les probabilités des valeurs de X_i sachant toutes les valeurs possibles de leurs parents. E est l'ensemble des arcs représentant les relations de dépendance directe entre les différents nœuds du graphe G .

Les réseaux bayésiens permettent de représenter d'une manière compacte une distribution de probabilités jointe associée à l'ensemble des variables en utilisant la notion d'indépendance. Une distribution de probabilité jointe sur n variables binaires est composée de 2^n entrées. La distribution de probabilités jointe est décomposée sous forme d'un produit des distributions de probabilités locales selon la règle de chaînage.

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1 \dots n} (P_{X_i} / Pa(X_i))$$

Où $Pa(X_i)$ représente l'ensemble des parents de X_i . La probabilité conditionnelle d'une valeur d'une variable X_i sachant la valeur d'une autre variable X_j peut être calculée par la loi de Bayes de la manière suivante :

$$P(X_i | X_j) = \frac{P(X_j | X_i) \cdot P(X_i)}{P(X_j)}$$

Les distributions de probabilités locales doivent satisfaire les conditions de normalisation :

- Si X_i est un nœud sans parent du réseau bayésien alors la distribution locale associée à X_i doit satisfaire la condition suivante :

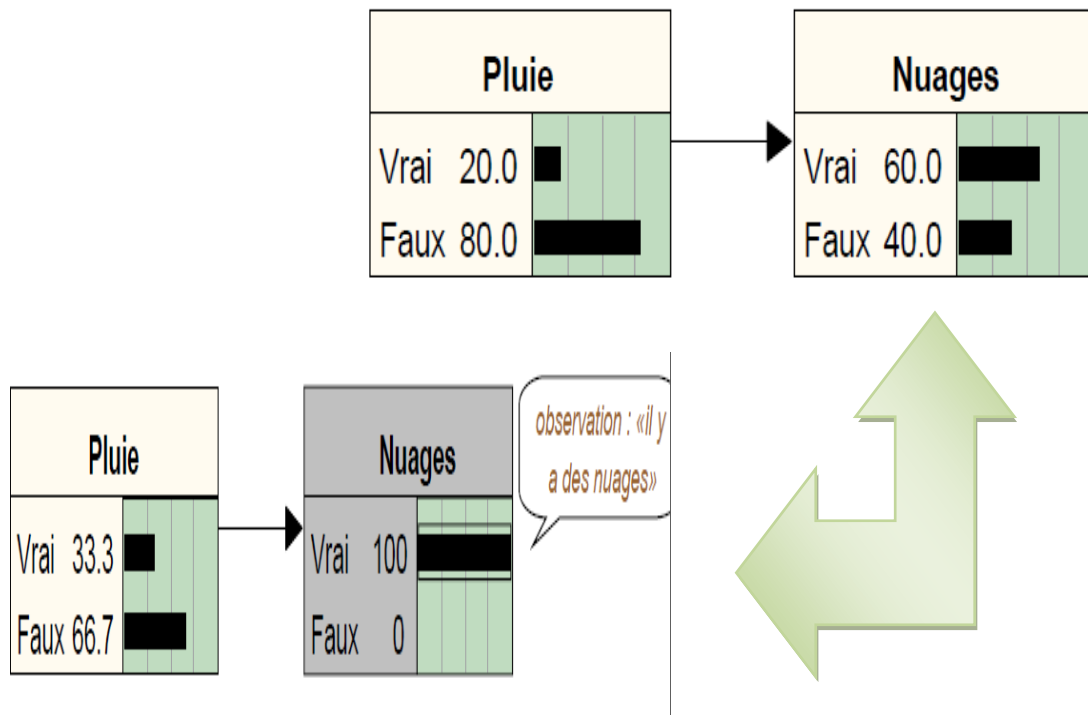
$$\forall x_i \in D_{x_i} \sum_{x_i} P(x_i) = 1$$

- Si X_i a des parents dans le réseau bayésien, alors la distribution conditionnelle associée à X_i doit satisfaire :

$$\sum_{x_i} P(x_i | Pa(X_i)) = 1$$

2. Exemple d'un Réseau Bayésien :

- On introduit des observations (par exemple : «il y a des nuages») Le réseau bayésien propage les observations introduites et met à jour les lois de probabilité conditionnelles des autres variables.



- On utilise **la formule de Bayes** ci dessus :

$$\Pr(A/B) = \frac{\Pr(A \text{ et } B)}{\Pr(B)} = \frac{\Pr(B/A) \Pr(A)}{\Pr(B)}$$

- On obtient les résultats suivants:

$$\Pr(\text{Pluie}/\text{Nuages}) = \frac{\Pr(\text{Nuages}/\text{Pluie}) \Pr(\text{Pluie})}{\Pr(\text{Nuages})} = \frac{1*0.2}{1*0.2+0.5*0.8} = \frac{1}{3}$$

| Pluie | | Nuages | |
|-------|------|--------|-----|
| Vrai | 33.3 | Vrai | 100 |
| Faux | 66.7 | Faux | 0 |

3. La construction d'un réseau Bayésien :

Plusieurs étapes sont à considérer dans la construction d'un réseau bayésien: [40]

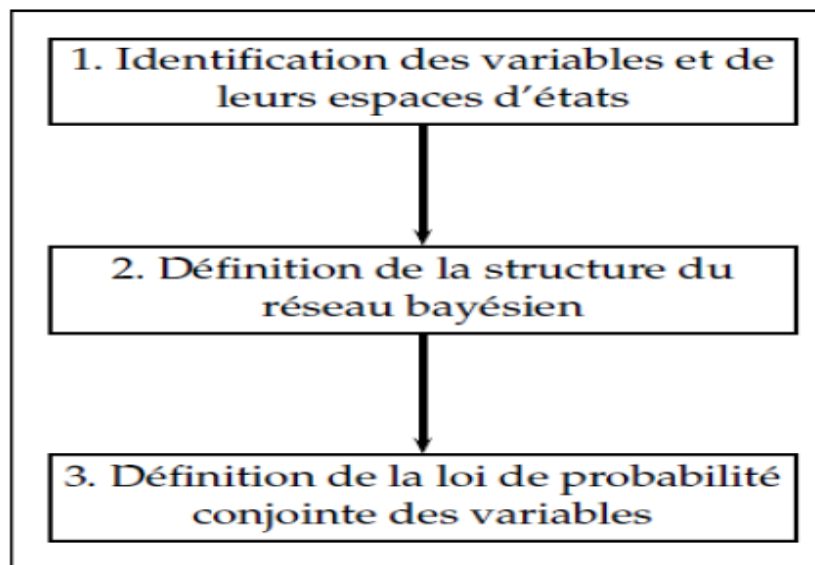


Figure II.9 : Étapes de construction d'un réseau bayésien.

- La première étape est *l'identification de variables* et pour chaque variable l'ensemble de ses valeurs possibles, pour cette étape l'intervention des experts du système est toujours nécessaire.

- La deuxième étape est *la définition de la structure du réseau bayésien*, trouver les liens d'influence entre les variables tout en s'assurant qu'il n'y a pas de boucle ou cycle.
- La dernière étape vise *la création des tableaux de probabilités pour les variables*, soit de variables sans parentes pour lesquelles des probabilités marginales doivent être définies, soit de variables qui ont de variables partantes et, dans ce cas, de probabilités conditionnelles sont définies.[40]

4. Intérêts des Réseau Bayésien :

- Outil de représentation graphique des connaissances.
- Représentation de l'incertain.
- Raisonnement à partir de données incomplètes : **Inférence**.
- Dés domaine d'application variés : Diagnostic, Fiabilité, Maintenance, Sécurité Informatique, Maitrise des risques

5. D-Séparation :

- **Principe :**

Déterminer si deux variables quelconques sont indépendantes conditionnellement à un ensemble de variables instanciées.

- **Définition :**

Deux Variable A et B sont d-séparées ou (bloqués) si pour tous les chemins entre A et B, Il existe une variable intermédiaire V différente de A et B telle que l'une de deux propositions est vrais :

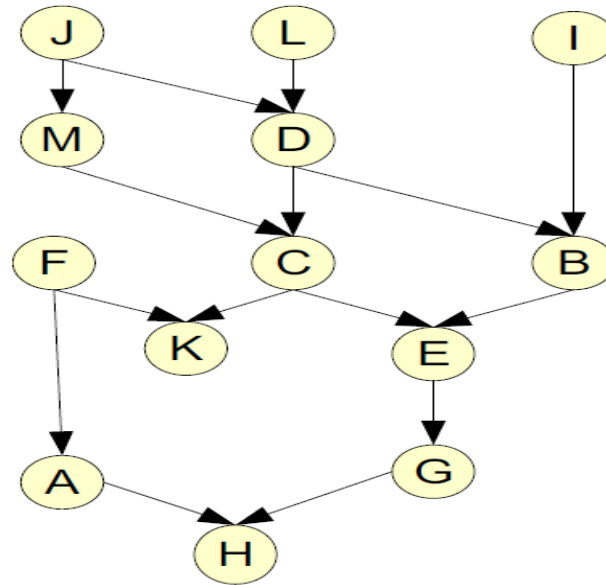
- ✓ La connexion est série ou divergente et V est instancié.
- ✓ La connexion est convergente et ni V ni ses descendants ne sont instanciés

Si A et B ne sont pas d-séparés, ils sont d-connectés. [42]

- **Exemples :**

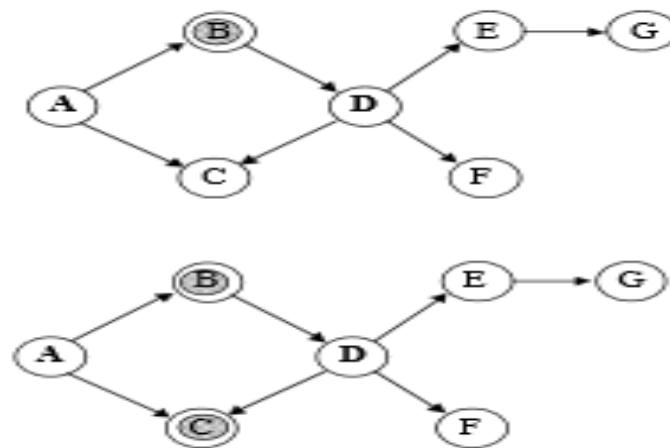
1. La connexion est série ou divergente et V est instancié.

La connexion est convergente et ni V ni ses descendants ne sont instanciés.



2. A est d-séparé de D par B : Comme B est la seule information connue dans ce graphe, une connaissance sur A ne modifiera pas la connaissance sur D : le circuit d'information de A à D est bloqué par B.

A est toujours d-séparé de D par B : Cependant, comme C est connu maintenant, un chemin de circulation de l'information est ouvert de A à D.



6. Apprentissage des Réseaux Bayésiens :

L'apprentissage d'un réseau bayésien consiste à définir la structure graphique et associer des tables de probabilités conditionnelles à chaque variable du problème à modéliser. Il s'agit donc d'apprentissage de structures et de paramètres.

- **Apprentissage de structures :**

L'apprentissage de structure d'un réseau bayésien consiste à identifier les nœuds et les relations possibles entre ces nœuds à partir des données d'apprentissage. La recherche de structure de réseaux bayésiens est un problème difficile [42], principalement à cause du fait que l'espace de recherche est exponentiel en fonction du nombre de variables décrivant le domaine. C'est pourquoi, de nombreux algorithmes d'apprentissage automatique ont été proposés : algorithme de recherche de causalité [43], algorithme de poids minimum [44], l'algorithme K2 [45], consistant k-graphs [46], etc. La recherche de la meilleure structure d'un réseau bayésien peut se faire par exemple en parcourant tous les graphes possibles, de leur associer une valeur quantitative (score), puis de choisir le graphe ayant le score le plus élevé [45].

- **Apprentissage de paramètres :**

L'apprentissage de paramètres d'un réseau bayésien consiste à associer une table de probabilités locales à chaque nœud de la structure du réseau préalablement élaborée ou apprise. Les paramètres peuvent être donnés directement par l'expert (ses connaissances subjectives) ou calculés à partir de données d'apprentissage. En présence d'un ensemble de données d'apprentissage et de la structure du réseau, il est simple de calculer les probabilités conditionnelles. Le calcul peut se faire de deux manières différentes selon la nature des données (données complètes ou incomplètes) :

1. Concernant le cas où toutes les données sont complètes (données observées), les probabilités sont calculées sur la base des fréquences qui représentent le nombre d'apparitions de chacune des valeurs que le nœud peut prendre. Il s'agit dans ce cas d'un apprentissage statistique basé sur le maximum de vraisemblance. D'autres méthodes peuvent être également utilisées. Nous citons par exemple des méthodes qui se basent sur des estimations Bayésiennes (maximum a posteriori et espérance a posteriori).
2. Lorsque les données sont incomplètes, c'est-à-dire que les variables sont complètement manquantes ou ne sont observées que partiellement, plusieurs traitements sont possibles selon la nature des données. Nous citons par exemple la méthode basée sur l'analyse des exemples disponibles qui consiste à calculer seulement les probabilités des variables X_i et les probabilités de leur parent $P_a(X_i)$. Ainsi, pour estimer $P(X_i | P_a(X_i))$, il suffit d'utiliser tous les exemples où X_i et $P_a(X_i)$

sont complètement observées. Le lecteur intéressé peut consulter [47] et [48] pour plus d'informations.

7. Inférence dans les Réseaux Bayésiens :

L'inférence dans un réseau bayésien concerne le calcul de la probabilité de n'importe quelle variable ou sous ensemble de variables à partir des autres variables observées. Il s'agit donc de déterminer les probabilités conditionnelles d'événements reliés par des relations d'influences. Les algorithmes d'inférence dans les réseaux bayésiens se répartissent en deux groupes: *algorithmes d'inférence exacte* et *algorithmes d'inférence approchée*. Les algorithmes d'inférence exacte exploitent les indépendances conditionnelles contenues dans le réseau pour calculer les probabilités a posteriori exactes [49] [55]. Concernant la deuxième catégorie d'algorithmes, les méthodes utilisées donnent des estimations approchées des probabilités a posteriori [50] [51].

8. Classification dans les Réseaux Bayésiens :

La classification est considérée comme un cas particulier d'inférence : une seule variable, dite variable de classe que nous symbolisons par C et les autres variables notées A_i , constituent les attributs. A partir des valeurs des attributs, la classification rend comme résultat la classe ayant la plus grande probabilité a posteriori $P(c_i=A)$. Comme exemples de classificateurs bayésiens, on trouve : classificateur naïf de Bayes (Naïve Bayes) [52], classificateur bayésien naïf augmenté TAN (Tree Augmented Naïve Bayes)[53], classificateur BAN (Bayésien Network Augmented Naïve Bayes) [52], les BMN (Multi-Nets Bayésiens) [54].

9. Avantages et Limites des Réseaux Bayésiens :

1. *Avantage :*

- La représentation de connaissance par des liens entre cause et effets et souvent plus naturelle que la représentation par règles de production.
- La représentation de connaissance est assez lisible (par opposition aux réseaux de neurones par exemple, ou même aux arbres de décision).
- Les types d'inférences réalisables à partir de la même représentation sont très variés (diagnostic, analyse de sensibilités....).
- Des méthodes d'apprentissages existants.

2. Limite :

- L'utilisation de probabilités et leur donnée par l'expert sont problématiques.
- Problème de variable continue.
- Complexité des Algorithmes.

Les réseaux bayésiens demeurent un outil puissant dans la modélisation de problèmes complexes et le raisonnement à partir de l'incertain. Nous avons introduit dans cette deuxième partie la notion de ces modèles graphiques probabilistes, leur principe de raisonnement ainsi que certaines méthodes d'apprentissage des réseaux bayésiens dans le cas des données complètes.

Conclusion :

Dans ce chapitre, nous avons présenté le concept général de SVM et les réseaux bayésiens et leur utilité dans différents domaines et en particulier pour les problèmes de classification.

Dans la première section nous avons introduit les Machines à Vecteurs Support leur fondement théorique et leurs différentes variantes sans être très exhaustifs. Nous nous sommes concentrés sur les principes classificateurs SVM qui étaient à l'origine conçus pour le classement binaire peut être utilisé pour classer les différents types d'attaques.

SVMs sont en avance dans plusieurs points, Elles sont très efficaces sur les données à nombre élevé d'attributs, même avec peu d'exemples. Elles n'imposent aucune limite sur le nombre d'attributs sauf les limites imposées par le hardware.

Et Dans la deuxième partie Nous nous sommes intéressés particulièrement aux Réseaux Bayésiens qui sont considérés parmi les modèles graphiques les plus utilisés dans les systèmes de détection d'intrusion durant ces dernières années, de nouveaux formalismes et techniques sont utilisés pour améliorer la détection et faire face à certains problèmes tels que la complexité du problème de détection due à la nature des données à analyser. En effet, les données qu'un administrateur réseau doit analyser sont souvent hétérogènes (différentes sources), d'une taille très importante et contiennent des informations incertaines, incomplètes, imprécises ou manquantes.

Chapitre II: Machine à Support Vecteur et Réseaux Bayésiens

Dans le chapitre suivant nous présentons l'approche proposée pour la détection d'intrusions fondés sur des architectures utilisant des méthodes de classification (SVM et réseaux bayésiens) qui distinguent automatiquement le comportement normal et anormal des systèmes, et les fonctions de croyances pour prendre la décision final.

*Chapitre III : Apports des fonctions de croyance dans la
Détection d’Intrusion*

Introduction

En sécurité informatique, la détection d'intrusion est l'acte de détecter les actions qui essaient de compromettre la confidentialité, l'intégrité ou la disponibilité d'une ressource.

La détection d'intrusion peut être effectuée manuellement ou automatiquement. Dans le processus de détection d'intrusion manuelle, un analyste humain procède à l'examen de fichiers de logs à la recherche de tout signe suspect pouvant indiquer une intrusion.

Lorsqu'une intrusion est découverte par un IDS, les actions typiques qu'il peut entreprendre sont par exemple d'enregistrer l'information pertinente dans un fichier ou une base de données, de générer une alerte par e-mail ou un message sur un pager ou un téléphone mobile.

Les réseaux bayésiens et les Machine à Vecteur Support(SVM) sont largement utilisés pour le problème de la détection d'intrusion. Beaucoup de travaux ont été présentés dans ce domaine, nous citons par exemple le travail de **M. Moorthy** dans [64]. Ils ont proposés de développer un système de détection d'intrusion hybride pour réseaux locaux sans fil, basée sur la logique floue, la détection anomalie est effectuée en utilisant la technique de réseau bayésien et la détection de l'utilisation abusive est réalisée en utilisant la technique Support Vecteur Machine (SVM). La décision globale de système est effectuée par un ensemble de règles floues.

Une autre approche proposée par *Salem Benfarhat, et all* [65] ou la corrélation d'alertes est un mécanisme indispensable pour la réduction du volume important des alertes et pour la détection des attaques coordonnées et complexes. Ils proposent une approche de corrélation d'alertes basée sur les réseaux bayésiens naïfs. Leur modélisation implique une légère contribution des connaissances d'experts. Elle tire profit des données disponibles, et fournit des algorithmes efficaces pour la détection et la prédiction des scénarios les plus plausibles.

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

Dans ce chapitre, on va étudier l’apport des fonctions de croyance dans la détection d’intrusion, l’idée est de développer un **IDS** Hybride qui regroupe par la théorie des fonctions de croyance, deux modules de détection, le premier basé sur les Machines à Vecteur Support **SVM** et le deuxième basé sur la technique des *réseaux bayésien*.

I. Théorie des fonctions de Croyance :

La théorie des croyances est issue des travaux de Dempster en 1967, repris par Shafer [57] sous le nom de theory of evidence, elle porte également le nom de théorie de Dempster-Shafer. Cette méthode repose sur la modélisation de la croyance en un évènement. Cette modélisation se réalise à partir de fonctions de masse permettant une bonne représentation de connaissances elle est souvent présentée d'un point de vue probabiliste, mais peut être vue comme un modèle formel de degrés de confiance dans le modèle transférable des croyances de Smet, qui a une approche plus subjective. Cette théorie permet particulièrement de bien modéliser l'incertitude, mais aussi l'imprécision.

Enfin du fait de sa modélisation, la théorie des croyances est plus adaptée à des problèmes de classification que des problèmes d'estimation,

Nous présentons ci-dessous, les étapes de la fusion d'informations pour la théorie des croyances.

1. Modélisation des fonctions de Croyance:

Le principe de la théorie des croyances repose sur la manipulation de fonctions définies sur des sous-ensembles et non sur des singletons comme dans la théorie des probabilités.

Ces fonctions sont habituellement à valeur dans $[0; 1]$, et sont appelées fonctions de masse ou encore masses élémentaires ou masses de croyance. [59]

2. Fonction de masse :

Considérons l'ensemble appelé cadre de discernement $D = \{d_1, d_2, \dots, d_n\}$, ensemble de toutes les décisions possibles par exemple l'ensemble de toutes les classes envisageables dans une problématique de classification. L'espace des fonctions de masse m est donné par l'ensemble de toutes les disjonctions possibles des décisions d_i noté :

Chapitre III : Apports des fonctions de croyance dans la Détection d'Intrusion

$$2^D = \{\phi, \{d1\}, \{d2\}, \{d1 \cup d2\}, \{d3\}, \{d1 \cup d3\}, \{d2 \cup d3\}, \{d1 \cup d2 \cup d3\}, \dots, D\}$$

La différence introduite par la généralisation de Dezert-Smarandache consiste à considérer les décisions di pas forcément disjonctives, il faut donc ajouter les toutes les intersections possibles entre deux éléments de 2^D . [59]

Nous définissons ici une fonction de masse comme une fonction définie sur 2^D à valeurs dans $[0; 1]$. Pour une source S_j , la fonction de masse m_j vérifie en général par construction :

$$\sum_{A \in 2^D} m_j(A) = 1$$

Ici, la différence avec les probabilités réside dans le fait que A peut être l'union de deux décisions $d1$ et $d2$. C'est grâce à ce principe que la théorie des croyances permet de modéliser les imprécisions. Une hypothèse souvent faite par commodité est celle d'un monde clos, ou monde fermé, c'est-à-dire que toutes les décisions possibles sont représentées dans D , et dans ce cas :

$$m_j(\phi) = 0.$$

Au contraire, si nous pouvons accepter le fait qu'une décision hors de D est envisageable, alors :

$$m_j(\phi) > 0.$$

Et dans cas on suppose un mode ouvert, c'est-à-dire que les décisions ne sont pas exhaustives.

3. Fonction de croyance :

La théorie des fonctions de croyance est une généralisation de la théorie bayésienne qui tient compte des notions d'incertitude et d'imprécision de l'information Toutes les hypothèses sont mutuellement exclusives et sont nommées *singletons*. [60]

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

Les fonctions de *plausibilité* ($Pl(.)$) et de *croyance* ($Bel(.)$) sont définies de 2^D vers $[0,1]$ comme suit :

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B)$$
$$bel(A) = \sum_{B \subset A} m(B)$$

Pour obtenir une fusion de l’information de deux sources différentes 1 et 2, il existe une combinaison de leur masses d’évidence appelée règle de Dempster :

$$(m_1 \oplus m_2)(A) = m_{1,2}(A) =$$
$$\frac{1}{1 - K} \sum_{B \cap C = A} m_1(B).m_2(C) \quad A, B, C \subset 2^\Theta$$

Où K est défini comme suit :

$$K = \sum_{B \cap C = \emptyset} m_1(B).m_2(C)$$

Le dénominateur $1 - K$ est un facteur de normalisation. Plus précisément K représente la mesure du conflit entre les sources 1 et 2. Plus K est important, plus les sources sont en conflit et moins la fusion a de sens. Si $K = 1$ alors le conflit est total et la fusion n’a pas de sens.

Si les sources sont en conflit fort (K est grand) alors la règle de Dempster peut conduire à des résultats erronés [60]. La raison de ce comportement de la règle de Dempster provient du fait que la masse d’évidence affectée à l’ensemble vide est nulle. Cette contrainte $m(\emptyset) = 0$ implique que l’intersection entre deux hypothèses est vide.

3.1. Mesure de conflit :

Comme on l’a vu, la valeur K permet de mesurer le conflit entre sources. Néanmoins, le conflit doit être suffisamment faible pour que la fusion ait un sens. Pourtant en utilisant la mesure de conflit K on peut arriver à des résultats erronés.

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

Ainsi pour deux paramètres fournissant exactement les mêmes masses d’évidence, on peut trouver un conflit K non nul entre les deux sources.

Par exemple supposons que nous avons 3 paramètres P1, P2 et P3 et deux classes C1 et C2 (on a donc $D = \{C1, C2\}$). Les masses d’évidence supposées connues pour les 2 classes sont respectivement pour P1, P2 et P3 : [60]

| | P1 | P2 | P3 |
|-------------------------------|---------------|---------------|---------------|
| m(c1) | $\frac{1}{3}$ | $\frac{1}{3}$ | 0 |
| m(c2) | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{1}{3}$ |
| m(Θ) | $\frac{1}{3}$ | $\frac{1}{3}$ | $\frac{2}{3}$ |

Nous pouvons calculer le conflit $K_{1,2}$ entre P1 et P2, et $K_{1,3}$ entre P1 et P3. Le calcul du conflit se simplifie et revient dans ce cas à la formule suivante :

$$K_{i,j}, i \neq j = \sum_{k,l,k \neq l} m_i(C_k) \cdot m_j(C_l)$$

Le conflit $K_{1,2}$ est égal à 0.222 alors que les paramètres P1 et P2 sont parfaitement en accord.

De plus le conflit $K_{1,3}$ est égal à 0.111. Ainsi le conflit entre deux paramètres fournissant la même distribution de masses d’évidence est non nul et même supérieur au conflit entre deux paramètres donnant des résultats différents.

3.2. Fonction de non croyance :

Une fonction de non croyance ou doute a été introduite dans [57] et étudiée dans [61], elle est donnée pour tout 2^D par :

$$dtj(A) = \sum_{B \cap A = \emptyset} m_j(B) = Cr(A^c)$$

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

Ou A^c représente le complémentaire de A.

3.3. Fonction de communalité :

La fonction de communalité introduite par Dempster [58] et nommée par Shafer [57] est une autre fonction en bijection avec la fonction de masse, qui permet un calcul simple dans l'étape de combinaison (et plus précisément pour la combinaison de Dempster).

Elle n'apporte donc pas d'information supplémentaire. Elle a été introduite surtout à des fins calculatoires ; son interprétation reste peu aisée. Elle est donnée, pour une source S_j , pour tout 2^D par :

$$Q_j(A) = \sum_{B \in 2^D} m_j(B)$$

4. Combinaison :

Plusieurs modes de combinaison ont été développés dans le cadre de la théorie des croyances. Il existe principalement deux types de combinaison : la combinaison conjonctive et la combinaison disjonctive, qui ont été déclinés en un grand nombre d'opérateurs de combinaison dont des combinaisons mixtes.

4.1. Combinaison conjonctive :

L'approche initiale est celle introduite par Dempster [58] et reprise par Shafer [57], elle combine les fonctions de masse en considérant les intersections des éléments de 2^D .

- **Règle orthogonale de Dempster-Shafer :**

La règle (ou somme) orthogonale de Dempster-Shafer permet de combiner deux fonctions de masse ou plus en une seule. Elle est donnée pour tout $A \in 2^D$ par :

$$M(A) = (m_1 \oplus m_2 \oplus)(A) = \sum_{B \cap C = A} m_1(B)m_2(C)$$

Pour les m fonctions de masse elle s'écrit :

$$M(A) = (m_1 \oplus m_2 \oplus m_3 \dots \oplus m_n) (A) = \sum_{B_1 \cap \dots \cap B_n = A} \prod_{j=1}^n m_j(B_j)$$

Chapitre III : Apports des fonctions de croyance dans la Détection d'Intrusion

4.2. Combinaison disjonctive :

D'autres approches de combinaison ont été proposées telle que la combinaison disjonctive [57]. La combinaison disjonctive est donnée non plus en considérant les intersections, mais les unions. Les éléments focaux de la combinaison s'obtiennent alors par des tables d'union. La combinaison de m fonctions de masse m_j est donnée donc pour tout $A \in 2^D$ par :

$$m(A) = \sum_{B_1 \cup \dots \cup B_n = A} \prod_{j=1}^n m_j(B_j)$$

Cette combinaison s'écrit simplement avec les fonctions d'implacabilités. En effet, pour tout $A \in 2^D$ nous avons :

$$M(A) = \prod_{j=1}^n b_j(A)$$

4.3. Combinaison mixte :

Pour chercher à conserver les avantages de la combinaison conjonctive et disjonctive, Dubois et Prade [62] ont proposé un compromis : une combinaison mixte. Cette combinaison est donnée pour tout $A \in 2^D, A \neq \emptyset$, par :

$$m(A) = \sum_{B_1 \cap \dots \cap B_n = A} \prod_{j=1}^n m_j(B_j) + \sum_{\substack{B_1 \cup \dots \cup B_n = A \\ B_1 \cap \dots \cap B_n = \emptyset}} \prod_{j=1}^n m_j(B_j)$$

Ce modèle suppose que le conflit provient du non fiabilité des sources et peut donc être modélisé par la fiabilité. Cette combinaison est donc un compromis raisonnable entre la précision et la fiabilité.

II. Approche proposée :

L'inconvénient de la détection d'intrusion d'anomalie sous réseau est le taux élevé de faux positifs. Ceci peut être résolu par une conception d'un Système de détection d'intrusion hybride qui regroupe à la fois des modules de détection basés pour réduire le taux de faux

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

positif. Dans ce chapitre, nous présentons notre solution pour le problème des faux positifs l'idée est d'hybrider deux classificateur utiliser la théorie de demster shafer.

Dans notre système de détection d'intrusion, la détection d'intrusion est effectuée en utilisant la technique de réseau bayésien et la technique Support Vecteur Machine (SVM). La décision globale de système est effectuée par la fusion des deux résultats en se basant sur la théorie de Dempster-Shafer [57] [58].

1. La description du modèle :

Dans les travaux de la détection d'intrusion, il ya des approches qui utilise plus qu'un classificateur [64], dans ce chapitre, nous proposons une nouvelle approche qui combine deux différents classificateurs dans deux modules (SVM et les Réseaux Bayésiens) qui forme le premier niveau, le deuxième niveau représente la phase de la fuzzification des sorties de chaque classificateurs pour initialiser les paramètres du niveau suivants, dans le troisième niveau on trouve le module de la fusion qui se produit par la théorie des fonctions de croyance .

2. La base théorique de notre approche :

Dans le domaine de data mining, nous avons différents types de classificateurs basé sur différentes méthodes et techniques. Chaque classificateur peut classer chaque connexion réseau comme un comportement normal ou une attaque avec divers taux d'erreur.

La performance des différents types de classificateurs est mesurée par leurs capacités de classer chaque connexion dans la bonne catégorie. Les quatre cas possibles sont : **vrai négatif, faux négatif, faux positif** et **vrai positif**.

Le vrai négatif **VN** est la classification correcte de la classe d'attaque, le faux négatif **FN** est la mauvaise classification de la classe d'attaque comme une classe normale, le faux positif **FP** est la mauvaise classification d'une classe normale comme une classe d'attaque, et le vrai positif **VP** est la classification correcte de la classe normale.

La modification de l'ensemble de données d'apprentissage ou l'utilisation d'un autre type de classificateur peut causer des modifications dans les résultats de la classification.

Chapitre III : Apports des fonctions de croyance dans la Détection d'Intrusion

Si nous utilisons deux classificateurs de types différent, l'intersection des résultats de classification peut nous permettre de diminuer le *FP* et *FN*, ce qui nous permet de construire un modèle de détection d'intrusion plus efficace.

3. La structure générale de notre modèle:

Dans cette section, nous présentons les différentes phases de notre système. L'objectif est de minimiser les *FP* par l'hybridation de deux classificateurs. Comme le montre la figure III.2.

Le premier niveau est composé de deux modules : le classificateur SVM et dans ce module on choisie d'utilisé le *SVM Light* comme outil de classification, et les réseaux Bayésiens et dans ce deuxième module on choisie d'utilisé *le NaiveBayes*.

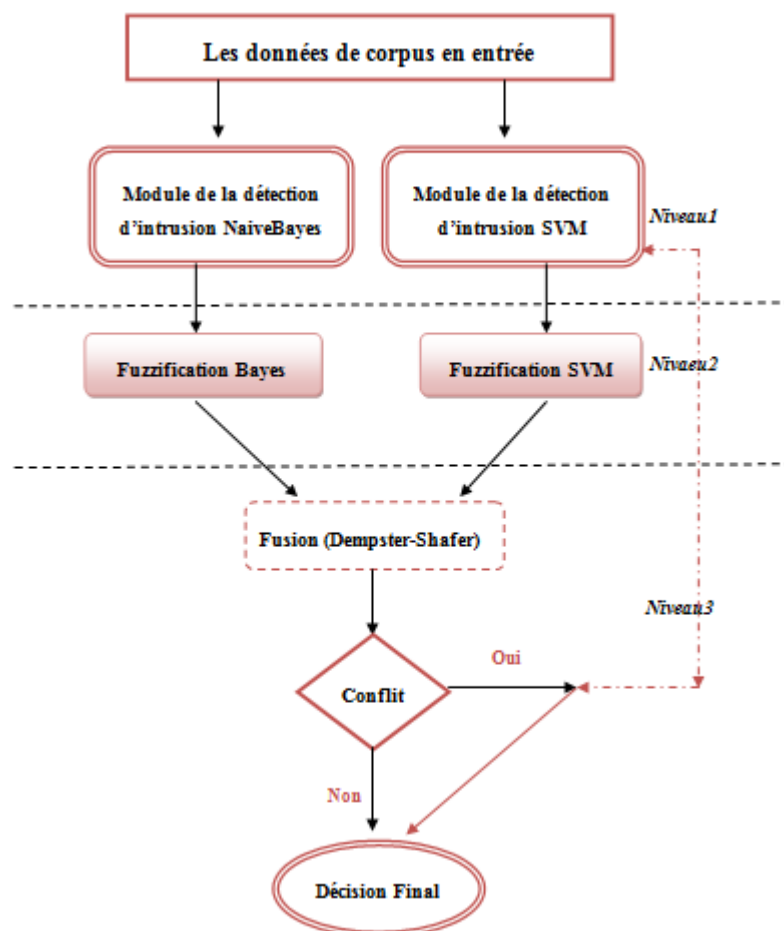


Figure III.1 : L'architecture générale de notre approche hybride.

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

Le deuxième niveau c’est la phase de fuzzification des sorties du premier niveau à l’aide des trapèzes associer a chaque un des classificateurs on intégrant des variable linguistiques (**N** pour normal, **AN** pour anormal et **LAN** pour légèrement anormal) comme illustre la figure III.3.

Le troisième niveau représente la phase de fusion qui regroupe les résultats de la fuzzification à l’aide de la théorie de **Dempster-Shafer**, et dans cette phase on a choisie d’utilisé la combinaison conjonctif pour calculer le taux de conflit qui est essentiel pour le calcul des mases anormal, normal, et légèrement anormal.

Les résultats de la fusion est considérée comme décision final, en cas de présence de conflit entre deux sources, on prend la décision du module SVM parce qu’il présente des meilleurs performances par rapport au module bayésien.

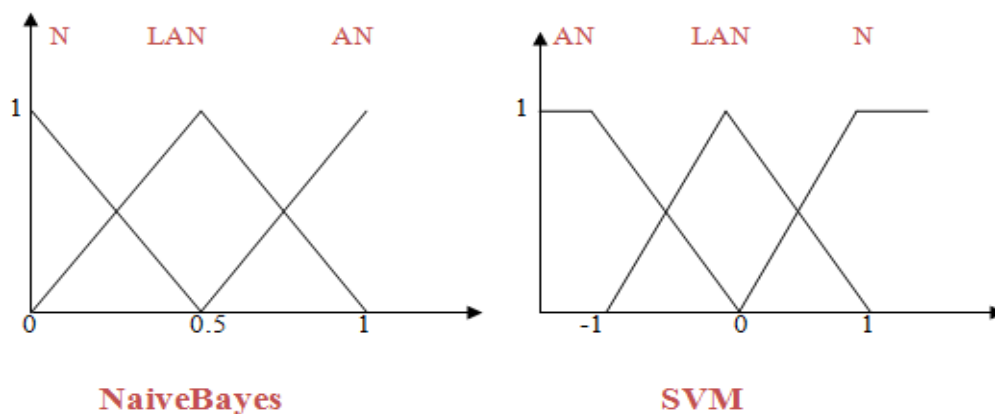


Figure III.2 : La phase de fuzzification.

3.1. Module de détection SVM :

Le SVM a été largement utilisé pour la détection d'intrusion, il passe par trois phases pour réaliser une classification correcte des données :

- **La première phase :** est la phase de traitement préalable des données ou le *prétraitement*, qui traite les données de corpus (données) on conversant les attributs alphanumérique à la forme numérique suivant a un codage représenter dans le *Tableau III.2* puis on réorganise les données comme suit : la classe actuel (par exemple 1 pour la classe

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

normal et -1 pour la classe d’attaque), 1 : attribut N°1, 2 :attribut N°2, etc jusqu’à le dernier attribut dans le corpus.

• **La deuxième phase** : est de la phase *formation* ou *l’apprentissage* 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* ou *classify*. Cette phase consiste à mesurer les performances des données de la deuxième phase a travers le modèle générer après l’apprentissage.

| URI_Ressource_type | code | | | Script_type | code | Writing_script | Code | Class | code | | |
|--------------------|------|------|----|-------------|------|----------------|------|------------|------|-----------|----|
| .swf | 1 | .php | 15 | none | 1 | none | 1 | normal | 1 | sqli | -1 |
| .exe | 2 | .tar | 16 | js | 2 | Cookie_set | 2 | R2L | -1 | Sqli-auth | -1 |
| other | 3 | .txt | 17 | vb | 3 | Doc_load | 3 | bo | -1 | new | -1 |
| none | 4 | .ico | 18 | | | Cookie_read | 4 | Value_mis | -1 | | |
| .htm | 5 | .ppt | 19 | | | | | Iv_R2L | -1 | | |
| .cgi-bin | 6 | .js | 20 | | | | | Il_R2L | -1 | | |
| .asp | 7 | .zip | 21 | | | | | Flooding | -1 | | |
| .pl | 8 | .cgi | 22 | | | | | url_R2L | -1 | | |
| .dat | 9 | .asm | 23 | | | | | Poor_DOS | -1 | | |
| .gif | 10 | .sh | 24 | | | | | DOS | -1 | | |
| .ps | 11 | .mov | 25 | | | | | Vul_scan | -1 | | |
| .pdf | 12 | .com | 26 | | | | | XSS | -1 | | |
| .css | 13 | .ini | 27 | | | | | Shell_cmds | -1 | | |

Tableau III.1 : le prétraitement des données en entrées a le SVM.

3.1. Module de détection RB :

Dans notre approche on choisie d’utilisé les réseaux Bayésiens naïf, et comme un outil de réalisation on a préféré le **Weka**.

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

La figure III.3 représente la structure d’un RB naïf, et le tableau III.2 représente les attributs du corpus qu’on a utilisé pour la construction d’un NaiveBayes.

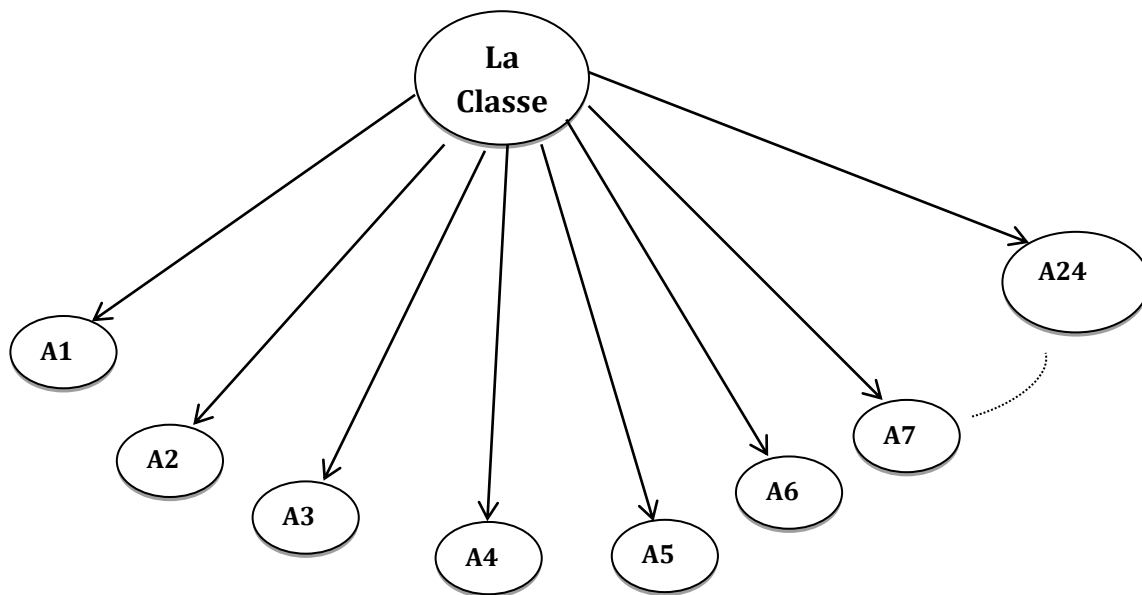


Figure III.3 : La structure du RB naïf.

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 déterminer :

| | Attribut | Valeurs |
|----|-----------------|---------|
| A1 | http_req_length | numeric |
| A2 | http_uri_length | numeric |
| A3 | num_safe_req | numeric |
| A4 | num_unsafe_req | numeric |

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

| | | |
|-----|-----------------------------|--|
| A5 | uri_ressource_type | .swf, .exe, other, none, .htm, cgi-bin, .asp, .pl, .dat, .gif, .ps, .pdf, .css, .jpg, .php, .tar, .txt, .ico, .ppt, .js, .zip, .cgi, .asm, .sh, .mov, .com, .ini |
| A6 | num_param | numeric |
| A7 | num_arg | numeric |
| A8 | response_code | 0, 200, 201, 202, 203, 204, 205, 206, 300, 301, 302, 303, 304, 305, 306, 307, 400, 401, 402, 403, 404, 405, 414, 416, 500 |
| A9 | response_time | numeric |
| A10 | script_type | none, js, vb |
| A11 | writing_script | none, cookie_set, doc_loc, cookie_rd |
| A12 | is_html_response | 0, 1 |
| A13 | directory_traversal | 0, 1 |
| A14 | shell_cmds | 0, 1 |
| A15 | sensitive_files | 0, 1 |
| A16 | default_login_passwd | 0, 1 |
| A17 | num_nonprintcar | numeric |
| A18 | sql_cmds_tricks | 0, 1 |
| A19 | css_scripts | 0, 1 |
| A20 | num_req_same_host | numeric |
| A21 | num_req_same_URI | numeric |
| A22 | inter_req_time_inter_val | numeric |
| A23 | req_same_host_diff_URI_rate | numeric |
| A24 | http_error_rate | numeric |

Tableau III.2 : Les attributs du corpus appliqué sur NaiveBayes.

4. Exemple illustratif :

Dans cette partie on va détailler trois cas différents de notre modèle le premier présent le cas où la classe actuelle est Normal, le premier Classificateur SVM le déclare comme Anormal, le deuxième Classificateur Bayésienne le déclare comme Normal et notre Approche

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

a détecté la connexion comme Normal. Ce cas est défini dans le corpus comme ligne numéro **7643**.

Le deuxième exemple présente le cas où la classe actuelle est Normal, le premier Classificateur SVM détecté comme Normal, le deuxième Classificateur NaiveBayes détecté comme Anormal et notre Approche a détecté la connexion comme Normal. Ce cas est défini dans le corpus comme ligne numéro **9118**.

Le troisième exemple présente le cas où la classe actuelle est Normale, le premier Classificateur SVM détecté comme anormal, le deuxième Classificateur NaiveBayes détecté comme Normal et notre Approche a détecté la connexion comme Légèrement Anormal. Ce cas est défini dans le corpus comme ligne numéro **15724**.

| Corpus | Attributs |
|---------------------------|--|
| ligne numéro 7643 | 156, 11, 1, 0, .txt, 0, 0, 404, 0.000829, none, none, 1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0.45, 1.00, 0.00, normal |
| ligne numéro 9118 | 487, 28, 1, 0, .htm, 0, 0, 200, 0.001145, js, none, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1.82, 1.00, 1.00, normal |
| Ligne numéro 15724 | 167, 11, 1, 0, .txt, 0, 0, 404, 0.001231, none, none, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 13, 8, 3.13, 0.38, 0.62, normal |

Tableau III.3 : Exemple d’échantillon du corpus.

5. Expérimentations :

Dans cette section nous avons effectué une série d’expérimentation avec un corpus qui représente l’ensemble de données de détection d’intrusion du projet PLACID (voir *Annexe A*).

On a utilisé le *Weka Data Mining Tools 3.7* et *SVMLight* pour la mise en œuvre de notre approche. La performance d’un IDS est mesurée par sa capacité de classer chaque connexion dans la bonne catégorie. Les indicateurs de performance les plus utilisés pour évaluer les systèmes de détection d’intrusion sont :

$$\text{Exactitude} = \frac{D+A}{\text{Total}}$$

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

$$\text{Faux positif} = \frac{B}{B+A}$$

$$\text{Faux négatif} = \frac{C}{C+D}$$

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.

Les résultats de simulations de notre approche sont représentés dans **le tableau III.4**.

Nous allons interpréter ici les différents résultats parvenus des pratiques faites régulièrement et qui vous ont été montrées auparavant. Ces commentaires reposent sur les épreuves des taux appartenant aux résultats.

Le tableau ci-dessus montre les différents résultats de test de notre approche sur les différentes dimensions du corpus.

| | SVM | R.Bayes | Approche proposé |
|----------------------------------|----------------|----------------|-----------------------------|
| Normales mal classés | 585 | 1487 | 151 |
| Anormales mal classés | 22 | 22 | 22 |
| Exactitude | 99.12% | 97.81% | 99.73% |
| Faux positif | 7.13% | 16.33% | 1.47% |
| Faux négatif | 0.0361% | 0.0367% | 0.0359% |

Tableau III.4 : Résultats de test sur les différentes mesurées.

Chapitre III : Apports des fonctions de croyance dans la Détection d’Intrusion

Nous avons commencé l’apprentissage avec un ensemble d’exemples d’enregistrements de fichier global 69015, dont 61378 normales et 7637 anormales. Après les nombreux tests faites pour le processus de classification, on a arrivé à obtenir des résultats satisfaisants.

D’après les résultats des expérimentations, on conclue que notre approche est la plus performante par rapport aux résultats du SVM et RB, et la on touche une bonne amélioration au niveau d’exactitude et la minimisation du taux des faux positifs.

La *figure III.4* montre la comparaison entre les critères de performances des classificateurs SVM et réseaux bayésiens et notre approche.

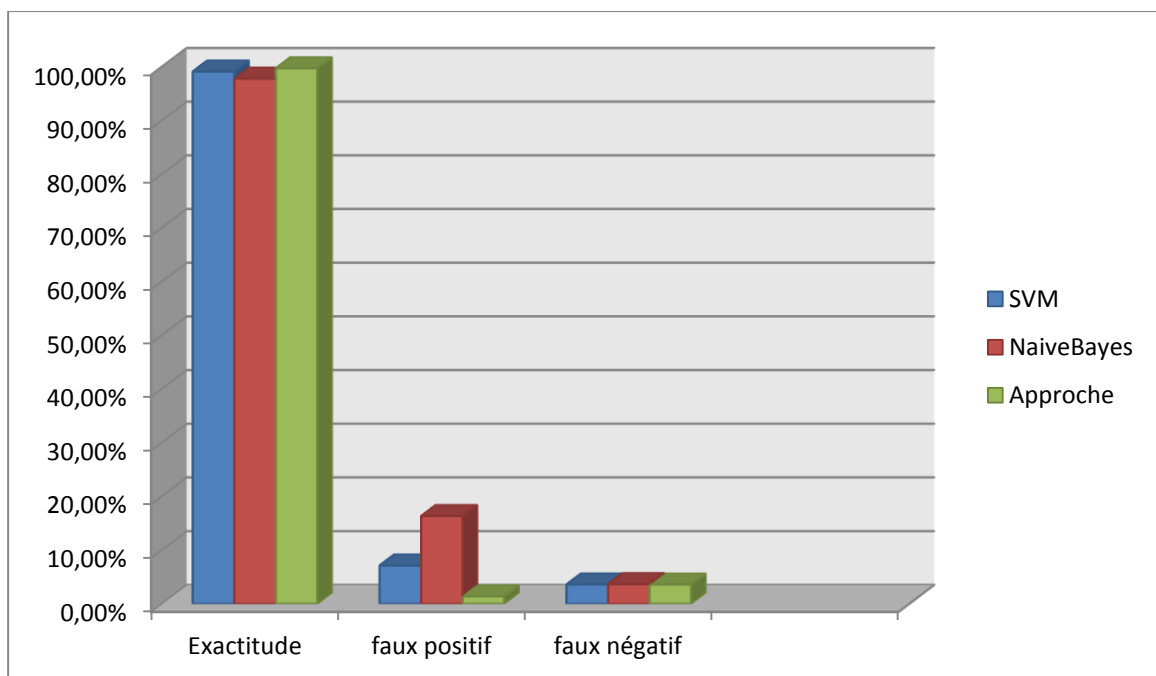


Figure III.4 : histogramme comparative des résultats de SVM, NaiveBayes, et l’Approche.

Conclusion :

Dans ce chapitre, nous avons présenté une nouvelle approche pour la détection d’intrusion basée sur la fusion des résultats de deux différents classificateurs en utilisant la théorie de fonction de croyance pour combinée les deux résultat dans ce niveau .

Notre approche donne des meilleurs résultats et augmente les performances d’un IDS par rapport aux deux classificateurs.

Chapitre IV : Implémentation et réalisation.

Introduction :

Dans cette section, nous abordons la partie implémentation d'un prototype de notre système avec les langages de programmation, logiciel, l'environnement de travail, la plateforme et les outils utilisés pour le développement de notre solution. Puis, nous allons présenter les différentes interfaces de notre application.

I. Outil de construction d'application :

Dans cette application nous avons utilisé :

1. les langages utilisés dans le développement :

Dans notre application, nous avons choisi d'utiliser le langage **JAVA**.

Le langage Java reprend en grande partie la syntaxe du langage C++, très utilisée par les informaticiens. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et référence.

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.).

Java permet de développer des applications client-serveur. Il a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (NetBeans/JDK), des machines virtuelles (JRE) applicatives multi plate-forme (JVM).

2 .Environnement de développement intégré :

NetBeans 8.0.2 :

Nous avons choisi NetBeans comme Environnement de Développement Intégré(EDI) open source lancé par SUN en juin 2009 qui permet de développer des applications Java, PHP, C, C++ et Ruby. Il comprend toutes les caractéristiques d'un IDE moderne.

NetBeans IDE 8.0.2 fournit sur les analyseurs de code de la boîte et les éditeurs pour

travailler avec la dernière Java 8 technologies--Java SE 8, Java SE Embedded 8, et Java ME Embedded 8.

3. Le Weka :

Weka est un ensemble de classes et d'algorithmes en Java implémentant les principaux algorithmes de data mining. Ce logiciel est développé en parallèle avec un livre data mining par I. Witten et E. Frank. [66]

Weka peut s'utiliser de plusieurs façons :

- Par l'intermédiaire d'une interface.
- Sur la ligne de commande.
- Par l'utilisation des classes fournies à l'intérieur de programmes Java : toutes les classes sont documentées dans les règles de l'art.

Les Données du Weka:

Les données sont sous un format ARFF (Attribut Relation File Format). Il est simple et il est facile de convertir des données issues d'un tableur ARFF (il y a même un convertisseur inclus dans Weka du format CSV vers le format ARFF).

Weka est installée sur vos machines : `user/local/weka-3-4/weka.jar`

Après avoir lancé, vous obtenez la fenêtre intitulée Weka GUI Chooser : choisissez l'Explorer.



Figure VI.1 : La fenêtre intitulée Weka GUI Chooser.

Chapitre IV : Implémentation et réalisation

La nouvelle fenêtre qui s'ouvre alors (Weka Knowledge Explorer) présente six onglets :

- **Preprocess** : pour choisir un fichier, inspecter et préparer les données.
- **Classify** : pour choisir, appliquer et tester différents algorithmes de classification : là, il s'agit d'algorithmes de classification supervisée qui est plutôt l'objet de l'option APE.
- **Cluster** : pour choisir, appliquer et tester les algorithmes de segmentation.
- **Associate** : pour appliquer l'algorithme de génération des règles d'association.
- **Select Attribut** : pour choisir les attributs les plus promoteurs.
- **Visualize** : pour afficher (en deux dimensions) certains attributs en fonction d'autres.

On cliquant sur l'onglet classify et on choisit le classifieur NaiveBayes:

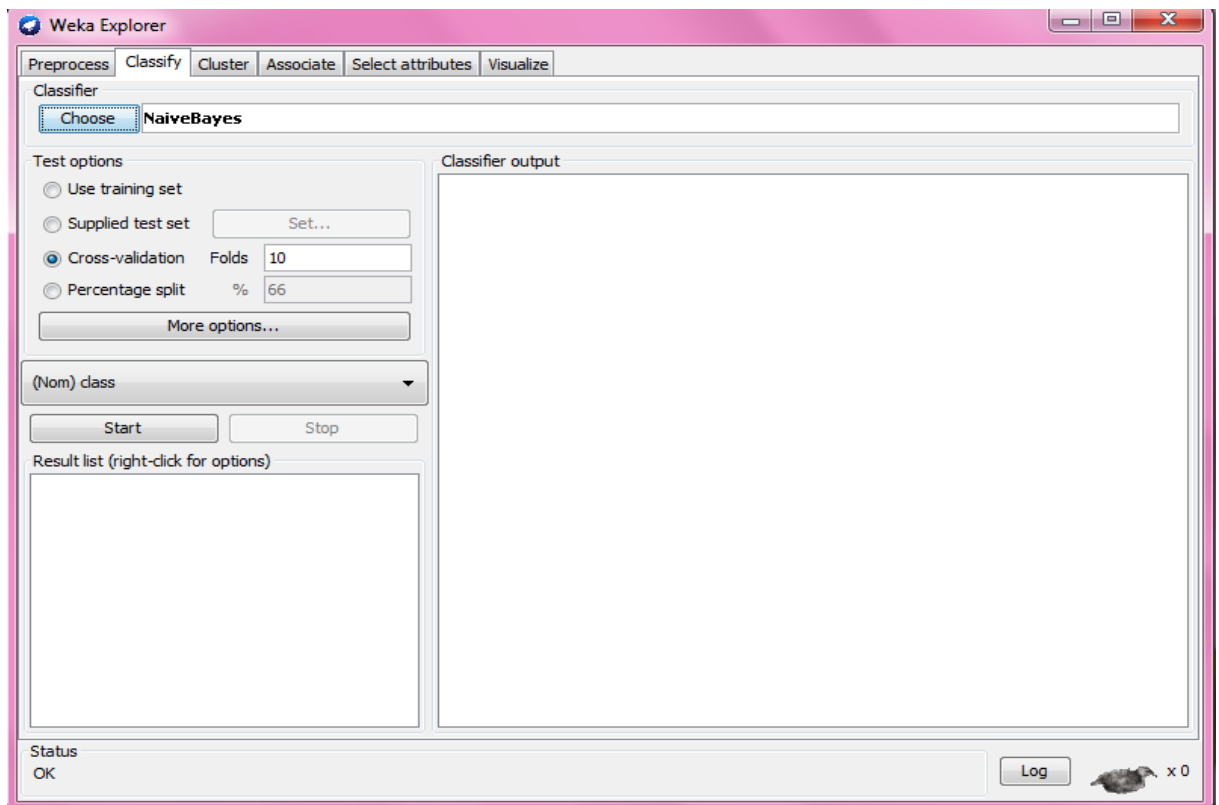


Figure IV.2 : Le classifieur NaiveBayes.

Les résultats de classification sont présentés sur la figure IV.3 on cliquant sur « Start ».

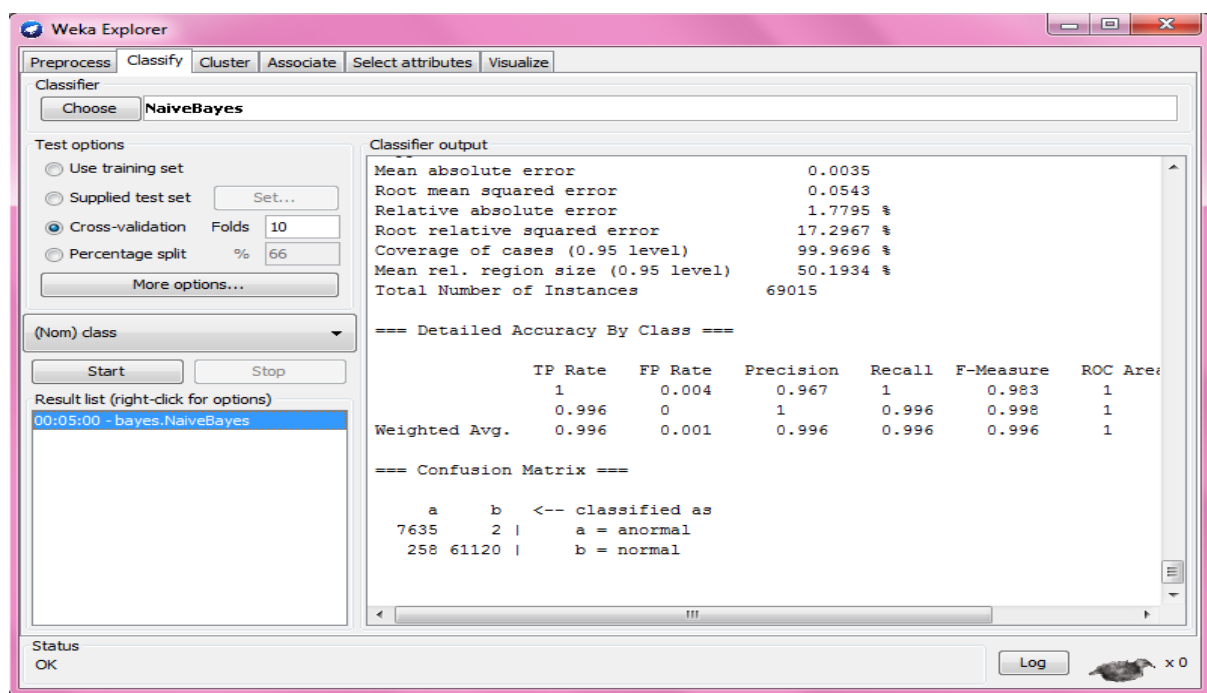


Figure IV.3 : Les résultats de classification par NaiveBayes.

On peut déduire la matrice de confusion d'après les résultats de l'application de NaiveBayes sur le corpus. *Tableau IV.1*

| A | B | ← Détecter comme : |
|-------------|--------------|--------------------|
| 7615 | 22 | A anormal |
| 1487 | 59891 | B normal |

Tableau IV.1 : La matrice de confusion pour NaiveBayes.

4. SVMLight :

Est une implémentation de Support Vector Machines (SVM) en C pour le problème de la reconnaissance des formes, pour le problème de la régression, et le problème de l'apprentissage d'une fonction de classement.

L'algorithme a des exigences de mémoire évolutive et peut gérer des problèmes avec plusieurs milliers de vecteurs de support de manière efficace.

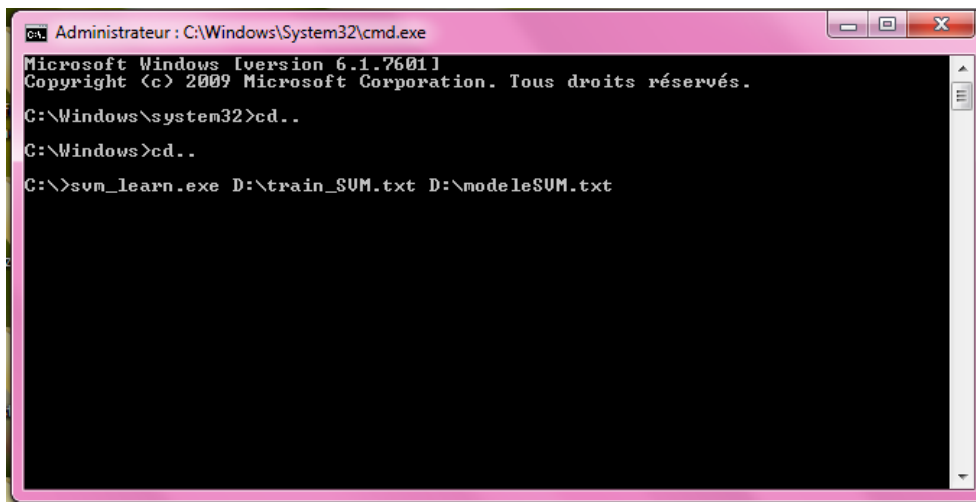
Les principales caractéristiques du programme sont les suivants :

- algorithme d'optimisation rapide.
- Résout classification et de régression des problèmes.

Chapitre IV : Implémentation et réalisation

- Résoudre les problèmes de classement (e. g. apprentissage des fonctions de récupération dans le moteur de recherche STRIVER).
- peut former SVMs avec des modèles de coûts et par exemple les coûts à charge.
- utilise la représentation de vecteur creux.

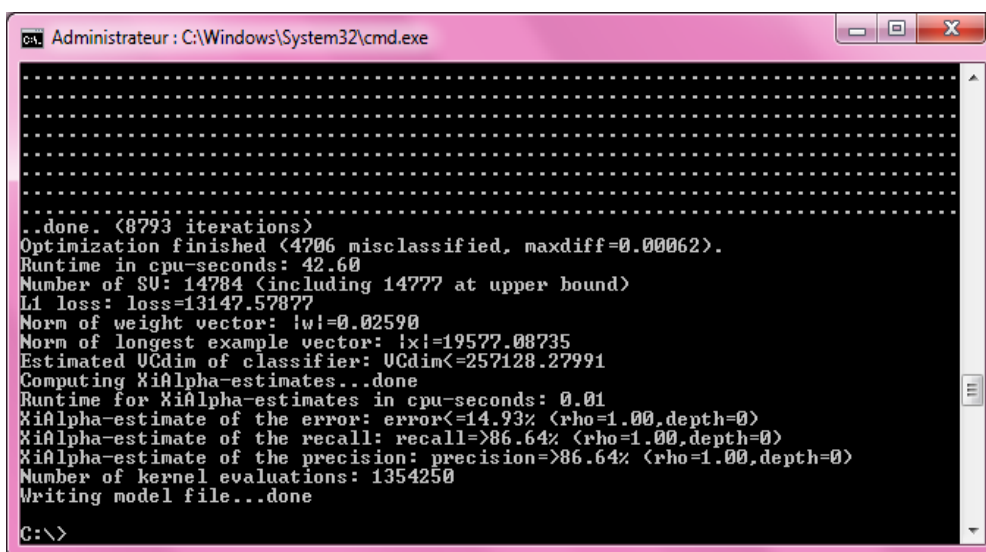
On lance le SVMLight par la console de commande CMD, et on choisie d'abord le `svm_learn.exe` pour faire un apprentissage sur le corpus train et définir le modèle qui est la sortie du programme comme illustre la figure IV.4 :



```
Administrateur : C:\Windows\System32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.
C:\Windows\system32>cd..
C:\Windows>cd..
C:\>svm_learn.exe D:\train_SUM.txt D:\modeleSUM.txt
```

Figure IV.4 : La fenêtre du `svm_learn`.

Après taper « Entrer » le `svm_learn` fait sont optimisation et la création du modèle.

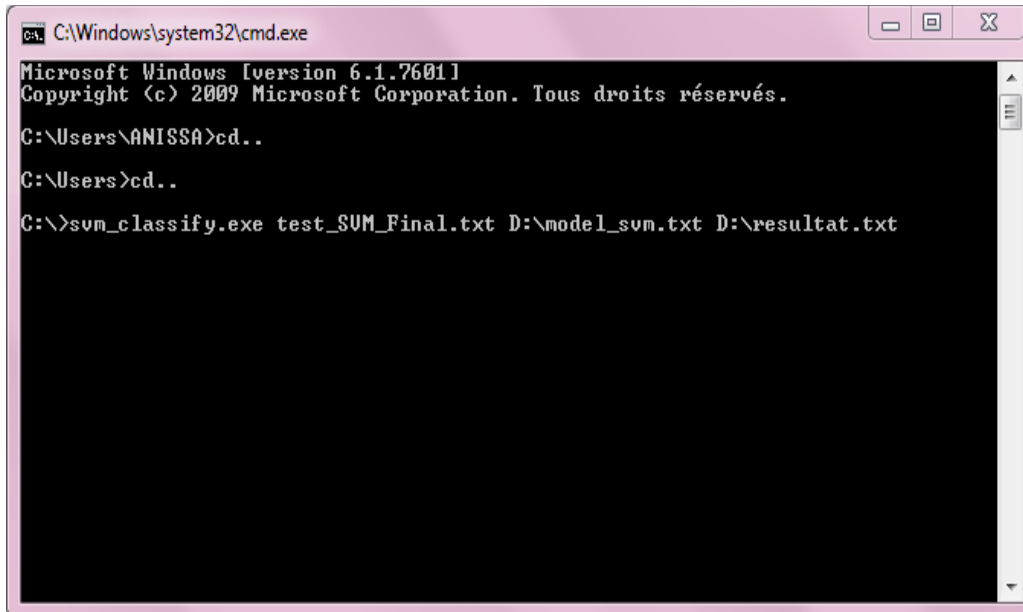


```
Administrateur : C:\Windows\System32\cmd.exe
.....
..done. (8793 iterations)
Optimization finished (4706 misclassified, maxdiff=0.00062).
Runtime in cpu-seconds: 42.60
Number of SU: 14784 (including 14777 at upper bound)
Li loss: loss=13147.57877
Norm of weight vector: |w|=0.02590
Norm of longest example vector: |x|=19577.08735
Estimated UCdim of classifier: UCdim<=257128.27991
Computing XiAlpha-estimates...done
Runtime for XiAlpha-estimates in cpu-seconds: 0.01
XiAlpha-estimate of the error: error<=14.93% (rho=1.00,depth=0)
XiAlpha-estimate of the recall: recall=>86.64% (rho=1.00,depth=0)
XiAlpha-estimate of the precision: precision=>86.64% (rho=1.00,depth=0)
Number of kernel evaluations: 1354250
Writing model file...done
C:\>
```

Figure IV.5 : l'optimisation et la création du modèle.

Chapitre IV : Implémentation et réalisation

Par suite, on passe à la phase du test par le `svm_classify.exe` qui s'applique sur le corpus du test on basant sur le modèle généré dans la phase d'apprentissage et on fin générer le fichier qui contient les résultats de classification.

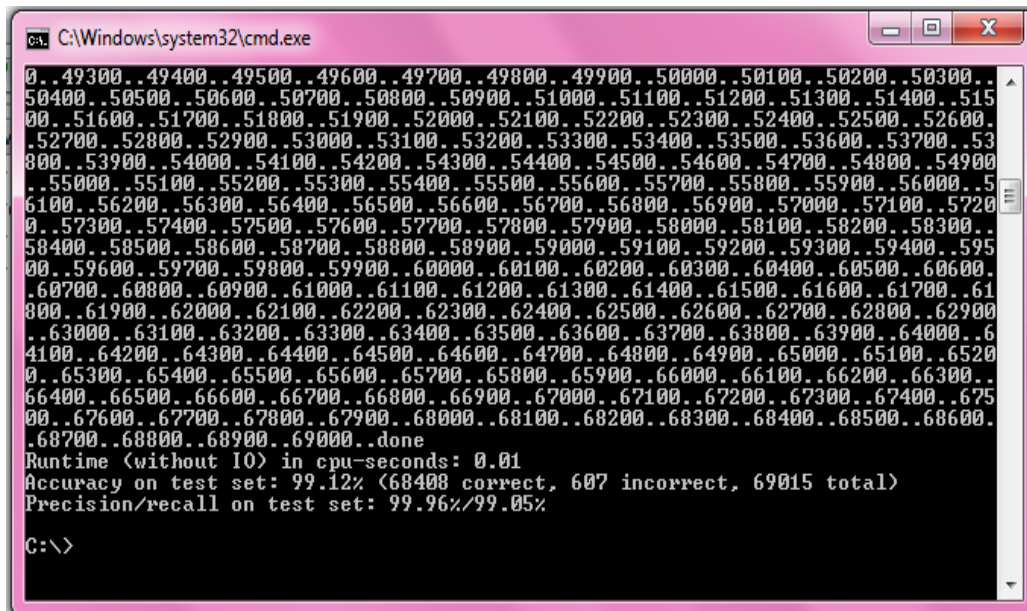


```
C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\ANISSA>cd..
C:\Users>cd..
C:\>svm_classify.exe test_SUM_Final.txt D:\model_svm.txt D:\resultat.txt
```

Figure IV.6 : La fenêtre du `svm_classify`.

On tapant la touche « Entrer », la ligne de commande sera exécuter et les résultats sont enregistrer au niveau de fichier `D:\resultat.txt`.



```
C:\Windows\system32\cmd.exe
0..49300..49400..49500..49600..49700..49800..49900..50000..50100..50200..50300..
50400..50500..50600..50700..50800..50900..51000..51100..51200..51300..51400..515
00..51600..51700..51800..51900..52000..52100..52200..52300..52400..52500..52600..
52700..52800..52900..53000..53100..53200..53300..53400..53500..53600..53700..53
800..53900..54000..54100..54200..54300..54400..54500..54600..54700..54800..54900
..55000..55100..55200..55300..55400..55500..55600..55700..55800..55900..56000..5
6100..56200..56300..56400..56500..56600..56700..56800..56900..57000..57100..5720
0..57300..57400..57500..57600..57700..57800..57900..58000..58100..58200..58300..
58400..58500..58600..58700..58800..58900..59000..59100..59200..59300..59400..595
00..59600..59700..59800..59900..60000..60100..60200..60300..60400..60500..60600..
60700..60800..60900..61000..61100..61200..61300..61400..61500..61600..61700..61
800..61900..62000..62100..62200..62300..62400..62500..62600..62700..62800..62900
..63000..63100..63200..63300..63400..63500..63600..63700..63800..63900..64000..6
4100..64200..64300..64400..64500..64600..64700..64800..64900..65000..65100..6520
0..65300..65400..65500..65600..65700..65800..65900..66000..66100..66200..66300..
66400..66500..66600..66700..66800..66900..67000..67100..67200..67300..67400..675
00..67600..67700..67800..67900..68000..68100..68200..68300..68400..68500..68600..
68700..68800..68900..69000..done
Runtime (without IO) in cpu-seconds: 0.01
Accuracy on test set: 99.12% (68408 correct, 607 incorrect, 69015 total)
Precision/recall on test set: 99.96%/99.05%

C:\>
```

Figure IV.7 : Les résultats de classification en fonction de précision, rappel et exactitude.

Chapitre IV : Implémentation et réalisation

On peut déduire la matrice de confusion d'après les résultats de l'application de SVMLight sur le corpus. *Tableau IV.2* :

| A | B | ← Détecter comme : |
|------|-------|--------------------|
| 7615 | 22 | A anormal |
| 585 | 60792 | B normal |

Tableau IV.2 : La matrice de confusion pour SVMLight.

II. Scénario d'exécution :

Ce scénario d'exécution présente un exemple d'utilisation de notre application

On lance l'application du NetBeans, la première interface contient un menu principale composé de :

- *Chargement* : la ou on charge les connexions à tester,
- *SVM* : l'utilisation de SVMLight sur la connexion choisie,
- *NaiveBayes* : l'utilisation de NaiveBayes sur la connexion choisie,
- *Hybride* : la réalisation de notre modèle hybride (SVM et RB)
- Et finalement *Quitter* : pour sortir de l'application.

Voici comme on le voit sur la **figure IV.8**.

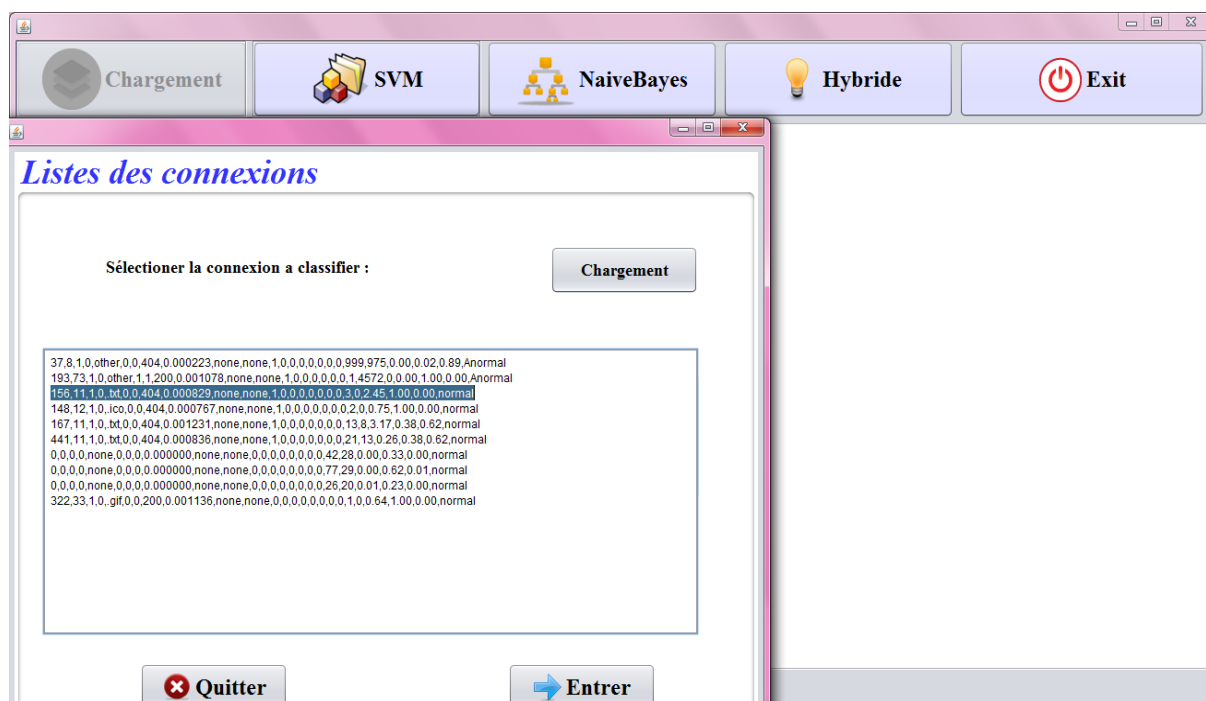


Figure IV.8 : interface principale.

Chapitre IV : Implémentation et réalisation

Si on clique sur le bouton chargement, une deuxième interface se visualise pour la sélection d'une connexion et la tester tout à l'heure par un des trois classificateurs présenter.

La figure IV.9 présente le résultat de SVM sur la ligne choisie accompagné par la classe actuel :

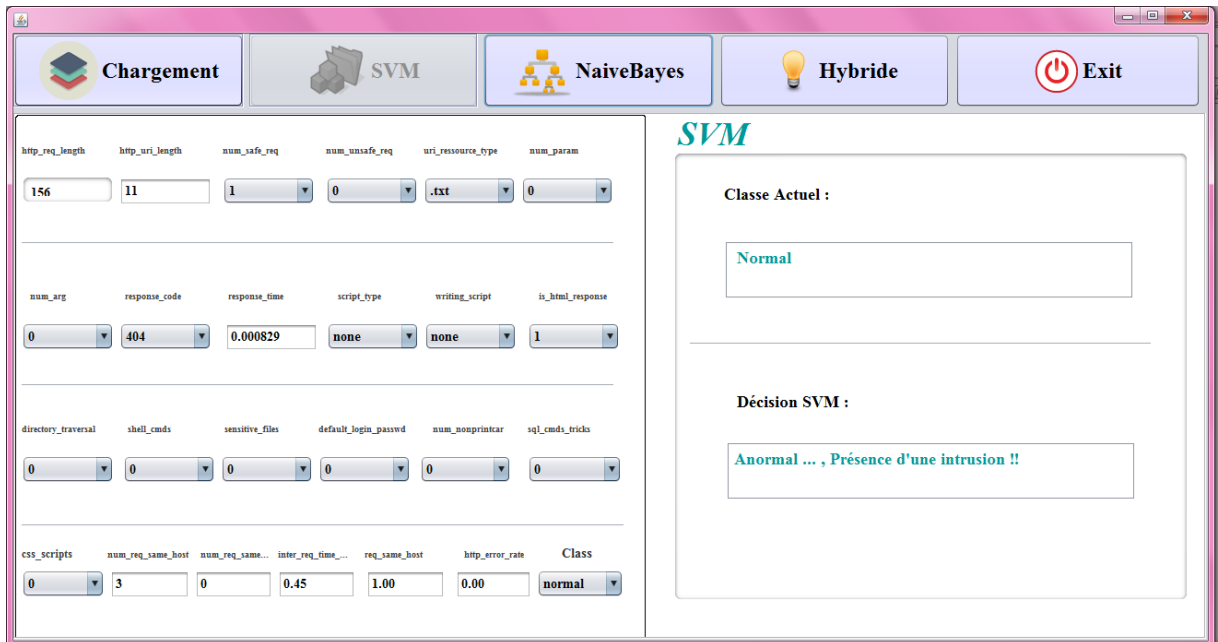


Figure IV.9 : Interface SVM.

La figure IV.10 présente le résultat de NaiveBayes sur la ligne et la classe actuel :

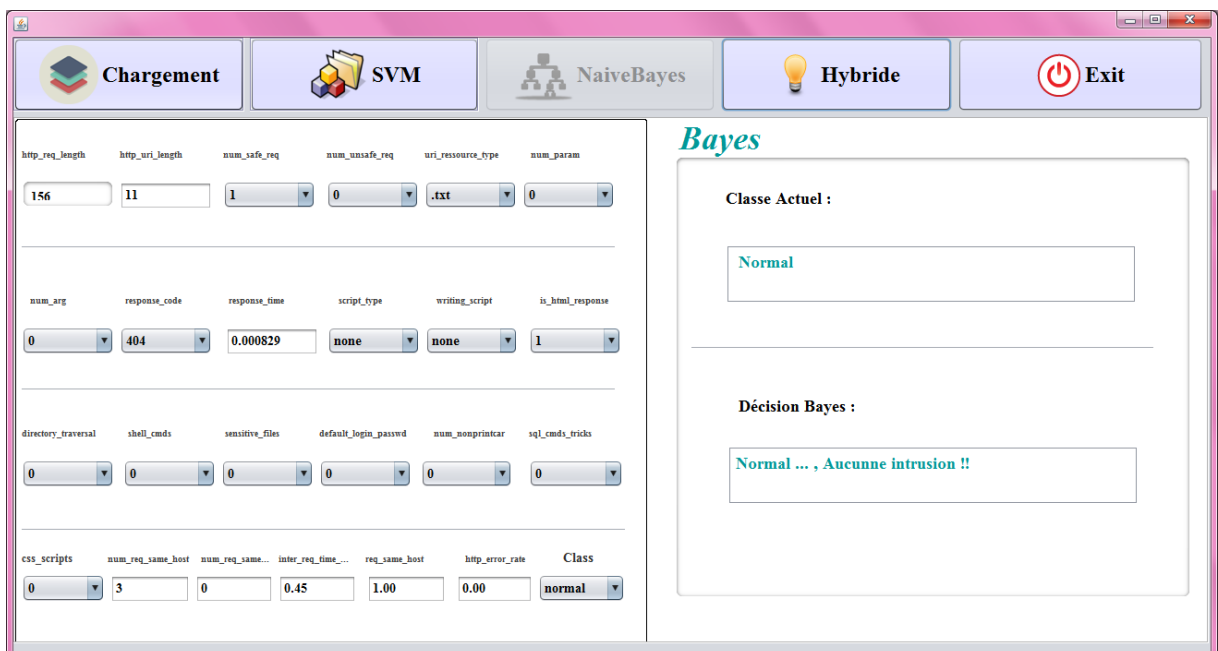


Figure IV.10 : Interface NaiveBayes

Chapitre IV : Implémentation et réalisation

Finalement, la figure IV.11 représente l'interface du modèle hybride. On voit sur la figure la nature de la classe actuelle « normal », la décision de SVM « anormal », la décision de NaiveBayes « normal », et la décision de notre modèle hybride « normal », donc il a bien classé cette connexion.

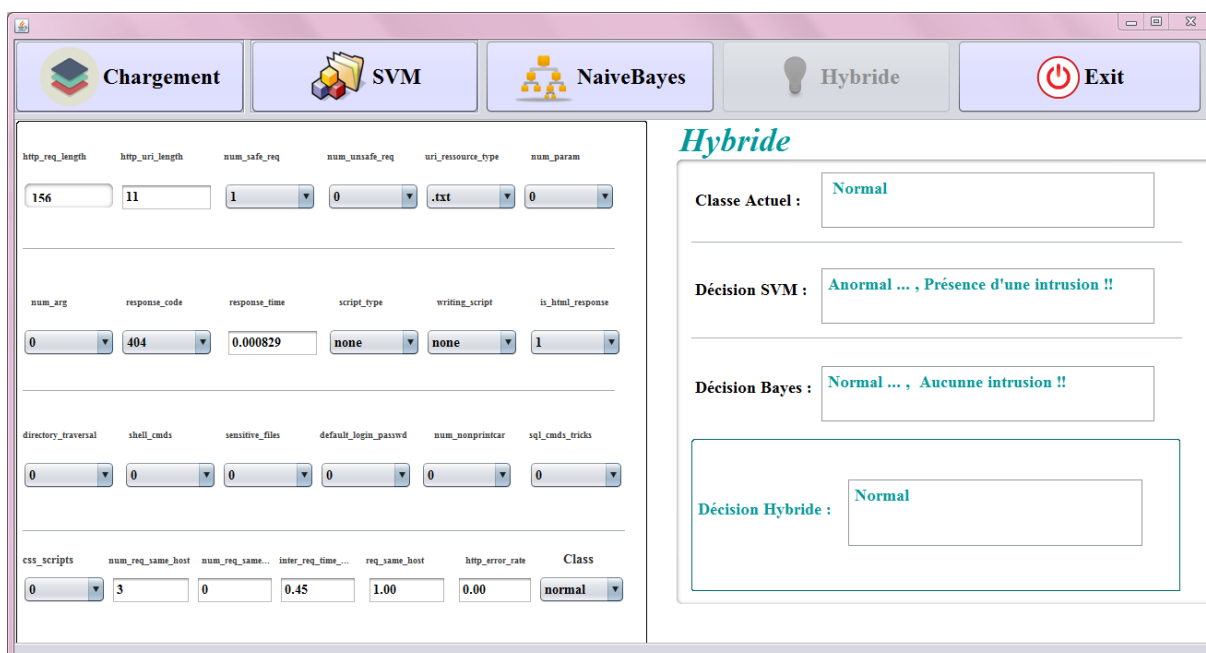


Figure IV.11 : Interface Hybride

Le Tableau IV.3 présente la matrice de confusion de notre approche hybride sur le corpus ;

| Légèrement anormal | A | B | ← Détecter comme : |
|--------------------|-------------|--------------|--------------------|
| 0 | 7615 | 22 | A anormal |
| 8 | 154 | 61216 | B normal |

Tableau IV.3 : La matrice de confusion pour le modèle hybride.

Conclusion :

D'après ce qu'on a présenté dans les chapitres précédents, la sécurité informatique et les systèmes de détection d'intrusion avec les machines à vecteur support (SVM) et les réseaux bayésien qui sont la base de notre étude, on a vu dans ce chapitre la conception et la réalisation d'un prototype de notre modèle pour qu'on puisse tirer l'avantage de fusionner les Apports des fonctions de croyance dans la détection d'intrusion

Chapitre IV : Implémentation et réalisation

SVM et RB pour la minimisation des faux positif et constater les résultats obtenus à la fin de l'exécution.

L'intégration de la théorie des fonctions de croyance dans la classification de système de détection d'intrusion améliore largement la performance de ces derniers au niveau d'exactitude et de minimiser le taux des fausses alertes (faux positif).

Conclusion Général :

Parmi les différents outils de sécurité informatique, on trouve le système de détection d'intrusion. Cet outil est devenu très indispensable pour tout réseau informatique, il nous permet de connaître toutes activités anormales qui peuvent présenter un danger pour notre système.

Le développement des systèmes de détection d'intrusion a passé par deux générations, la première génération ad hoc, cette génération à montrer beaucoup de limites avec la rapidité et l'augmentation du trafic réseau. La deuxième génération a été proposé a fin de traiter les problèmes de la première génération, ou les technique de data mining ont été utilisé. Malgré la puissance et l'efficacité des techniques de data mining ; les systèmes de détection d'intrusion de la deuxième génération souffrent de certains limites comme la nécessité de faire une mise à jour régulière, la nécessité de préparer les données d'apprentissage, la difficulté de détecter les nouvelles formes d'attaques... etc. Les systèmes de détection d'intrusion nommés adaptatifs sont proposés afin de traiter ces limites.

Dans les travaux de la détection d'intrusion, il ya des approches qui utilise plus qu'un classificateur, Dans ce mémoire, nous avons proposés un modèle hybride, proposons une nouvelle approche qui combine deux différents classificateurs dans deux modules (SVM et les Réseaux Bayésiens) qui forme le premier niveau, le deuxième niveau représente la phase de la fuzzification des sorties de chaque classificateurs, dans le troisième niveau on trouve le module de la fusion qui se produit par la théorie des fonctions de croyance .

Notre approche proposée a montré des très hautes performances par rapport a d'autre travaux de recherche.

Dans nos futurs travaux, nous allons résoudre le problème de conflit de la fonction de croyance au niveau du module de fusion. Plus précisément K représente la mesure du conflit entre deux sources. Plus K est important, plus les sources sont en conflit et moins la fusion a de sens. Si $K = 1$ alors le conflit est total et la fusion n'a pas de sens, et sa pose un problème ou la fusion n'aura plus d'utilité.

Finalement, on espère que ce modeste travail a donné un aperçu sur ce domaine de recherche très actif, intéressant, et défiant. Et à montrer l'utilité de la théorie de croyance dans la détection d'intrusion.

Bibliographie :

- [1]: sécurité des systèmes d'information, <https://SécuritéDesSystèmesInforamtion.fr/> consulté 05/01/2016
- [2]: C. Llorens, L. Levier, D. Valois. *Tableaux de bord de la sécurité réseau*. Eyrolles 2^{ème} édition, 2006.
- [3]: ISO-7498-2. *Information processing systems - open systems interconnection- basic reference model - part 2: Security architecture*. Technical report, International Organization for Standardization, Geneva, Switzerland, 1989.
- [4]: International Standard Organisation, «ISO/IEC. TR 13335-1: Guidelines for the Management of IT Security (GMITS): Part 1— Concepts and Models for IT Security», 2000.
- [5]: N. Krawetz. *Introduction to Network Security*. Charles River Media, 2006.
- [6]: J.F. Carpentier. *La sécurité informatique dans la petite entreprise - Etat de l'art et Bonnes Pratiques*. Editions ENI, 2009.
- [7]: National Institute of Standards and Technology, « Risk Management Guide for Information Technology Systems», NIST, Special publication 800-30, July 2002.
- [8]: Donald L.P., « Sécurité des systèmes d'information, protection globale de l'entreprise» CampusPress, 2000. ISBN 2-7440-0948-2
- [9]: CERT – Computer Emergency Response Team, www.cert.org consulté le 15/01/2016
- [10]: CSI – Computer Security Institute www.gocsi.com consulté le 15/01/2016
- [11]: G. McGraw, « Managing software security risks», Computer Volume 35, Issue 4, March 2002. Page(s):99 – 101.
- [12]: G. McGraw, « Software security», Security & Privacy Magazine, IEEE Volume 2, Issue 2, Mar-Apr 2004. Page(s):80 – 83.
- [13]: E. Farman, Rapport de stage, Cursus AFPA TSRIT 2011-2012, Lieu: Hôtel-de-ville de Pertuis du 30/01/2012 au 16/02/2012.

- [14][12mimoun]: *livre de la sécurité informatique*, février 2004. URL : <http://www.securiteinfo.com>. Consulté le 20/01/2016
- [15]: V. Remazeilles. *La sécurité des réseaux avec CISCO*. Editions ENI, Février 2009.
- [16][13minoun] : J. Postel. *RFC 793- Transmission Control Protocol-* Internet Engineering Task Force- www.ietf.org. Standard, 1981.
- [17][14mimoun] : J. Postel. *RFC 792- Internet Control Message Protocol-* Internet Engineering Task Force- www.ietf.org. Standard, 1981.
- [18]: [9mimoun] : A. Serhrouchni et C. Llorens. *Mesure de la sécurité "logique" d'un réseau d'un opérateur de télécommunications*. PhD thesis, Novembre 2005.
- [19]: [16mimoun] : J.P. Anderson. *Computer security threat monitoring and surveillance*. Technical report, James P. Anderson Company, Fort Washington, USA, Avril 1980.
- [20]: [1mimoun] : S. Benfarhat, T. Kenaza et A. Mokhtari *.Modèles graphiques probabilistes pour la corrélation d'alertes en détection d'intrusions*. PhD thesis, 2011.
- [21]: [Denning 1987] Denning, D. E. (1987). An intrusion-detection model. 13(2):222–232
- [22] : [Meier 2004] Meier, M. (2004). Intrusion detection systems list and bibliography.
- [23] :[Bace et Mell 2001] Bace, R. et Mell, P. (2001). Intrusion detection systems. Rapport technique, National Institute of Standards and Technology (NIST).
- [24]: [Wood ET Erlinger 2007] Wood, M. ET Erlinger, M. (2007). Intrusion detection message exchange requirements. IETF Intrusion Detection Exchange Format Working Group. Request for Comments. Reference: 'rfc4766'.
- [25]:<http://www.inetdoc.net/guides/tutoriel-cu/tutoriel.securite.protection.nids.html>
- [26]: <http://www.linuxfocus.org/Francais/May2003/article292.shtml>
- [27]: H. Debar, M. Dacier, et A. Wespi. *Towards a taxonomy of intrusion detection systems*. Computer Networks, Elsevier, 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.

- [29]: http://igm.univ-mlv.fr/~dr/XPOSE2009/Sonde_de_securite_IDS_IPS/IDS.html
consulté le 17/03/2016
- [30]: Nathalie Dagorn. Détection et prévention d'intrusion : présentation et limites. [ResearchReport] 2006. <inria-00084202>
- [31]: Guillaume Claudio 2004 une méthode de classification non supervisé pour l'apprentissage des règles et la recherche d'information.
- [32]: Cours_KIS_2012-2013_SVM. Versailles St Quentin, France. janvier 2012-2013
- [33]: Yiming yang Xin Lui , A re_examination of text categorization method.
- [34]: [8]Thorsten Joachims transductive interface for text classification using support vector machines.
- [35]: H.Mohamadally et B.Fomani , " SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges" Versailles St Quentin, France . janvier 2006.
- [36]: Abdelhamid DJEFFAL, Utilisation des méthodes supports vector achine (SVM) dans l'analyse de base de données ,thèse de doctorat .
- [37]: J. Weston c. Watkins Multi class support vector machines.
- [38]: J. Callut, «Implémentation efficace des Support vector Machines pour la classification » Mémoire présenté en vue de l'obtention du grade de Maître en informatique. Université libre de Bruxelles . département informatique, 2003
- [39]: A. Cornuéjols, L. Miclet, Y.Kodratoff, « Apprentissage Artificiel, Concepts et algorithmes » ISBN 2-212-11020-0 , 2002.
- [40]: Essai présenté au CeFTI en vue de l'obtention du grade de maître en génie logiciel (maîtrise en génie logiciel incluant un cheminement de type cours en génie logiciel)
- [41] Thèse de doctorat (fusion de données avec les réseaux bayésiens pour la modélisation du système dynamique et son application en télémédecine).
- [42]: D. M. Checkering. *Learning Bayesian networks is NPcomplete*. Dans Learning from Data : Artificial Intelligence and Statistics V. Springer-Verlag, 1996.

- [43]: P. Spirtes, C. Glymour et R. Scheines. *Causation, Prediction, and Search, Second Edition (Adaptive Computation and Machine Learning)*. The MIT Press, Janvier 2001.
- [44]: C. Chow et C. Liu. *Approximating discrete probability distributions with dependence trees*. Information Theory, IEEE Transactions on, 1968.
- [45]: G. F. Cooper et E. Herskovits. *A Bayesian Method for the Induction of Probabilistic Networks from Data*. Mach. Learn., 1992.
- [46]: A. M. Carvalho et A. L. Oliveira. *Learning Bayesian Networks Consistent with the Optimal Branching*. Dans Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA). Los Alamitos, USA, 2007.
- [47]: P. Leray. *Réseaux bayésiens : apprentissage et modélisation de systèmes complexes*. Mémoire présenté en vue de l'obtention de l'habilitation à diriger des recherches, département ASI, INSA, Rouen, 2006.
- [48]: David Heckerman. *A tutorial on learning with Bayesian networks*, 1999.
- [49]: S;L. Lauritzen et D. J. Spiegelhalter. *Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems*. Journal of the Royal Statistical Society, Series B, 1988.
- [50]: R.D. Shachter et M.A. Peot. *Simulation Approaches to General Probabilistic Inference on Belief Networks*. Dans Proceedings of the Conference on Uncertainty in Artificial Intelligence. Amsterdam, The Netherlands, 1990.
- [51]: R. Fung et D.B. Favero. *Backward Simulation in Bayesian Networks*. Dans Proceedings of the Conference on Uncertainty in Artificial Intelligence. San Francisco, USA, Juillet 1994.
- [52]: N. Friedman, D. Geiger, et M. Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, 1997.
- [53]: E. J. Keogh et M.J. Pazzani. *Learning the Structure of Augmented Bayesian Classifiers*. International Journal on Artificial Intelligence Tools, 2002.

- [54]: J. Cheng et R. Greiner. *Learning Bayesian Belief Network Classifiers : Algorithms and System*. Dans Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of Intelligence : Advances in Artificial Intelligence, 2001.
- [55]: J. Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Morgan Kaufmann Publishers Inc.. San Francisco, USA, 1988.
- [56] : ZONE-project_SVM_Ameni_Bouaziz (Catégorisation automatique de news à l'aide de techniques d'apprentissage supervisé).
- [57]: G. Shafer : A mathematical theory of evidence. Princeton University Press, 1976.
- [58]: A.P. Dempster : Uper and Lower probabilities induced by a multivalued mapping. Anals of Mathematical Statistics, 38:325339, 1967.
- [59]: la fusion d'information Polycopié de cours ENSIETA - Réf. : 1484 Arnaud MARTIN. Janvier 2005
- [60] : Théorie des fonctions de croyance pour la fusion et l'évaluation de la pertinence des sources d'informations: application à un bioprocédé fermentaire (1 Groupe de Recherche en Informatique et Mathématiques Appliquées des Antilles et de la Guyane, Université des Antilles et de la Guyane 97159 Pointe-à-Pitre Guadeloupe, France ; 2 Laboratoire d'Analyse et d'Architecture des Systèmes, 31077 cedex 04 Toulouse, France,).
- [61]: A. Josang : The Consensus Operator for Combinning Beliefs. Arti_cial Intelligence Journal, 141(1-2):157_170, 2002.
- [62]: D. Dubois et H. Prade : Representation and combination of uncertainty with belief functions and possibility measures. Computational Intelligence, 4:244_264, 1988.
- [63]: <http://svmlight.joachims.org/> consulter 07/05/2016
- [64] : Une exploration de données de système hybride de détection d'intrusion sur la base de réseaux locaux sans fil, Moorthy, M .; Sathiyabama, S.juillet 2012 International Journal of Computer Applications, 01/07/2012, vol. 50, p19
- [65] : Salem Benfarhet et all Réseaux Bayésiens naïfs pour la détection des attaques coordonnées Author manuscript, published in "Journées Francophone sur les Réseaux Bayésiens, Lyon : France (2008)"

[66] : weka Licence Maitrise d'Informatique de Lille d'informatique de Lille. Maitrise d'informatique (2003/2004) Intelligence artificielle.

Annexe A : Projet PLACID

Le projet PLACID (Probabilistic graphical model and Logics for Alarm Correlation in Intrusion Detection) est un projet de l'agence nationale de recherche (ANR).

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 corrélér des alertes. Une partie des contributions de cette thèse s'inscrit dans le cadre de ce projet.

Attributs du corpus du projet PLACID :

Le corpus du PLACID est constituer de 24 attributs et la classe actuelle qui est soit une attaque soit normal.

Les attaques définit dans ce corpus sont : {R2L, bo, value_mis, iv_R2L, il_R2L, flooding, URL_R2L, poor_DoS, DoS, vul_scan, XSS, shell_cmds, sqli, sqli_auth, normal, new}

Le tableau suivant montre la description des attributs du corpus PLACID et leurs valeurs.

| | Attribut | Valeurs |
|-----|---------------------|--|
| A1 | http_req_length | numeric |
| A2 | http_uri_length | numeric |
| A3 | num_safe_req | numeric |
| A4 | num_unsafe_req | numeric |
| A5 | uri_ressource_type | .swf, .exe, other, none, .htm, cgi-bin, .asp, .pl, .dat, .gif, .ps, .pdf, .css, .jpg, .php, .tar, .txt, .ico, .ppt, .js, .zip, .cgi, .asm, .sh, .mov, .com, .ini |
| A6 | num_param | numeric |
| A7 | num_arg | numeric |
| A8 | response_code | 0,200,201,202,203,204,205,206,300,301,302,303,304,305,306,307,400,401,402,403,404,405,414,416,500 |
| A9 | response_time | numeric |
| A10 | script_type | none, js, vb |
| A11 | writing_script | none, cookie_set, doc_loc, cookie_rd |
| A12 | is_html_response | 0,1 |
| A13 | directory_traversal | 0,1 |
| A14 | shell_cmds | 0,1 |

| | | |
|-----|---------------------------------|---------|
| A15 | sensitive_files | 0,1 |
| A16 | default_login_passwd | 0,1 |
| A17 | num_nonprintcar | numeric |
| A18 | sql_cmds_tricks | 0,1 |
| A19 | css_scripts | 0,1 |
| A20 | num_req_same_host | numeric |
| A21 | num_req_same_URI | numeric |
| A22 | inter_req_time_inter val | numeric |
| A23 | req_same_host_diff_ URI_rate | numeric |
| A24 | http_error_rate | numeric |