(

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Centre Universitaire BELHADJ BOUCHAIB d'Aïn-Témouchent

Institut des Sciences
Département des Mathématiques et de l'Informatique

Mémoire

Pour l'obtention du Diplôme de Master en Informatique
Option : Réseaux et Ingénierie des Données (RID)

Présenté par :
M. DJEBLI AbdeNour
M. BENBAKOUR Yacine

SYSTÈME DE DÉTECTION D'INTRUSION INFORMATIQUE ET LA DÉTECTION D'ERREURS

Encadrant :
Dr.BELGRANA Fatima zohra
Maître de Conférences Classe "B" au C.U.B.B.A.T.

Soutenu en 2019

Devant le jury composé de :

Président :	M. BENZERBADJ Ali (M.C.B)	C.U.B.B.A.T.
Examinateurs :	Melle. BERRAKAM Fatima (M.A.A)	C.U.B.B.A.T.

Remerciement

Remerciement

Tout d'abord nous remercions le bon dieu tout puissant qui nous a guidé dans nos pas, qui nous a donnés le courage, la volonté, et la patience pour surmonter les épreuves que nous avons rencontré tout au long de nos vies .

Nous tenons à exprimer nos sincères gratitudes et reconnaissances à notre encadrante Mme BELGRANA Fatima Zohra maître de conférences au CUBBAT pour avoir accepté de dériger ce travail, pour son aide, sa disponibilité et ses conseils judicieux .

Nous remercions Mr BENZERBADJ Ali maître de conférences au CUBBAT d'avoir accepté d'examiner ce modeste travail. Nous remercions Mlle BERRAKAM maître assistante au CUBBAT d'avoir accepté d'examiner ce travail.

En fin, nous remercions toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

Dédicace

Je dédie ce modeste travail à : Mes chers parents qui m'ont énormément encouragé et soutenu pendant la réalisation de ce travail, que je ne pourrai assez remercier de m'avoir apporté le nécessaire. Mes chers frères , ma sœur et toute la famille. A mes chers amis. Tous ceux qui m'ont aidé de près ou de loin surtout dans les moments les plus difficiles.

A tous, je dédie ce modeste travail...

Mr. BENBAKOUR Yacine

Je dédie ce présent travail : A mes parents.

A tous ceux qui m'ont aidé de près ou de loin, . . .

A tous et à toutes, je dédie ce modeste travail. Qu'ils trouvent ici l'expression de toute mon affection..

Mr. DJEBLI AbdelNour

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

Introduction générale

Introduction générale

L'informatique et en particulier l'Internet jouent un rôle grandissant dans notre société. Un grand nombre d'applications critiques d'un point de vue de leur sécurité sont déployées dans divers domaines comme le domaine militaire, la santé, le commerce électronique, etc. La sécurité des systèmes informatiques devient alors une problématique essentielle tant pour les individus que pour les entreprises ou les états.
La meilleure façon de protéger un réseau ou un système informatique est de détecter les attaques et de se défendre avant même qu'elles ne puissent causer des dommages. Les systèmes de détection d'intrusions sont alors l'une de ces contre-mesures les plus efficaces. Leur rôle est de reconnaître des intrusions ou des tentatives d'intrusions par des comportements anormaux des utilisateurs ou par la reconnaissance d'attaque à partir du flux des données du réseau.
Les méthodes de détection d'intrusions reposent sur l'observation d'un certain nombre d'événements et sur l'analyse de ceux-ci. Il s'agit premièrement de collecter les informations que l'on souhaite analyser. Ces informations proviennent des fichiers journaux du système, d'applications spécifiques (tels que des serveurs web, serveurs ftp, serveurs de courriers électroniques, etc.) ou des sondes mises en place par les outils de détection d'intrusions tels des « sniffers » réseau, des modules spécifiques à certaines applications ou au système d'exploitation. On distingue généralement deux grands types de sources de données et l'on peut classer les SDI suivant leurs sources : les flux réseau (SDI basé réseau SDIR) et les données système ou applicatives (SDI basé Hôte ou SDIH) [Eddy, 2007].
Différentes méthodes et approches ont été adoptées pour la conception de systèmes de détection d'intrusions qui dérivent généralement de l'Intelligence Artificielle (IA) telle que les Réseaux de Neurones Artificiels (RNAs), les Séparateurs à Vaste Marge(SVM), les méthodes de partitionnement, la machine à états finis ou même les modèles statistiques.
Tout au long de notre travail nous allons nous situer dans une approche de SDIR. Notre principal objectif est de proposer une nouvelle technique de détection d'intrusion tout en réduisant les taux de faux positifs et de faux négatifs ce qui est traduit par une amélioration de la précision, du rappel ainsi que du taux de réussite. Notre approche proposée est basée principalement sur les réseaux de neurones et plus précisément du type FBR (RN à Fonctions de Base Radiales) caractérisé par sa fonction d'activation (Fonction Gaussienne) connue pour sa robustesse au

bruit. Notons toute fois que la notion du temps dans ce contexte est très importante ainsi que l'utilisation des Systèmes Multi-Agents(SMAs) permet non seulement une réduction du temps de traitement grâce au parallélisme mais aussi une distribution d'expertises ainsi qu'une simulation du SDIR.
Afin d'assurer le bon fonctionnement de notre SDI même en présence d'erreur qui représente dans notre cas les faux positifs et les faux négatifs, nous nous sommes inspirés du domaine de sureté de fonctionnement en introduisant un nouveau moyen pour renforcer la protection contre les attaques. Le flux concerné est d'abord détecté via à une double analyse puis il est non seulement bloqué mais aussi inclus dans une nouvelle base d'apprentissage. Ce mécanisme permet de déceler les états incorrects de notre SDIN suite à une mauvaise détection.
Afin de réaliser l'apprentissage de notre FBR et de valider notre approche nous allons utiliser le jeu de données CICIDS2017, une base de données réaliste, récente, qui contient plusieurs types d'attaque et qui est étiquetée ce qui permet une comparaison avec la vérité-terrain et par conséquent une évaluation de notre SDIN. Nous avons notamment réalisé une étape de prétraitement via la méthode K-means afin de créer les bases de test et d'apprentissage de telle sorte qu'elles soient le plus similaire possible.
Notre mémoire est organisé comme suit :
Le deuxième chapitre présente les systèmes de détection d'intrusion, leurs fonctionnements et leurs classifications selon différents critères à savoir l'emplacement du SDI, les méthodes de détection, les types de réponses ainsi que la fréquence d'utilisation.
Le troisième chapitre décrit les méthodes de classification ainsi que d'autres techniques destinées particulièrement aux SDIs où nous présentons un état de l'art de ce dernier. Nous nous focalisons par la suite sur les approches que nous avons adoptées où nous introduisons les RNAs, leurs techniques d'apprentissage et leurs architectures. Nous introduisons aussi quelques concepts de bases sur les SMAs puis aussi quelques travaux ayant fait appel à ce paradigme en vue d'une détection.
Dans le quatrième chapitre, nous allons présenter les étapes nécessaires à l'aboutissement de notre SDIN et cela à partir de l'étape de prétraitement du jeu de données CICIDS201 jusqu'aux applications, méthodes et données utilisées. Nous présentons ensuite notre application, ou nous illustrons les résultats obtenus en évaluant les performances de notre SDIN RBF-SMA accompagnés de quelques discussions.

Chapitre 2

Systèmes de détection d'intrusion

Sommaire

2.1 Introduction

La sécurité des systèmes d'information encourage les entreprises à chiffrer leurs données et les informations sensibles. Ces outils défensifs sont très essentiels à la protection des données numériques d'une immense majorité d'utilisateurs honnêtes ; cette structure est chargée de coordonner et d'aider les entreprises à se protéger des cybers attaques. Les experts se protègent du vol de données suite à une attaque informatique ou un intrus qui a été infiltré grâce à des failles de sécurité des systèmes .Il existe aussi d'autres solutions techniques qui sont nombreuses et complémentaires comme les systèmes de détection d'intrusion(SDI) qui surveillent les connexions au réseau de l'entreprise et protègent l'information des activités malveillantes, filtrent les connexions entrantes et sortantes du réseau informatique selon des règles établies par l'administrateur . Cette technique est efficace pour empêcher l'intrusion.

2.2 Définition, motivation et limites des SDIs

2.2.1 Définition [Touati ,2016]

La détection des intrusions est le processus de surveillance des événements qui se trouvent dans un système des ordinateurs ou du réseau et en les analysant pour détecter les signes des intrusions, définis comme des tentatives pour compromettre la confidentialité, l'intégrité,la disponibilité ou éviter des mécanismes de sécurité de l'ordinateur ou du réseau.
L'intrusion est causée par les attaques accédant au système via Internet, autoriser l'utilisateur du système qui essaye de gagner les privilèges supplémentaires pour lesquels ils n'ont pas été autorisés, et autoriser les utilisateurs qui abusent des privilèges donnés. Le système de détection des intrusions est un logiciel ou un matériel qui automatise des surveillances et les processus analysés .
Les SDIs protègent un système contre les attaques, les mauvaises utilisations et les compromis. Ils peuvent également surveiller l'activité du réseau, analyser les configurations du système et du réseau contre toute vulnérabilité, analyser l'intégrité de données et bien plus.
Il existe deux types généraux de systèmes : un SDI basé sur l'hôte (SDIH) et un SDI basé sur le réseau (SDIR).
Un SDIR est souvent une application matérielle autonome qui inclut des fonctionnalités de détection de réseau. Il s'agira généralement de capteurs matériels situés à divers endroits du réseau. Il peut également s'agir d'un logiciel installé sur divers ordinateurs connectés via le réseau. Le SDIR analyse les paquets de données entrants

et sortants et offre une détection en temps réel.
Un SDIH analyse le trafic en provenance et à destination de l'ordinateur spécifique sur lequel le logiciel de détection d'intrusion est installé. Un système basé sur un hôte a également la capacité de surveiller les fichiers système clés et toute tentative d'écraser ces fichiers.

2.2.2 Motivation

Parmi les principales fonctionnalités du SDI est que ce dernier donne une image claire sur ce qui se passe dans le réseau et le système, il peut détecter, reconnaitre les attaques et alerter le système tout en offrant une plus grande flexibilité et intégrité à l'infrastructure de sécurité existante. Un SDI permet une analyse en temps réel en surveillant le réseau ou les systèmes en temps réel. Il peut également prendre des réponses actives, comme le blocage des adresses IP, l'arrêt des connexions. Il peut aussi enregistrer les sessions en format spécifié. Plus important encore, l'SDI fournit des lignes directrices qui aident dans le développement de la politique de sécurité de l'organisation. Il fournit et améliore le processus d'enquête à l'aide de journaux de sessions et la corrélation des événements. Il permet de manière efficace de donner des rapports pour la gestion le flux d'informations [Chikouche ,2012].
Parmi les raisons pour lesquelles il est intéressant de faire appel aux SDIs nous citons :

- Un SDI offre une description qui fournit des informations potentielles sur les interventions et les tentatives réalisées, permettant d'améliorer le diagnostic, le rétablissement et la correction des facteurs de causalité.
- Il contrôle la qualité de la sécurité et de la gestion, en particulier dans les grandes entreprises
- Il permet de détecter les cibles des attaques.
- Il permet d'éviter les problèmes en renforçant la protection contre les risques découverts d'attaques ou d'usage abusif du système.
- Un SDI permet de détecter les attaques et les violations de la sécurité qui ne sont pas couvertes par d'autres mesures de sécurité. [Philip ,1998]

2.2.3 Limites [Chikouche ,2012]

C'est l'une des questions les plus courantes pour les nouveaux chercheurs en matière de détection d'intrusion.La plupart des gens pensent que le SDI peut protéger leur réseau, cela n'est pas vrai, en tenant compte des observations suivantes :

- Il est tout simplement actif, pas pro actif (optimiste).

• Il ne peut pas empêcher l'attaque.
• Il n'est pas automatisé, il a besoin d'importantes ressources humaines pour sa gestion.
• Il ne peut pas offrir une protection complète pour les ressources. C'est juste une couche supplémentaire.
• Il ne peut pas compenser les lacunes des protocoles réseau.
• Il ne peut pas protéger tous les types d'attaques. Il a des limites.
• Il ne peut pas résister à des volumes élevés et des vitesses élevées de trafic.

2.3 Classification des SDIs

Il existe plusieurs types d'SDI disponibles aujourd'hui, caractérisés par différentes approches en matière de surveillance et d'analyse. Chaque approche a ses avantages et ses inconvénients. Toutes les approches peuvent être décrites en matière de modèle SDI. Parmi les critères de classification on retrouve :

• L'emplacement du SDI
• Méthodes de détection
• Les types de réponse
• La fréquence d'utilisation.

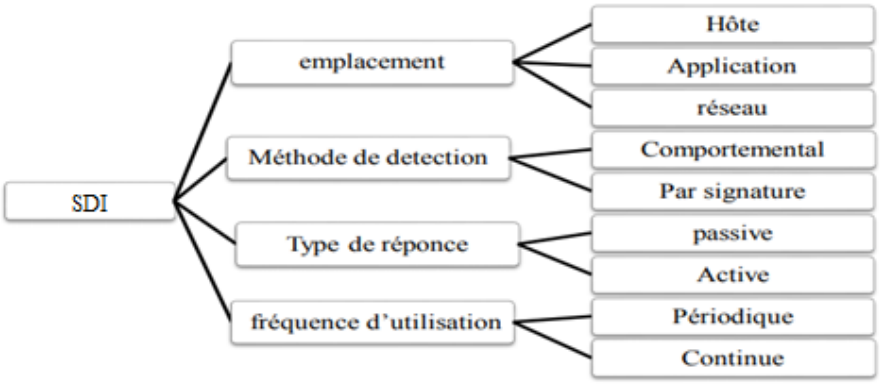

Figure 2.1 – Taxonomie de classification [Bourouh et kanoun ,2017]

2.3.1 L'emplacement de l'SDI [Wespi et al ,1999]

Le moyen le plus courant de classer le SDI est de le regrouper en fonction de l'emplacement de la source d'information où il fonctionne. Les principales sources

d'information sont : les paquets réseaux capturés à partir des réseaux, les jeux de puces LAN, les systèmes d'exploitation et les fichiers critiques.

1.3.1.1 Les Systèmes de Détection d'Intrusion Hôte SDIHs [Abbass et Aouadi ,2016]

Les SDIHs analysent le fonctionnement et l'état des machines sur lesquelles ils sont installés afin de détecter les attaques. L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent être levées

Avantages des SDIHs :

Parmi les avantages des SDIHs, nous citons :

- Surveiller les événements locaux pour héberger et détecter les attaques qui n'ont pas été vues par SDIR.
- Opérez dans un environnement où le trafic réseau est crypté, lorsque des sources d'information basées sur l'hôte sont créées avant le cryptage des données ou le décryptage des données dans l'hôte de destination.
- Il n'est pas possible d'accéder à SDIH via un commutateur réseau
- Lors de l'examen du SDIH sur le SE, ils peuvent détecter des chevaux de Troie ou d'autres attaques liées au programme.

Inconvénients des SDIHs :

Les SDIHs présentent quelques inconvénients qui sont :

- Difficulté de gestion : les informations doivent être configurées et gérées pour chaque hôte surveillé
- Puisque les sources d'information du SDIH résident sur l'hôte qui est la cible de l'attaque, l'SDIH peut être attaqué et neutralisé comme une partie de l'attaque.
- Un SDIH peut être neutralisé par certaines attaques de type Dos.
- Un SDIH n'est pas bon pour la surveillance du réseau entier parce qu'il ne voit que les paquets du réseau reçus par ses hôtes

1.3.1.2 Les Systèmes de Détection d'IntrusionApplication SDIAs [Zaidi ,2013]

Un raffinement du SDI basé hôte est le SDI basé Application (SDIA). Alors que le SDIH examine un système unique pour la modification de fichier, le système SDI basé application examine une application afin de détecter les événements anormaux. Il fait habituellement cet examen en regardant le fichier créé par l'application à la recherche d'occurrences anormales, tels que des utilisateurs dépassant leur autorisation, des exécutions de fichiers non valides ou d'autres activités indiquant un problème lors

de l'interaction normale entre les utilisateurs, l'application et les données .
En suivant l'interaction entre les utilisateurs et les applications, le SDIA est capable de retracer une activité spécifique jusqu'à des utilisateurs individuels. Un avantage unique ce type d'SDI est sa capacité à afficher des données cryptées .

Avantages des SDIAs :

Parmi les avantages des SDIAs nous citons :

- Le SDIA connaît les utilisateurs spécifiques et peut observer l'interaction entre l'application et l'utilisateur, cela lui permet d'attribuer des activités non autorisées à des utilisateurs spécifiques et connus.
- Le SDIA est capable de fonctionner même lorsque les données entrantes sont cryptées, car il est capable de fonctionner au moment du processus lorsque les données ont été décryptées par les applications avant d'être recrypté pour le stockage.

Inconvénients des SDIAs :

- Les SDIs d'applications sont moins capables de détecter les altérations logicielles et peuvent être capturées par un code de cheval de Troie ou d'autres formes d'utilisation frauduleuse. Il est généralement recommandé d'utiliser SDIA dans une combinaison avec SDIH et SDIR.

1.3.1.3 Les Systèmes de Détection d'Intrusion Réseaux SDIRs

Les systèmes de détection d'intrusion sur le réseau sont de plus en plus importants pour la sécurité du réseau. Les logiciels antivirus et les pare-feu ne peuvent bloquer que les entrées non autorisées dans un système. Les systèmes de détection d'intrusion recherchent des modèles d'activité réseau pour identifier les activités malveillantes. La nécessité de cette catégorie de système de sécurité est apparue en raison des changements apportés aux méthodes de piratage en réaction aux stratégies précédentes visant à bloquer les activités malveillantes [Cooper ,2018].

Avantages des SDIRs : [TranTay et Hugo ,2005]

Parmi les avantages des SDIRs, nous citons :

- Le SDIR peut surveiller un grand réseau.
- Le déploiement du SDIR a peu d'impact sur un réseau existant. Les SDIRs sont habituellement des dispositifs passifs qui écoutent le réseau sans interférer l'opération normale d'un réseau. Ainsi, il est habituellement facile de monter en rattrapage un réseau pour inclure le SDI avec l'effort minimal.
- Le SDIR peut être très sûr contre l'attaque et peut savoir même la source de l'attaquant.

Inconvénients des SDIRs : [Abbass et Aouadi ,2016] Malgré leurs avantages,

les SDIRs présentent certains inconvénients à savoir :

- Un SDIR ne peut pas traiter tous les paquets circulant sur un grand réseau.
- Quelques fournisseurs essayent à implémenter le SDIR sur le matériel pour qu'il marche plus rapidement.
- Le SDIR ne peut pas analyse des informations chiffrées (cryptées). Ce problème a lieu dans les organisations utilisant le VPN.

2.3.2 Méthodes de détection d'intrusion

Pour une bonne gestion d'un système de détection d'intrusion, alors il est nécessaire de connaitre le fonctionnement d'un tel système, plusieurs questions se posent, il s'agit alors de savoir comment détecter une intrusion et comment différencier un flux malveillant d'un flux normal. Il existe deux techniques de détection d'attaque, la première se base sur la détection des signatures d'attaques dans des paquets c'est alors l'approche basée sur les connaissances. Tan dit que la deuxième consiste à détecter une activité suspecte dans le comportement de l'entité surveillée (utilisateur, machine, service, application), il s'agit alors de l'approche comportementale

[Labed ,2006]

1.3.2.1 L'analyse comportementale [Robin et Paxson ,2010]

Dans l'analyse comportementale, on construit un modèle du comportement naturel du système. . Cet échantillon s'appelle une définition des comportements naturels qui serviront de référence dans la détection. Lors de la surveillance du système, toute déviation significative de son comportement actuel contrôlé par rapport au comportement de référence normal signifie qu'il s'agit d'une attaque. Cette approche possède un certain nombre d'avantages et d'inconvénients .

Avantages de l'analyse comportementale :

- Elle ne nécessite aucune connaissance préalable des attaques.
- Elle permet de détecter les abus de privilèges.
- Elle permet la production d'informations pouvant servir à identifier des signatures pour une analyse basée sur les connaissances.

Inconvénients de l'analyse comportementale

- Un pourcentage élevé de fausses alarmes positives est généré en raison du comportement inattendu des utilisateurs du réseau.
- Cette approche nécessite des phases d'apprentissage qui distinguent les schémas comportementaux normaux.
- Les avertissements issus de cette approche ne sont pas importants.

1.3.2.2 L'analyse par signature [Debar et al,2000]

Cette approche consiste à rechercher dans l'activité de l'élément surveillé les empreintes (ou signatures) d'attaques connues. Ce type d'SDI est purement réactif ; il ne peut détecter que les attaques dont il possède la signature. De ce fait, il nécessite des mises à jour fréquentes. De plus, l'efficacité de ce système de détection dépend fortement de la précision de sa base de signature. C'est pourquoi ces systèmes sont contournés par les pirates qui utilisent des techniques dites "d'évasion" qui consistent à maquiller les attaques utilisées. Ces techniques tendent à faire varier les signatures des attaques qui ainsi ne sont plus reconnues par le SDI.

Avantages de l'analyse par signature

- L'analyse par signature est très efficace pour détecter des attaques sans produire un grand nombre de fausses alarmes.
- Peut rapidement et sûrement diagnostiquer l'utilisation d'un outil spécifique ou une technique d'attaque
- Ceci peut aider les responsables de sécurité à donner la priorité aux mesures correctives .

Les inconvénients de l'analyse par signature

- L'analyse par signature peut seulement détecter les attaques connues, dont les signatures sont introduites dans le système, donc le système de détection doit être constamment mis à jour avec les signatures des nouvelles attaques.
- Beaucoup de systèmes adoptant cette approche sont conçus pour employer un nombre limité de signatures qui peuvent être définies, ce qui les empêchent de détecter des variantes de ces attaques .

2.3.3 Types de réponses [Chikouche ,2012]

Il existe deux types de réponses, en fonction du SDI utilisé

- Une réponse passive est disponible pour tous les SDIs,
- Une réponse active est implémentée sous une certaine forme .

1.3.3.1 Réponse active [Mharouach ,2005]

Une réponse efficace vise à arrêter l'attaque au moment de la détection. Pour cela, il existe deux techniques, la reconfiguration du pare-feu et l'interconnexion TCP.

La reconfiguration du pare-feu permet d'empêcher le trafic malveillant, en fermant le port de l'utilisateur ou en bloquant l'adresse de l'attaquant. Cette fonctionnalité dépend du modèle de pare-feu utilisé et tous les modèles ne permettent pas la

configuration par SDI . De plus, cette reconfiguration ne peut être effectuée qu'en fonction des capacités du pare-feu.
SDI peut également mettre fin à une session entre un attaquant et son appareil cible, afin d'empêcher le transfert de données ou de modifier le système de l'attaquant.

1.3.3.2 Réponse passive [Labed ,2006]

Une réponse passive d'un SDI consiste à enregistrer les interventions détectées dans le fichier journal qui sera analysé par le responsable de la sécurité. Certains SDI autorisent l'enregistrement de la connexion spécifiée comme malveillante. Cela supprime les failles de sécurité pour empêcher la répétition des attaques enregistrées.

2.3.4 Fréquence d'utilisation [wikimemoires ,2012]

La dernière caractéristique des systèmes de détection d'intrusions est leur fréquence d'utilisation : **périodique** ou **continue**.

- Certains systèmes de détection d'intrusions analysent périodiquement les fichiers d'audit à la recherche d'une éventuelle intrusion ou anomalie passée. Cela peut être suffisant dans des contextes peu sensibles (on fera alors une analyse journalière, par exemple).
- La plupart des systèmes de détection d'intrusions récents effectuent leur analyse des fichiers d'audit ou des paquets réseau de manière continue afin de proposer une détection en quasi temps-réel. Cela est nécessaire dans des contextes sensibles (confidentialité) et/ou commerciaux. C'est toutefois un processus coûteux en temps de calcul car il faut analyser à la volée tout ce qui se passe sur le système.

2.4 Outils de la sécurité et comparatif

Les intrusions se produisent lorsque des attaquants atteignent des systèmes en ligne, des utilisateurs autorisés, ou tentent d'obtenir plus de privilèges non autorisés, ou des utilisateurs autorisés qui abusent de leurs privilèges. La détection d'intrusion améliore la sécurité du réseau en ajoutant une couche de protection.
La détection d'intrusion permet aux organisations de protéger leurs systèmes contre les menaces associées à une connectivité réseau accrue. Le SDI est considéré comme complément nécessaire à l'infrastructure de sécurité de chaque organisation. Bien que la technologie de détection d'intrusion ne fournisse pas une protection complète contre les attaques, elle favorise une approche de défense en profondeur, une tendance moderne en matière de sécurité des réseaux.

2.4.1 Pare-feu

C'est un dispositif de sécurité réseau, qui fonctionne selon une politique de sécurité qui a été défini et configuré. Il est également impératif qu'il ne puisse pas offrir une protection complète contre le trafic malveillant. Il surveille et contrôle les applications et le flux de données (paquets). Le pare-feu fournit le contrôle d'accès du trafic internet à partir de l'intérieur et de l'extérieur, mais il est insuffisant pour stopper une intrusion. Par exemple, lors de l'exploitation d'une faille d'un serveur Web, le flux HTTP sera autorisé par le firewall puisqu'il n'est pas capable de vérifier ce que contiennent les paquets. La différence fondamentale entre un pare-feu et un SDI, c'est que le pare-feu offre une protection active contre les attaques, alors que les SDIs peuvent déclencher une alerte et détecter les attaques. [Chikouche, 2012].
Il existe trois types de pare-feu :[Burgermeisteret et Jonathan ,2006]

1. Les systèmes à filtrage de paquets sans état : analyse les paquets les uns après les autres, de manière totalement indépendante.
2. Les systèmes à maintien d'état (stateful) : vérifient que les paquets appartiennent à une session régulière. Ce type de firewall possède une table d'états où est stocké un suivi de chaque connexion établie, ce qui permet au firewall de prendre des décisions adaptées à la situation. Ces firewalls peuvent cependant être outrepassés en faisant croire que les paquets appartiennent à une session déjà établie.
3. Les firewalls de type proxy : le firewall s'intercale dans la session et analyse l'information afin de vérifier que les échanges protocolaires sont conformes aux normes.

2.4.2 Anti-virus [Sécurité-Malwares]

Un antivirus est un programme de sécurité réseau et système, capable de détecter les virus, les vers, les Troyens et parfois les spywares qui peuvent infecter un ordinateur. L'antivirus devrait être mise en place sur l'ordinateur, mais il existe aussi des tests d'infection virale disponibles sur le web. Parmi les critères importants d'un antivirus, on retrouve la fréquence des mises à jour qui devrait être quotidienne. La différence fondamentale entre un Anti-virus et un SDI, c'est que l'Anti-virus offre une protection active contre les attaques, alors que les SDIs peuvent déclencher une alerte et détecter les attaques.

2.5 Conclusion

Dans ce chapitre, nous avons présenté les systèmes de détection d'intrusion, à savoir leurs fonctionnements et leurs critères de classification. Il est clair pour nous que ces systèmes sont maintenant nécessaires aux entreprises pour assurer la sécurité des technologies de l'information en effectuant les tâches associées à d'autres équipements de sécurité.

Chapitre 3

État de l'art sur les SDI

Sommaire

3.1 Introduction

La classification est une technique largement utilisée dans la recherche d'information, diagnostic médical, apprentissage automatisé, aide à la décision et dans de nombreux domaines de recherche en intelligence artificielle.

Dans la sécurité informatique, la classification est utilisée pour classifier une masse de données système ou un flux réseau, elle a pour objectif d'apporter une aide à la décision. Les SDIs utilisent généralement des méthodes de classification afin de pouvoir détecter tout comportement suspect tout en réduisant le taux de fausses alertes. Il s'agit alors de faire appel aux méthodes issues de l'intelligence artificielle telle que les Réseaux de Neurones Artificiels RNAs, les Algorithmes Évolutionnaires AE, ...

Dans la première partie de ce chapitre, nous nous concentrerons sur les méthodes de classification où nous allons introduire sa taxonomie ainsi que quelques concepts de base, nous verrons aussi les principaux travaux ayant utilisé ces techniques. Dans la deuxième partie, nous présenterons les RNAs, où nous commencerons par un petit historique, quelques définitions, ses techniques d'apprentissage et ses architectures sans oublier d'introduire un état de l'art des RNA dans le domaine des SDIs.

3.2 Méthodes de classification

Dans le contexte de la détection d'intrusion, le terme **classification** est utilisé pour exprimer une **distinction** ou une **identification** d'attaques. Il existe plusieurs méthodes de classification dont la taxonomie est présentée dans la figure (2.1) . Parmi ces méthodes on retrouve les méthodes exclusives qui consistent à ce qu'un objet ne peut être que dans une seule classe dans la partition finale .Nous allons voir dans la section suivante les différentes catégories de cette classe.

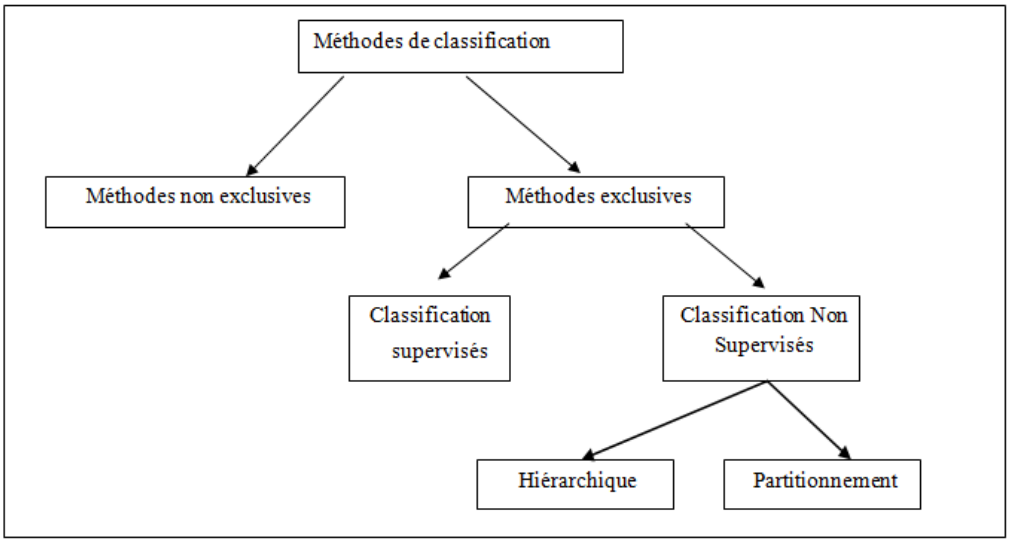

Figure 3.1 – Taxonomie de classification [Bourouh et kanoun ,2017]

3.2.1 Classification non supervisée [Bourouh et kanoun ,2017]

La classification non supervisée (clustering) consiste à affecter les individus considérés similaires au même groupe. Les classes ne sont pas connues à l'avance, et les exemples disponibles sont non étiquetés. Le but est donc de constituer des classes, parmi les méthodes de classification non supervisée .

2.2.1.1 Classification hiérarchique

a. Classification Ascendante Hiérarchique (CAH) [Wikistat] Il s'agit de regrouper itérativement les individus, en commençant par le bas (les deux plus proches) et en construisant progressivement un arbre (dendrogramme), regroupant finalement tous les individus en une seule classe. Ceci suppose de savoir calculer, à chaque étape ou regroupement, la distance entre un individu et un groupe ainsi que celle entre deux groupes. Ceci nécessite donc, pour l'utilisateur de cette méthode,

de faire un choix supplémentaire : comment définir la distance entre deux groupes en connaissant celles de tous les couples d'individus entre ces deux groupes .

b. Classification Descendante Hiérarchique [Hérault ,2012] Il s'agit d'une dichotomie de la classe entière jusqu'à obtenir tous les éléments terminaux. Au départ tous les individus sont dans le même groupe. À chaque étape, un groupe est séparé en deux. Il faut un critère de séparation

2.2.1.2 Partitionnement

Le partitionnement consiste à diviser le jeu de données en k clusters, où le nombre entier k doit être spécifié par l'utilisateur. En général, l'utilisateur exécute l'algorithme pour une gamme de k-valeurs. Pour chaque k, l'algorithme effectue la classification et donne également un «indice de qualité», qui permet à l'utilisateur de sélectionner une valeur par la suite .[Struyf et Rousseeuw ,2004]
Un des algorithmes les plus connus de cette catégorie est l'algorithme K-moyennes (K-Means) qui consiste à regrouper N points de données en K clusters disjoints, où K est un paramètre prédéfini. D'autres algorithmes similaires ont été développés par la suite tels que l'algorithme des centres mobile [Forgey ,1965] et la méthode des nués dynamiques . [Bonner ,1964]

a. Méthode des K-means La classification k-means a été introduite par MacQueen [MacQueen ,1967] , le principe de la méthode consiste a construire k classes à partir d'un ensemble de n individus [Mokeddem et Oumiloud ,2014], tout en minimisant la quantité :

$$\sum_{r=1}^{k} \sum_{Xi \in Cr} (X_i - G_r)^2 \tag{3.1}$$

- Xi est un individu dans une classe
- Cr est la classe numéro r
- Xi est un individu dans une classe
- Gr est le centre de classe Cr

La méthode K-means a été largement utilisée dans la détection des intrusions, parmi ces travaux, nous citons le SDI introduit par Al-Yaseen et son équipe . [Al-Yaseen et al ,2016] a proposé un modèle qui traite des problèmes réels de détection d'intrusion dans l'analyse des données et de classer les données du réseau en comportements normaux et anormaux. Cette étude propose un modèle de détection d'intrusion hybride à plusieurs niveaux qui utilise une machine à vecteurs de soutien et une

machine d'apprentissage extrême pour améliorer l'efficacité de la détection des attaques connues et inconnues. Un algorithme K-means modifié est également proposé pour construire un jeu de données d'entraînement de haute qualité qui contribue de manière significative à l'amélioration des performances des classifieurs. Le K-means modifié est utilisé pour construire de nouveaux petits jeux de données d'apprentissage représentant l'intégralité de l'ensemble de données original de la base KDD Cup 1999 . [Mokhtari et Moulkhaloua ,2017]

b. Méthode des centres mobiles [Panda et Patra ,2007] Cette méthode consiste à construire une partition en k classes en sélectionnant k individus comme centres des classes tirés au hasard de l'ensemble d'individus. Après cette sélection, on affecte chaque individu au centre le plus proche en créant k classes, les centres des classes seront remplacés par les centres de gravité et les nouvelles classes seront créées par le même principe

Algorithme

K le nombre maximum de classes désiré.

Début

(1) Choisir k individus au hasard (comme centre des classes initiales)

(2) Affecter chaque individus au centre le plus proche Ce qui donne une partition en k classes P1 = C1, . . . , Ck

(3) On calcule les centres de gravité des chacune des classes de P1 Ce qui donne k nouveaux centres de classes.

(4) Répéter l'étape (2) et (3) jusqu''a deux itérations successives donnent la même partition

(5) Editer la partition obtenue.

Fin

c. Méthode Nuées dynamiques D'après Ludovic Mé [Ludovic ,2015] cette méthode est une généralisation de la méthode des centres mobiles .L'algorithme suivant détaille ses différentes étapes :

Algorithme

Données :

k le nombre maximum de classes désirées.

I le nombre maximum d'itérations désirées.

Début

(1) Donner une famille de noyaux N0

(2) Déduire par g une famille de classes C0 (une partition)

(3) Calculer la valeur de H(C0, N0)

(4) A l'aide de h déterminer une nouvelle famille de noyaux N1 telle que :

$$\text{H(C0, N1)} \leq H(C0, N0)$$

(5) Calculer g(N1) = C1 tel que :

$$\text{H(C1, N1)} \leq H(C0, N1)$$

j=j+1

jusqu'à j=i

Fin

Melle Lynda SELLAMI [SELLAMI, 2009] a présenté un travail sur le datamining intitulé « Approche Data Mining pour la Détection d'Intrusions » dont les systèmes de détection d'intrusions en utilisant la méthode de classification automatique la segmentation par les nuées dynamiques. La solution proposée par cet algorithme dépend de la partition initiale et qui a pour objectif de partitionner l'espace en classes (connues).

3.2.2 Classification supervisée [Projet ,2007]

Par classification supervisée, on entend le problème suivant : à partir d'un échantillon ni d'objets étiquetés/classés, on désire construire une fonction capable d'étiqueter/classer au mieux de nouveaux objets (ne faisant pas partie de l'échantillon initial)

2.2.2.1 Algorithme des K Plus Proches Voisins (K-PPV)

La méthode des K-PPV pondérés figure parmi les méthodes à base de voisinage, elle offre dans ce contexte des performances très intéressantes. Il s'agit d'un algorithme d'apprentissage automatique supervisé. Dans un contexte de classification d'une nouvelle observation x, l'idée fondatrice simple est de faire voter les plus proches voisins de cette observation. La classe de x est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation x .

[Mathieu et Dupas ,2010]

Donc l'algorithme des K-PPV permet d'obtenir de très bons résultats de classification dans le cas ou la base d'apprentissage utilisée est correctement étiquetée .

[Projet ,2007]

a. L'algorithme 1-PPV [Mathieu ,2010] La méthode du plus proche voisin est une méthode non paramétrique où une nouvelle observation est classée dans la classe d'appartenance de l'observation de l'échantillon d'apprentissage qui lui est la plus proche, au regard des covariables utilisées. La détermination de leur similarité est basée sur des mesures de distance. Formellement, soit L l'ensemble de données à disposition ou échantillon d'apprentissage.

La distance **d** entre deux classes Ci et Cj est définie par :

$$d(Ci, Cj) = \inf_{\substack{X \in Ci \\ Y \in Cj}} d(X, Y); \tag{3.2}$$

Sachant que la distance Euclidienne :

$$d^2(X, Y) = (\sqrt{\sum_{i=1}^{n}(x - y)^2})Cj \tag{3.3}$$

b. K-PPV classifieur paresseux [Projet ,2007] L'algorithme k-ppv est un algorithme de la famille des algorithmes dits « paresseux » : à l'inverse de beaucoup d'autres méthodes d'apprentissage automatique (telles que la régression logistique, les réseaux de neurones articiels, ... etc.) aucun apprentissage ne prend réellement place, c'est-à-dire qu'il n'y a pas de phase de détermination de paramètres d'une fonction par le biais d'une optimisation mathématique. Le principe est le suivant : étant donné une base d'apprentissage étiquetée correctement et un entier k, le classieur K-ppv détermine la classe d'un nouvel objet en lui attribuant la classe majoritaire des k objets lui ressemblant le plus dans la base d'apprentissage.
On note deux aspects importants de l'algorithme k-ppv : d'une part, à chaque nouvelle classication, il est nécessaire de parcourir l'ensemble de la base d'apprentissage, ce qui n'est pas vraiment très efficace (surtout que, habituellement, on cherche à avoir la base d'apprentissage la plus grande possible an d'avoir un meilleur classieur). D'autre part, un point crucial de cet algorithme est la fonction de distance utilisée pour mesurer la proximité des objets où il n'existe pas de distance/similarité universellement optimale, une bonne connaissance du problème traité guide généralement le choix de cette distance/similarité.
L'algorithme des K-PPV est le suivant :

Algorithme

Début :
(1) On cherche à classer le point y

(2) Pour (chaque exemple (x,C(x)) de l'ensemble d'apprentissage) Faire

(3) Calculer la distance d(x,y) entre x et y

Fin pour

Dans les k points proches de y compter le nombre d'occurrences de chaque classe
Attribuer à y la classe qui apparaît le plus souvent

Fin.

3.3 Les réseaux de neurones

3.3.1 Les neurones formels

Un "neurone formel" (ou simplement "neurone") est une fonction algébrique non linéaire et bornée, dont la valeur dépend des paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie". Un neurone est donc avant tout un opérateur mathématique, dont on peut calculer la valeur numérique par quelques lignes de logiciel. On a pris l'habitude de représenter graphiquement un neurone comme indiqué sur la figure (2.2) .[Djeriri ,2017]

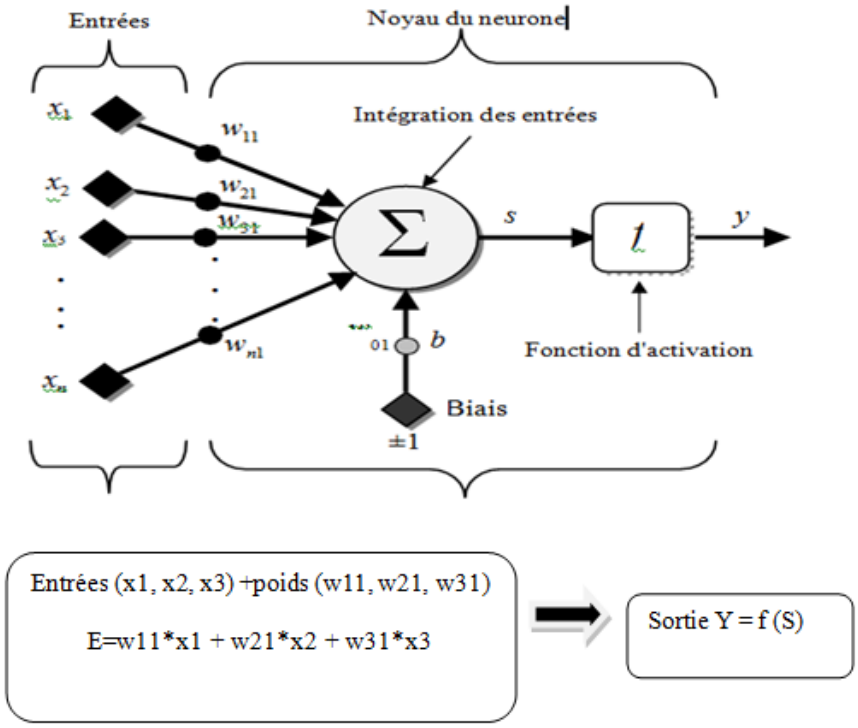

Figure 3.2 – Modèle d'un neurone artificiel [Rfia, 2012]

De façon générale, on définit un neurone formel par les cinq paramètres suivants [Rfia, 2012] :
1-La nature des entrées (booléenne ou réelle).
2-La fonction d'entrée totale, définissant le prétraitement effectué sur les entrées .
3- La fonction d'activation du neurone définissant son état interne en fonction de la somme pondérée de ses entrées.
4-La fonction de sortie calculant la sortie du neurone en fonction de son état d'activation.

5- La nature des sorties du neurone.

3.3.2 Modèles des réseaux de neurones

Le Modèle de Hopfield

Ce modèle très simple est basé sur le principe des mémoires associatives. C'est d'ailleurs la raison pour laquelle ce type de réseau est dit associatif (par analogie avec le pointeur qui permet de récupérer le contenu d'une case mémoire).
Le modèle de Hopfield utilise l'architecture des réseaux entièrement connectés et récurrents (dont les connexions sont non orientées et ou chaque neurone n'agit pas sur lui-même). Les sorties sont en fonction des entrées et du dernier état pris par le réseau. [Djeriri ,2017]

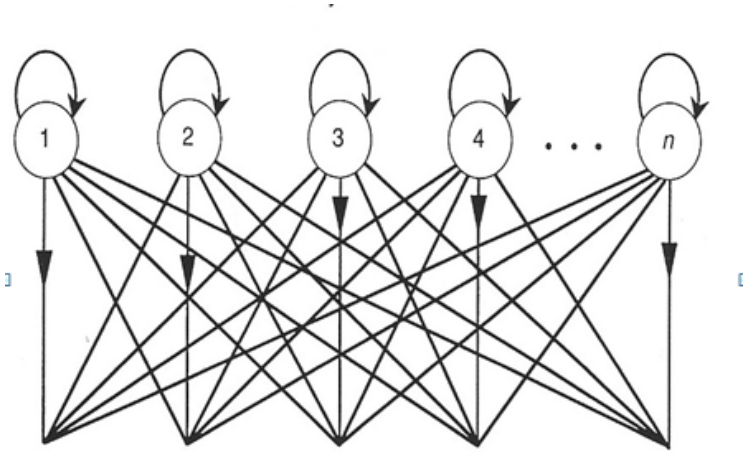

Figure 3.3 – Le Modèle de Hopfield [Sheynikhvich, 2017]

Le Modèle de Kohonen

Ce modèle a pour objectif de présenter des données complexes et appartenant généralement à un espace discret de grandes dimensions dont la topologie est limitée à une ou deux dimensions. Les cartes de Kohonen sont réalisées à partir d'un réseau à deux couches, une entrée et une sortie. Notons que les neurones de la couche d'entrée sont entièrement connectés à la couche de sortie figure (2.4). Les neurones de la couche de sortie sont placés dans un espace d'une ou de deux dimensions en général, chaque neurone possède donc des voisins dans cet espace. Et qu'enfin, chaque neurone de la couche de sortie possède des connexions latérales récurrentes dans sa couche (le neurone inhibe, les neurones éloignés et laisse agir les neurones voisins) [Djeriri, 2017].

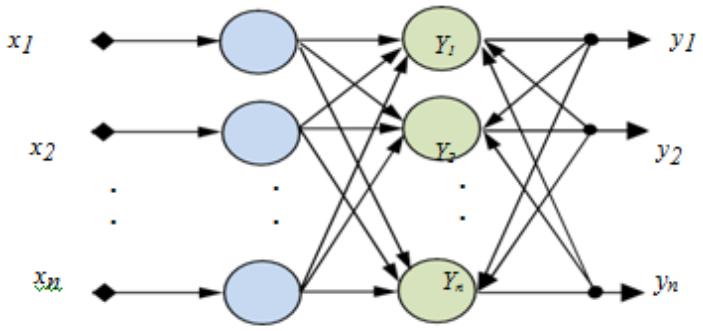

Figure 3.4 – Le modèle de Kohonen [Djeriri ,2017]

Ce modèle a été utilisé dans le domaine de détection d'intrusion [Ludovic, 2015]. Le chier d'audit, vu comme une chaine de caractères, peut être découpé en sous chaines de longueur constante. Ces sous chaines sont appelées des patterns d'activité. Chaque pattern correspond à un comportement élémentaire de l'utilisateur qui l'a généré.
À l'aide de la carte de Kohonen, il est possible de déterminer l'ensemble des comportements élémentaires de chaque utilisateur du système. À cette n, une base d'apprentissage est utilisée qui constituée par des patterns d'activité résultant de l'observation de l'utilisateur pendant une période signicative de son activité. La carte de comportement de l'utilisateur (CCU) est obtenue une fois l'apprentissage terminé en comptant le nombre de fois qu'a été élu chaque neurone de sortie lorsque l'on présente toutes les entrées de la base d'exemples.
Lorsque le réseau a été entrainé, on lui présente des patterns d'activités nouvellement acquises. Le niveau d'activation de chaque neurone donné en fonction de sa position permet de construire la carte de détection de l'utilisateur (CDU).
Si l'excitation des neurones de la CDU est faible, le pattern n'est pas connu. Il est donc anormal au sens de l'approche comportementale et une alerte doit être émise vers l'administrateur [Ludovic, 2015].

Le modèle de Perceptron [Djeriri ,2017]

Le réseau dans ce modèle est formé de trois couches : Une couche d'entrée, fournissant des donnés à une couche intermédiaire, chargée des calculs, cela en fournissant la somme des impulsions qui lui viennent des cellules auxquelles elle est connectée, et elle répond généralement suivant une loi définie avec un seuil, elle-même connectée à la couche de sortie (couche de décision), représentant les exemples à mémoriser. Seule cette dernière couche renvoie des signaux à la couche intermédiaire, jusqu'à ce que leurs connexions se stabilisent

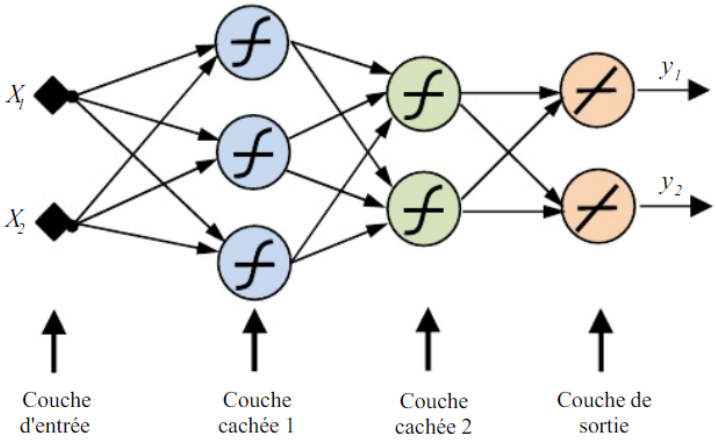

Figure 3.5 – Le modèle de Perceptron

Le modèle ADALINE [Djeriri ,2017]

L'ADALINE est un réseau à trois couches : une d'entrée, une couche cachée et une couche de sortie. Ce modèle est similaire au modèle de perceptron, seule la fonction de transfert change, mais reste toujours linéaire. Les modèles des neurones utilisés dans le perceptron et l'ADALINE sont des modèles linéaires. On dit que deux classes A et B, sont linéairement séparables si on arrive à les séparer par une droite coupant le plan en deux figure (2.6). Le problème est résolu avec les réseaux multicouches, car il peut résoudre toutes sortes de problèmes qu'ils soient linéairement séparables ou non.

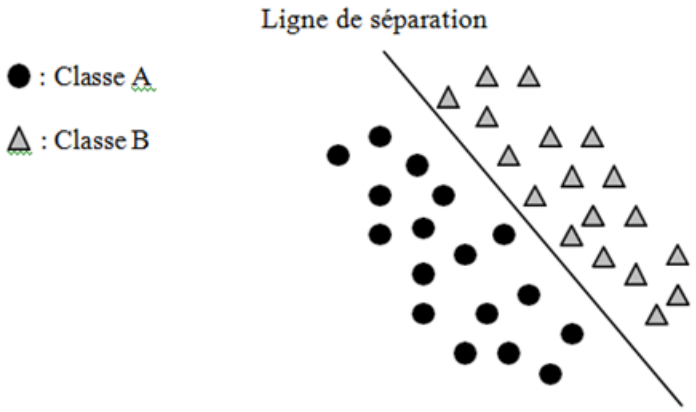

Figure 3.6 – La séparation linéaire entre la classe A et B

Le modèle à base radiale (Radial Basis Function (RBF))

Les réseaux de neurones RBFs, sont principalement utilisés pour résoudre des problèmes d'approximation de fonctions dans des espaces de grandes dimensions. Ils sont lus adaptés, en raison d'apprentissage local. Ce type d'apprentissage peut rendre le processus d'entrainement bien plus rapide que dans le cas d'un MLP, qui apprend de façon globale. [rfia ,2012]

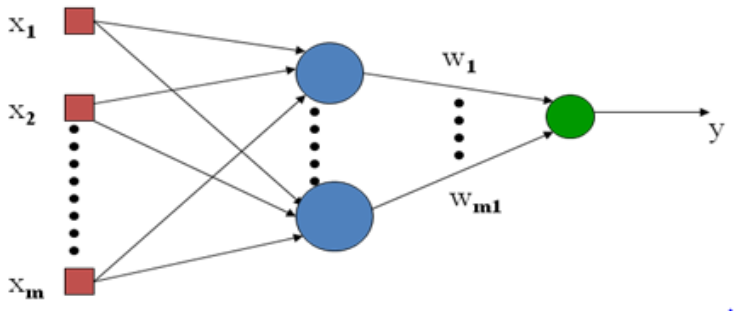

Figure 3.7 – Le modèle à base radiale RBFs [Djeriri ,2017]

3.3.3 Les RNAs et les SDI

Les réseaux de neurones sont largement utilisés dans la détection d'anomalies dans les systèmes de détection d'intrusion afin d'exploiter leur capacité d'apprentissage. Panda et al.[Panda et al., 2012] proposent une approche hybride incluant une combinaison de classificateurs dans l'objectif d'améliorer les performances globales du modèle tout en réalisant plusieurs hybridations. Le système est constitué de deux étapes, dans la première il s'agit d'utiliser un algorithme de sélection supervisé ou non supervisé pour le filtrage de la base de données NSL-KDD afin de sélectionner les attributs les plus significatifs où ils ont utilisé les arbres de décision et l'analyse en composantes principales. La deuxième étape consiste à appliquer un algorithme permettant la détection d'intrusion où ils ont utilisé le réseau de neurones de fonction radiale RBF, les Séparateurs à Vaste Marge SVM ainsi que la méthode Forêts aléatoires.

Plusieurs d'autres auteurs se sont intéressé à appliquer le réseau de neurones du type RBF au sein des SDIs afin de réduire le temps de réponse ainsi que les fausses alarme. Cet RN est construit de trois couches : une couche d'entrée, une couche cachée dotée d'une transformation Gaussiennes et une couche de sortie. L'objectif était alors d'optimiser ce modèle en ajustant les paramètres du réseau tel que les

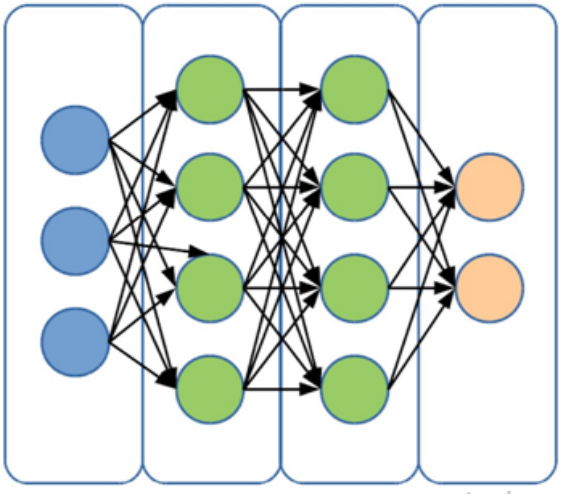

Figure 3.8 – Conception finale du réseau neuronal artificiel

poids entre la couche cachée et la couche de sortie, le nombre de neurones dans la couche caché et même le choix des centres [Sheth et al., 2014]. Pour cela plusieurs méthodes ont été utilisées tel l'algorithme MGIN [Zhong, Jiang et al 2006] , K PPV [Yichun.et al., 2012] [Tian et al., 2009] [Peng, Yichun, et al 2014]ou la méthode d'Optimisation par essaims particulaires PSO [Chen, et al., 2009].

Gautam et son équipe [Gautam et al., 2016] propose deux modèles de réseaux de neurones destinés à la détection d'intrusion sur hôte. Il s'agit du réseau de Neurones de régression généralisée (RNRG) ainsi que le Perceptron Multi-Couches PMC. Ces modèles ont été appliqués en utilisant le journal log généré sur une machine personnelle et les résultats de détection ont été évalués via la matrice de confusion montrant ainsi que le MLP permet une meilleure précision que le RNRG.

Shenfield et al [Shenfield et al., 2018] propose un RN en vue d'une détection d'intrusion au sein d'un réseau. Les auteurs proposent une approche hors ligne pour détecter du shellcode dans des données où ils ont adopté un MLP avec deux couches cachées dotées chacune de 30 neurones. L'évaluation du SDI a été réalisée via une validation croisée.

3.4 Les systèmes Multi Agent(SMA)

Les Systèmes Multi-agents SMA sont actuellement très utilisés, particulièrement dans les applications complexes nécessitant l'interaction entre plusieurs entités. Ainsi, la construction, la maintenance et l'exécution des systèmes complexes sont des tâches délicates et nécessitent la coopération de plusieurs intelligences entre elles, plutôt que de tenter de les attribuer à une seule intelligence monolithique. Cette complexité s'est exprimée soit de façon structurelle, soit de façon comportementale, soit, plus généralement, les deux à la fois .[Drogoul ,2005]

Un SMA est un système distribué, constitué d'un ensemble d'agents organisés et situés dans un environnement où ils interagissent entre eux. Les SMAs œuvrent dans plusieurs domaines d'application, on peut les trouver dans les systèmes de production, les différentes tâches de contrôle, contrôle de trafic dans réseaux informatique, les diagnostiques multi nivaux,. . . etc.

3.4.1 Agent

Un agent est une entité logicielle qui exécute un bloc d'instructions d'une façon autonome dans un environnement et qui doit communiquer avec d'autres agents pour atteindre un objectif global (ou local). Il existe trois types d'agent :

- **Les agents cognitifs** : ce type d'agent dispose d'une base de connaissances comprenant diverses informations liées à leurs domaines d'expertise et à la gestion des interactions avec les autres agents et leur environnement.
- **Les agents réactifs** : contrairement aux agents cognitifs, ces agents interagissent d'une manière naïve. Ils ne peuvent que réagir à des stimuli simples provenant de leur environnement, et leur comportement est simplement dicté par leurs relations avec leur entourage sans que ces agents ne disposent d'une représentation des autres agents ou de leur environnement.
- **Les agents hybrides** : en général, la différence entre des agents réactifs et des agents cognitifs peut être expliquée par le compromis : efficacité/ complexité. La complexité des systèmes réactifs exige le développement de nouvelles théories dans le domaine de la coopération, de la communication et de la compréhension de nouveaux phénomènes tels que l'émergence. Toutefois, il est maintenant possible de concevoir des systèmes hétérogènes comportant les deux types de comportements (cognitif et réactif) : on parlera alors d'agents hybrides .[Méziane ,2007]

3.4.2 Interaction entre agents [Pesty et al., 2001]

Parmi les interactions entre agents que l'on peut citer :

a. Communication : un agent doit être capable de communiquer avec les autres agents. Il existe deux types de communication :

- communication indirecte : par partage d'information via l'environnement.
- communication directe : par envoi de messages.

b. Coopération : la coopération entre les agents consiste à décomposer les tâches en sous-tâches puis à les répartir entre les différents agents

c. La coordination : permet d'éviter le comportement chaotique entre agents et maintenir la cohérence de l'ensemble

d. La négociation : un processus permet de s'accorder sur des situations communes ou sur des plans des échanges

3.4.3 Caractéristique d'un System Multi Agents

Selon le comportement de ses agents un SMA peut être :

- **ouvert** où tous les agents sortent et entrent librement .
- **fermé** dans lequel les agents restent les mêmes sans aucun changement .
- **Homogène** où les agents sont construits sur un même modèle.
- **Hétérogène** contenant des agents différents qui sont construits sur des modèles différents

3.4.4 Plate-forme SMA

Des environnements ou des plates-formes de développement multi-agents permettent d'améliorer le succès de la technologie SMA. Les plates-formes SMA permettent aux développeurs de concevoir et de réaliser leurs applications sans perdre de temps à réaliser des fonctions de base pour la création et l'interaction entre agents. Dans la plupart des cas, il est nécessaire de se familiariser avec les différents concepts théoriques des systèmes multi-agents. Parmi les plates-formes fournies en tant que programme gratuit, certaines ont été utilisées dans le développement de nombreuses applications [theses.univ, 2008] :

- **JADE** (Java Agent Développent Framework) est une plate-forme multi-agent développée en Java par CSELT qui a comme but la construction des systèmes multi-agent et la réalisation d'applications conformes à la norme FIPA .
- **MACE** [Gasser et al. ,1987] est le premier environnement de conception et d'expérimentation de différentes architectures d'agents dans divers domaines

d'application. Dans MACE, un agent est un objet actif qui communique par envoi de messages. Les agents existent dans un environnement qui regroupe tous les autres agents et toutes les autres entités du système.

• **ZEUS** : est une plate-forme multi-agent conçue pour développer des applications collaboratives. ZEUS est écrit dans le langage Java et il est fondé sur les travaux de la FIPA.

3.4.5 Simulation avec les SMAs

La simulation SMA est le domaine qui étudie les méthodes et algorithmes, dans le cadre conceptuel et opérationnel des systèmes. La simulation permet de construire des techniques de représentation et de production des connaissances ainsi que de calcul d'événements [Benotmane et Benglia, 2010].
Plusieurs formes de parallélismes sont exploitables dans les systèmes informatiques, nous allons voir plus de détails dans ce qui suit.

Simulation parallèle [Bouzid ,2001]

La simulation parallèle est un terme désignant une nouvelle fonctionnalité qui permet aux utilisateurs d'exécuter plusieurs simulations simultanément. Il fournit une interface utilisateur unique pour afficher et gérer toutes les simulations.
Deux types de simulation sont distingués : les simulations dites continues et les simulations dites discrètes. Les simulations multi-agents sont des simulations discrètes. Elles se décomposent en deux grandes classes : les simulations discrètes par pas de temps (time-driven) et les simulations discrètes par événements (event-driven). Les pas de temps dans une simulation discrète (time-driven) se définissent comme des intervalles de temps réguliers qui sont parcourus au cours de la simulation. Le pas de temps peut prendre différentes unités (secondes, minutes, jours...). Lors de l'incrémentation d'un pas de temps, tous les agents de la simulation sont mis à jour et/ou exécutés. En d'autres termes, nous incrémentons toute la simulation d'une unité de temps comme présenté dans la Figure (3.9).

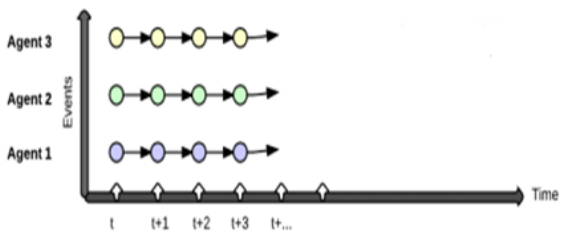

Figure 3.9 – Simulations discrètes par pas de temps [Christophe ,2014]

Pour leur part les simulations discrètes par événements (event-driven) utilisent des listes ordonnées d'événements, et traitent donc le premier événement qui est chronologiquement dans la liste. Ceci permet de dérouler la simulation comme présenté dans la Figure (3.10).

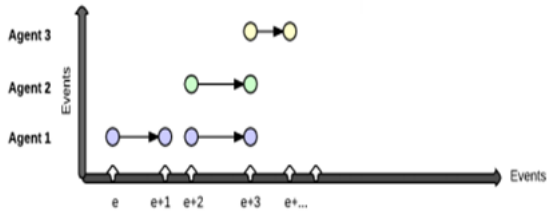

Figure 3.10 – Simulations discrètes par événement [Christophe ,2014]

Lorsqu'un system devient complexe et de nature distribuée, les systèmes multi-agents peuvent être alors le modèle informatique adéquat pour le décrire. Le type de simulation est généralement déterminé par le système que nous souhaitons modéliser. Par exemple si nous souhaitons modéliser le comportement de personnes dans une ville, il convient d'utiliser la simulation par pas de temps afin d'observer l'évolution au cours du temps pour chaque agent et pour l'ensemble de la simulation.
Les premiers ordinateurs étaient séquentiels, exécutant ainsi les instructions l'une après l'autre. Le parallélisme se manifeste actuellement de plusieurs manières : en juxtaposant plusieurs processeurs séquentiels ou en exécutant simultanément des instructions indépendantes.
Grâce aux multiprocesseurs il est possible d'avoir une forme d'architecture parallèle, un fonctionnement sous un ordinateur multiprocesseur est dit parallèle quand chaque processus est exécuté sur un processeur ou on obtient une grande puissance de calcul par rapport à un ordinateur uni processeur .
On retrouve notamment la notion multitâche où un système d'exploitation permet d'exécuter de façon simultanée, plusieurs programmes informatiques. On parle également de multiprogrammation.
Pour ce faire il n'est pas nécessaire d'avoir un système multiprocesseur .
Il s'agit alors d'une alternance rapide d'exécution des processus présents en mémoire c'est là où la simultanéité apparente se présente. Ces commutations peuvent être initiées par les programmes eux-mêmes (multitâche coopératif) ou par le système d'exploitation lors d'événements externes (multitâche préemptif) .

3.4.6 Les SMAs et les SDIs

Le paradigme des SMAs a été exploité à son tour afin de réaliser des SDIs ou **Al-Yaseen et son equipe** [Al-Yaseen et al. ,2016] ont suggéré une proposition d'une méthode appelée système multi-agents en temps réel permettant au système RTMAS-AIDS de détecter les infiltrations à adapter, basée sur un système multi-agents, afin de permettre au système de détection d'intrusion de s'adapter à des attaques inconnues et en temps réel. Cette méthode utilise des modèles hybrides à plusieurs niveaux et des évaluations hybrides pour détecter un comportement normal et des attaques connues.

Adriana et al [Adriana et al., 2017] ont proposé une combinaison entre un comportement réactif et l'apprentissage 'hylaa'« A hybrid and learning agent architecture for network intrusion détection » sur une architecture agent logiciel . En apprenant de nouvelles règles réactives et en les stockant dans la base de connaissances de l'agent. « hylaa » pourrait s'adapter à son environnement et identifier de nouvelles intrusions non précisées dans la conception du système.

3.5 Autres méthodes

3.5.1 Machine à états finis [Ali Khan et al, 2011]

On l'appelle aussi parfois automates à états finis, est un graphe orienté et étiqueté dont les sommets représentent des états et les arêtes représente les transitions entre ces états.
On modélise un flux d'exécution à l'aide d'un automate de type bien particulier de machine à états finis appelé automate accepteurs pour déterminer si l'entrée qui leur est fournie est correcte ou non .Ce type de modélisation a été utilisé au niveau des appels système des processus .
Chaque état de l'automate représente le couple (numéro de l'appel système, adresse de l'appel).
La phase d'apprentissage se fait à l'aide de traces d'exécution obtenues en utilisant l'outil trace qui permet notamment de connaître pour chaque appel le compteur ordinal du processus. C'est ce dernier qui contient l'adresse à partir de laquelle l'appel système a été effectué. Durant la phase de détection, à chaque fois qu'un appel système est effectué, on récupère le compteur ordinal du processus afin de construire un nouvel état. Puis, on regarde s'il existe au sein de l'automate à états fini une transition possible depuis l'état courant vers ce nouvel état. Si ce n'est pas

le cas une alerte est alors levée. Enfin, le nouvel état construit devient le dernier état connu. Si celui-ci n'est pas présent dans l'automate, alors il est remplacé par un nœud initial à partir duquel tous les états valides sont atteignables. Le processus normal de vérification reprendra au prochain état valide.

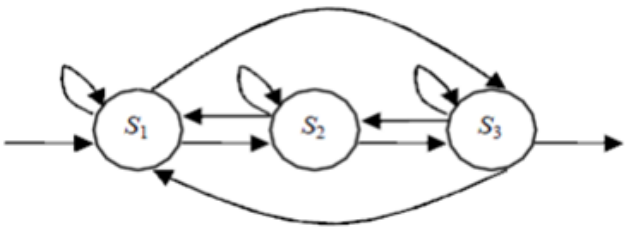

Figure 3.11 – Automate d'état finis

3.5.2 Modèle statistique [Bouzid ,2001]

Pour modéliser l'enchaînement des appels système exécutés il est préférable d'utiliser un modèle de Markov caché. Premièrement, on considère que les traces d'exécution des programmes analyseur durant une fenêtre de temps sont en situation de fonctionnement normal. Ensuite, le modèle créé est utilisé à l'exécution durant la phase de détection pour lever une alerte lorsque qu'un appel système observé n'a pas été prédit. Un seuil doit être défini puisqu'il s'agit d'un modèle statistique, ce dernier est défini pour savoir à partir de quelle probabilité une alerte doit être levée. Ce paramètre va influencer le taux de faux positifs et de faux négatifs dans ce type d'approche.
Un modèle de Markov caché est un modèle statistique décrivant l'évolution d'une variable aléatoire selon un processus de paramètres inconnus où il n'est pas nécessaire de connaître l'ensemble des états passés de la variable pour prédire probabilistiquement son prochain état. Ce modèle, ne détecte que les intrusions qui modifient l'enchaînement des appels système exécutés.

3.6 Conclusion

Nous avons présenté dans ce chapitre un état de l'art sur les différentes méthodes de classification à savoir les approches supervisées et non supervisées ou nous avons présenté par la suite les RNAs en mettant l'axe sur quelques travaux réalisés en vue d'une détection d'intrusion.
Nous avons abordé quelques concepts de base des SMAs et un bref aperçu des travaux effectués via ce paradigme dans le domaine des SDIs. Finalement, nous avons introduit quelques méthodes destinées principalement aux SDIs telles que les automates à états finis et les modèles statistiques.

Chapitre 4

Approche proposée, résultats et discussion

Sommaire

4.1 Introduction

Dans le domaine de la sureté de fonctionnement, des terminologies peuvent présenter de manière claire les entraves qui peuvent mener à un mauvais fonctionnement d'un système. Ainsi, trois types d'entraves causalement dépendantes sont définis ; les fautes d'une part, qui sont la cause initiale d'un mauvais fonctionnement ; les erreurs qui sont les conséquences directes de l'activation des fautes et qui peuvent produire causalement d'autres erreurs par propagation dans le système ; ces erreurs peuvent enfin induire des défaillances qui caractérisent la sortie du système de sa spécification.

Dans ce domaine, les fautes qui influencent l'état du système sont souvent des fautes externes intentionnelles avec volonté de nuire au système dans le but de violer ses propriétés de sécurité. Ces fautes sont alors appelées des attaques.

Face à ces erreurs et ces fautes, l'un des moyens d'assurer le bon fonctionnement d'un système est de mettre en place des mécanismes ayant pour objectif d'assurer le bon fonctionnement du système même en présence d'erreurs, et incluant souvent des mécanismes de détection d'erreurs. Ceux-ci ont pour rôle de détecter des états incorrects du système, soit pendant son fonctionnement, soit après. La notion de mécanismes de détection d'erreurs peut donc tout naturellement être étendue à des erreurs dues à des attaques. On nomme ce type de détection la détection d'intrusion [Totel ,2012].

Dans ce chapitre nous allons présenter notre approche proposée en vue d'une détection d'intrusion dans un réseau informatique en utilisant deux niveaux de détection via les Réseaux de Neurones (RNs) afin de renforcer la sécurité ainsi qu'un Système Multi-Agents (SMA).

4.2 Présentation de l'approche proposée

Afin d'aboutir à notre SDIR, nous avons adopté dans approche proposée un réseau de neurones du type RBF doté d'une fonction Gaussienne connue pour sa robustesse au bruit. Les étapes d'apprentissage et de test sont réalisées sur la base de données **CICIDS2017** [iscxdownloads ,2017], une base de données réaliste, récente et qui contient plusieurs types d'attaque. L'avantage principal de cette base et qu'elle est étiquetée ce qui permettra une bonne évaluation. Malheureusement la base n'est pas divisée en base de test et base d'apprentissage, pour remédier à cela nous avons procédé à une étape de prétraitement via la méthode K-means.

Afin d'assurer une bonne simulation du système nous avons fait recours aux SMAs. Ces derniers offrent la possibilité de parallélisme ce qui réduit le temps de traitement des taches indépendantes à savoir la récupération et le prétraitement de la base de données ainsi qu'au niveau de la détection. Les SMAs assurent aussi une certaine décentralisation de l'expertise.

Dans ce qui suit nous allons détailler notre approche proposée où nous verrons toutes les étapes nécessaires à l'aboutissement de notre SDIR à partir du prétraitement jusqu'aux applications, méthodes et données utilisés. Nous présenterons ensuite notre application, ou nous illustrerons les résultats obtenus accompagnés de quelques discussions. La figure suivante montre l'organigramme de notre approche proposée.

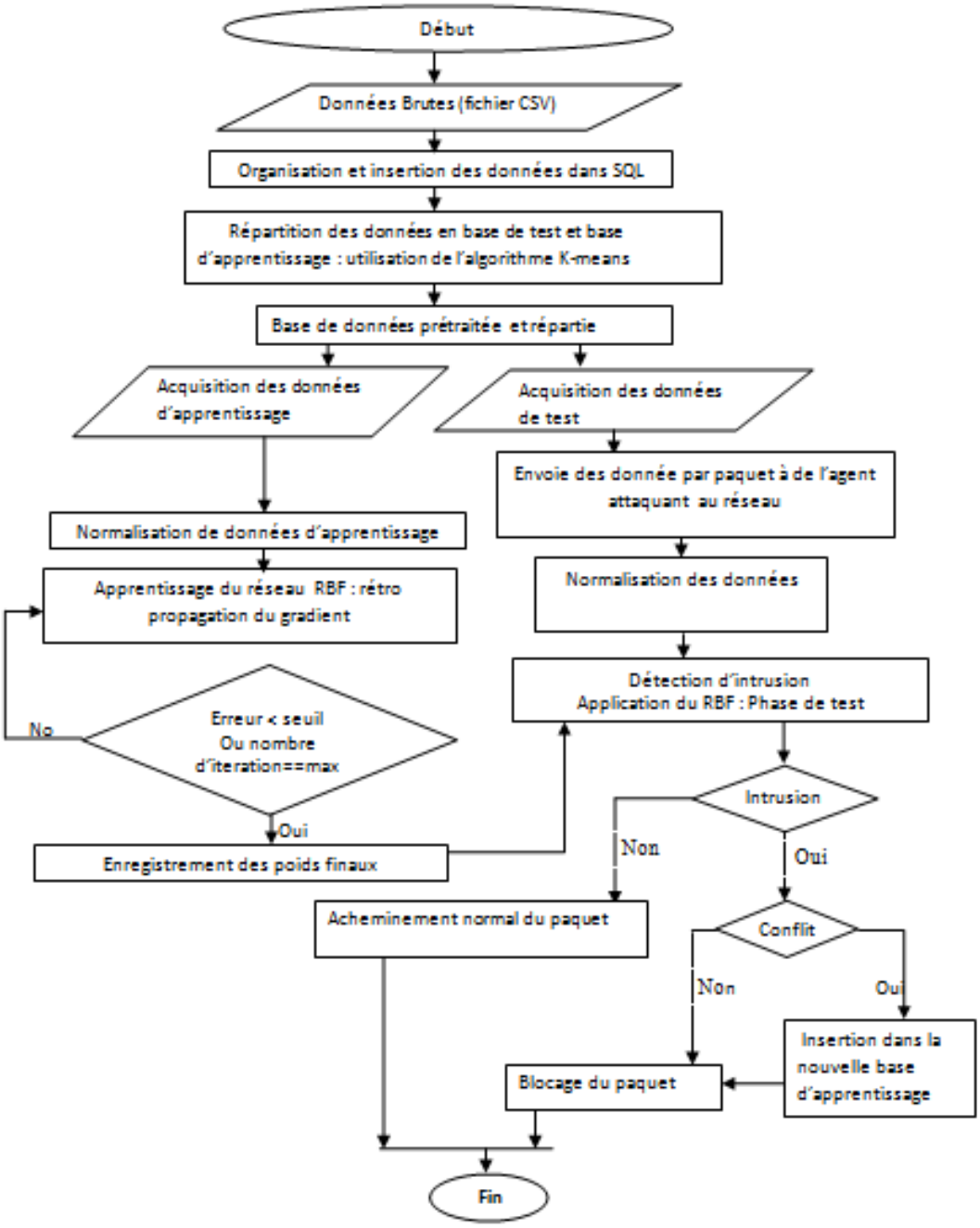

Figure 4.1 – Organigramme de notre SDIR : SMA-RBF

4.3 Prétraitement de Données

La détection des anomalies a été la principale préoccupation de nombreux chercheurs en raison de son potentiel de détection d'attaques nouvelles. Cependant, son adoption dans des applications réelles a été entravée par la complexité du système, qui nécessite un nombre considérable de tests, d'évaluations et de réglages avant le déploiement. L'exécution de ces systèmes sur de véritables traces réseau étiquetées avec un ensemble complet et étendu d'intrusions et de comportements anormaux est la méthodologie la plus idéaliste pour les tests et l'évaluation. Pour remédier à ces problèmes, une approche systématique a été conçue pour générer des ensembles de données permettant d'analyser, de tester et d'évaluer des systèmes de détection d'intrusion, en mettant l'accent sur les détecteurs d'anomalies basés sur le réseau[Sharafaldin ,2018] .

4.3.1 Présentation de la base de données utilisée

Dans notre travail nous utilisons le jeu de données CIC-IDS-2017[iscxdownloads ,2017], qui comprend des flux de réseau étiquetés, y compris des charges utiles complètes de paquet au format pcap. Ce jeu de données comporte des attaques communes et récentes et qui ressemblent aux véritables données du monde réel (PCAP). Il inclut également les résultats de l'analyse du trafic réseau avec des flux étiquetés. L'utilisation d'un jeu de données réaliste et récent était notre priorité absolue dans la construction de notre SDI. Dans ce qui suit, nous allons présenter les attaques contenues dans ce fichier accompagnées de quelques descriptions :

- Bot :[Opendata ,2018] L'attahe botnet utilise une série d'ordinateurs connectés pour attaquer ou détruire un réseau, un périphérique réseau ou un environnement informatique .
- DDos,DoS :[infosecinstitute ,2017] Destributed Denial-of-Service ,Denial-of-Service empêche les demandes légitimes adressées à une ressource réseau en consommant la bande passante ou en surchargeant des ressources informatiques.
- Dos Hulk :[Whatismyip ,2013] Les attaques Dos Hulk (Denial-of-Service Http Unbearable Load King), sont générer par l'outil Denial-of-Service Hulk, ces attaques peuvent provoquer une panne de serveur mal configuré ou d'application mal faite .
- Dos slowloris :[Darkmoreops ,2014] les attaques Denial-of-Service Slowloris sont générer par le programme Slowloris, qui permet à un attaquant de submerger un serveur cible en ouvrant et en maintenant de nombreuses connexion

http simultanées entre l'attaquent et la cible.

• Ftp patator :[Cloudflare ,2015] Patator est un script Python pour les attaques par force brute.

• Dos Goldeneye : GoldenEye est un outil de test HTTP DoS, Il génère des attaques Goldeneye.

• Portscan :[knowledgebase,2013] est un processus qui envoie des demandes de clients à une plage d'adresses de port de serveur sur un ordinateur hôte, dans le but de trouver un port actif .

• Web Attack force brute :[knowledgebase,2013] attaquant utilise un ensemble de valeurs prédéfinies pour attaquer une cible et analyser la réponse jusqu'à ce qu'il réussisse .

Table 4.1 – Extrait de la liste des fonctionnalités d'un flux du trafic avec description [OpenDATA ,2018]

Nom de la fonctionnalité	**La description**
fl_dur	Durée du flux
tot_fw_pk	Nombre total de paquets dans le sens aller
tot_bw_pk	Nombre total de paquets dans la direction arrière
tot_l_fw_pkt	Taille totale du paquet dans le sens aller
fw_pkt_l_max	Taille maximale du paquet dans le sens aller
fw_pkt_l_min	Taille minimale du paquet dans le sens aller
fw_pkt_l_avg	Taille moyenne du paquet dans le sens aller
fw_pkt_l_std	Taille de déviation standard du paquet dans le sens aller
Bw_pkt_l_max	Taille maximale du paquet en arrière
Bw_pkt_l_min	Taille minimale du paquet en arrière
Bw_pkt_l_avg	Taille moyenne du paquet en arrière
Bw_pkt_l_std	écart type de la taille du paquet en arrière
fl_byt_s	débit en octets correspondant au nombre de paquets transférés par seconde
fl_pkt_s	débit de paquets qui correspond au nombre de paquets transférés par seconde
fl_iat_avg	Temps moyen entre deux flux
fl_iat_std	Temps d'écart type deux flux
fl_iat_max	Temps maximum entre deux flux
fl_iat_min	Temps minimum entre deux flux
fw_iat_tot	Temps total entre deux paquets envoyés dans le sens aller
fw_iat_avg	Temps moyen entre deux paquets envoyés dans le sens aller

4.3.2 Étapes de prétraitement

a. Étape 1 : insertion SQL Nous avons remarqué que la lecture de la base de donnée telle quelle était vraiment gourmande en temps vu qu'il s'agit de lire à partir d'un fichier qui est à la fois volumineux et sous format csv. Pour pallier à cela et durant cette première étape nous avons récupéré les données du dossier de la BDD CICIDS2017 [iscxdownloads ,2017] contenant neufs fichiers volumineux de type csv afin de les stocker dans une base de données SQL. Et pour réduire encore le temps de lecture des fichiers et d'insertion des données dans la base de données SQL, nous avons utilisé neuf agents réactifs (threads), ces agents fonctionnent de façon parallèle départageant ainsi la tache de lecture et d'insertion.
L'agent qui fait l'insertion « Agent insertion SQL » est bien détaillé dans la section (4.4.2.a) Une fois cette étape est achevée, notre base de données SQL est organisée comme suit :

Table 4.2 – Les tables des paquets avec et sans intrusion

.

Nom de la table	**Contenue**	**Nombre de ligne**	**Nombre colonne**
benign	Les attaques benign	2272389	79
Bot	Les attaques bot	1966	79
Dos	Les attaque de type Dos	128027	79
DDos	Les attaque de type DDos	10293	79
Dos_Hulk	Les attaque de type Dos_Hulk	230124	79
Dos_slowloris	Les attaque de type Dos_slowloris	5796	79
Ftp_patator	Les attaque de type Ftp_patator	7938	79
Prtscan	Les attaque de type Prtscan	7938	79
Web_force_brut attaque	Les attaque de type attaque web par force brute	1507	79

Table 4.3 – Quelques tables utilisées dans la normalisation

.

Nom de la table	**Contenue**	**Nombre de ligne**	**Nombre colonne**
Norm_benign	Normalisation de la table benign	Tout dépend la taille d'échantillon normalisé : Taille < 2272389	78
Norm_Bot	Normalisation de la table Bot	Tout dépend la taille d'échantillon normalisé : Taille < 1966	78
Norm_Dos	Normalisation de la table Dos	Tout dépend la taille d'échantillon normalisé : Taille < 128027	78
Norm_DDos	Normalisation de la table DDos	Tout dépend la taille d'échantillon normalisé : Taille < 10293	78
Norm_Dos_Hulk	Normalisation de la table Dos_Hulk	Tout dépend la taille d'échantillon normalisé : Taille < 230124	78
Norm_Dos_slowloris	Normalisation de la table Dos_slowloris	Tout dépend la taille d'échantillon normalisé : Taille < 5796	78

b. Étape 2 : application de k-means L'étape d'apprentissage du réseau de neurones RBF est une étape cruciale et qui dépend principalement de la base d'apprentissage qui doit être le plus représentatif possible, englobant ainsi une variété d'exemples ce qui assure lors de l'étape de validation d'avoir de bons résultats.
Hélas la base choisie n'est pas divisée en base de test et base d'apprentissage, afin de résoudre ce problème nous avons procédé à la classification des données via l'algorithme K-means pour par la suite la départager ce qui permet une certaine similarité entre les deux ensembles.
Cette classification a été réalisée avec un logiciel d'analyse de données et de statistiques où nous avons exporté nos tables. Une fois la classification de ces données est achevée on les a réinsérées à nouveaux dans la base de données et plus précisément dans les tables qui leurs sont appropriées.

Table 4.4 – Extrait des valeurs min et max des paquets

.

Nom de la table	Contenue	Nombre de ligne	Nombre colonne
Minmax_Dos	Le minimum et le maximum de chaque colonne de la table Dos	2	78
Minmax_DDos	Le minimum et le maximum de chaque colonne de la table DDos	2	78
Minmax_Dos_Hulk	Le minimum et le maximum de chaque colonne de la table Dos_Hulk	2	78
Minmax_Dos_slowloris	Le minimum et le maximum de chaque colonne de la table Dos_slowloris	2	78
Minmax_Ftp_patator	Le minimum et le maximum de chaque colonne de la table Ftp_patator	2	
Minmax_Prtscan	Le minimum et le maximum de chaque colonne de la table Prtscan	2	78
Minmax_Web_force_brut attaque	Le minimum et le maximum de chaque colonne de la table Web_force_brut attaque	2	78
Probaminmax	Deux valeur générer aléatoirement avec une qui est plus petite que l'autre valeur	1	2

c. Étape 3 : la normalisation À ce stade du prétraitement, il s'agit de mettre les valeurs des différents champs des paquets (flux) dans l'intervalle [0.1] puis les réinsérer dans des tables SQL que nous avons crées spécialement pour stocker les paquets normalisés.
La normalisation est appliquée car les données varient dans des échelles différentes. À l'issue de cette transformation, les valeurs normalisées seront comprises dans un intervalle fixe [0,1].

4.4 Notre système de détection d'intrusion réseau

4.4.1 RBF adopté

Un réseau de neurones est avant tout un outil de traitement de l'information. À force d'apprentissage, on peut le former à des tâches de reconnaissance, classification, approximation, prévision,..... etc .[academia ,2014]
Dans notre approche nous avons opté pour réseau de neurones de type RBF qui est Composé d'un total de trois couches. La première s'appelle la couche d'entrée et se compose de nœuds sources (ou unités sensorielles). La quantité de ces nœuds sources est égale à la dimension du vecteur d'entrée. La seconde est la couche cachée, qui consiste en unités non linéaires. Celles-ci sont directement connectées à chacune des unités sensorielles de la couche d'entrée. Le réseau RBF ne possède qu'une seule couche masquée dotée des fonctions d'activation RBF. Enfin, la couche de sortie est utilisée pour combiner linéairement les sorties de la couche cachée et donner la réponse du réseau aux données d'entrée.

a. L'architecture de notre RBF La couche d'entrée dans notre réseau est fixé à 78 neurones d'entrée, qui corresponde au nombre d'attributs des flux de données de la base CICIDS2017.
Le nombre de neurones au niveau de la couche cachée dans notre réseau de neurones RBF a été fixé à 10 neurones, et cela après plusieurs tests avec différents nombres de neurones au niveau de cette couche cachée dont l'objectif était d'optimiser le temps d'apprentissage ainsi que les résultats.
Nous avons en fait réalisé deux niveaux de détection d'intrusion utilisant alors deux RBFs qui se différencient principalement au niveau de la dernière couche ainsi qu'au niveau des poids déterminés. Le premier niveau alors comporte un RBF avec un seul neurone dans la couche de sortie permettant ainsi de déterminer l'existence ou non de l'intrusion figure (4.2 .a).
Quant au deuxième niveau, il comporte un RBF doté de 9 neurones dans la couche

de sortie, ou 8 neurones représentent les types d'attaques et le dernier représente l'état normal figure (4.2 .b).

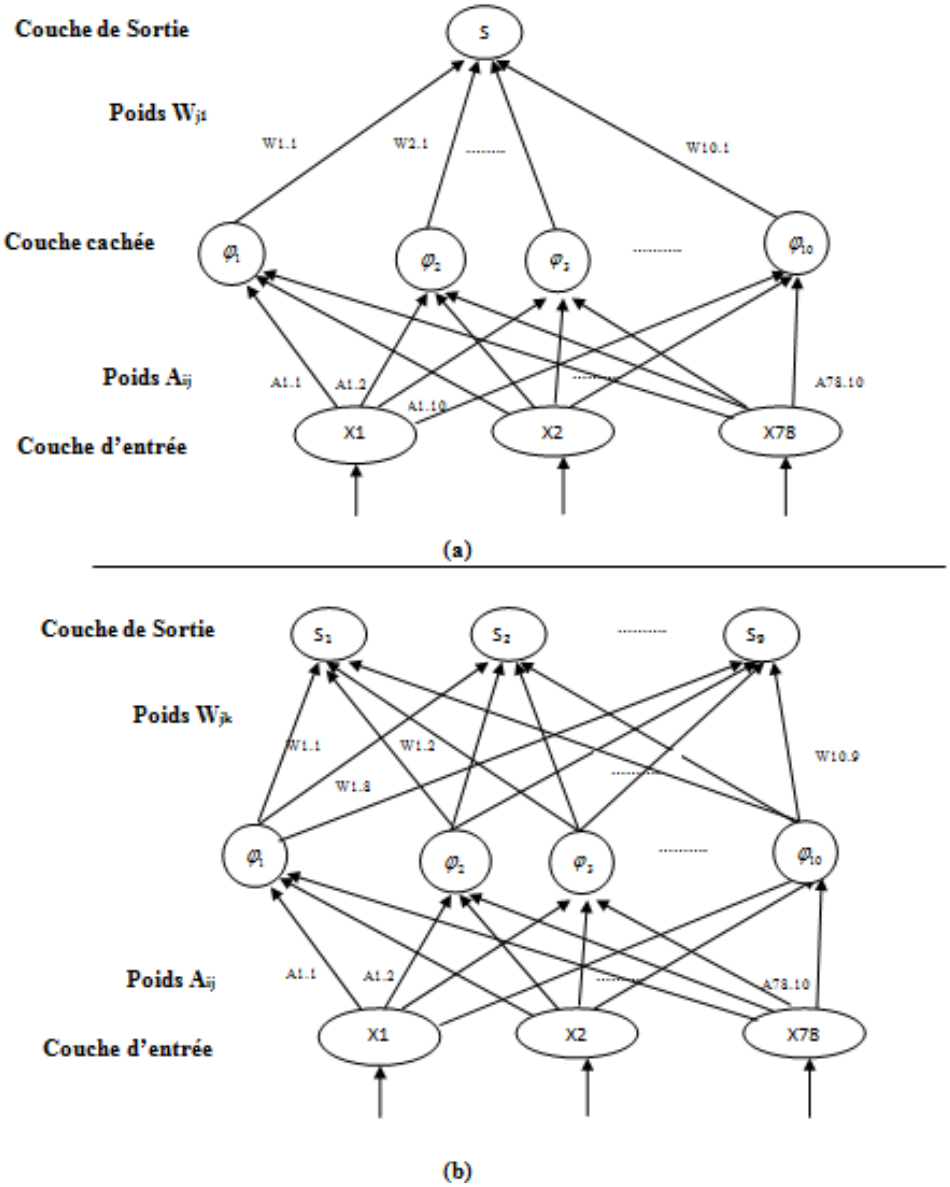

Figure 4.2 – a.Architecture de notre RBF du premier niveau b.Architecture de notre RBF dans le deuxième niveau de détection

La fonction Gaussienne dans la couche cachée est définit comme suit :[Kumar Dash, Ajit Kumar Behera, 2016]

$$\varphi(r) = \exp(\frac{r^2}{2\sigma^2}) \tag{4.1}$$

Tel que : σ : *est.l'étendue*

$$\sigma = \frac{Distance.max.entre.deux.centres.arbitraires}{\sqrt{Nombre.de.centres}} = \frac{d_{max}}{\sqrt{m_1}} \quad (4.2)$$

$$r = \|x - c\| \quad (4.3)$$

et

$\|x - c\| : est.la.distance.entre.x = (x_1, ...m_1).et.le.vecteur.\mathbf{c} = (c_1, ..., c_m)$

Avec cj le centre d'un attribut, m le nombre d'entrée égale à 78 .
Quand à l'unité de sortie ou fonction d'activation/sortie linéaire est calculée comme suit :
.

$$S_k = \sum_{j=1}^{10} w_{jk}\varphi_j(||x - c||) \quad (4.4)$$

tel que :
.
Les Wjk représentent les poids entre la couche cachée et la couche de sortie avec j apartien à [1,10] et k dans le RBF du premier niveau égale à 1 quand au RBF du deuxième niveau apartien à [1,9]
.
Quand aux poids Aij entre la couche d'entrée et la couche cachée sont mis à 1.

b. Phase d'apprentissage du RBF L'apprentissage ou la formation d'un réseau de neurones est un processus par lequel le réseau s'adapte à un stimulus en ajustant correctement les paramètres, ce qui entraîne la production de la réponse souhaitée.
Par conséquent, afin d'obtenir une précision d'approximation / classification similaire et en plus du nombre requis d'unités de RBF, les paramètres suivants sont déterminés par le processus d'apprentissage du RBF :

1. Le nombre de neurones dans la couche cachée est choisi après plusieurs essais.
2. Les coordonnées du centre de chaque couche cachée RBF. Il existe différentes techniques de classification qui peuvent être utilisées pour localiser les centres RBF (ci). Dans notre travail nous avons utilisé l'algorithme de classification K-means. Partitionner N points de données de Xn en j groupes.
3. Le rayon (propagation) de chaque RBF dans chaque dimension.
4. Les poids Aij sont mis à 1, alors que les poids Wjk du RBF sont déterminés par la rétro-propagation du gradient comme suit :

$$[w_{1k}...w_{10k}]^T = \Phi^+[d_1...d_N]^T \tag{4.5}$$

Tel que : $[\mathrm{w}_{1k}...w_{10k}]^T$:

La transposé du vecteur (matrice) des poids qui lies la couche cachée avec la couche de sortie.

$[\mathrm{d}_1...d_N]^T$:

La transposé du vecteur des sorties du réseau.

$$\Phi^+ = 1 \begin{bmatrix} \varphi_1(||x_1 - c_1||)...\varphi_{10}(||x_1 - c_{10}||) \\ ... \\ \varphi_1(||x_m - c_1||)...\varphi_{10}(||x_m - c_{10}||) \end{bmatrix}$$

equation

(3.6)

Une fois ces poids sont déterminés, ils sont stockés dans une table de la base de données pour les réutilisé lors de l'étape de test.

4.4.2 Système multi-agents proposé

Une question de conception commune à tout SDI est de savoir comment maximiser les avantages et minimiser les pénalités associées aux approches basées sur le réseau.
Afin d'atteindre nos objectifs de détection d'intrusion et implémenter efficacement notre SDI nous avons fait recours aux SMAs en tant qu'environnement de système. Cette mise en œuvre facilitera l'échange de données entre les agents de notre SDI, permettra d'assurer le parallélisme des taches indépendantes et puis aussi garantir une certaine distribution d'expertise. L'architecture générale de notre SDIN est illustrée par la Figure (4.3) .

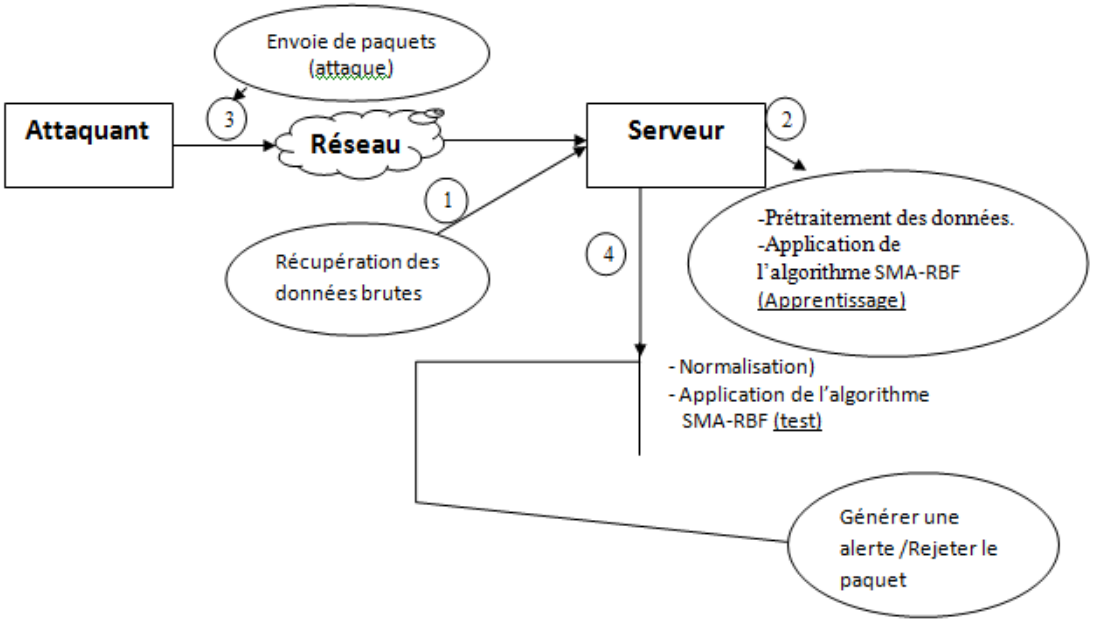

FIGURE 4.3 – Architecture générale de notre SDI

a. Les agents de notre SMA

Les agents sont des entités intelligentes qui fonctionnent dans divers domaines. Un agent est caractérisé par son architecture interne qui consiste aux mécanismes qui régissent son comportement. Ce point de vue interne correspond aux étapes de perception, de raisonnement et d'action avec une mention particulière à la décision d'action lors de l'étape de raisonnement. Il englobe ainsi l'ensemble des mécanismes, plus ou moins complexes, produisant le comportement de l'agent [Patakyet Damas Gruska ,2014]

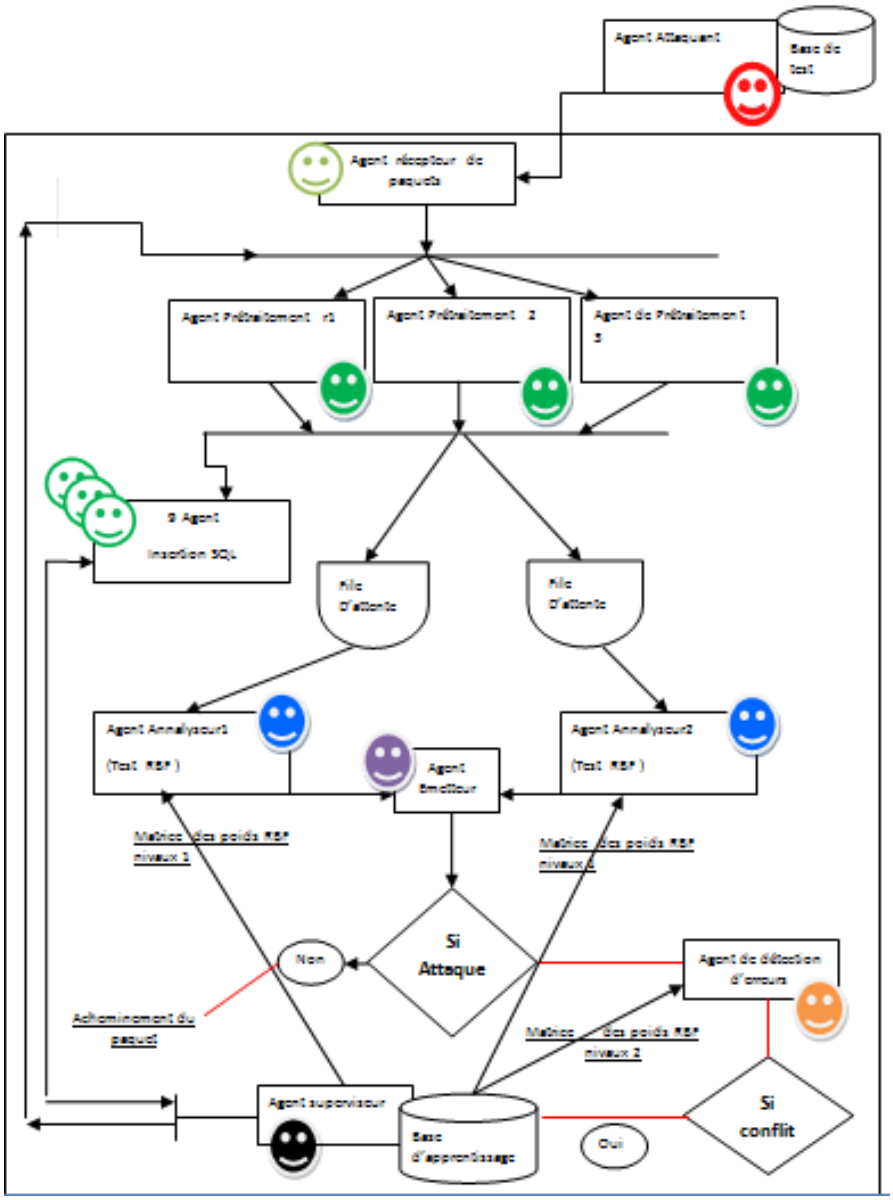

Figure 4.4 – Architecture générale et mode de fonctionnement de notre SMA-SDI

1) Agent Attaquant Cet agent a un comportement d'un attaquant qui veut exploiter les vulnérabilités des machines au sein du réseau, il se trouve dans une machine à part qui est connectée au réseau. Il a comme tâche d'envoyer un ensemble de paquets de chaque type d'attaque ainsi que des paquets normaux qui ne contienne aucune anomalie. La sélection des paquets se fait à partir de la base de données CICIDS2017[iscxdownloads ,2017] et plus précisément la base de test vers une machine victime.

2) Agent Superviseur Cet agent se situant au niveau du serveur. Son rôle est de réaliser la phase d'apprentissage avec RBF puis envoyer les poids aux agents analyseurs. Il s'agit de l'agent principal unique responsable du lancement des autres agents du système qui sont :

- L'agent de prétraitement.
- L'agent insertion SQL.
- L'agent Récepteur.
- L'agent Analyseur.
- L'agent de détection d'erreur.
- L'agent Emetteur.

3) Agent de prétraitement L'agent contrôleur existe en trois instances qui font le même travail. Le rôle de l'agent contrôleur est de récupérer les données de la liste des paquets reçus par l'agent récepteur et de les normaliser .Une fois ce processus est finis, il commence à envoyer les données normalisées aux agents Analyseurs de façon aléatoire.

Nous avons appliqué la normalisation car les données varient dans des échelles différentes. A l'issue de cette transformation, les valeurs normalisées seront comprises dans un intervalle fixe [0,1]. Le but d'avoir un tel intervalle restreint est de réduire l'espace de variation des nouvelles valeurs et par conséquent réduire l'effet des valeurs aberrantes.

Notre transformation se fait grâce à la formule suivante :

$$(X_{new}) = \frac{(X_{max} - X_{old})}{(X_{max} - X_{min})} * (X_{amax} - X_{amin}) \tag{4.7}$$

Avec :

- Xmin : la plus petite valeur observée dans la colonne.
- Xmax : la plus grande valeur observée dans la colonne.
- Xold : La valeur qu'on cherche à normaliser.

- Xamin : C'est la valeur minimale générée aléatoirement, elle se trouve dans l'intervalle [0 , 1].
- Xamax : Valeur généré aléatoirement ,elle se trouve dans l'intervalle]Xamin , 1] .
- Xnew : c'est la valeur final de normalisation

4) Agent Récepteur Le rôle de cet agent, dans un premier temps est de recevoir les paquets envoyés par l'attaquant puis les enregistrer dans un tableau où il commence à les délivrer au réseau où ils seront récupérés par les agents de prétraitements en premier lieu.
L'état de chaque paquet est à 0 par défaut, une fois délivré à un agent prétraitement, l'agent récepteur met la valeur de son état à un pour qu'il ne le transmet pas à nouveau

5) Agent Analyseur Les poids obtenus suite à la phase d'apprentissage du RBF du premier niveau sont enregistrés afin d'être utilisés par les agents analyseur lors du test. Si le paquet ne contient aucune intrusion alors il est acheminé à sa destination sinon il est récupéré par l'agent de détection d'erreurs.

6) L'agent de détection d'erreurs Si un des agents analyseurs détecte une anomalie, le paquet concerné est récupéré par l'agent de détection d'erreurs où il sera analysé une deuxième fois afin de déterminer le type d'attaque .Si la réponse est paquet normal cela veut dire qu'il existe un conflit alors le paquet en question est réinséré dans la base d'apprentissage.
Le paquet n'est acheminé à sa destination que lorsqu'il est détecté par l'agent analyseur comme paquet normal.

7) Agent Emetteur L'agent permet d'afficher la liste des paquets reçus et ceux bloqués

4.5 Présentation de notre application

4.5.1 Les outils utilisés

a. Caractéristiques techniques du matériel utilisé

Processeur : Intel®Core (™) i3-6006U

Mémoire : 4.00 Go

Système d'exploitation : Windows 7 64 bits

b. Environnement de développement (IDE)

Les avantages offerts par la programmation orientée objet et la programmation multiThreads sont les principales raisons pour lesquelles nous avons choisi le langage de programmation JAVA, mais il faudrait toutefois noter que l'utilisation de la plate-forme jade a renforcé ce choix. Effectivement cette dernière est entièrement programmée en JAVA d'où l'implémentation du SMA ne peut être qu'avec ce langage.

Afin d'implémenter notre SDIR, nous avons utilisé l'IDE NetBeans 8.2 qui comporte des bibliothèques de classes très riches comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, contrôles, menus, graphismes), la gestion des exceptions, la variété des types, la gestion des fichiers, l'ajout automatique des déclarations d'importation, etc.. .Tout cela nous a été utile dans la conception de l'ensemble de notre système de détection d'intrusion.

C. Plate-forme Jade

lors de notre approche proposée nous avons utilisé les SMAs par conséquent nous avons fait appel à une plate-forme multi-agents et plus précisément la plate-forme JADE.

JADE pour Java Agent Développement Framework est un Framework logiciel permettant le développement d'applications d'agents conformes aux spécifications FIPA pour l'interopérabilité des systèmes multi-agents intelligents. JADE fonctionne sous tous les systèmes d'exploitation, inclut tous les composants Obligatoires qui contrôlent un SMA [Duro et Hoarau ,2004]. C'est un projet Open Source, LGPL License contrôlée par Telecom Italia Lab (TI-LAB), qui reste propriétaire du projet.

d. PhpMyAdmin

Dans notre travail nous avons fait recours aux SGBD MySQL afin de réduire le temps d'accès aux données qui sont d'un volume assez important. Pour cela nous avons utilisé le ***phpMyAdmin*** qui est un logiciel open source de gestion de bases de données MySQL qui permet de gérer les BDDs. Le logiciel

dispose d'une interface intuitive et personnalisable qui permet ainsi de créer, copier et déplacer les bases de données, d'éditer les tables et les champs, d'exécuter une requête SQL,...

4.5.2 L'interface principale de notre application

1) Machine 1 (serveur)

Sur cette machine réside l'agent superviseur. Dans ce qui suit nous allons montrer son IHM ainsi que les principales fonctionnalités proposées dans notre SDIR.

a. Fenêtre principale cette fenêtre apparait au lancement de l'application donnant accès à d'autres fonctionnalités sous forme de boutons :

- **Le bouton « Chargement »** : permet le chargement de la base de données qui est en format « CSV ».
  dans des listes afin de l'insérer dans phpMyAdmine en format « SQL ».
- **Le bouton « Apprentissage »** : ce bouton va nous mener sur une autre fenêtre, il s'agit de la fenêtre qui concerne l'apprentissage de notre RBF, cette dernière est illustrée dans la figure (4.5).
- **Le bouton « Lancer les agents »** : pour lancer les autres agents et les rendre actif, on doit cliquer sur le bouton lancement.

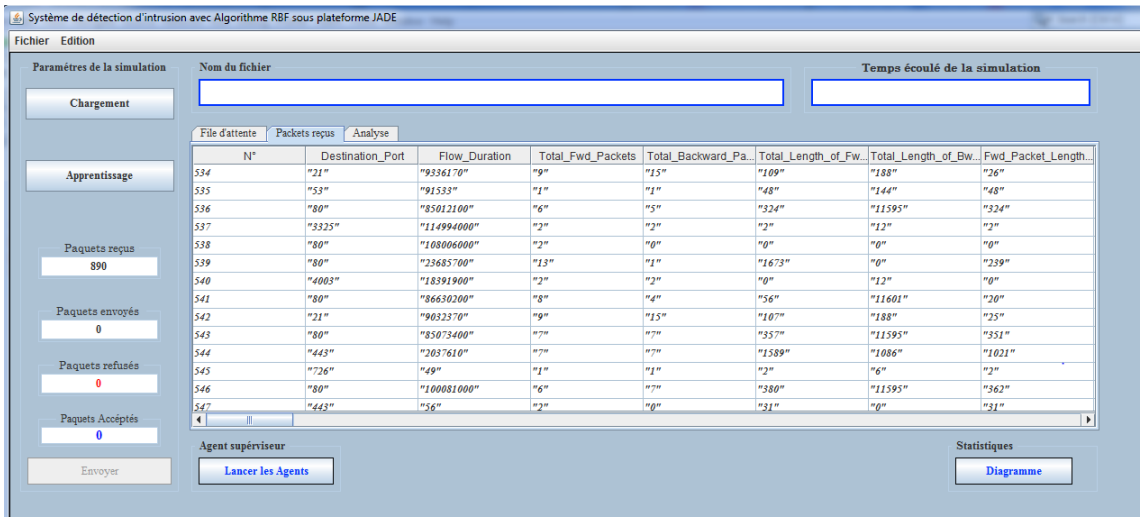

FIGURE 4.5 – Fenêtre principale de notre SDIN (Agent superviseur)

b. Fenêtre d'apprentissage à partir de cette fenêtre nous pouvons démarrer la normalisation de la base de données (CICIDS2017), déterminer tous les paramètres d'apprentissage du réseau de neurones RBF et démarrer la phase d'apprentissage

tout en spécifiant la taille de la BDD qu'on souhaite utiliser et cela par attaque. La fenêtre comporte les boutons suivants :

- **Le Bouton « Initialisation »** : permet de spécifier la taille de la BDD d'apprentissage(en nombre de paquets) par type d'attaque.
- **Le Bouton « Lancer »** : sélection de la BDD d'apprentissage à partir des tables SQL selon les proportions précisées dans l'échantillonnage.
- **Le Bouton « Normalisation »** : permet de procéder à la normalisation de la base d'apprentissage et les insérer dans des tables SQL.
- **Le Bouton « Générer Matrice »** : Afficher la base de données normalisée sous forme d'un tableau .
- **Le Bouton « Lancer RBF »** : lancer la phase d'apprentissage de notre RBF sur la base de données sélectionnée et normalisée, une fois cette phase est terminée la matrice des poids générée est affichée dans le tableau « Poids » et les sorties du RBF dans le tableau « Classe de sortie ».
- **Le Bouton « Enregistré (Wi) »** : enregistrement des tableaux« Poids » et « Classe de sortie » dans la base de données SQL.

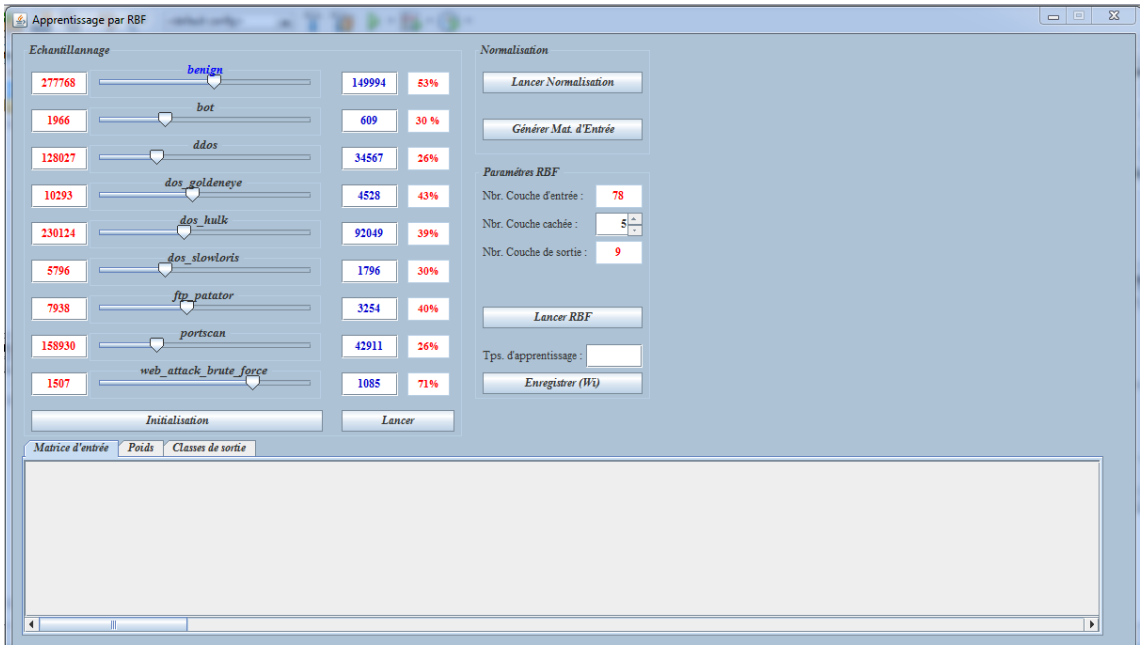

FIGURE 4.6 – Fenêtre d'apprentissage

c. Fenêtre de l'agent récepteur L'agent récepteur existe en une seule instance, son rôle est de récupérer les paquets à partir du réseau qui lui ont été destiné et de les envoyer aux agents de prétraitement.

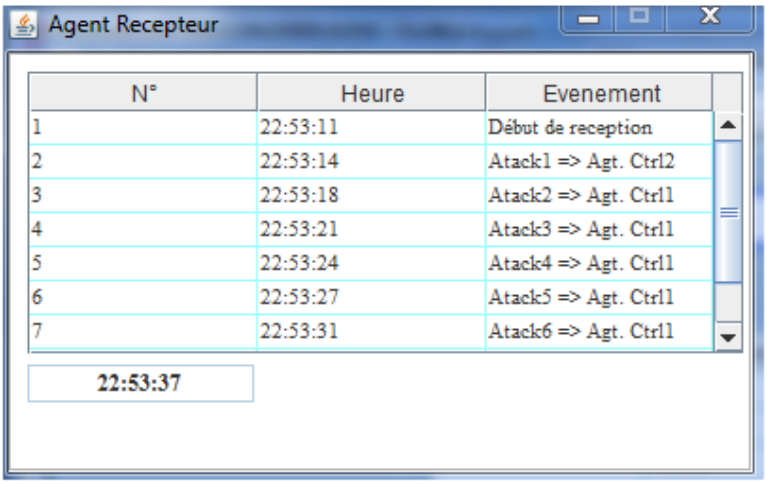

Figure 4.7 – Fenêtre de l'agent récepteur

d. Fenêtre de l'agent de prétraitement Cet agent existe en trois instances dans notre système de détection d'intrusion, il est responsable de la normalisation des paquets qu'ils lui sont délivrés par l'agent récepteur et les transférer aux agents analyseurs.

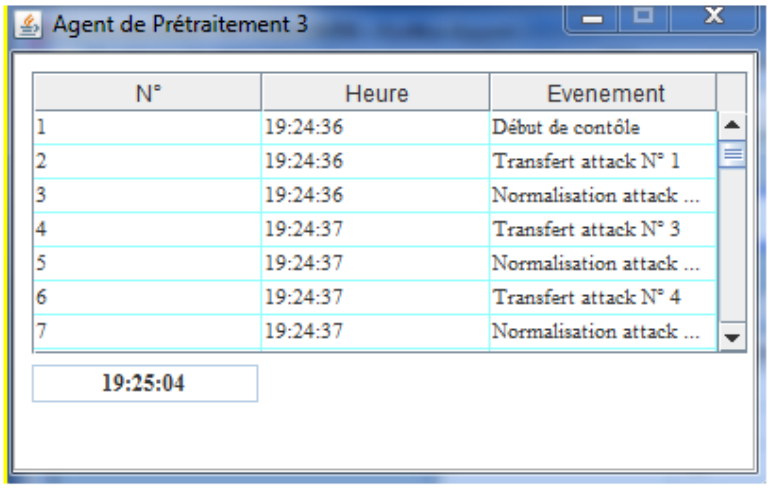

Figure 4.8 – Fenêtre de l'agent de prétraitement

e. Fenêtre de l'agent analyseur La Figure (4.9) montre l'agent responsable de la détection d'intrusion dans notre système. Notre système dispose de deux agents analyseurs

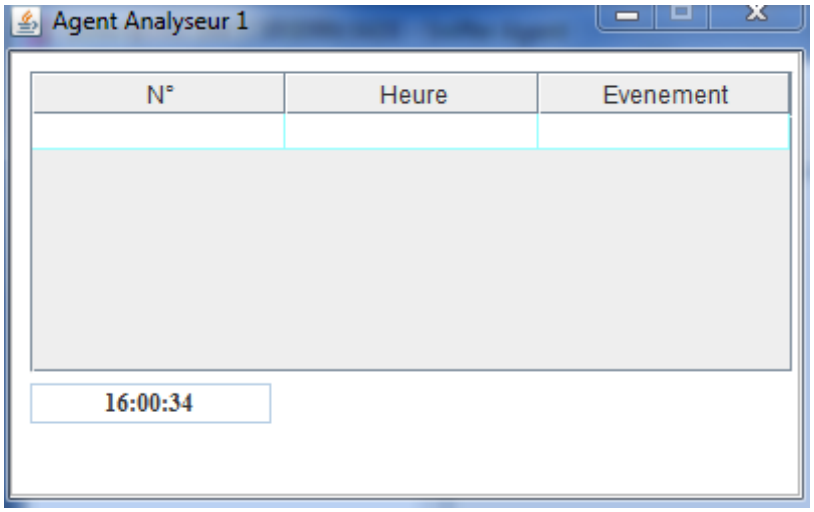

Figure 4.9 – Fenêtre de l'agent analyseur

f. Fenêtre de l'agent émetteur Le rôle de l'agent qui apparaît dans la figure ci-dessous Figure (4.10) est d'afficher l'état des paquets bloqués ayant été détectés comme intrusion ou ceux qui doivent être acheminés.

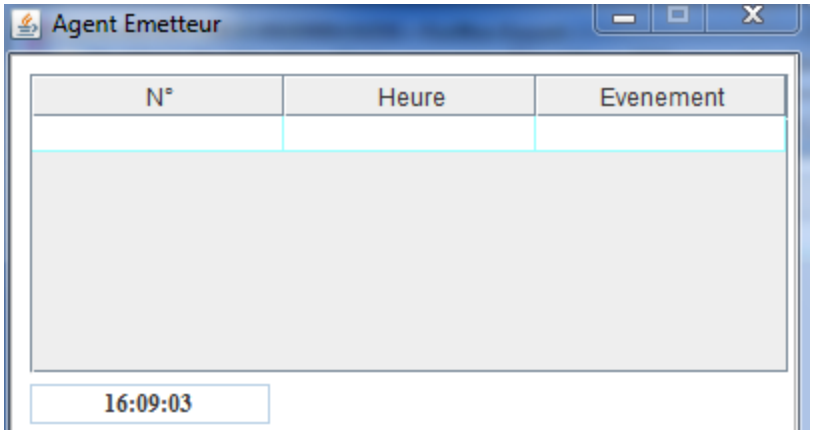

Figure 4.10 – Fenêtre de l'agent émetteur

2) Machine 2 (attaquant)

Sur cette machine se trouve l'attaquant, c'est-à-dire qu'à partir de cette machine que l'attaquant essayera d'exploiter les vulnérabilités de la machine victime en envoyant des paquets malveillants ou non via le réseau.

a. Fenêtre Principale cette fenêtre est utilisée dans le but de lancer l'agent attaquant d'où l'importation et le lancement des attaques. Elle permet également la visualisation du résultat de l'SDIR que nous proposons. Les principaux Boutons de cette fenêtre sont :

- **Le Bouton « Importer les attaques »** : permet importer les fichiers qui contiennent les paquets normaux et les paquets avec intrusion.
- **Le Bouton « Générer des attaques »** : récupère à partir des fichiers importés des paquets normaux et les paquets d'attaques et les mettre dans **la liste des intrusions**, le nombre de paquet mis dans cette liste dépend du pourcentage précisé par l'attaquant.
- **Le Bouton« Lancer l'attaque »** : permet d'envoyer les paquets qui ont été déjà générés dans la liste des intrusions, à **la machine 1**, via le réseau.

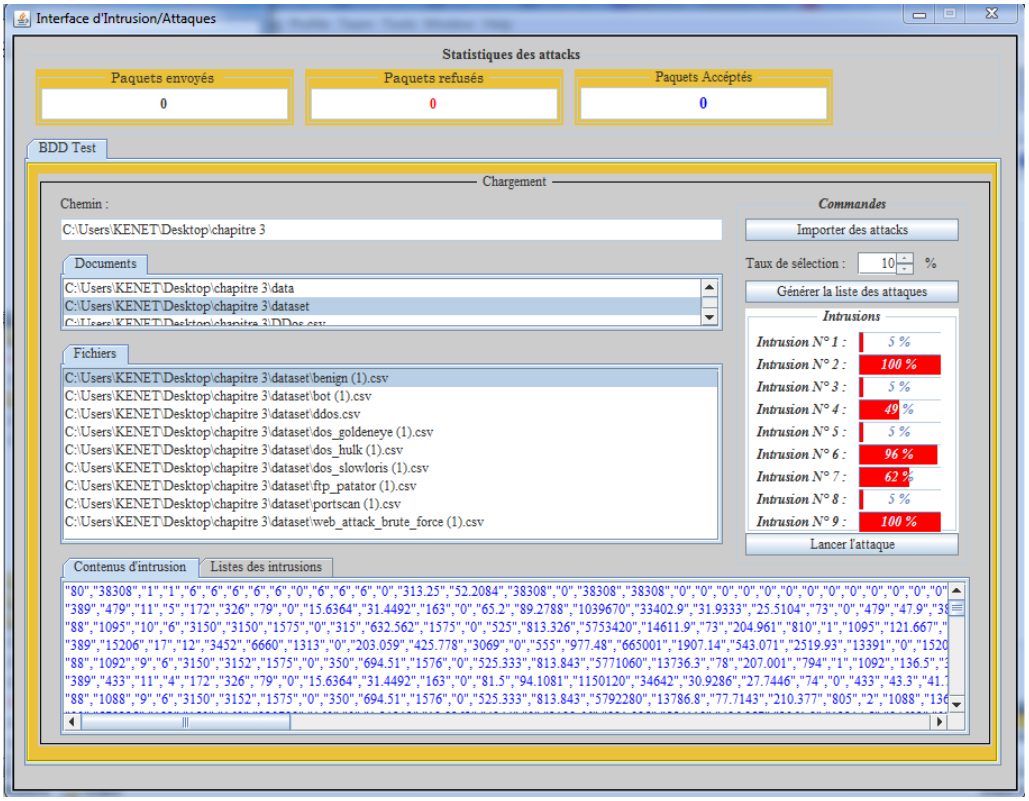

Figure 4.11 – La fenêtre de l'agent 'attaquant

4.6 Les mesures d'évaluation

La matrice de confusion est utilisée pour mesurer les performances de l'algorithme RBF. Ceci fournit une visualisation de la performance du classificateur sur le jeu de données en entrée. Un certain nombre de mesures de performance différentes, y compris le rappel et la précision, sont dérivées de la matrice de confusion. Le tableau 4 montre la structure de cette matrice. Les 4 cas possibles sont :[Bijone ,2016]

Vrai positif (VP) : une attaque correctement détectée par le test.
Faux positif (FP) : une activité normale détectée comme attaque par le test.
Vrai négatif (VN) : une activité normale correctement détectée par le test.
Faux négatif (FN) : une attaque détectée comme activité normale par le test.

Table 4.5 – Matrice de Confusion

.

Classe prédite			
		Positive	**Négative**
Classe Actuel	**Positive**	VP	FP
Classe Actuel	**Négative**	FN	VN

4.6.1 La précision

Cette métrique, également relative à chaque catégorie, renseigne sur la probabilité qu'une prédiction d'une catégorie donnée soit correcte.

$$Précision = \frac{\text{VP}}{VP + FP} * 100\% \tag{4.8}$$

4.6.2 Le taux de détection (Rappel)

C'est le rapport entre le nombre d'intrusions correctement détectées et le nombre total d'intrusions. Et décrit par la formule :

$$Rappel = \frac{\text{VP}}{FN + VP} * 100\% \tag{4.9}$$

4.6.3 Le taux de faux positif (FP)

Le taux des fausses alertes est calculé comme le rapport entre les nombres de trafic normal qui sont incorrectement classés comme intrusions et le nombre total de trafic normal.

$$FP=\frac{\text{FP}}{VN+FP}*100\% \tag{4.10}$$

4.6.4 Le taux de réussite (Accuracy)

Nous avons évalué cet algorithme en utilisant le taux de réusite comme métrique de performance. L'exactitude, dans ce cas, représente le taux de précision globale de la classification de l'ensemble de données de testes. Elle traduit le rapport entre les détections correctes et les détections totales obtenues. Elle est donné par :

$$Accuracy=(\frac{\text{VP}+\text{VN}}{FN+FP+VP+VN})*100\% \tag{4.11}$$

4.7 Discussion des résultats

4.7.1 Paramètres de test

Nous avons effectués plusieurs tests dans notre expérimentions en utilisons des parties de la Base de données de test CICIDS2017 [iscxdownloads ;2017].
Nous avons expérimentés notre SDI trois fois avec des parties différentes de la base de données de test.
Les paramètres de test de l'RBF sont comme suit :
Paramètre de RBF 1 :

Table 4.6 – Paramètres d'apprentissage du RBF1

Paramètre	**Valeur**
Nombre de neurone dans la couche d'entrée	78
Nombre de neurone dans la couche cachée	10
Nombre de neurone dans la couche de Sortie	1

Paramètres du RBF 2 :

Table 4.7 – Paramètres d'apprentissage du RBF2

Paramètre	**Valeur**
Nombre de neurone dans la couche d'entrée	78
Nombre de neurone dans la couche cachée	10
Nombre de neurone dans la couche de Sortie	9

Table 4.8 – Evaluation des résultats obtenus avec algorithme RBF avec SMA

Taux de la base de test	Nombre de neurone de la couche cachée RBF1	Nombre de neurone de la couche cachée RBF2	Précision	Rappel	Faux Positif %	Accuracy	Taux d'Erreur
10%	10	10	97%	89,9%	11,11%	94,28%	0,7%
20%	10	10	97,8%	98,9%	11,4%	98%	0,1%
30%	10	10	98,2%	98,5%	3,4%	98,6%	0,11%

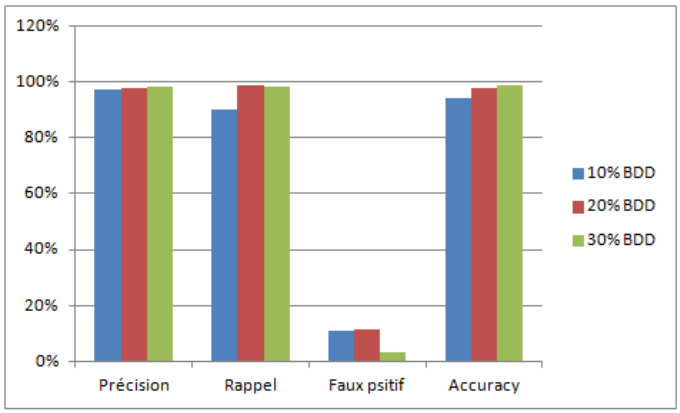

Figure 4.12 – Comparaison des résultats obtenus avec l'algorithme RBF sur les trois expériences

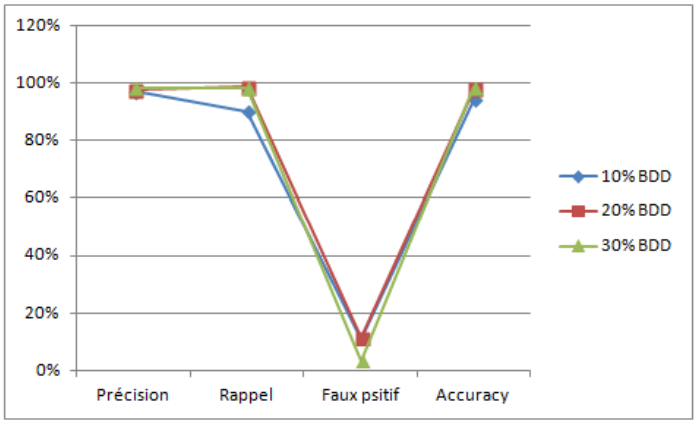

Figure 4.13 – Evaluation des résultats obtenus avec RBF-SMA

La matrice de confusion indique le nombre d'instances affectées à chaque classe. Le tableau(4.9) montre combien d'instances pour chaque classe ont reçu différentes classifications. La somme des diagonales représente la quantité d'échantillons correctement classés. Par exemple, la quantité totale d'échantillons pour RBF qui ont été correctement classés est la somme de **52450, 645, 42619, 3356, 76632 , 1891, 2634, 2619** et **487**.

Table 4.9 – Matrice de confusion pour RBF en tant que classificateur sur la base de test choisie

	benign	**Bot**	**Dos**	**DDos**	**Dos Hulk**	**Dos slow-loris**	**Ftp pa-ta-tor**	**Port scan**	**Web force brut at-taque**	**Erreur %**
benign	**52450**	**1**	**10**	**5**	**20**	**0**	**0**	**10**	**4**	**0,11%**
Bot	**1**	**645**	**0**	**0**	**4**	**0**	**1**	**0**	**0**	**0,91%**
Dos	**10**	**0**	**42619**	**40**	**10**	**5**	**1**	**0**	**0**	**0,15%**
DDos	**5**	**0**	**40**	**3356**	**7**	**13**	**0**	**0**	**0**	**1,8%**
Dos Hulk	**20**	**4**	**10**	**7**	**76632**	**20**	**4**	**0**	**1**	**0,08%**
Dos slowloris	**0**	**0**	**5**	**13**	**20**	**1891**	**3**	**0**	**1**	**2,01%**
Ftp pa-tator	**0**	**1**	**1**	**0**	**4**	**3**	**2634**	**9**	**2**	**0,75%**
Prtscan	**10**	**0**	**0**	**0**	**0**	**0**	**9**	**2619**	**7**	**0,98%**
Web force brut attaque	**4**	**0**	**0**	**0**	**1**	**1**	**2**	**7**	**487**	**2,98%**

4.8 Conclusion

Dans ce chapitre, nous avons présenté toutes les étapes que nous avons parcourues afin d'aboutir à notre SDIR. Nous avons alors détaillé les méthodes que nous avons adoptées dans notre approche proposée et que nous avions appliquée durant la phase d'implémentation.

Vu la taille énorme de la base de données où elle contient plus de trois millions de paquets nous avons procédé à une étape de prétraitement qui englobe la normalisation de données et leurs insertions à SQL ce qui a permis un accès plus rapide aux données.

Sachant que le temps est un atout important, nous avions alors fait recours aux SMAs profitant ainsi de l'une de leurs particularités les plus importantes qui est le parallélisme. Nous avons utilisé plusieurs agents procédant à la récupération et le prétraitement des donnés ce qui permet de réduire considérablement le temps d'accès à la BDD.

La base bien qu'elle soit étiquetée elle n'est malheureusement pas répartie en base de test et base d'apprentissage or nous l'avions répartie nous-mêmes en réalisant

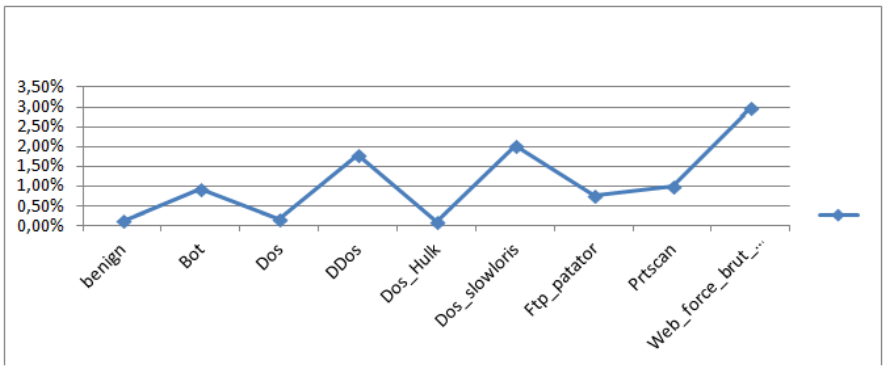

Figure 4.14 – Evaluation d'erreur obtenue par les résulta

d'abord un pré-classification via k-means pour assurer une certaine similarité entre les ensembles et pour garantir une bonne performance de notre SDI.
Nous avons également réalisé deux niveaux de détection via les RBF afin de sécuriser encore plus le réseau et de prendre en charge ainsi les cas de conflit en les enregistrant dans la nouvelle base d'apprentissage .Quant au SMA ils nous ont permis une distribution d'expertise, une réduction du temps au niveau de la récupération de la BDD ainsi qu'au niveau de détection où il suffit encore d'augmenter le nombre des agents analyseurs.

Comparaison Au cours du chargement du jeu de données dans la les tables SQL. Nous avons fait une comparaison avec et sans agent dans des machines de différentes caractéristiques.

Caractéristiques de la machine utilisée				
Processeur	**Mémoire**	**Système d'exploitation**	**Temps de chargement /h**	
			Avec les agents	Sans les agents
Intel R Core (TM) i3-6006U	4.00 Go	Windows 7 64 bits	126 h	Impossible
Intel Core i7-7500U	8.00 Go	Windows 7 64 bits	53 h	236 h

Figure 4.15 – Comparaison de temps d'insertion des tables SQL dans des différentes machines

Conclusion générale

Chapitre 5

Conclusion générale et perspective

La sécurité des réseaux informatiques reste un problème très sensible à cause des attaques informatiques successives qui augmentent jour en jour, cette augmentation présente un problème majeur pour les réseaux informatiques. De plus, la détection en temps réel est devenue plus importante. Par conséquent, différentes méthodes ont été développées .

Ce projet nous a permis dans le deuxième chapitre de découvrir les systèmes de détection d'intrusion à savoir leurs fonctionnements et leurs critères de classification .

Afin de réaliser le SDI, nous avons adopté un System Multi-Agent tenant compte des caractéristiques qu'il fournit.

Notre approche se repose sur un nouveau paradigme de SMA basées sur l'algorithme K-means et les neurones artificielles RBF .

Pour tester et évaluer notre travail, nous avons choisi de travailler avec la base de données récente CICIDS2017, spécialement conçue pour évaluer SDI.

Les résultats obtenus sont satisfaisants, la plupart des objectifs du thème ont été atteint. Notre SDIN SMA-RBF parvient à distinguer les types d'attaques en plus de la détection, il parvient notamment à assurer un deuxième niveau de sécurité en faisant recours à la détection d'erreurs. Mais il reste des perspectives et des perfectionnements à réaliser tels que la minimisation des attributs des paquets via la sélection des attributs les plus significatifs.

Bibliographie

[**Abbass et Aouadi** ; **2016**] Abbass et Aouadi.« Détection d'intrusions dans les réseaux LAN : IDS Snort sous LINUX », Mémoire fin d'etude,univeristé d'Alger,Algérie(2016).

[**Adriana et al.** ; **2017**] Adriana Leite.« A hybrid and learning agent architecture for network intrusion detection ». In :Journal of Systems and Software , pages 59 à 80 Rosario Girardi(2017).

[**Ali Khan et al** ; **2011**] M. Sadiq Ali et Khan SMAqil Burney.« Techniques efficaces de FSM pour IDS afin de réduire les attaques système ». In : International Journal of Computer Applications(2011).

[**Al-Yaseen et al.2017**] Al-yaseen , ZA Othman..« Real-time multi-agent system for an adaptive intrusion detection system ». In : MZA Nazri Pattern Recognition Letters 85, 56-64(2017)

[**Al-Yaseen et al.** ; **2017**]Al-Yaseen, Othman, et Nazri.« Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system ». In : Expert Systems with Applications, 67, 296-303 ,(2017)

[**Barbara** ; **2001**] D.Barbara, J.Couto, S.Jajodia, et N.Wu.« ADAM : A test bed for exploring the use of data mining in intrusion detection ». In : SIGMOD, vol30, no4(2011)

[**Benotmane et Benglia** ; **2010**]Benotmane.M.SD et Benglia. M.E.« Simulation par SMA d'un système multi Robots Flexible pour l'exploration

l'identification, recherche et traitement ». mémoire de fin étude, Université Kasdi Merbah Ouaregla, algerie(2010).

[**Bonner** ; **1964**]Bonner, R. E.« On some clustering techniques ». In : IBM journal of research and development, Article 8(1), 22-32 (1964).

[**Bourouh et kanoun** ; **2017**] BOUROUH Mouloud et KANOUN Zakaria.« Détection d'intrusions à base des réseaux de neurones et algorithmes génétiques », mémoire de fin d'études ,Telemcen, Algerie(2017)

[**Bouzid** ; **2001**] Bouzid Makram.« Contribution à la modélisation de l'interaction agent /environnement », Thèse de doctorat,Université Henri Poincaré Nancy 1. Faculté des sciences et techniques (autre partenaire),Françe(2001) .

[**Burgermeister et Jonathan** ; **2006**] David Burgermeister et Jonathan Krier.« Les systèmes de détection d'intrusions », EC Ezin, HA Djihountry. In :International Journal of Computer (2011)

[**Bijone** ; **2016**]. Manu Bijone.« A Survey on Secure Network : Intrusion Detection Prevention Approaches ». In : American Journal of Information Systems. Vol. 4, No. 3, pp 69-88. http ://pubs.sciepub.com/ajis/4/3/2(2016)

[**Chen et al.** ; **2009**] Chen, Zhifeng et Peide Qian.« Intelligent Information Technology Application », IITA 2009. Third. In :International Symposium on. Vol. 1. IEEE,(2009)

[**Chikouche** ; **2012**] Chikouche Soumia .« Système de détection d'intrusion basé sur la classification comportementale des processus », mémoire fin d'etude , Mesila,Algérie(2012)

[**Debar et al** ; **2000**] Magnus Almgren, Herve Debar, et Marc Dacier.« lightweight tool for detecting web server attacks ». In : Proceedings of the Network and Distributed System Security Symposium (NDSS'2000), pages 157170, San Diego(2000)

[**Duro et Hoarau** ; **2004**]. Duro, O., Hoarau, W.« Langage de Communication pour Agents de la FIPA », Master Recherche Informatique, Université

Paris Sud XI (2004)

[**Djeriri** ; **2017**] Youcef Djeriri.« Les Réseaux de Neurones Articiels », mémoire fin d'etude, Sidi Bel Abbès,Algerie(2017)

[**Drogoul** ; **2005**] A. Drogoul.« Systemes multi-agents », Projet MIRIAD, Rapport technique, OASIS/LIP6, UniversiteParis6 France

[**Forgey** ; **1965**]Forgey, E.« Cluster analysis of multivariate data : Efficiency vs. interpretability of classification ». In : Biometrics, Article 21(3), 768-769.(1965)

[**Gasser** ; **et al.** ; **1987**] Gasser, L., Braganza, C., et Herman, N. « A flexible testbed for distributed AI research », Rapport :Distributed AI.(1987)

[**Gautam** ; **et al.** ; **2016**] Gautam, et Om, H.« Computational neural network regression model for Host based Intrusion Detection System ».In :Perspectives in Science, Article 8, 93-95(2016)
[**Herault** ; **2012**] Bruno Hérault.« méthodes de classfication » LS6 – BBB – Biomathématiques–Semestre 2(2012)

[**Labed** ; **2006**] LABED Ines.« Proposition d'un systéme immunitaire artificiel pour la détection d'intrusions »,Thèse de doctorat,Université de Telemcen Algerie(2006)

[**Ludovic** ; **2015**] Ludovic Mé.« Utilisation de cartes de Kohonen pour détecter des intrusions dans un système informatique », Article,France (2015)

[**MacQueen** ; **1967**] MacQueen, J.« Some methods for classification and analysis of multivariate observations ». In :Proceedings of the fifth Berkeley symposium on mathematical statistics and probability ,Article 14, (pp. 281-297)(1967)

[**Mathieu-Dupas** ; **2010**] Mathieu-Dupas.« Algorithme des k plus proches voisins pondérés et application en diagnostic » In :42èmes Journées de Statistique.Rapport(2010)

[**Meziane** ; **2007**]Imane Méziane Tani.« Stratégie de rendez-vous dans les systèmes multi-agents », mémoire fin d'etude ,Université Abou bekr Belkaid-

Faculté de Technologie Département d'Informatique , Tlemcen, Algerie(2007)

[**Mokhtari et Moulkhaloua** ; **2018**] Mokhtari .S.D, Moulkhaloua.A, encadré par Belgran.F.Z.« Système DE Détection D'intrusions Informatiques par Système Multi-Agents », mémoire de master, Centre Universitaire Belhadj Bouchaib d'Aïn-Témouchent ,Algerie(2018)

[**Mokeddem et Oumiloud** ; **2014**] Mokeddem.A et Oumiloud.H.« Classification non supervisée : Application de k-means , mémoire de fin étude, Université Abou Bakr Belkaid– Tlemcen Faculté de Technologie Département d'Informatique, Algerie

[**Mounzer** ; **2007**] Mounzer Boubou.« Contribution aux méthodes de classification non supervisée via des approches prétopologiques et d'agrégation d'opinions », Thèse de doctorat,Université Claude Bernard - Lyon Françe(2007)
.
[**Mharouach** ; **2005**] Emira MHAROUECH.« Etude et développement d'un outil d'analyse de sécurité des logs », Mémoire de fin étude ,Tunisie(2005)

[**Panda et Patra** ; **2007**] M. Panda et M.R. Patra.« Network intrusion detection using naive bayes». In : International Journal of Computer Science and Network Security . Article 7, 258-263(2007)

[**Panda et al.** ; **2012**] Panda, M., Abraham, A.et Patra, M.« A hybrid intelligent approach for network intrusion detection ». In :Procedia Engineering, Article 30, 1-9.

[**Pataky et Damas Gruska 2014**]Pataky et Damas Gruska.« Multi agent heterogeneous intrusion detection system », Article 11, 122-127 ,(2014)

[**Peng** ; **Yichun** ; **et al 2014**].« Application Study on Intrusion Detection System Using IRBF » Journal of Software 9.1 : 177183(2014)

[**Pesty et al.** ; **2001**] Pesty, S., Webber, C., et Balacheff, N.« Baghera : une architecture multi-agents pour l'apprentissage humain » Agents Logiciels, Cooperation, Apprentissage et Activité Humaine ALCAA, 204-214 (2001)

[**Philip** ; **1998**] A. Phillip, Porras et Alfonso Valdes.« Live trac analysis of tcp/ip gateways. Proc ».In :ISOC Symposium on Network and Distributed System Security (NDSS98). San Diego, CA

[**Projet** ; **2007**] « Projet informatique ,Algorithme des k-plus-proches-voisin Projet informatique »

[**Robin et Paxson** ; **2010**] Robin Sommer et Vern Paxson.« Outside the closed world Article :" On using machine learning for network intrusion detection ». In :Security and Privacy (SP), IEEE Symposium on, pages 305–316,(2010)

[**Shenfield et al.** ; **2018**] Day et Ayesh, A. « Intelligent intrusion detection systems using artificial neural networks ». ICT Express, 4(2), 95-99.(2018)

[**Sharafaldinet et al.** ; **2018**]Sharafaldin, , Arash Habibi Lashkari et Ali A. Ghorbani « Vers la génération d'un nouveau jeu de données de détection d'intrusion et la caractérisation du trafic d'intrusion » Article,(2018)

[**Sheth et al.** ; **2014**] Sheth, Shah, et Yagnik.« A survey on RBF neural network for intrusion detection system ».In : Journal of Engineering Research and Applications, Article 4(12), 17-22.(2014)

[**Struyf**; **Rousseeuw** ; **2004**] .Anja Struyf et J. Rousseeuw .« Clustering in an Object Oriented Environment ». In : U.I.A. Universiteitsplein 1,Artcle B-2610 Antwerp, Belgium

[**Tian et al.** ; **2009**] Tian Jingwen, Meijuan Gao, et Fan Zhang .« Network intrusion detection method based on radial basic function neural network ». In :E-Business and Information System Security, . EBISS'09. International Conference on. IEEE,(2009)

[**Totel** ; **2012**] Eric Totel .« Techniques dedétection d'erreur appliquées à la détection d'intrusion » Article-Hal(2012)

[**TOUATI** ; **2016**] TOUATI.« Détection d'intrusions dans les réseaux LAN : Installation et configuration de l'IDS-SNORT», mémoire de Master professionnel en Informatique , Bejaia ,Algerie(2016)

[**Tran anTay** ; **Hugo** ; **2005**] M. Tran VanTay et M. DOMINGUEZ Hugo.« LE SYSTÈME DE DÉTECTION DES INTRUSIONS ET LE SYSTÈME D'EMPÊCHEMENT DES INTRUSIONS » , rapport de fin étude, Québec Canada(2005)

[**Ugur Halici**] Ugur Halici.« Artificial Neural Network. Chapter 9.Radial basis function Network » .EE543 lecture notes. Metu EEE. Ankara 139

[**Wespi** ; **1999**] Wespi H.Debar.« a revised taxonomy for intrusion detection systems » Article 2 (1999).

[**Wenke** ; **Stolfo** ; **2000**] Wenke Lee et Salvatore J.Stolfo. « A Framework for constructing features and models for intrusion detection systems », ACM transactions on Information and system security (TISSEC), vol.3, Issue 4 (2000)

Webography

[**academia** ; **2014**] https ://www.academia.edu/15721018/Radial Basis Function Networks RBFN ,**visité le** : 04/2019

[**Cloudflare** ; **2015**] https ://www.cloudflare.com/learning/ddos/ddos-attack-tools/slowloris ,**visité le** : 03/2019

[**Cooper** ; **2018**] http ://www.comparitech.com ,**visité le** : 03/2019

[**Darkmoreops** ; **2014**] https ://www.darkmoreops.com/2014/11/22/dos-website-with-goldeneye ,**visité le** : 03/2019

[**Infosecinstitute** ; **2017**] https ://resources.infosecinstitute.com/popular-tools-for-brute-force-attacks/gref ,**visité le** : 03/2019

[**Iscxdownloads** ; **2017**] https ://iscxdownloads.cs.unb.ca/iscxdownloads/CIC-IDS-2017/CI-IDS-2017,**visité le** : 03/2019

[**knowledgebase** ; **2013**] https ://knowledgebase.paloaltonetworks.com/KCSArticleDetail id=kA10g000000,**visité le** : 03/2019

[**knowledgebase** ; **2013**]https ://knowledgebase.paloaltonetworks.com/KCSArticleDetail id=kA10g000000C,**visité le** : 03/2019

[**Opendata** ; **2018**] https ://registry.opendata.aws/cse-cic-ids2018 , **visité le** : 03/2019

[**Securite Malwares**]http ://imss www.upmf grenoble.fr/prevert/SpecialiteIHS/Documents/ Securite Malwares.pdf ,**visité le** : 03/2019

[**Sheynikhvich** ; **2017**] http ://www.aging vision action.fr/avateaching/Slides/projet9.pdf ,**visité le** : 04/2019

[**techopedia** ; **2016**] https ://www.techopedia.com/definition/29948/botnet-attack ,**visité le** : 03/2019

[**Rfia** ; **2012**] https ://www.rfia2012.files.wordpress.com , **visité le** : 03/2019

[**theses.univ** ; **2008**] http ://theses.univlyon2.fr/documents/getpart.phpid=lyon2.2008.nfaouiehpart=152187, **visité le** : 03/2019

[**Wikistat**] https ://www.math.univ-toulouse.fr/besse/Wikistat/pdf/st-m-explo-classif.pdf ,**visité le** : 03/2019

[**wikimemoires 2012**] http ://wikimemoires.net/2012/08/les-systemes-de-detections-dintrusions-securite-des-reseaux, **visité le** :02/2019

[**Whatismyip** ; **2013**]https ://www.whatismyip.com/port-scanner , **visité le** : 04/2019