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 :
Mr. MOKHTARI Sidi Mohammed
Mr. MOULKHALOUA Abdelmadjid

Système DE Détection D'intrusions Informatiques par Système Multi–Agents.

Encadrant :
Dr. BELGRANA Fatima Zohra
Maitre De Conférence "B" à C.U.B.B.A.T.

Soutenu en 2018

Devant le jury composé de :

Président :	Mme. BENOSMAN Amina (M.C.B)	C.U.B.B.A.T.
Examinateur :	Mr. BENDIABDALLAH Mohammed Hakim (M.A.B)	C.U.B.B.A.T.
Encadrant :	Mme. BELGRANA Fatima Zohra (M.C.B)	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 donné le courage, la volonté, et la patience pour surmonter les épreuves que nous avons rencontré tout au long de nos vie.

nous tenant à exprimer nos sincère gratitude et reconnaissance à notre encadrant : Mme Belgrana Fatima Zohra maitre de conférence a CUBBAT pour avoir accepté de diriger ce travail, pour son aide, sa disponibilité et ses conseils.

Nous remercions Mlle BENOSMAN Amina maitre assistante à CUBBAT d'avoir accepté d'examiner ce travail. Nous remercions Mr Bendiabdellah Mohamed Hakim maitre assistant à CUBBAT d'avoir accepté d'examiner ce travail. Enfin nous remercions toute personnes ayant contribuées de prés ou loin à la réalisation de ce travail.

MOKHTARI & MOULKHALOUA

Dédicaces

Dédicaces

Je dédie ce modeste travail à :
Mes chers parents qui m'avoir 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 soeur et toute la famille.
A mes chers amis.
Tous ceux qui m'ont aidé de prés ou de loin surtout dans les moments les plus difficiles.
A tous, je dédie ce modeste travail...

Mr. MOULKHALOUA Abdelmadjid

Par courtoisie et par respect il est de mon devoir de dédier ce projet.
A mon cher père qui m'a enfournée de mon amour et affection. qu'il trouve dans ce modeste travail, le fruit de ses efforts et ses sacrifices.
A ma très chère mère pour sa patience, son amour et ses conseils.qu'elle trouve ici l'expression de ma gratitude.
Mon cher frère Mokhtar, ma soeur Hanaa ainsi toute la famille et amis avec mon amour et mes vœux de bonheur.

Mr. MOKHTARI Sidi Mohammed

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Introduction générale

L'air de la technologie est le titre le plus approprie au siècle au quel nous vivons. L'informatique accouplé avec le réseau est la tendance qui a envahi tous les domaines sans exception sois la médecine, l'administration, finance et même le domaine militaire qui est un peu délicat.

Cette tendance a permis d'avoir un monde interconnecte et a facilité la communication , le stockage et l'échange de données et d'informations.

En prenant compte des nombres phénoménales d'ordinateurs et d'appareils connectés sa vient directement à la tête l'impossibilité de tous sécurisées. C'est pour sa qu'apparais des vulnérabilités exploité négativement par des personnes mal intentionnés.

Cette exploitation est notamment dangereuse pour les différentes pertes causées par des experts ou des amateurs avec quelques billets d'argents. Les pertes peuvent aller jusqu'à paralyser un pays.

En Outre pour venir a bout de ce phénomène plusieurs chercheurs et boites de développements ont mis en œuvre multiples techniques et mécanismes pour parvenir de détecter et même contrer ces attaques .Parmi les techniques on distingue les pare-feu permettant de filtré le trafic ainsi l'utilisation de la cryptographie afin d'assurer l'authentification et les proxys ... etc. Hélas ces derniers ont des limites et des incapacités et même certaines attaques sont très développées et utiliser ces vulnerabilités non seulement pour nuire au système et même au travail des mécanismes de sécurité.

L'objectif de ce travail est de présenter, d'une part, le lien qui peut exister entre les systèmes de détection d'intrusions et la sécurité des réseaux et de proposer d'autre part, de nouvelles techniques basées sur les systèmes Multi-agents.

Après l'introduction générale, dans le premier chapitre nous allons voir quelques notions sur la sécurité informatique âpres on entamera la partie des systèmes de détection d'intrusion (SDI) avec leurs différentes classifications .Ainsi l'architecture de base et a la fin l'efficacité des SDI.

Le deuxième chapitre sera consacré aux systèmes multi-agents (SMA) avec quelques notions introductives aux domaines, les différents types d'agents ainsi que leurs travaux .A la fin nous terminons avec les domaines d'applications.

Le troisième chapitre montre l'analogie entre l'objectif des systèmes de détection d'intrusions et les SMA , ainsi les différents travaux réalisés jusqu'à maintenant avec les algorithmes utilisés .A la fin de ce chapitre nous introduisons l'approche utilisée .
Dans le quatrième et dernier chapitre nous parlerons sur l'approche étudié avec les différentes méthodes et algorithmes adoptées (k-ppv , MLP) tout en exposant les résultats obtenus .

Chapitre 1

Sécurité informatique et Système de détection d'intrusions

1.1 Introduction

De nos jours l'informatique et les réseaux ont envahi tous les domaines de la vie quotidienne, voire même qu'ils sont devenu indispensables. Ce développement est géré par des personnes spécialistes du domaine nommé informaticien.

Entre ces derniers existe certains mal intentionnés qui travaillent jour et nuit afin de nuire au bon fonctionnement de ces systèmes et d'obtenir des informations personnelles ou concernant la vie privée. Les cibles préférées sont les infrastructures gouvernementales et nationales.

Ce milieu technologique a des vulnérabilités à des attaques et des données malicieuses courantes. Ces derniers ne cessent de se développer et d'évaluer du jour à l'autre ce qui rend la protection plus dur à y parvenir.

Certains développeurs on prévoyait et mise en œuvre des astuces et solutions afin de sécuriser les différents systèmes. Car l'évaluation des systèmes est déterminée à partir de sa fiabilité.

Alors la protection est un point majeur pour les administrateurs ce qui a mené à utiliser et de développer certains outils pour assurer cette dernière . Parmi les outils les plus fréquents on distingue :

— Les mécanismes de chiffrement pour assurer l'intégrité.
— Les pare-feu utilisé pour filtrer le trafic réseau.
— L'antivirus pour se protéger contre malveillant et les données malicieuses.
— Les détections d'intrusion pour détecter certains comportements illégaux.

Dans ce chapitre nous nous sommes intéressés à la sécurité informatique avec ses différents types ainsi qu'aux SDIs avec leurs diverses méthodes qui permettent à mettre fin aux attaques.

1.2 Sécurité informatique

Le monde moderne inter-connecté et développé grâce à l'être humain contient des faiblesses exploitées par d'autres personnes dont le but est d'y accéder à l'information pour tentative de les détruire, sensible circulant dans le réseau, les exposés, ou les voler.

En exploitant ces faiblesses des attaques potentiellement dangereuses ont vu le jour, tous en vison des infrastructures délicates telles que les secteurs militaires, économiques. Les cybers attaques ont pris un saut depuis les loisirs vers des projets militaires il s'agit de la cyber guerre.

1.2.1 Anatomie d'une attaque

La plupart des attaques suivent le même politique de fonctionnement en passant par 4 étapes [Étienne Duris, 2004] :

— Identification de cible : n'importe quelle attaque doit passer par cette étape, elle consiste à récolter des informations sur la cible.
— Le scanning : le but est d'assemblée les renseignements sur la cible, elle permet même de voire l'IP de la cible. Mais certaines techniques arrivent même à endommager un système ou ralentir le réseau.

— L'exploitation : à partir des informations récolter cette dernière a permis de les exploiter dans le but de trouver des failles présentes soit au niveau protocolaire système ou sur le réseau.
— La progression : C'est l'étape finale qui permet à l'attaquant d'achever le travail établi dans les étapes précédentes, c'est la réalisation de l'attaque.

1.2.2 Différents types d'attaque

Les chercheurs du domaine ont classé les différentes attaques connues dans des classes, déterminées par le genre de faille exporté par cette attaque. Il existe cinq formes d'attaque décrites comme suit : [Cole et al, 2005]

— **L'attaque passive :** ce genre regroupe tous les actions qui permettent l'analyse et le décryptage du trafic, la surveillance des communications et la divulgation des informations. Ces attaques peuvent provoquer la confession des informations ou des données sans la permission et la conscience de la cible, et même permis d'obtenir des numéros de carte de crédit ou d'intercepter des mots de passe.

— **L'attaque active :** ce type englobe toute démarche de détourner ou arrêter les fonctions de protection, en introduisant un code malicieux afin de changer où dérober des informations. En plus des conséquences des attaques passives celle-ci peut entraîner un déni de service ou la modification de données.

— **L'attaque de proximité ou externe :** cette attaque représente l'utilisation du réseau ou dû système sans y avoir des droits d'accès grâce à une entrée clandestine afin d'obtenir des informations, les modifier ou même décliner l'accès à d'autre utilisateurs.

— **L'attaque interne :** elle concerne toute tentative d'espionnage ou d'utilisation frauduleuse d'informations. Elle peut être involontaire par le manque de connaissances ou du à la mauvaise manipulation.

— **L'attaque de distribution :** cette attaque consiste à introduire un code malveillant dans un produit comme une porte dérobée pour obtenir un accès frauduleux à des informations ou un système.

1.2.3 Quelques exemples d'attaque

1. **Déni de service**

   Une attaque par déni de service et un mode pour rendre un site web ou une application dis-fonctionnelle, les cibles de ce dernier ne sont pas les victimes attaquées (les portes, serveur) mais ce sont les clients qui ne parviennent pas à y accéder. Cette attaque peut arriver à n'importe quel moment il suffit juste d'être connecté au réseau. Pour parvenir à cette attaque il faut submerger l'infrastructure cible de faux trafic dans le but de débloquer le ralenti le traitement [Maxime Beaudoin, 2001].

2. **Empoisonnement du cache ARP**

   La mystification ARP (spoofing en anglais), utilisable plus sur les réseaux LAN et particulièrement le réseau du type Ethernet utilisant le protocole TCP/IP. Cette attaque est basée sur l'accouplement d'une IP avec une adresse MAC non correspondante, pour que la victime soit redirigée vers une autre machine.

3. **Injection SQL**

   Les injections SQL sont des vulnérabilités permettant d'exécuter des commandes non prévues. Cette attaque exploite une faille de sécurité d'une application web tous en injectant une requête SQL malicieuse pouvant compromettre sa sécurité.

4. **Attaque de l'homme au milieu**

   Plus connue sous le nom de « Man in the middle », cette attaque permet d'être une tierce partie entre deux stations communiquant dans le but de détourner le trafic réseau. Après le détournement, les données peuvent être modifiées, endommagées ou justement lises. Cette attaque même transparente pour les clients. [Jonathan Krier, 2006]

1.3 Systèmes de détection d'intrusion (SDI)

1.3.1 Définition

Un SDI est un ensemble de composants logiciels et matériel dont la fonction est d'analyser et de détecter toute tentative d'effraction qui soit volontaire ou non. [Jonathan Krier, 2006]

La détection des intrusions est une tâche très complexe, cette complexité est due à la variété des environnements et les différents paramètres entrant dans le processus de détection. [Abdelhalim Zaidi, 2001]

Vu la diversité, il existe plusieurs travaux de recherche qui ont abouti à de multiples architectures et approches, ces architectures sont détaillées un peu plus loin.

1.3.2 L'audit de sécurité

Les audits sont produits par le système d'exploitation, et permettent d'enregistrer tout une partie des actions effectuées. Dans les réseaux il est difficile voire impossible d'avoir une base d'audit permettant d'estampiller et d'enregistrer certains évènements. Les intrusions sont détectées à partir des données analysées.

D'après [Mé et Alanour, 1996], il s'agit de collecter des informations concernant les différentes composantes (processus, mémoire, fichiers, entrées/sortie, etc.. . .) du système afin de comprendre le « qui a fait, à quoi, quand et comment ? » afin d'empêcher les intrusions possibles. [Ali K, 2001]

1.3.3 Classification des SDI

Le domaine des SDI est très nouveau, tous les spécialistes du domaine ont entamé des recherches dans le but de développer plusieurs types de SDI. Selon Hervé Debar et son équipe [Hervé Debar et al, 2000] cinq critères de classification ont été introduits et résumé qui sont présentés dans le schéma suivant :

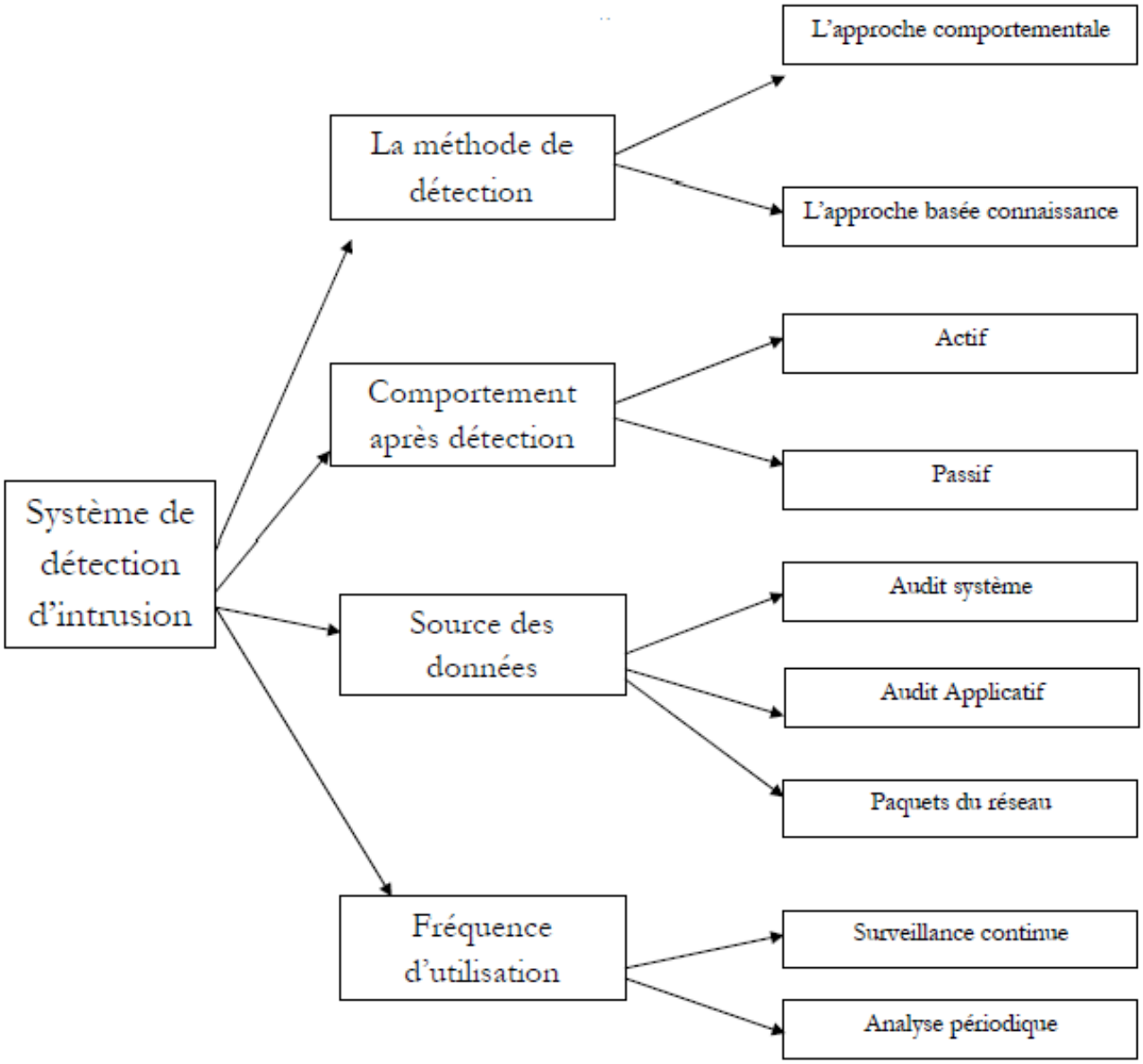

FIGURE 1.1 – Taxonomie D'un SDI. [Debar et al , 2000]

1. **Méthodes de détection**

— **Approche par scénario**

Elle se base sur la connaissance accumulée sur des attaques spécifiques et la vulnérabilité du système. Le système de détection d'intrusion contient des informations sur les vulnérabilités et cherche toute tentative de les exploiter. Si le SDI détecte une tentative, une alarme est déclenchée .En d'autres termes, toute action qui n'est pas explicitement reconnue comme une attaque est considérée comme acceptable. Par conséquent, la précision des SDI basée sur l'approche par scénario est bonne .Cependant, cette précision dépend toujours de la mise à jour des connaissances sur les attaques qui doivent être régulière [Debar et al 2000].

L'inconvénient majeur de cette approche est qu'il faut toujours garder la base de connaissances à jour pour pouvoir reconnaitre des attaques dues à des vulnérabilités et que même si la base est mise à jour constamment il y a de nouvelles attaques que le SDI ne peut pas détecter parce qu'elles ont des signatures spécifiques non reconnues au niveau des bases.

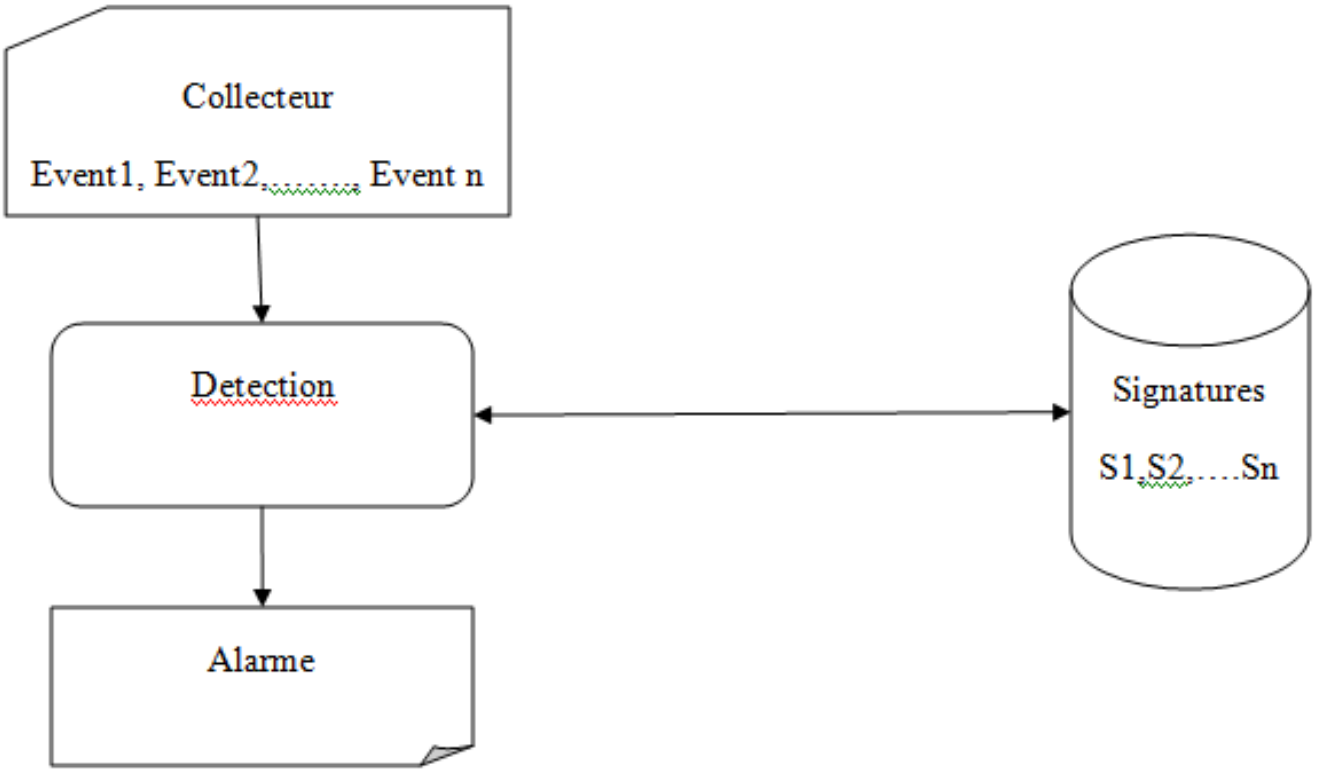

FIGURE 1.2 – Modèle de détection pour l'approche par scénarios

— **Approche comportementale**

Ayant été proposée par Andersson en 1980, puis révisée et développée par Denning en 1987 l'approche comportementale consiste à détecter des anomalies. Cette approche définit un profil spécifique à un utilisateur et considère la déviation significative de l'activité courante de ce dernier par rapport au profil de comportements normaux comme une intrusion.

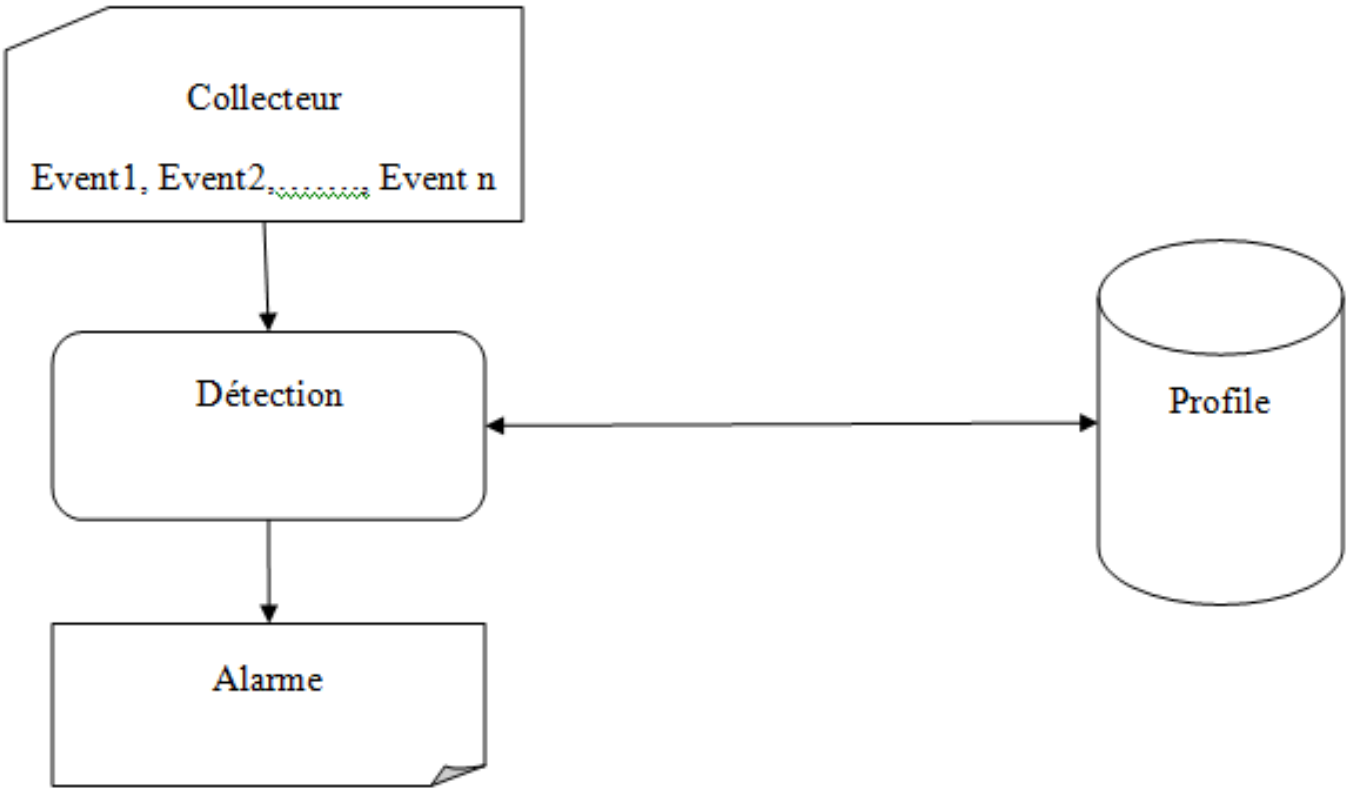

FIGURE 1.3 – Modèle de détection pour l'approche comportementale

2. **Comportement après détection**

Ce sont les actions réalisées après la détection, Le SDI offre deux types de réponses, la première et une réponse passive et la deuxième est active.

Dans le cas de réponse passive, le SDI renvoie un message d'alerte indiquant l'intrusion mais c'est l'utilisateur qui se charge des actions à déployer contre ce type d'attaque.

Mais dans le deuxième cas les actions sont automatisées et en temps réel, c'est-à-dire que le SDI prend des contres mesures directement sans attendre la permission ou la réaction de l'utilisateur.

En comparant entre les deux réponses on distingue que le deuxième type de réponse est plus efficace que la première qui est passive.

3. **Les sources de données**

Les données filtrées et analysées par les SDIs proviennent de différents milieux, ces dernières ont un rôle majeur pour spécifier le SDI le plus approprié. Or il existe plusieurs sources de données.

— **Audit Système (hôte ou HIDS)**

Les audits systèmes sont produits par le système d'exploitation d'un hôte. Ces données permettent à un SDI de contrôler les activités d'un utilisateur sur un hôte. Elles peuvent être également de plusieurs types. L'avantage de ces données systèmes réside dans leur fiabilité et leur granularité fine, qui permettent un diagnostic précis des actions effectuées sur un hôte par un attaquant [Krier, 2006].

Cependant, le volume d'événements généré par les audits systèmes est très volumineux ce qui implique un impact très important sur les performances de la machine surveillée. Les SDIs qui se basent sur cette catégorie des sources de données sont appelés : SDI basé hôte «Host Based Intrusion Detection System». [SLIMANI Ahmed, 2011]

— **Audit Applicatif**

Dans ce cas le SDI traite les activités liées à une application, le but alors est de détecter toute activité non conforme à l'utilisation normale de l'application cible. Le SDI peut être installé dans l'hôte ou est placé l'application ou bien dans un serveur en relation directe avec l'application [Zaidi, 2013].

— **Audit Réseau (SDIR)**

Il s'agit dans cette catégorie d'analyser de manière passive les flux en transit sur le réseau et de détecter les intrusions en temps réel. Un NIDS (Network IDS) écoute donc tout le trafic réseau, puis l'analyse et génère des alertes si des paquets semblent dangereux.

Les NIDS étant des SDIs plus intéressants et les plus utiles du fait de l'omniprésence des réseaux dans notre vie quotidienne, ce document se concentrera essentiellement sur ce type d'IDS [Krier, 2006].

— **Audit hybride**

C'est une combinaison qui permet de grouper dans un seul SDI la surveillance d'un réseau et des hôtes. Selon le besoin, les sondes de détection vont jouer le rôle d'un NIDS ou bien d'un HIDS, et elles seront placées dans des points stratégiques du réseau. Ces sondes communiquent les alertes à un nœud central du réseau qui va faire la corrélation des alertes pour confirmer les résultats des sondes. [Zaidi. 2013]

1.3.4 Architecture de base d'un SDI

1. **Architecture CIDF**

Le projet CIDF (Common Intrusion Detection Framework) [P. Porras et al, 1998] a visé le développement des protocoles et des interfaces de programmation d'application, pour permettre le partage de l'information et des ressources entre les projets de recherche. En effet, les modules dans CIDF échangent des données dans un format standard, qui est basé sur un langage de communication spécifique : Common Intrusion Spécification Langage (CISL) [R. Feiertag et al, 1999 ; C. Kahn et al, 1998]. En plus, l'architecture CIDF permet la réutilisation des composants du SDI déjà développés.

L'architecture CIDF, utilise quatre modules : générateur d'événement, analyseur d'évènement, unités de réponse et une base de données d'événements. Les trois premiers modules jouent les mêmes rôles que ceux cités dans la section Tandis que la base de données des événements est utilisée pour le stockage des événements et des données analysées. [Zaidi, 2013]

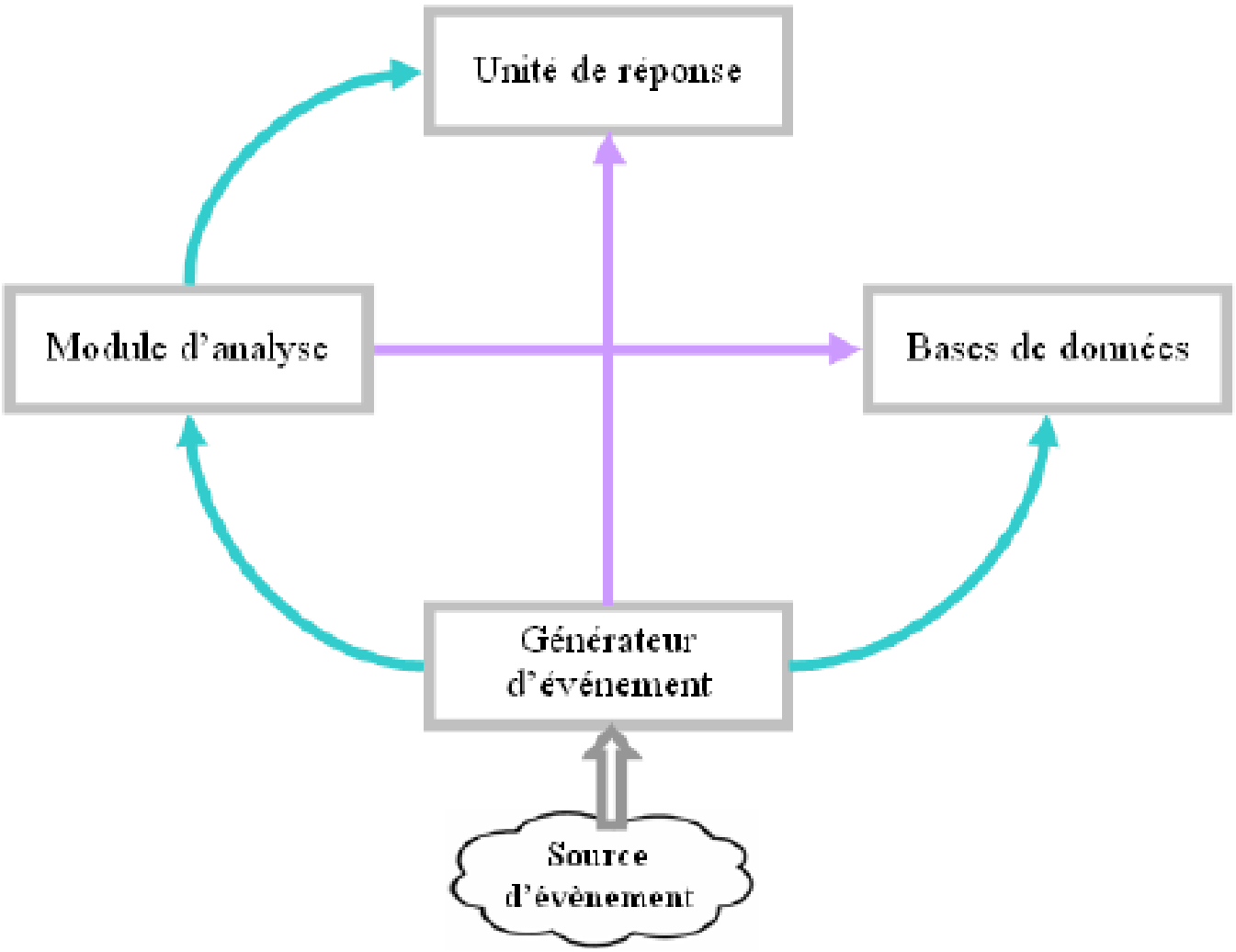

FIGURE 1.4 – L'architecture CIDF. [Zaidi, 2013]

2. **Architecture IDWG**

Ce sujet a été le centre d'intérêt du groupe IDWG (Intrusion Détection Working Group) de l'IETF. IDWG a proposé le modèle général des SDIs qui se compose de senseur (Collecteur), d'analyseur, et de manager (administrateur) [Ahmim , 2014].

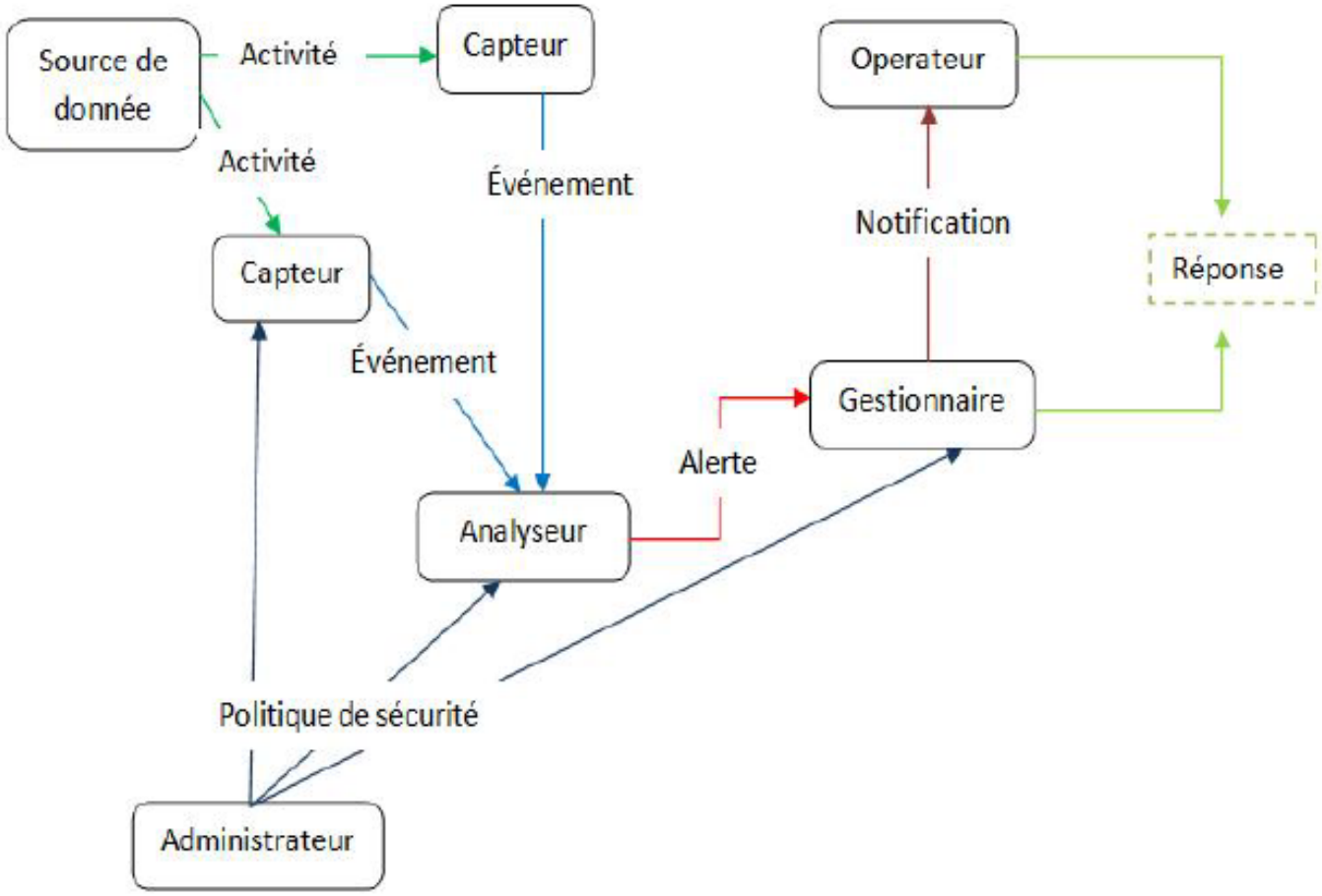

FIGURE 1.5 – L'architecture IDWG. [J. Zimmermann et al , 2002]

1.3.5 L'efficacité des IDSs

Philip défini trois critères pour évaluer l'efficacité des systèmes de détection d'intrusion [Philip, 1998] décrits comme suit :

— **L'exactitude (accuracy) :** on parle de l'exactitude quand le système de détection d'intrusion déclare comme malicieuse une activité légale. Ce critère correspond au faux positif.

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

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

Debar dans [Debar et al, 2000] a rajouté également les deux critères suivants :

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

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

1.4 Conclusion

Dans ce chapitre nous avons expliqué ce que c'est la sécurité informatique et différents types d'attaques visant cette dernière. Ainsi que les différents types d'SDI. Les SDI peuvent être classifié selon différents critères tels que : la source d'information, la méthode d'analyse, la réponse après détection, etc. Les différents types d'SDI sont appropriés pour des situations et des usages précis.

A la fin nous avons présenté l'efficacité des SDI, en plus Une nouvelle branche de l'intelligence artificielle nous a particulièrement attiré par la possibilité d'utiliser les systèmes multi agents pour la détection d'intrusion, c'est pour sa que nous allons présenter cette nouvelle air dans le chapitre suivant.

Chapitre 2

Système multi-agents

2.1 Introduction

Nous allons présenter dans ce chapitre quelques notions de base sur les systèmes multi-agents (SMAs), branche importante, relativement jeune, de l'intelligence artificielle distribuée (IAD) [Weiss 99], où nous détaillerons le concept d'agent, le principe de communications entre plusieurs agents ainsi que l'efficacité de cette approche pour l'intelligence artificielle distribuée.

L'IAD est une branche de l'intelligence artificielle qui s'articule autour de trois axes [Weiss 99] [Jamont 05] :

La résolution distribuée des problèmes qui s'intéresse à la manière de diviser un problème en un ensemble d'entités distribuées et coopérants et à la manière de partager la connaissance du problème afin d'en obtenir la solution.

L'intelligence artificielle parallèle qui développe des langages et des algorithmes parallèles visant ainsi l'amélioration des performances des systèmes d'IA.

Les systèmes multi-agents qui privilégient une approche décentralisée de la modélisation et mettent l'accent sur les aspects collectifs des systèmes.

2.2 Historique

Les systèmes multi-agents sont actuellement très utilisés, particulièrement dans les applications complexes nécessitant l'interaction entre plusieurs entités. [ARSÈNE SABAS 2001]

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]

D'un autre côté, certains problèmes, comme la gestion décentralisée d'un réseau électrique ou la coopération de robots, sont naturellement distribués or une résolution distribuée sollicitée. Enfin, certains systèmes naturels, aussi bien physiques, biologiques que humains, ne sont pas modélisables par des outils standard et leur compréhension passe par la simulation du comportement d'un grand nombre d'entités atomiques. [Ricordel 2001]

L'étude des comportements collectifs et la répartition de l'intelligence sur des agents autonomes, capables de s'organiser et d'interagir pour résoudre des problèmes, sont alors les motivations principales de l'évènement. Dans les années quatre-vingt, des systèmes multi-agents, qui obéissaient à une évolution logique des concepts ayant répondu au besoin sans cesse accru, dans le temps, de l'être humain, de recourir à l'outil informatique.

Dans un premier temps, lorsque l'objectif des personnes qui utilisaient l'outil informatique était de simplifier la programmation, on s'est attelé à élaborer des langages dits évolués passant par l'agrégation d'instructions en procédures puis en fonctions lorsqu'elles retournaient des résultats. Ensuite, un besoin en dynamisme des applications et en généricité s'est fait sentir et on a assisté à l'émergence de la programmation structurée suivie de l'introduction du concept

d'objet. Un manque d'autonomie n'a pas tardé à se faire sentir à son tour, induit par la répartition et la distribution des ressources, et la réponse n'a pas non plus tardé avec l'avènement de la notion d'agent. Très peu de temps après, la coopération a amené les informaticiens à créer des systèmes à agents. La prise en compte d'objectifs globaux et l'utilisation de l'émergence ont alors abouti à l'introduction du paradigme multi-agents qui, intégrera alors le processus de Génie Logiciel. [Sahnoun et al 97]

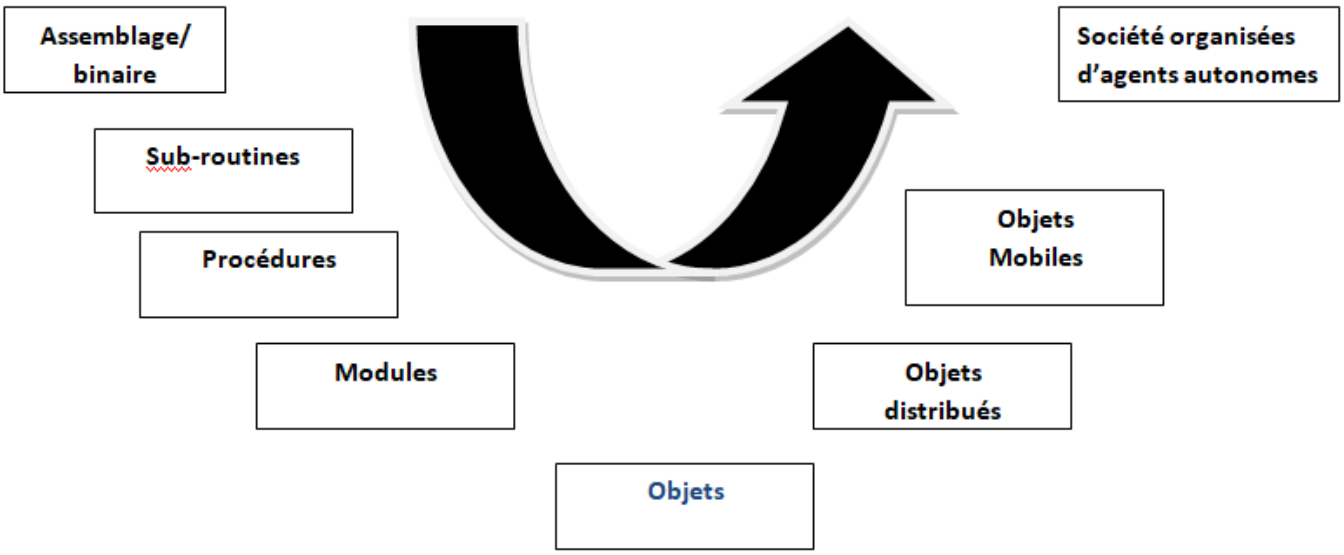

FIGURE 2.1 – Evolution des concepts à travers le temps. [Ferber 2005]

2.3 Définitions d'un SMA

Un système multi-agents est défini comme un système distribué composé d'un ensemble d'agents intelligents qui interagissent le plus souvent selon des modes de coopération, de concurrence ou de coexistence. Il s'agit d'un système composé d'un ensemble organisé d'agents, situés dans un certain environnement et interagissant selon certaines relations.
La figure suivante, extraite de [Jennings et al 2000], donne une vue canonique d'un SMA :

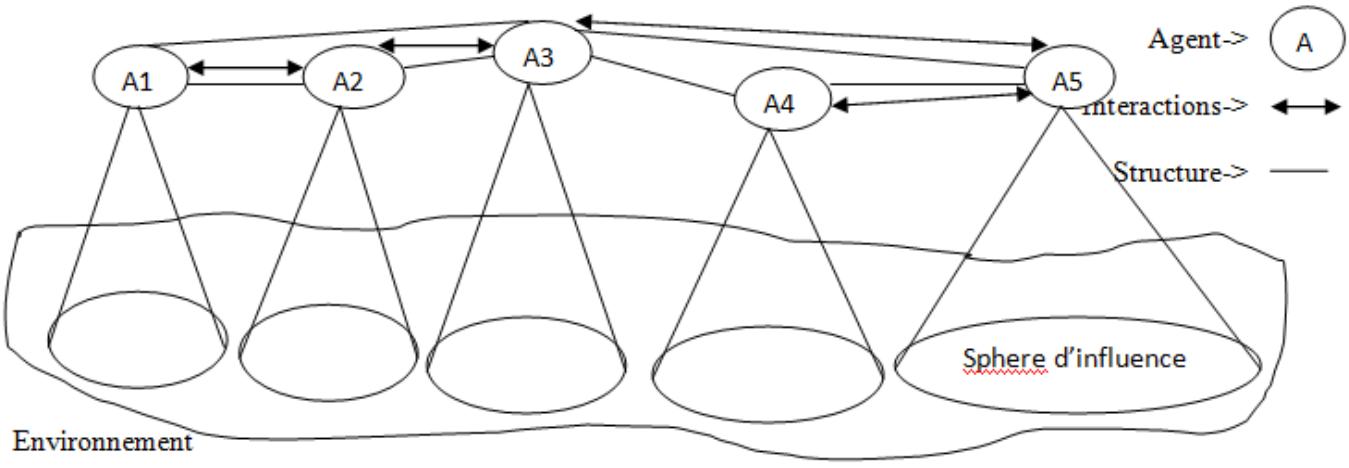

FIGURE 2.2 – Vue canonique d'un SMA. [Jennings et al 2000]

2.4 Definitions d'un Agent

Minsky [Minsky 85] utilise le terme d'agent pour désigner un processus pouvant effectuer des tâches simples ne demandant ni esprit ni réflexion. Un agent est une entité qui agit de manière rationnelle par rapport à ses buts et à ses croyances, dans [Demazeau et al 90] et [Demazeau et al 91].

Un agent est une entité qui fonctionne continuellement et de manière autonome dans un environnement, qui peut être partiellement connu et où d'autres processus se déroulent et d'autres agents existent, dans [Shoham 93].

Un agent est une entité logicielle persistante destinée à réaliser un but spécifique, dans [Smith et al 94].

Coen [Coen 04] voit un agent comme un programme engagé dans un dialogue pour négocier et coordonner un transfert d'information.

Maes [Maes 95], comme Wooldridge et Jennings [Wooldridge et al 95] définissent un agent comme un système informatique agissant, dans un environnement, d'une façon autonome, pour réaliser les tâches et objectifs pour lesquels il a été conçu.

Pour Franklin [Franklin 96], un agent est un objet qui agit, en plus de raisonner.

Dans [Russell 97], un agent est une entité qui perçoit et agit sur son environnement.

Enfin, un agent, dans [Ferber 05]est une entité physique (hard) ou logicielle (soft) située dans un environnement (réel ou virtuel) qui est capable d'agir dans son environnement, de percevoir et partiellement se représenter son environnement (et les autres), de communiquer avec d'autres agents, de mû par ses tendances internes (buts, recherche de satisfaction), de se conserver et se reproduire, et qui présente un comportement autonome qui est la conséquence de ses perceptions, de ses représentations et de ses communications.

2.4.1 Types d'agents et caractéristiques

Les systèmes multi-agents sont en général classés en deux principales familles : les systèmes cognitifs constitués principalement d'agent cognitif et les systèmes réactifs hébergeant des agents réactifs [N.KAbachi 1996], d'autres systèmes dit hybrides contentant les deux catégories d'agents.

1. **Agents cognitif**

   Les agents cognitifs disposent 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 sont généralement «intentionnels » c'est-à-dire qu'ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts. Dans ce cadre, comme le précise Ferber [feber 2005] ,les problèmes de coopération ressemblent étonnamment à ceux de petits groupes d'individus, qui doivent coordonner leur activité, et sont parfois amenés à négocier pour résoudre leurs conflits.

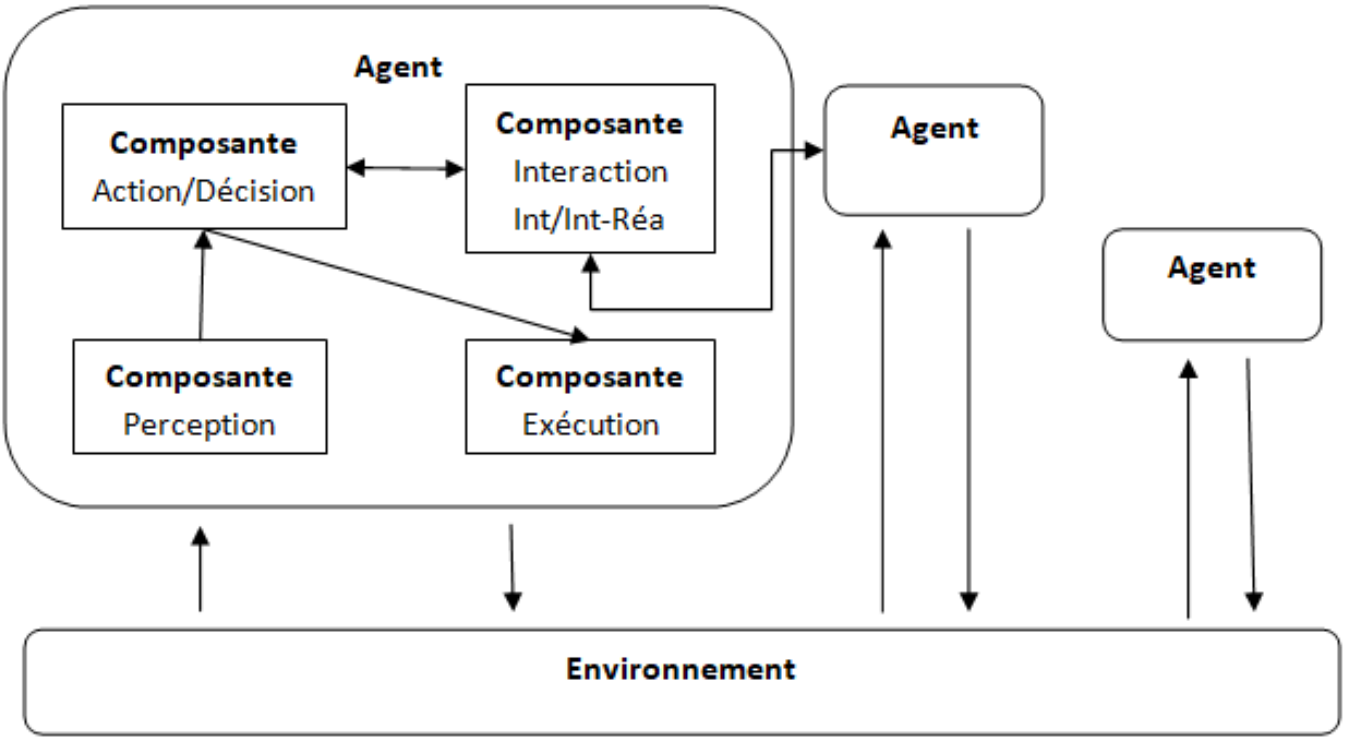

FIGURE 2.3 – Schéma d'agent cognitif. [Imane Méziane Tani , 2007]

2. **Agents réactif**

Les agents réactifs contrairement aux agents cognitifs ne sont pas « intelligents » pris individuellement. Ils ne peuvent que réagir à des stimuli simples provenant de leur environnement, et leur comportement est alors simplement dicté par leurs relations avec leur entourage sans que ces agents ne disposent d'une représentation des autres agents ou de leur environnement. Cependant, du fait, de leur nombre, ces agents réactifs peuvent résoudre des problèmes qualifiés de complexes. Les travaux sur ces agents s'intéressent plus à la modélisation d'une société d'agents qu'à l'agent lui-même. Les analogies que les chercheurs ont établies sont celles de la vie artificielle, de l'éthologie (la fourmilière, la termitière, la ruche d'abeilles), etc.. .

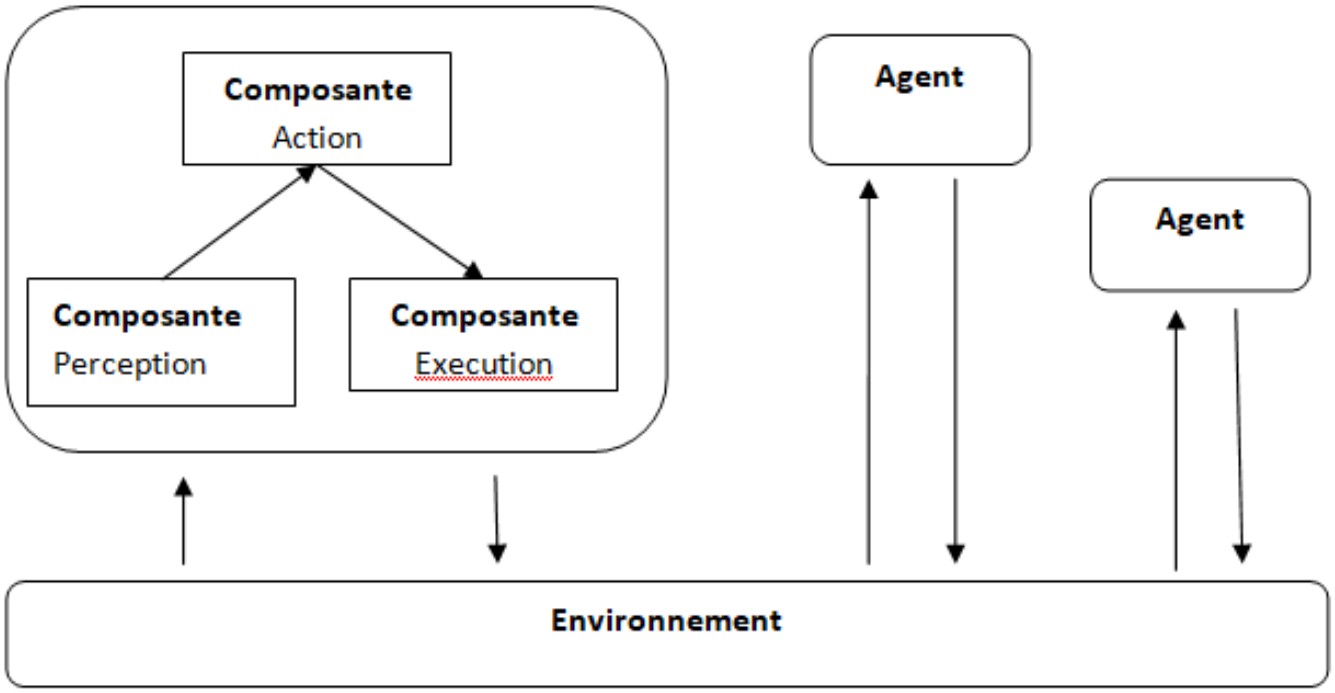

FIGURE 2.4 – Structure d'un agent réactif dans un environnement. [Imane Méziane Tani , 2007]

AGENTS COGNITIFS	AGENTS REACTIFS
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire locale
Agents complexes	Fonctionnement stimulus/action
Nombre d'agents réduit	Nombre d'agents élevé

TABLE 2.1 – Les agents cognitifs vs réactifs. [Reichgelt, 90]

3. Agent hybride

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 [Imane Méziane Tani , 2007].

Dans ce sens, nous pouvons citer les travaux de [Ferguson, 92 ; Muller & Pischel, 94 ; Bussman & Demazeau, 94]. La majorité des modèles d'agents hybrides présentés par ces auteurs, propose de décomposer chaque agent en différents modules réactifs et cognitifs avec un module spécifique qui contrôle l'activation des autres modules. Cette approche est intéressante et semble apporter une solution adéquate pour modéliser les systèmes complexes dont l'environnement est dynamique, mais elle ne résout pas clairement le problème d'interaction entre les différents modules. Le problème, pour.Wooldridge et. Jennings [Wooldridge & Jennings, 95], est alors de définir les mécanismes et les stratégies du module de contrôle interne de l'agent pour soit gérer les interactions entre ses

différents modules, soit imposer un séquencement temporel global interne à l'agent.

2.5 Quelques caractéristiques d'un agent [Ferber 95]

Nous allons voir dans ce qui suit quelques caractéristiques d'agents qui dépendent certainement du type de ce dernier, parmi ces caractéristiques nous citons :

2.5.1 Intentionnalité

L'intentionnalité d'un agent est définie comme la déclaration des buts à atteindre et des moyens mis en œuvre pour y parvenir. Les intentions sont modélisées par un état mental complexe repris des travaux de Cohen et Levesque [Cohen & Levesque, 88 ; 90]. Selon ces auteurs, les intentions doivent satisfaire deux contraintes : des contraintes temporelles, et des contraintes sur des buts :

— Des contraintes temporelles sous-jacentes à la notion d'intention supposent que des intentions émises par un agent prennent en compte des croyances dans le passé et un désir final à réaliser dans le futur.

— Des contraintes sur des buts supposent une relation d'ordre concernant les intentions.

Plus précisément, un désir est réalisé en un ensemble d'intentions complexes définies par des buts à satisfaire. Les buts sont alors décomposés en sous buts jusqu'à l'obtention de sous buts élémentaires.

2.5.2 Autonomie

Définie comme la capacité propre d'un agent de pouvoir atteindre son but, indépendamment des autres agents. C'est, en d'autres termes, l'aptitude d'un agent d'avoir un comportement spontané, de prendre l'initiative pour parvenir à ses objectifs.

2.5.3 Flexibilité

C'est la possibilité pour un agent de modifier son comportement et sa morphologie.

2.5.4 Adaptabilité

Il s'agit de la capacité d'un agent d'améliorer son fonctionnement individuel où collectif pour faire face à un environnement perpétuellement évolutif, ce qui lui permet d'acquérir une « personnalité » qui l'individualise, et le différencie des autres agents.

2.5.5 Rationalité

Un agent rationnel, utilise efficacement les ressources à sa disposition pour sélectionner une action dont l'exécution lui permet d'atteindre un de ses buts.

Du point de vue de l'IA, et comme l'indique Newell [Newell, 82], le principe de rationalité consiste à faire en sorte que « si un agent sait que l'une de ses actions lui permet d'atteindre ses buts, alors il sélectionnera cette action ». Les agents rationnels disposent de critères d'évaluation de leurs actions, et sélectionnent selon ces critères les meilleures actions qui leurs permettent de prendre la meilleure décision. De tels agents sont capables de justifier leurs décisions.

2.5.6 Engagement

Un agent est engagé à accomplir les actions qui satisfont un but à atteindre et se donne les moyens d'y parvenir. L'engagement se présente comme l'un des concepts clés de l'action collective dans le cas de SMA cognitifs .Ce concept a été introduit en IA [Fikes, 82], et analysé en détail par Alan Bond [Bond,89 ; 90], [Rao & Georgeff, 92], [Cohen & Levesque, 87 ; 90].

Pour Jacques Ferber, les engagements caractérisent les dépendances (devoirs, contraintes...) qui lient les agents cognitifs par rapport à eux-mêmes, mais surtout par rapport aux autres lorsqu'ils décident d'accomplir une action, de rendre un service et, d'une manière générale, lorsqu'ils ont l'intention de faire quelque chose. Si les agents ne prenaient pas d'engagements, il leur serait impossible d'avoir une représentation adéquate d'un état futur du monde et donc de planifier leurs propres actions en anticipant sur l'avenir.

2.5.7 Croyances

En général, les croyances décrivent l'état du monde du point de vue d'un agent, et donc la manière dont il se représente son environnement, les autres agents ainsi que lui même. Notons que toute information qui fait l'objet d'une croyance peut être remise en question, ce type de connaissance n'est pas objectif. Chaib-Draa [Chaib-Draa, 89] lui donne le sens d'une information pertinente pour l'agent.

2.5.8 Coopération

La coopération entre les agents consiste à décomposer les tâches en sous-tâches puis à les répartir entre les différents agents, ils existent différents types de coopération qui dépendent du besoin du SMA :

— Coopération par partage de tâches et de résultats.

— Commandes : un agent supérieur décompose le problème et reçoit les solutions partielles.

— Appel d'offre : un agent supérieur décompose le problème et envoie la liste aux agents puis reçoit les propositions et par conséquent choisis l'agent exécutant, puis coopération par commandes.

— Compétition : un agent supérieur décompose le problème et envoie la liste aux agents, il reçoit les solutions puis fait le tri.

2.5.9 Communication

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

— communication indirecte : Partage d'information Via l'environnement.
— communication directe : Envoi de messages.

Les agents doivent avoir des capacités à manipuler un langage commun.

Pour échanger les informations et les connaissances, les agents utilisent des ACL (Agent Communication Language). Il existe deux principales ACL, il s'agit du KQML (Knowledge Query and Manipulation Language) [Finin, Labrou] dont le un message comporte un contenu, le langage (Java, XML, etc.), la performative ainsi que l'ontologie puis le deuxième ACL est le FIPA-ACL :

1. **FIPA-ACL**

   FIP A est un groupe multidisciplinaire poursuivant la standardisation de la technologie agent. Ce groupe a mis sur pied une série de spécifications pour diriger le développement des SMA. Les plus importantes sont la spécification de la gestion d'agent (Agent Managment) [FIP97a] et la spécification de langage de communication agent (ACL) [FIP97b]. Le langage FIP A-ACL suit le style de KQML (utilisant des performatifs issus de la théorie des actes de langage et quelques paramètres complémentaires), mais avec une sémantique mieux spécifiée.

   Le langage prévoit aussi l'utilisation de protocoles d'interaction. TI existe une bibliothèque de protocoles standards [RIBOO].

   L'approche de FIP A pour le développement des SMA est basée sur un paradigme minimal pour la gestion des agents dans un environnement ouvert. Ce paradigme est décrit en utilisant un modèle référentiel (qui spécifie un environnement normatif dans lequel les agents existent et opèrent) et une plate-forme agent (qui spécifie une infrastructure pour le déploiement et l'interaction des agents).

2. **Spécifications FIPA-ACL**

   la communication entre agents est basée sur le passage du message, où les agents communiquent en formulant et en envoyant des messages individuels les uns aux autres. La liste de contrôle d'accès FIPA spécifie un langage de message standard en énonçant l'encodage, la sémantique et la pragmatique des messages. La norme ne définit pas un mécanisme spécifique pour le transport interne des messages. Au lieu de cela, puisque différents agents pourrait fonctionner sur différentes plates-formes et utiliser différentes technologies de mise en réseau, FIPA précise que les messages transportés entre les plates-formes doivent être codés sous une forme textuelle. Il est entendu que l'agent a

un moyen de transmettre cette forme textuelle. La syntaxe de l'ACL est très proche de le langage de communication largement utilisé KQML. Cependant, malgré la similarité syntaxique, il y a différences fondamentales entre KQML et ACL, la plus évidente étant l'existence d'un sémantique formelle pour ACL qui devrait éliminer toute ambiguïté et la confusion de l'utilisation de la langue.[Bellifemine et al , 1999]

2.6 Quelques Modèles SMA

C'est tout récemment que plusieurs groupes de chercheurs et d'industriels indépendants ont commencé à poursuivre la standardisation de la technologie multi-agents. Parmi ces groupes on peut citer :

— Object Manager Group (OMG).
— Foundation for Intelligent Physical Agents (FIPA).
— Knowledgeable Agent-oriented System (KAOS).
— General Magic Group. [ARSÈNE SABAS 2001]

La notion de plate-forme est liée à l'implémentation des systèmes multi-agents : elle constitue un réceptacle au sein du quel les agents peuvent évoluer, c'est une infrastructure logicielle utilisée comme environnement pour le déploiement et l'exécution d'un ensemble d'agents. Les plates-formes SMA permettent de gérer le cycle de vie des agents et dans lequel les agents ont accès à certains services [Arlabosse, 2004].

2.7 plateforme Jade

JADE (Java Agent Développement Framework) est un framework logiciel pour faciliter le développement d'applications d'agents conformes aux spécifications FIPA pour l'interopérabilité systèmes multi-agents intelligents. Le but de JADE est de simplifier le développement tout en assurant conformité standard grâce à un ensemble complet de services système et d'agents. Pour atteindre de tels un objectif, JADE offre la liste de fonctionnalités suivante au programmeur de l'agent :

— Agent Platform conforme à la FIPA, qui comprend l'AMS (Agent Management System), le DF (Directory Facilitator) et ACC (Agent Communication Channel). Tous ces trois les agents sont automatiquement activés au démarrage de la plate-forme de l'agent.

— Plate-forme d'agent distribué. La plateforme de l'agent peut être divisée sur plusieurs hôtes (à condition que il n'y a pas de pare-feu entre eux). Une seule application Java, et donc seulement une Java Machine virtuelle, est exécutée sur chaque hôte. Les agents sont implémentés comme un thread Java et Les événements Java sont utilisés pour une communication efficace et légère entre les agents sur le même hôte. Les tâches parallèles peuvent toujours être exécutées par un agent, et JADE planifie ces tâches dans un manière plus efficace (et encore plus simple pour le programmeur qualifié) que Java Virtual La machine fait pour les threads.

— Un certain nombre de DFs (Directory Facilitator) conformes à la FIPA peuvent être démarrés au moment de l'exécution afin de implémenter des applications multi-domaines, où la notion de domaine est logique comme décrit dans la partie 1 du FIPA97.

— Interface de programmation pour simplifier l'enregistrement de services d'agent avec un ou plusieurs domaines(c'est-à-dire DF).

— Mécanisme de transport et interface pour envoyer / recevoir des messages vers / depuis d'autres agents.

— Protocole IIOP conforme à FIPA97 pour connecter différentes plates-formes d'agents.

— Transport léger des messages ACL à l'intérieur de la même plate-forme d'agent, car les messages sont transféré en tant qu'objets Java plutôt que sous forme de chaînes, afin d'éviter les procédures de démariage. Lorsque l'expéditeur ou le destinataire n'appartient pas à la même plate-forme, le Le message est automatiquement converti vers / depuis le format de chaîne compatible FIPA. De cette façon, celala conversion est cachée aux agents d'exécution de l'agent qui ont seulement besoin de traiter avec la même classe de Objet Java.

— Bibliothèque de protocoles d'interaction FIPA prêts à être utilisés.

— Enregistrement automatique des agents auprès de l'AMS.

— Service de dénomination conforme à la FIPA : les agents de démarrage obtiennent leur GUID (Globally UniqueIdentifiant) de la plate-forme.

— Interface utilisateur graphique pour gérer plusieurs agents et plateformes d'agent à partir du même agent.

L'activité de chaque plateforme peut être surveillée et enregistrée. [bellifemine,1999]

2.8 Conclusion

Dans ce chapitre, nous avons introduit quelques définitions ainsi que les principaux concepts de base sur les systéme multi-agents. Les SMA sont des systèmes performants et flexibles, permettant la modélisation distribuée des traitements et des connaissances. Ils représentent actuellement un champ de recherche très actif. C'est une architecture permettant le contrôle global du système orchestrant l'activité de ses composants et leur fournit un moyen d'échanger et de stocker les résultats de leurs activités. Ces avantages représentent la raison pour laquelle nous avons adopté cette approche.

Chapitre 3

ETAT de l'ART

3.1 Introduction

Dans le domaine de la détection d'intrusion différentes recherches ont été réalisées en faisant appel à différent techniques qui dérivent principalement de L'intelligence artificielle (IA) ainsi que l'intelligence artificielle distribuée .Plusieurs chercheurs se sont mis au travail afin d'atteindre cet objectif et d'améliorer pour détection de toute les activités illégales ou suspectes dans un réseau.

Selon l'approche d'analyse des données adoptée par les SDIs, ces derniers peuvent être classés en deux grandes catégories : la détection d'intrusions par signature et la détection d'intrusions comportementale [MAJORCZYK, 2008].

Dans la première catégorie l'approche est basée sur la comparaison entre le comportement normal de l'utilisateur et les anomalies. Parmi les méthodes les plus utilisés nous citons les méthodes statistiques, les systèmes experts, les réseaux de neurones et l'immunologie [Debar et al 2000].

Contrairement à la détection d'intrusions par anomalie qui apprend le comportement normal, la détection d'intrusions basée sur la connaissance cherche directement les activités intrusives. Plusieurs méthodes ont été adoptées dans cette catégorie notamment.

3.2 L'IA mis à la disposition des SDIs

Le modèle proposé par Denning [Denning , 1987] est la base de développement des SDI .les méthodes et les techniques utilisées dans cette génération dépend de l'approche soit comportementale ou par scenario.

3.2.1 La technique basée sur les statistiques

La méthode la plus utilisée pour construire les SDI basés comportementale est la méthode statistique [Javitz et al., 1993] [Helman et al., 1992] [Helman et Liepins, 1993]. Elle consiste à calculer la nature et la conduite de l'utilisateur par un nombre des prototypes dans le temps. Ces prototypes comprennent l'utilisation du processeur, le temps de connexion et déconnexion et l'occupation de la mémoire. L'échantillonnage peut aller de quelques minutes jusqu'à un mois. Ensuite il sera considéré comme une moyenne de normalité donc si le seuil est différent de cet échantillonnage sa sers une anomalie.

Haystack [Haystack Labs 1997] utilise également l'approche statistique. Il effectue un ensemble de mesures X = (x1 ; x2 ; . . . . . ; xn) puis représente sous forme d'un vecteur de Bernoulli v toute mesure xi qui dépasse un seuil prédéfini < ti, min, ti, max >. Ensuite le vecteur v est multiplié par un vecteur poids p spécifique à chaque intrusion. Le produit obtenu représente le score d'anomalie et il est considéré élevé si la probabilité d'obtenir des scores plus petits est grande.

3.2.2 Techniques basée sur support vecteur machine (SVM)

[E Kabir et al 2018] propose une nouvelle approche basé sur un échantillonnage avec une machine à vecteur de support au moins carré (LS-SVM). La prise de décision est effectuée en deux étapes. Dans la première étape, l'ensemble de données est divisé en sous-groupes arbitraires prédéterminés. L'algorithme proposé sélectionne des échantillons représentatifs de ces sous-groupes de sorte que les échantillons reflètent l'ensemble des données. Un schéma d'allocation optimal est développé sur la base de la variabilité des observations au sein des sous-groupes. Dans la seconde étape, la machine à vecteurs de support des moindres carrés (LS-SVM) est appliquée aux échantillons extraits pour détecter les intrusions. Nous appelons l'algorithme proposé une machine à vecteur de support des moindres carrés basée sur l'allocation optimale (OA-LS-SVM) pour l'IDS. Pour démontrer l'efficacité de la méthode proposée, les expériences sont réalisées sur la base de données KDD 99 qui est considérée comme une référence de facto pour évaluer la performance de l'algorithme de détection des intrusions. Toutes les classes binaires et multi class sont testées et notre approche proposée obtient une performance réaliste en termes de précision et d'efficacité. Enfin, une sortie est également montrée l'utilité de l'algorithme proposé pour les ensembles de données incrémentaux.

[Yassine Maleh et al 2015] utilise détection d'anomalie basée sur un algorithme de machine à vecteurs de support (SVM) et un ensemble de règles de signature pour détecter les codes malveillants comportements et fournir des IDS légers et globaux. Les résultats de la simulation montrent que le modèle proposé peut détecter des anomalies événements efficacement et a un taux de détection élevé avec moins de fausses alarmes.

3.2.3 Techniques basée sur les Systèmes Experts(SEs)

Les premiers prototypes des systèmes de détection d'intrusions utilisent les systèmes experts pour détecter les scénarios d'attaques. La base de règles traduit les connaissances des experts ce qui permet de transformer les faits extraits des sources d'audit et d'en déduire la présence des attaques.

La base des règles peut être conçue de deux façons :

— la première est basée sur le comportement quotidien de l'utilisateur, qui est enregistré sur un intervalle de temps. Après toutes les activités courante sont comparées avec cette base afin de détecter les anomalies .Wisdom et Sense [Vaccaro and liepins ,1989] sont des SDI basés sur ce type de fonctionnement.

— la deuxième est basées sur la comparaison des activités de l'utilisateur par rapport a un ensemble de règles qui décrivent le bon fonctionnement .donc toute activité non conforme est déclaré comme tentative d'intrusion. Cette approche est moins efficace que la première en cas d'une énorme quantité .

Sebring et al [M. Sebring et al 1988] introduise le P-BEST (Production Based Expert System Toolset) qui a été initialement développé pour le système MIDAS puis employé par IDES [T. F.et al ,1988] , EMERALD [P. G. Neumann 1999]. Par ailleurs, Le Charlier définit le langage RUSSEL (Rule based Sequence Evaluation Language) pour décrire les signatures d'attaques dans le système ASAX [N. Habra et al 1992].

Les langages [harba et al 1992] sont un outil naturel pour la modélisation de la connaissance que les experts ont recueilli sur les attaques. Cette méthode permet de surfer a la recherche des anomalies et toute tentative d'intrusion.

Le point fort des systèmes expert est la définition des attaques en dehors des intrusion. Et il reste les plus utilisées dans la détection surtout dans les SDI commerciale.

3.2.4 Réseaux de Neurones (RNs)

Les réseaux de neurones sont utilisés dans la détection d'anomalies afin d'exploiter Leurs capacités d'apprentissage.

[James Canady ,1998]est le premier chercheur à proposer un système de détection d'intrusion basé sur l'analyse des protocoles réseau à base de réseaux de neurones. Il utilise les perceptrons multicouches (MLP) pour l'apprentissage.[Lekagning et al 2016]

L'idée de base est d'utiliser les mécanismes d'apprentissage des Réseaux de neurones pour apprendre les profils de comportements normaux des utilisateurs ou d'un système.[SLIMANI Ahmed, 2011]

Réseaux de neurone sont des méthodes de raisonnement inductif. Cette technologie est basée sur la structure cellulaire du cerveau humain. Comme dans le cerveau humain, les connexions entre les cellules sont renforcées ou affaiblies en fonction de leur capacité de rendement " production" des résultats. Le système utilise un algorithme pour « apprendre » de l'expérience. Après avoir appris suffisamment, il a la capacité pour identifier ou classer [J.Greensmith, J. Feyereisl , 2008]

3.2.5 Les systèmes immunitaires

Système immunitaire artificielle inspiré par les systèmes immunitaires naturels. Leur efficacité a encouragé aux chercheurs d'étudier et d'apprendre les mécanismes immunitaires pour la mise en œuvre de systèmes artificiels qui peuvent efficacement détecter les intrusions [Mr Ameur BENNAOUI, revoir la référence].

Forest [Forest et al., 1997] est la première à introduire l'immunologie dans le domaine informatique. Sa méthode consiste à décrire le comportement normal ou le "soi" via une séquence finie d'appels systèmes. Les séquences appelées N-gram servent de base pour comparer les appels systèmes des processus lors d'une phase de surveillance.

Wespi, Dacier et Debar [Wespi et al 2000] considèrent un cas plus général en analysant les événements d'audit. Ils génèrent des séquences d'événements de tailles variables pour modéliser l'état normal du système. Ensuite un motif est sélectionné s'il existe d motifs qui le suivent directement sinon le score d'anomalie est incrémenté de 1 et une alarme est déclenchée lorsque le score dépasse le seuil toléré.

Les erreurs de configuration dans un service est l'inconvénient majeur de cette méthode donc l'attaquant peut accéder sans autorisation en utilisons des mesures légitimes.

3.2.6 L'analyse des signatures

L'analyse des signatures a le même fonctionnement que les systèmes experts mais les connaissances sont exploiter d'une manière différente.

Le principe de cette méthode est que les séquences d'événements d'audits générer peuvent décrire le scenario d'attaques , alors ils seront considère comme trace d'audit pour le système. Cette technique permet une mise en œuvre très efficace et applicable pour les produits de détections commerciaux [Haystack Labs 1997].

L'inconvénient majeur de cette méthode tous comme les autres basée sur les scenarios est la nécessité des mises à jour régulière et fréquentes. En plus il faudrait représenter toutes les attaques avec multiples et différentes signature possible.

3.2.7 L'analyse de l'état transition

L'analyse de l'état transition est une technique proposée par Porras et Kemmerer, elle a été implémentée dans un système UNIX au premier temps [Porras and Kemmerer 1992] , puis dans l'autre environnement . Cette technique consiste à décrire les attaques avec un ensemble d'objectifs et de transitions , puis les respecter sous forme de diagramme de transition d'états.

Nous constatons que la description des scénarios d'attaques sous forme de machines à états finis est un moyen plus facile que la définition de règles dans un système expert. Cependant, la méthode ne permet d'exprimer que de simples relations qui portent sur des séquences d'événements[Ali kartit , 2011].

3.2.8 Algorithmes génétiques

2.8 Algorithmes génétiques : Les algorithmes génétiques ont été proposes par John Holland en 1975 [Holland 1975] .leurs but est de trouver une solution proche de la solution optimale d'un problème donnée.

Les algorithmes génétiques ont été utilisés dans l'approche par scenario afin de trouver des signatures des attaques prédéfinies dans les traces d'audit de sécurité [Mé1995].cette approche permet la traduction des signatures d'attaques d' audits de sécurité en problème d'optimisation afin de garantir de trouver une performante salutation au nombre d'attaques.

3.2.9 Algorithme K-means

[Wathiq Laftah Al-Yaseen et al 2016] propose 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 classificateurs. Le K-means modifié est utilisé pour construire de nouveaux petits

jeux de données d'entraînement représentant l'intégralité de l'ensemble de données d'entraînement original, réduire significativement le temps de formation des classificateurs, et améliorer les performances du système de détection d'intrusion. Le jeu de données populaire KDD Cup 1999 est utilisé pour évaluer le modèle proposé. Comparé à d'autres méthodes basées sur le même ensemble de données, le modèle proposé montre une grande efficacité dans la détection d'attaque, et sa précision (95,75%) est la meilleure performance jusqu'à présent.

3.2.10 SDI basés sur SMA

Un système de détection d'intrusion adaptatif qui peut détecter des attaques inconnues dans le trafic réseau en temps réel est une préoccupation majeure. Les systèmes conventionnels de détection d'intrusion adaptative sont coûteux en termes de ressources informatiques et de temps, car ces systèmes doivent être recyclés avec des attaques connues et inconnues. [Wathiq Laftah Al-Yaseen et al 2016]

[Adriana Leite, et al 2017] propose une architecture d'agent logiciel combinant raisonnement par cas, comportement réactif et apprentissage appelée HyLAA (A hybrid and learning agent architecture for network intrusion détection). Grâce à son mécanisme d'apprentissage, HyLAA peut s'adapter à son environnement et identifier de nouvelles intrusions non précisées dans la conception du système. Ceci est fait en apprenant de nouvelles règles réactives en observant les bonnes solutions récurrentes à la même perception à partir du système de raisonnement par cas, qui sera stocké dans la base de connaissances de l'agent. L'efficacité de HyLAA pour détecter les intrusions en utilisant un comportement de raisonnement basé sur les cas, l'exactitude du classificateur appris par le composant d'apprentissage et la performance et l'efficacité de HyLAA pour détecter les intrusions utilisant le comportement hybride avec apprentissage et sans apprentissage ont été évaluées, respectivement, en effectuant quatre expériences. Dans la première expérience, HyLAA a montré une bonne efficacité pour détecter les intrusions. Dans la deuxième expérience, les classificateurs appris par le composant d'apprentissage ont présenté une grande précision. Le comportement de l'agent hybride avec apprentissage et sans apprentissage (troisième et quatrième expériences, respectivement) présentait une plus grande efficacité et un équilibre entre performance et efficacité, mais seul le comportement hybride présentait une meilleure efficacité et performance tant que l'agent apprenait.

[Wathiq Laftah Al-Yaseen et al 2016] a proposé une méthode appelée système multi-agent en temps réel pour un système de détection d'intrusion adaptative RTMAS-AIDS, basée sur un système multi-agent, est proposée pour permettre au système de détection d'intrusion de s'adapter à des attaques inconnues. Temps réel. Cette méthode utilise les modèles de classification hybride SVM multi-niveau et ELM pour détecter le comportement normal et les attaques connues. Un modèle SVM adaptatif, dans lequel les processus s'exécutent en parallèle et sont distribués dans MAS, est également utilisé pour détecter et apprendre de nouvelles attaques en temps réel. Les résultats montrent que la méthode proposée réduit de manière significative le coût de formation de la détection d'attaques inconnues par rapport à la méthode conventionnelle. En outre, les résultats d'analyse du jeu de données populaire KDDCup'99 montrent que RTMAS-AIDS peut mieux détecter les attaques Probe, R2L et U2R que le système multi-agent non recyclé utilisant les modèles hybrides SVM et ELM multi-niveaux, ainsi que la SVM hybride multi-niveau et ELM. RTMAS-AIDS a montré une précision de détection significativement améliorée qui a atteint 95,86% et peut détecter et apprendre des attaques inconnues plus rapidement (jusqu'à 61%) que les deux autres méthodes (MAS-MLSE et MLSE).

[NedaAfzali et al 2014] propose une approche basée sur un agent utilisant les paradigmes du système immunitaire artificiel (AIS) en tant que mécanisme efficace pour un SDI distribuée. Les paradigmes AIS sont la sélection négative, la sélection clonale, la théorie du danger et le réseau immunitaire. Ces paradigmes sont très réussis pour l'IDS d'anomalie. Les agents AIS proposés sont capables d'apprentissage, d'auto-adaptation, de mobilité sur plate-forme, d'autonomie et de collaboration. Le système proposé (MAIS-IDS) a été conçu en utilisant ces agents puissants et collaboratifs. Ce système a des agents mobiles et statiques avec des agents de détection en tant qu'acteurs principaux dans MAIS-IDS. Les cycles de vie des agents sont déterminés en utilisant les algorithmes immunitaires proposés dans des phases spécifiques. Les caractéristiques essentielles de MAIS-IDS sont le clonage, la mutation, la migration, la collaboration et le caractère aléatoire. MAIS-IDS a été évalué à l'aide d'un réseau d'hôtes virtualisés , d'un hyper viseur de machine virtuelle à base de noyau (KVM) et d'un orchestre de gestion.

3.2.11 Autres travaux

Dans cette partie nous allons citer des auteurs et leurs travaux qui ont mélangée entre plusieurs algorithmes afin de parvenir à une meilleure détection d'intrusion.

Parmi eux [Tarfa Hamed et al 2018] proposent un NIDS basé sur une méthode de sélection de caractéristiques appelée RGC (Récursive Feature Addition) et technique de bigram. Le système a été conçu, mis en œuvre et testé .Le modèle a été testé sur l'ensemble de données ISCX 2012, qui est l'un des ensembles de données les plus connus et les plus récents pour la détection d'intrusion. De plus, une technique de bigram est proposée pour coder les fonctions de chaîne de charge utile en une représentation utile qui peut être utilisée dans la sélection de caractéristiques. En outre, ils ont proposée une nouvelle métrique d'évaluation appelée (combinée) qui combine la précision, le taux de détection et le taux de fausses alarmes d'une manière qui aide à comparer différents systèmes et à sélectionner les meilleurs d'entre eux. Le système basé sur la sélection des fonctionnalités a montré une amélioration notable de la performance en utilisant différentes métriques.

Kummar [Kummar et spafford 1994] utilise les réseaux de pétri pour représenter les scénarios d'attaques. Les transitions sont étiquetées par des appels systèmes alors que les jetons évoluent chaque fois qu'un événement permet de tirer une transition. Un jeton possède une couleur qui est une évaluation des variables.

Les avantages des réseaux de pétri est que même les signatures complexe peuvent être facilement écrites néanmoins il est couteux en terme calcul et alourdit le processus.

La spécification des exécutions normales des processus constitue une autre technique de la détection d'intrusions comportementale. Ko, Fink et Levittet [C. Ko et al 1994] s'intéressent aux programmes avec privilège "super utilisateur" et développent un langage de spécification qui se base sur la logique des prédicats et les expressions régulières. Cependant la réalisation de ces spécifications est une tâche assez difficile. Les auteurs proposent une méthode utilisant une logique inductive pour synthétiser directement les spécifications à partir des traces valides [C. Ko et al 2000].

Dans ce contexte, Nuansri [N. Nuansri et al 1999] construit un diagramme de transition d'états pour représenter les changements de privilège. Il définit ainsi un ensemble de règles dont la violation révèle la présence d'une attaque.

3.3 Conclusion

Nous avons présenté dans ce chapitres un état de l'art présentant différente techniques qui dérive généralement de 'l'IA et de l'IAD qui sont utilisé en vue d'une détection d'intrusions. Nous avons abordé les deux principales approches citées dans la littérature : l'approche comportementale et l'approche basée sur la connaissance. Chacune de ces méthodes hélasse présentent des avantages et des inconvénients.[ali kartit , 2011]

Chapitre 4

Expérimentation et Résultats

4.1 Introduction

Dans ce chapitre nous allons présenter nos deux approches proposées destinées à la détection d'intrusion dans un réseau informatique .La première approche est réalisée via l'algorithme K-ppv avec un SMA tandis que la deuxième approche nous avons utilisé un RN et un SMA. Nous avons présenté dans les chapitres précédant des définitions et un état sur ces méthodes que nous avons adoptées. Nous allons maintenant dans ce chapitre et dans un premier temps détailler nos deux contributions et afin de prouver l'efficacité de notre travail nous allons dans un deuxième temps exposer nos résultats de détection d'intrusion. Pour cette raison nous avons utilisé une BDD fourni par DARPA en 1999 qui regroupe différents types de paquets simulant intrusion ainsi que d'autres paquets normaux. Une validation de nos approches est notamment réalisée via une comparaison avec un travail de la littérature.

4.2 Approches proposées

Les réseaux de neurones artificiels sont inspirés à partir des neurones naturels, leur efficacité a encouragé plusieurs chercheurs à étudier les mécanismes d'apprentissage pour la mise en œuvre de systèmes artificiels qui peuvent efficacement détecter les intrusions au niveau réseau.

Quant à la méthode des "K-ppv" reste actuellement la méthode la plus utilisée surtout lors de classifications des grands fichiers de données. Cette méthode à l'avantage d'être efficace et très rapide à l'instar d'autres algorithmes de classifications tels que la classification hiérarchique [Mustapha HADD, 1999]. La particularité de la méthode des "k-ppv" est que le nombre de classes doit être spécifié préalablement ce qui représente un autre avantage dû au fait nous connaissions le nombre de classes de notre BDD.

Lors de la réalisation d'un SDI le temps est très importants ors L'utilisation des systèmes multi-agents (SMA) avec chacune de ces deux méthodes permet une meilleure gestion du système ainsi qu'une bonne simulation aboutissant ainsi à des résultats satisfaisant en un temps raisonnable, notant toutefois que l'architecture du SMA que nous avons introduit est la même pour nos deux approches proposées.

Les étapes d'apprentissage ainsi que la validation de notre SDI sont réalisées s via la base de données KDD99 [KDD99] qui regroupe plusieurs types de paquets de connexion tcp/ip simulant plusieurs types d'intrusions ainsi que d'autres paquets normaux.

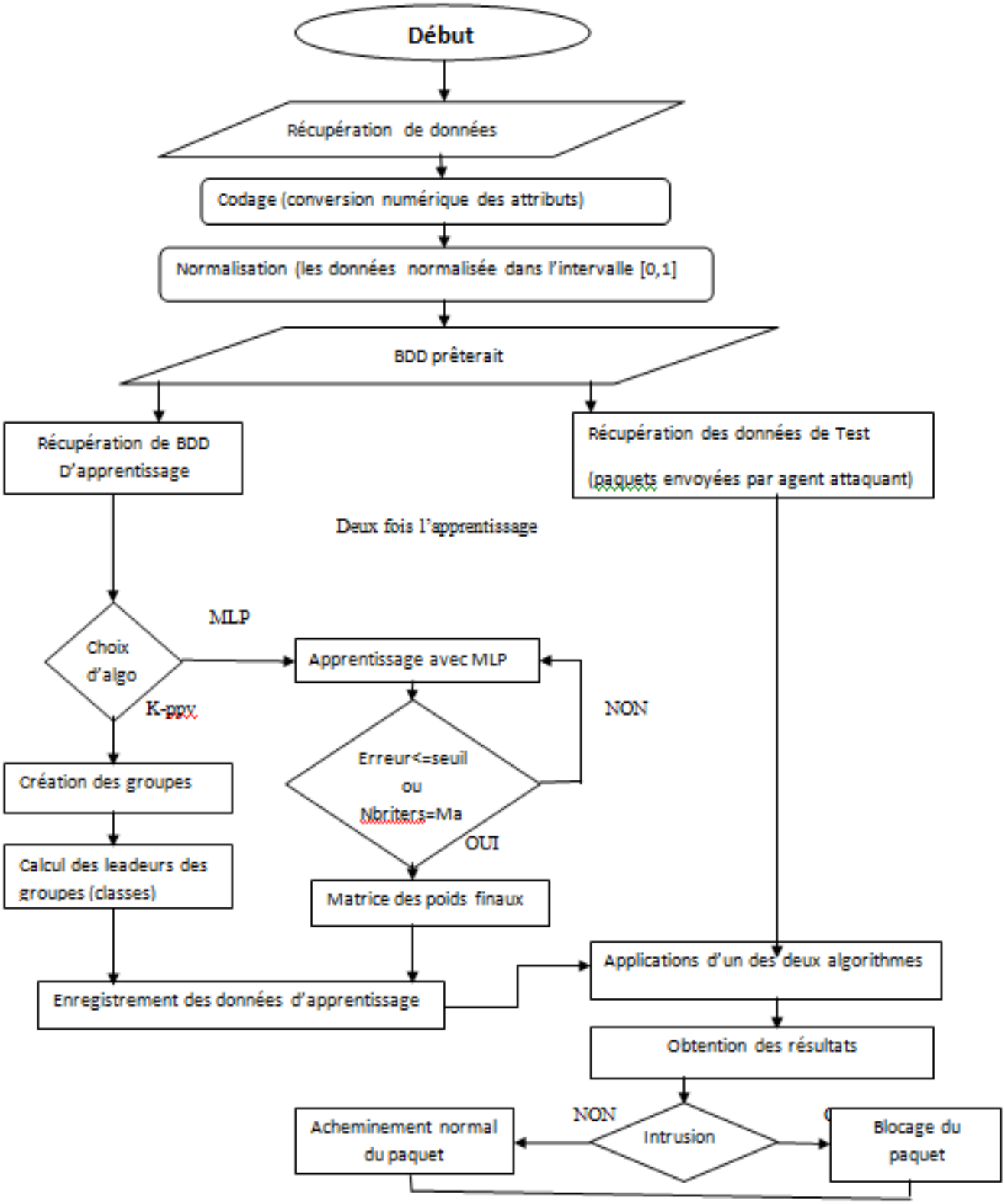

FIGURE 4.1 – Organigramme de nos deux approches proposée SMA et K-PPV ainsi que SMA et MLP

4.2.1 SMA proposé

Comme nous l'avons déjà détaillée dans le chapitre 2 les systèmes multi-agents sont dotées de plusieurs caractéristiques et selon le type d'agent utilisé il existe plusieurs types d'SMA. Le nôtre est hybride parce qu'il contient des agents cognitifs (intelligents) dont les taches sont stratégiques et complexes, ils sont destinée principalement à l'apprentissage ainsi qu'aux tests des deux algorithmes K-ppv et MLP, et d'autres agents dits réactifs qui servent à l'acquisition de la BDD

La communication entre agents cognitifs est réalisée avec des messages FIPA-ACL envoyés entre eux afin de garantir un bon fonctionnement du logiciel et par partage d'environnement pour les agents réactifs.

Selon les besoins de notre application, nous avons utilisé neuf différents types d'agents, chaque agent de notre SMA est doté d'un ou plusieurs comportements qui sont décrits comme suit :

1. **Agent global**

   L'Agent global, c'est l'agent qui s'exécute en premier et gère le système et les agents lors de leurs exécutions.

2. **Agent attaquant**

   L'agent attaquant récupère la BDD de test (KDD 99), cette dernière contiens 3 millions de paquets d'ou sa lecture prend un temps considérable, pour palier à cela nous avons utilisé les threads où la lecture est réalisée de façon parallèle ce qui permet gagner du temps.

3. **Agent serveur (agent superviseur)**

   Afin de garantir un bon apprentissage et une meilleure utilisation des données brutes de la BDD utilisée il est nécessaire dans un premier temps d'exécuter une étape de pré-traitement. La BDD est constituée de plusieurs paquets, chaque paquet est représenter par un vecteur contenant plusieurs variables de type différents.

   L'agent superviseur récupère les paquets à partir de la BDD d'apprentissage qui contient un demie millions de paquets tous en coopérant avec d'autre agents qui assurent l'étape de pré-traitement pour garantir une efficacité en temps. Il s'agit de trois agents spécialisés où chacun de ces agents effectue une des taches suivante :

   — La réception des paquets.
   — Le codage des paquets.
   — La normalisation des données de ces paquets reçus.

   C'est à ce niveau que l'apprentissage aura lieu en utilisant les RNS ou l'algorithme K-mens. L'or de notre première contribution via les RNs ces paquets seront introduits dans le RN dans une étape d'apprentissage pour obtenir la meilleure combinaison possible de poids qui vont être par la suite utilisés dans la détection d'intrusions dans une étape de

test qui sera exécutée par l'agent analyseurs.

Dans notre deuxième contribution via la méthode K-ppv il est nécessaire de déterminer les paquets représentant les leadeurs pour cela une classification est nécessaire. Il s'agit d'un apprentissage supervisé réalisé avant la détection d'intrusions. Une fois les poids du RN ou les leadeurs du K-ppv sont déterminés ils sont enregistré dans un fichier qui sera par la suite utilisé l'ors du de l'étape test.

Notant toute fois que la connexion entre l'agent attaquant et l'agent serveur est faite via les sockets (connexion client/serveur).

4. **Agent récepteur**

   C'est l'agent responsable de la récupération des données de la base d'apprentissage, vu que la base utilisée est assez importante (1/2 millions de paquets) nous avons profité de la notion de parallélisme sachant que chaque agent est représenté par un thread afin de gagner en temps d'exécution.

5. **Agent Codage**

   Nous avons remarqué que les types de variables sur les champs du même paquet sont différents (numériques et alphanumériques) sachant que les entrées de l'algorithme k-ppv et MLP sont numériques alors nous avons procédé à la codification des champs contenant du texte.

   C'est l'agent codage qui est responsable sur ces taches dans chacune des étapes d'apprentissage et test.

   Puisque les entrées de l'algorithme k-means et MlP sont numériques alors nous avons procédé à la codification des champs contenant du texte ("protocol-type", "service" et "flag".).

   Les tableaux suivants présentent les différents codes utilisés :

   — **codage de l'attribut protocol_type**

   Cet attribut contient trois catégories distinctes : ICMP, UDP , TCP , raison pour laquelle nous avons utilisé une codification numérique simple (ordre croissant de nombre entiers) .

Etat normal	**codification**
ICMP	0
TCP	1
UDP	2

TABLE 4.1 – Codification de l'attribut « protocol-type »

— **codage de l'attribut service**

Cet attribut contient 70 catégories distinctes, alors nous avons utilisé la même codification précédentes (ordre croissant de nombre entiers), est ce choix repose sur sa simplicité.

Etat normal	codification	Etat normal	codification	Etat normal	codification
aol	0	http_443	23	printer	46
auth	1	http_8001	24	private	47
bgp	2	imap4	25	red_i	48
courier	3	IRC	26	remote_job	49
csnet_ns	4	iso_tsap	27	rje	50
ctf	5	klogin	28	shell	51
daytime	6	kshell	29	smtp	52
discard	7	ldap	30	sql_net	53
domain	8	link	31	ssh	54
domain_u	9	login	32	sunrpc	55
echo	10	mtp	33	supdup	56
eco_i	11	name	34	Systat	57
ecr_i	12	netbios_dgm	35	Telnet	58
efs	13	netbios_ns	36	Tftp_u	59
exec	14	netbios_ssn	37	Tim_i	60
finger	15	netstat	38	Time	61
ftp	16	nnsp	39	urh_i	62
ftp_data	17	nntp	40	urp_i	63
gopher	18	ntp_u	41	uucp	64
harvest	19	other	42	uucp_path	65
hostnames	20	pm_dump	43	vmnet	66
http	21	pop_2	44	whois	67
http_2784	22	pop_3	45	X11	68
Z39_50	69				

TABLE 4.2 – Codification de l'attribut « service»

— **codage de l'attribut flag**

Le même principe est utilisé pour ces champs.

Etat normal	codification	Etat normal	codification
OTH	0	S0	5
REJ	1	S1	6
RSTO	2	S2	7
RSTOS0	3	S3	8
STSP	4	SF	9
SH	10		

TABLE 4.3 – Codification de l'attribut «Flag»

6. **Agent normalisation**

Cet agent s'occupe de normaliser les valeurs obtenues après le codage car elles sont

très variées et très large, c.a.d il y a des attributs qui prennent de grandes valeurs et d'autres de petites valeurs et surtout après l'étape de codage. Ce qui peut nuire au bon fonctionnement du système de détection d'intrusion, pour cela nous avons opté pour une normalisation des valeurs dans un intervalle [0,1].

Nous avons utilise la fonction de normalisation suivante [M BOUROUH ;2017] :

$$valnouv = \frac{(valanc - Minanc)}{(Maxanc - Minanc)} \times (Maxnouv - Minnouv) + Minnouv \qquad (4.1)$$

Où :

— valanc : est la valeur à normaliser.
— valnouv : est la valeur après la normalisation.
— Minanc : est la limite inférieure de l'intervalle à que valanc appartient
— Maxanc : est la limite supérieure de l'intervalle à que valanc appartient.
— Minnouv : est la limite inférieure de l'intervalle à que valnouv va appartenir.
— Maxnouv : est la limite supérieure de l'intervalle à que valnouv va appartenir.

7. **Agent analyseur**

Nous avons utilisé plusieurs agents analyseurs qui communiquent avec l'agent superviseur via échange de messages. Ils sont responsables sur l'étape d'analyse des paquets et par conséquent la détection de l'intrusion si elle existe.

Après plusieurs expérimentations nous sommes parvenus à une relation entre le nombre d'agents analyseurs et le nombre d'agents clients (pc) où pour chaque deux agents clients un analyseur est responsable de la détection.

Une fois que l'agent superviseur ai déterminé les poids du RN ou les leadeurs du K-ppv, les agents analyseurs restent en attente des paquets venant de l'attaquant et qui visent un ordinateur précis. L'adresse IP du pc de visé dans notre simulation est choisie aléatoirement parmi les adresses disponibles sur notre réseau. Seuls les paquets sans intrusion arrivent à destination, les autres paquets sont bloqués par l'agent analyseur exécutant ainsi k-ppv ou RN. Dans ce deuxième cas un message de retour sera envoyé à l'attaquant l'informant que le paquet a été refusé.

8. **Agent client**

Il existe plusieurs agents client, chacun de ces agents représente un ordinateur (pc). Les paquets avec intrusion âpres avoir été analysés sont bloqués et par conséquent l'agent client ne reçoit que les paquets normaux affichant ainsi le paquet reçu avec l'heure d'arrivée.

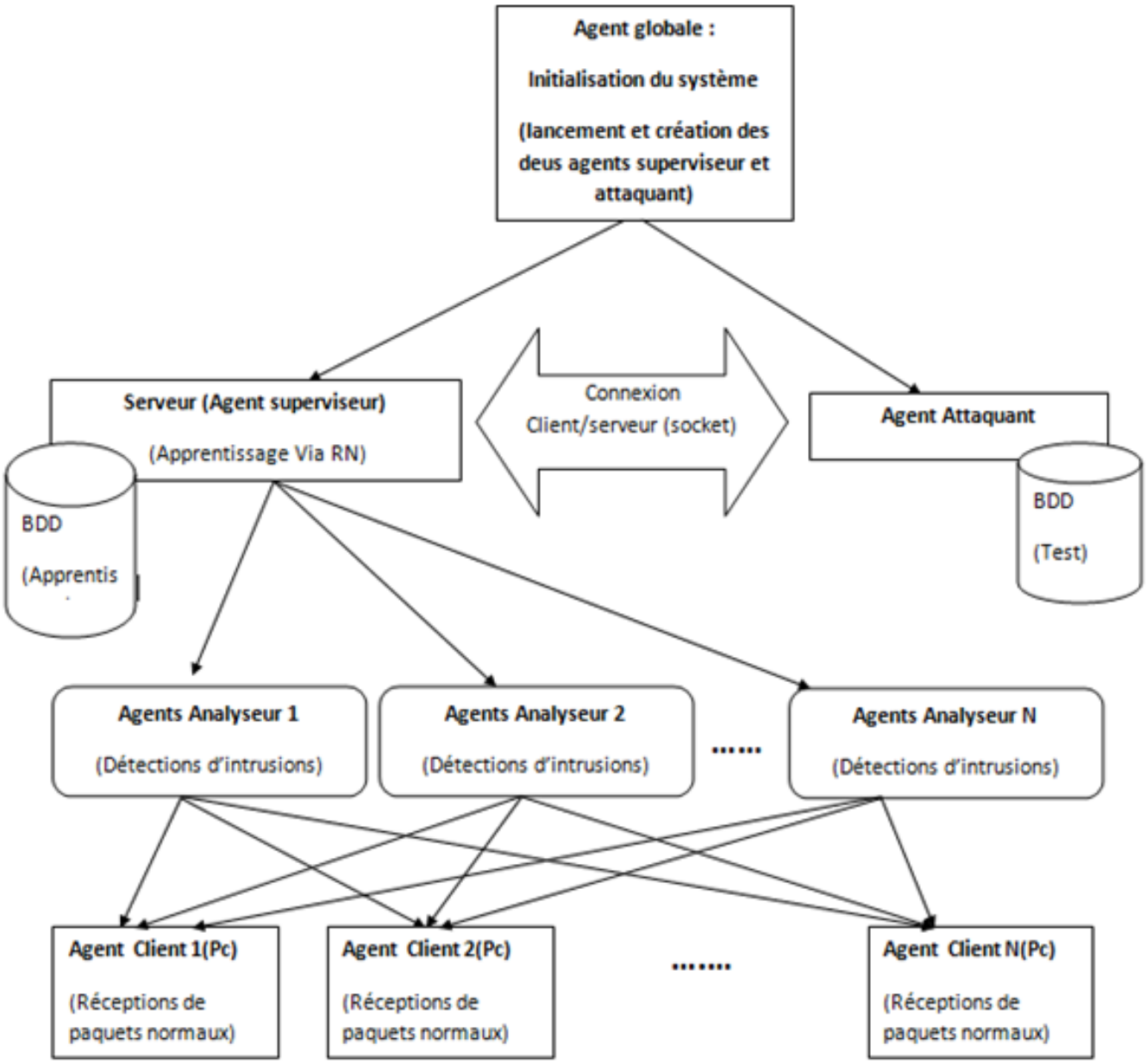

FIGURE 4.2 – Architecture de notre SMA proposé

4.2.2 Base de données utilisée KDD99

KDD 99 est une base de données dédiée à la détection d'intrusions basées sur l'ensemble de données issues du DARPA 98 qui fournit des données étiquetées pour les chercheurs travaillant dans ce domaine. C'est le seul jeu de données étiqueté publiquement disponible (open source) et il est considéré comme un standard pour l'évaluation des SDIs [kayacik, 2005].

En 1998, les laboratoires de MIT Lincoln ont organisé un programme d'évaluation des systèmes de détection d'intrusion DARPA dans le but d'examiner et d'évaluer les recherches dans la détection d'intrusion. Pour cela, ils ont installé un environnement pour acquérir des connexions TCP/IP pendant neufs semaines dans un réseau local (LAN) simulant un LAN typique de l'Armée de l'Air des États-Unis. Le réseau a été actionné mais en lui injectant de multiples attaques [KEN98]. Pour chaque connexion TCP/IP, 41 attributs quantitatifs et qualitatifs ont été extraits [STO00] .La compétition "KDD Intrusion Detection 1999" utilisait un sous-ensemble de 494021 enregistrements ce qui représente 10% de la base de données globales [SLIMANI Ahmed 2010].

Les principaux types d'attaques de l'ensemble de données KDD99 sont :

— Probing : surveillance et sondage.
— DOS (Denial Of Service) : déni de service.
— R2L (Remote to User) : accès non autorisé à partir d'une machine distante.
— U2R (User to Root) : accès non autorisé pour avoir le privilège d'un administrateur.

4.2.3 Algorithme K-PPV [Oumiloud et al, 2013]

Les algorithmes K-PPV représentent un ensemble des techniques qui visent à deviner l'appartenance d'un individu à une classe en s'aidant uniquement des valeurs qu'il prend. Elle construit un modèle représentatif d'un certain nombre de données organisées en classes que l'on appelle généralement le corpus d'apprentissage, puis d'utiliser ce modèle afin de classer de nouvelles données, c'est a dire de prédire leur classe au vu de leurs caractéristiques (appelées paramètres). La construction du modèle relève de l'apprentissage automatique supervisé, l'ensemble des membres constituant le corpus d'apprentissage étant sont étiqueté.

La méthode des plus proches voisins (notée parfois k-PPV ou k-NN pour (k-Nearest-Neighbor)) consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des plus proches voisins parmi les individus déjà classés. L'individu est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode nécessite de choisir une distance, la plus classique est la distance euclidienne et le nombre de voisins est un paramètre à prendre en compte.

Cette méthode supervisée est souvent performante, cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte [Berrani, SA et al 2002].

— **Adaptation de K-PPV avec notre SDI**

Afin de réaliser notre SDI nous avons opté pour deux algorithmes très connu dans le domaine informatiques afin de parvenir a la détection.

Attaque	Échantillons	Catégorie
smurf.	280790	Dos
neptune.	107201	Dos
back.	2203	Dos
teardrop.	979	Dos
pod.	264	Dos
land.	21	Dos
normal.	97277	Normal
satan.	1589	Probe
ipsweep.	1247	Probe
portsweep.	1040	Probe
nmap.	231	Probe
warezclient.	1020	r2l
guess_passwd.	53	r2l
warezmaster.	20	r2l
imap.	12	r2l
ftp_write.	8	r2l
multihop.	7	r2l
phf.	4	r2l
spy.	2	r2l
buffer_overflow.	30	u2r
rootkit.	10	u2r
loadmodule.	9	u2r
perl.	3	u2r

TABLE 4.4 – Les étiquettes de classe qui apparaissent dans un Jeu de données "10% KDD".[kayacik 2005]

En prenant en considération les caractéristiques de la BDD utilisée tout au long de notre étude (KDD 99), nous avons déduit que la méthode K-ppv est plus approprié parce que les données brutes sont classifiées en classes (clusters).

Rappelons toute fois qu'avant d'utiliser le corpus , il faudrait passer par l'étape de pré-traitement .

Dans l'algorithme des K-ppv les membres de la même classe sont regroupés en prenant compte de la grandeur ainsi que nombres de vecteurs inclus dans notre corpus (494021 paquets), où nous avions calculé un leadeur pour chaque classe afin d'éviter le calcul avec tous les membres du corpus. Afin de déterminer les leadeurs nous avons calculé la moyenne de tous les membres :

$$Leadeur = \frac{1}{n}\sum_{i=1}^{n} vecteur(i) \tag{4.2}$$

Ensuite pour l'étape de détection, la similarité ou non entre deux objets est mesurer par une méthode largement étudiées « distance euclidienne ». [B Taconet et al ; 2006]

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y(i) - x(i))^2} \tag{4.3}$$

L'agent analyseur calcule la distance euclidienne du vecteur d'entré envoyé par l'agent attaquant et les leadeurs qui sont calculés et enregistrés dans un fichier afin de déterminer si il s'agit d'une intrusion ou non .

4.2.4 Réseaux de neurones

Les réseaux de neurones sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle et reliés par des poids. Ces poids de connexion gouvernent le fonctionnement du réseau. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Les réseaux de neurones ont plusieurs avantages dans la mise en oeuvre d'un système de détection d'intrusion. Ils sont très efficaces et rapides dans la tâche de classification [G. DREYFUS 1998]. Ils sont capables d'apprendre et d'identifier facilement les nouvelles menaces qui leur sont soumises. Les réseaux de neurones sont capables de traiter les données incomplètes, imprécises et provenant de sources multiples. La rapidité naturelle des réseaux de neurones permet de réduire les dommages lorsque la menace est détectée [Cannady,J 1998]. L'utilisation des réseaux de neurones permet d'extraire les relations non linéaires qui existent entre les différents champs d'un paquet et de détecter en temps réel les attaques complexes [Vladimir Golovko et al 2005]. Les réseaux de neurones, après avoir appris correctement, ont une bonne capacité de généralisation, c'est-à-dire qu'ils sont capables de calculer avec précision les sorties correspondantes même pour des données qui n'ont pas été apprises. La flexibilité qu'offrent les réseaux de neurones est également l'un des atouts pour la détection d'intrusion [Yousef Abuadlla et al 2005].

La classification est l'étape de décision ; le choix de la classe dont la représentation ou le modèle est le plus proche. Il existe de nombreuses méthodes de classification, cependant on se limite dans ce qui suit au classificateur neuronal.
Les réseaux de neurones artificiels constituent une imitation du cerveau humain dans la résolution des problèmes que les êtres vivants savent résoudre. Ils se ressemblent en deux points essentiels ; la connaissance est acquise au travers d'un processus d'apprentissage, les poids des connexions entre les neurones sont utilisés pour mémoriser la connaissance [H. Abdi 1994]. Il existe plusieurs types de RN nous avons opté pour le Perceptron MultiCouche MLP.

1. **Perceptron multicouches**

   Le perceptron multicouche se compose d'une couche d'entrée (rétine du perceptron), d'une ou plusieurs couches intermédiaires (dites couches cachées), et d'une couche de sortie. Le calcul de l'activation dans le réseau s'effectue en propageant l'activation initiale de la couche d'entrée jusqu'à la couche de sortie [B. Gosselin 1996].

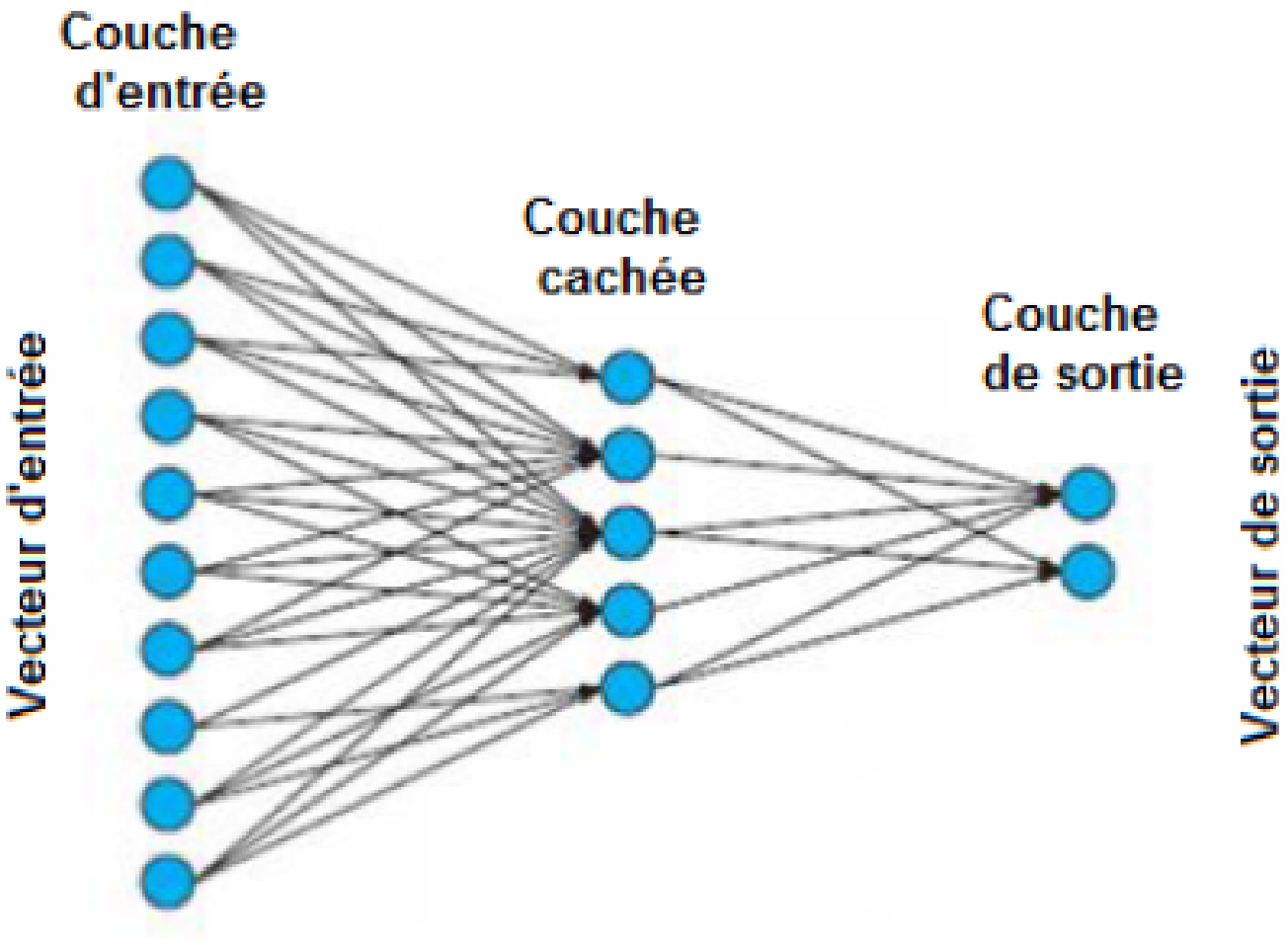

FIGURE 4.3 – Perceptron multicouches

2. **Architecture de notre MLP**

Notre MLP est constitué d'une couche d'entrée, une couche cachée et une couche de sortie décrite comme suit :

— **Couche d'entrée :** Les valeurs d'entrés représente des vecteur de 41 attributs
— **Couche cachée :** Après plusieurs expérimentions nous avons opté pour une seule couche cachée avec un nombres de neurones.
— **Couche de sortie :** Notre MLP contient deux neurones de sortie une pour les paquet sans intrusion (normaux) et la seconde pour les paquets avec intrusions.
— **Fonction d'activation :** nous avons utilisé la fonction sigmoïde définit par :

$$f(x) = 1/(1 + e^{(-\alpha x)}) \tag{4.4}$$

L'architecture de notre MLP est modélisée par les schémas suivants :

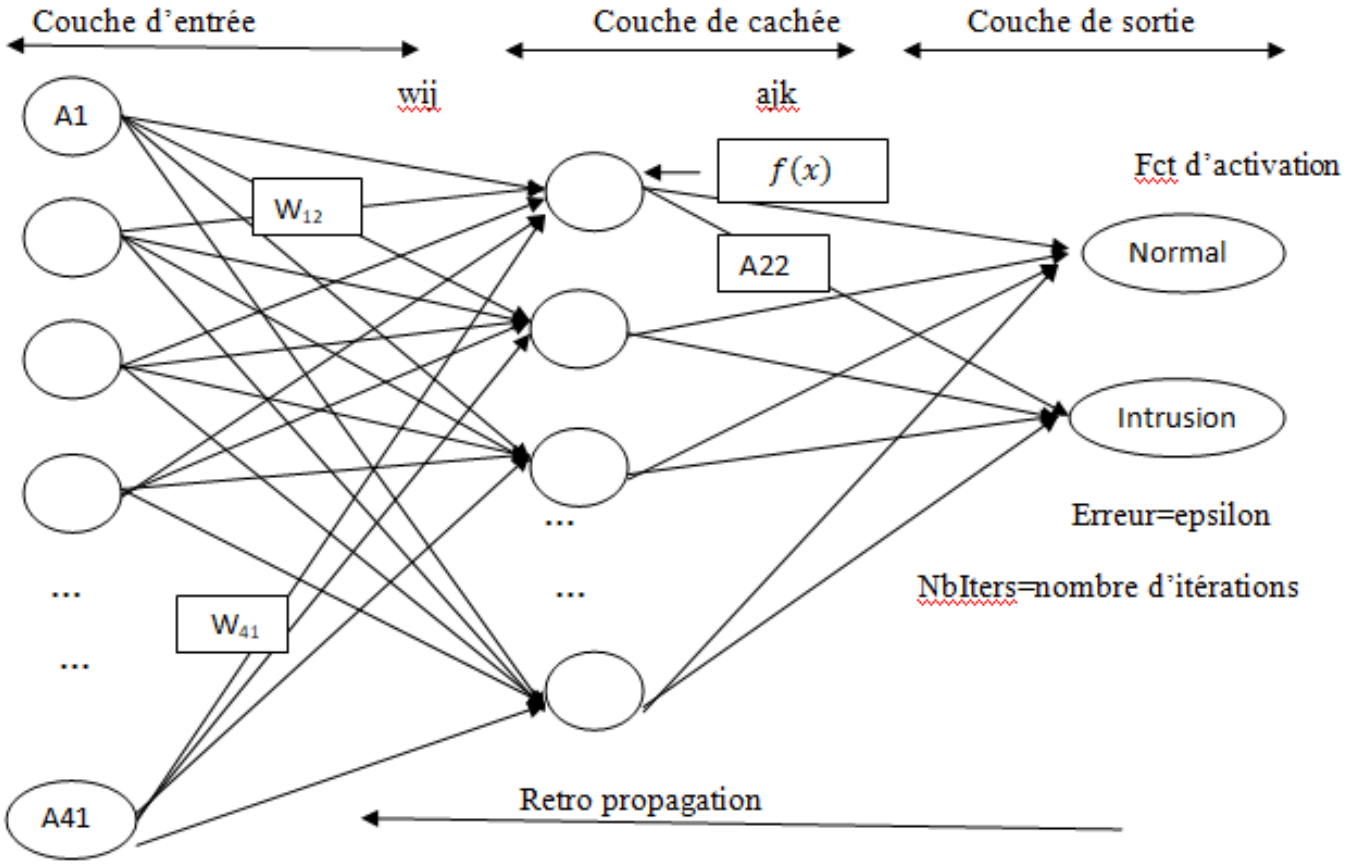

FIGURE 4.4 – Architecture du MLP proposée

3. **L'étape d'apprentissage de notre MLP**

   L'étape d'apprentissage est une des principales taches de l'agent superviseur, où avant de passer à l'étape de détection il faudrait d'abord effectuer un apprentissage du MLP afin de calculer les matrices de poids Wij et Ajk et les sorties.

   Au début les matrices d'entrées sont initialisée aléatoirement.

   Pour l'apprentissage nous avons utilisé la base de d'apprentissage KDD99 training contenant 1/2 millions de paquets traitée à l'étape de prétraitement. Les paquets sont introduits l'un après l'autre afin d'y parvenir au poids finaux qui seront enregistré et utilisé dans la phase de test.

4. **L'étape de test de notre SDI via le MLP**

   Après avoir récupéré les paquets envoyés par l'agent attaquant, ils seront prétraitée en premier et ensuite analysés en second par les agents analyseurs afin de détecter toute éventuelle menace d'intrusion et ceci en utilisons les poids finaux enregistrés après l'apprentissage du MLP.

4.3 Présentation du logiciel, exécution et résultats expérimentaux

Dans cette partie nous allons présenter les différentes étapes du logiciel conçu via le langage java sous l'environnement de développement NetBeans en utilisant la plate-forme jade, puis par la suite nous illustrerons quelques résultats obtenus via nos deux approches K-ppv et sma ainsi que PMC et SMA présenté précédemment.

4.3.1 Environnement de développement

1. **NetBeans**

   NetBeans est un environnement de développement intégré (IDE) open-source pour le développement avec Java, PHP, C ++ et d'autres langages de programmation. NetBeans est également appelé une plate-forme de composants modulaires utilisés pour développer des applications de bureau Java.

2. **Plateforme Jade**

   Jade est un Intergiciel pour le développement d'applications pair à pair d'agents intelligents sur des plateformes fixes, téléphones mobiles, ...Elle satisfait les spécifications de la FIPA-ACL. JADE fonctionne sous tous les systèmes d'exploitation, inclut tous les composants Obligatoires qui contrôlent un SMA [Duro & Hoarau, 2004]. C'est un projet Open Source, LGPL License contrôlée par Telecom Italia Lab(TILAB), qui reste propriétaire du projet .

3. **Materiel**

   Notre application a été réalisé sous le système exploitation Windows 10 64 bits. Le développement a été fait sur une machine dotée d'un processeur Intel (R) Core I5 M380@ 2.53GHz 2.53 GHz RAM 8 GO.

   Nos simulations SMA doivent être réalisées sur un réseau informatique, pour cela nous avons utilisé deux machines avec les mêmes performances décrites précédemment, sachant que notre agent attaquant réside sur une machine et les agents serveur et analyseurs sur une deuxième machine, quant aux agents clients ils sont dans la même machine que l'agent serveur durant nos simulations mais chaque agent client devrait normalement représenteér une machine « qui peut être ciblé par l'agent attaquant ».

4.3.2 Présentation du logiciel et exécution

Pour l'exécution de l'application conçue il faut tout d'abord lancer les deux agents serveur et attaquant sur nos deux machines, nous allons dans ce qui suit détailler les différentes étapes nécessaires à la simulation de nos approches proposées.

1. **Machine contenant l'agent serveur**

   La machine contenant l'agent serveur est responsable sur la détection ou l'ordre d'exécution des taches est comme suit :

   — **Interface de l'Agent serveur**

   L'interface de l'agent serveur contient tous les principaux paramètres de la simulation ainsi un fichier log qui représente un journal final de la détection regroupant les paquets et leurs états , elle affiche aussi l'analyseur responsable sur cette détection.
   Cette interface permet aussi de lanceur les agents clients (pc) et analyseurs.

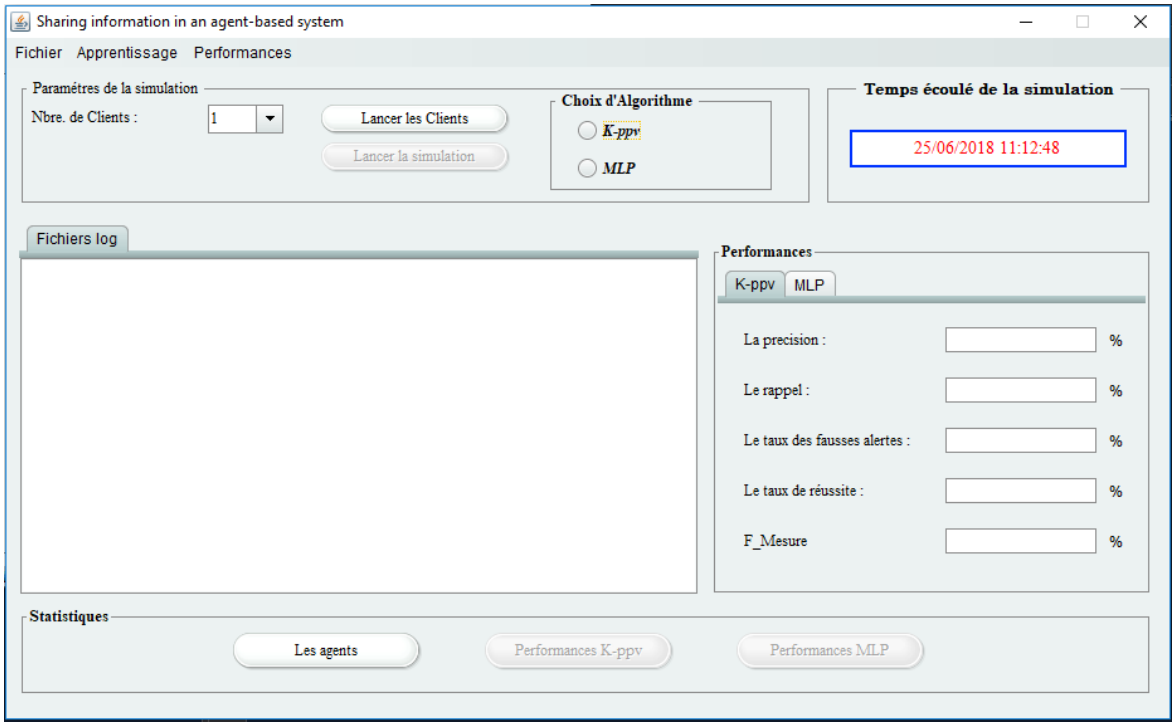

FIGURE 4.5 – Interface de l'Agent serveur

— **Lancement des agents**

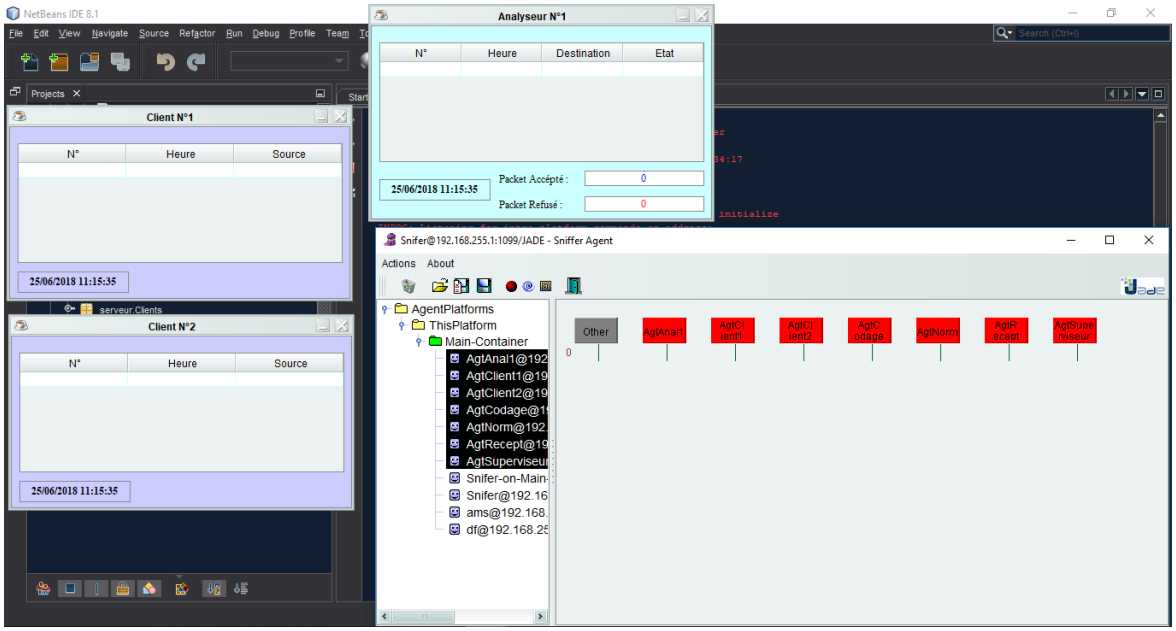

FIGURE 4.6 – Interface Lancement des agents (jade)

— **agents analyseurs et clients (pc)**

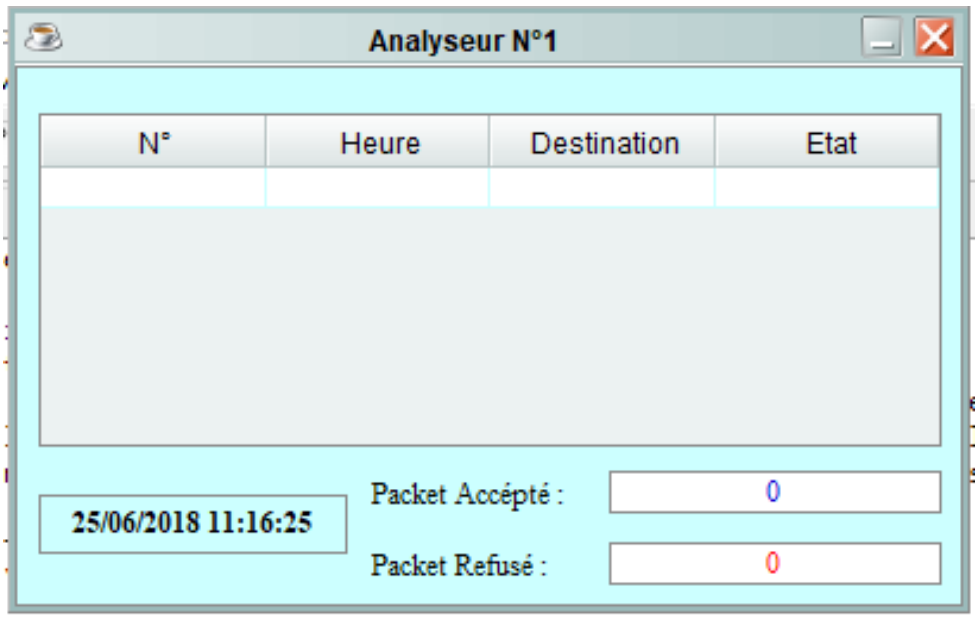

FIGURE 4.7 – Agent Analyseur

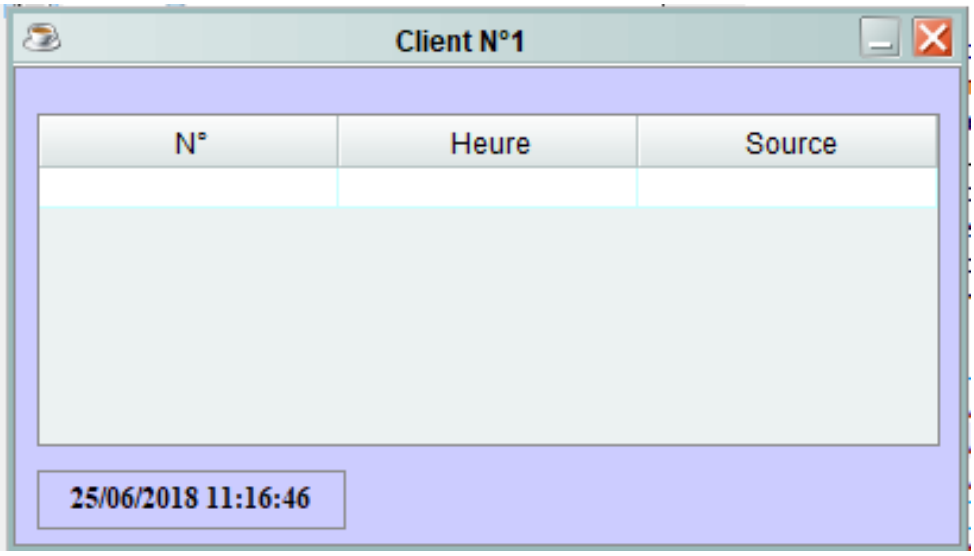

FIGURE 4.8 – Agent client(pc)

— **Interface Prétraitement**

Cette seconde étape consiste à traiter sur les données brutes de bases afin qu'elle soit exploitable par la suite. Cette étape est assurée par des agents spécialistes.

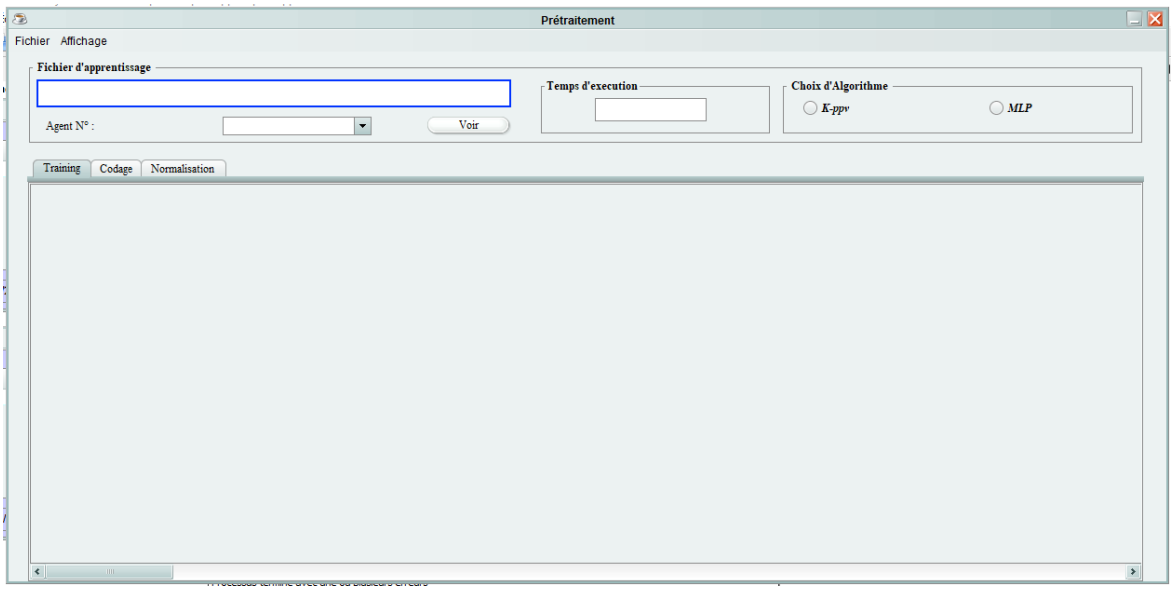

FIGURE 4.9 – Interface Prétraitement

— **Chargement de la base (agent recepteur)**

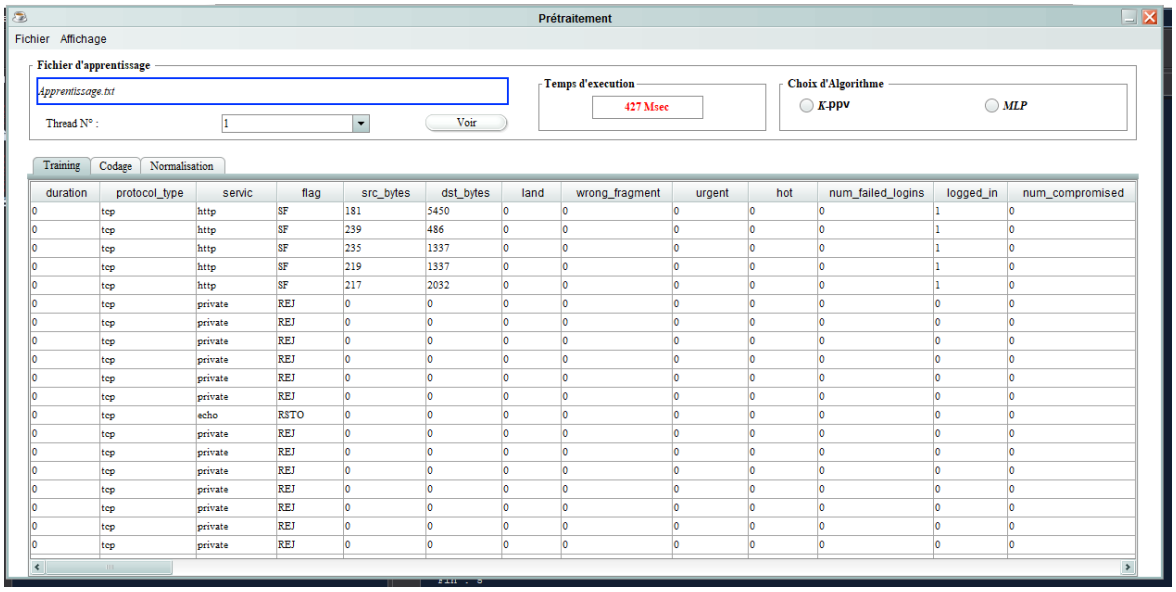

FIGURE 4.10 – Interface Prétraitement (chargement BDD)

— **Codage de la Base (agent codage)**

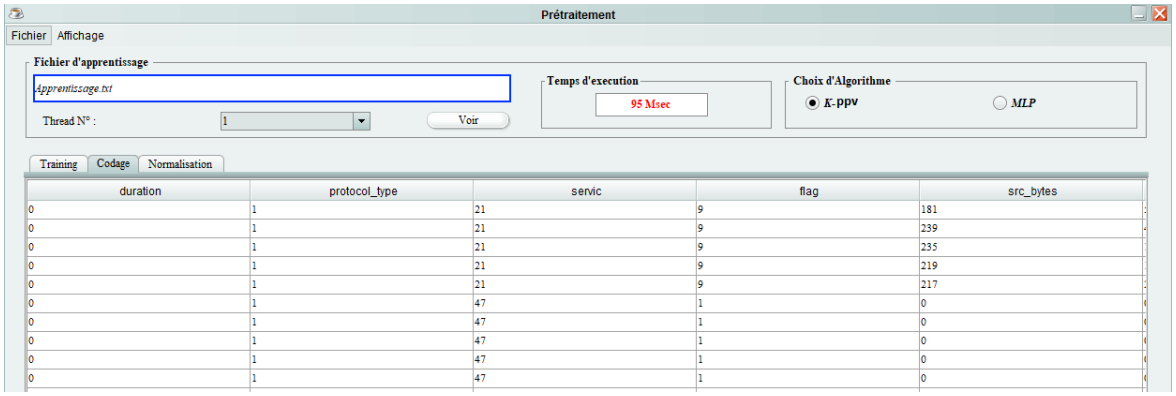

FIGURE 4.11 – Interface Prétraitement (Codage BDD)

— **Normalisation de la Base (agent normaliseur)**

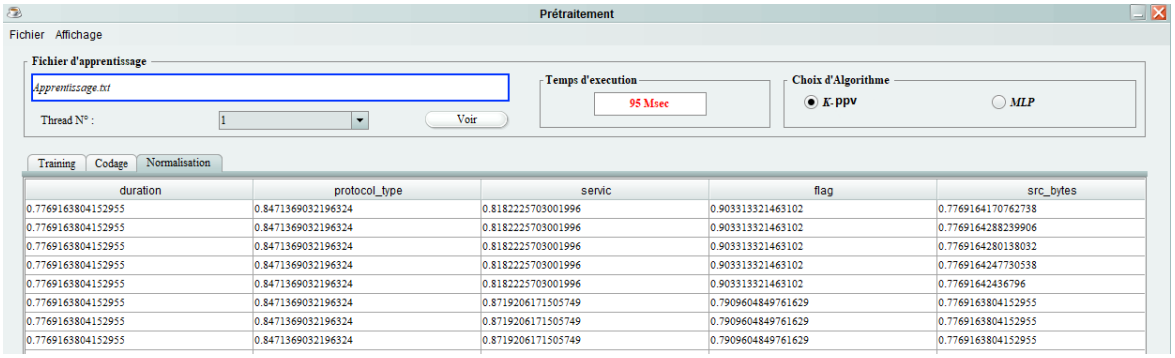

FIGURE 4.12 – Interface Prétraitement (Normalisation BDD)

— **Interface Algorithme K-ppv**

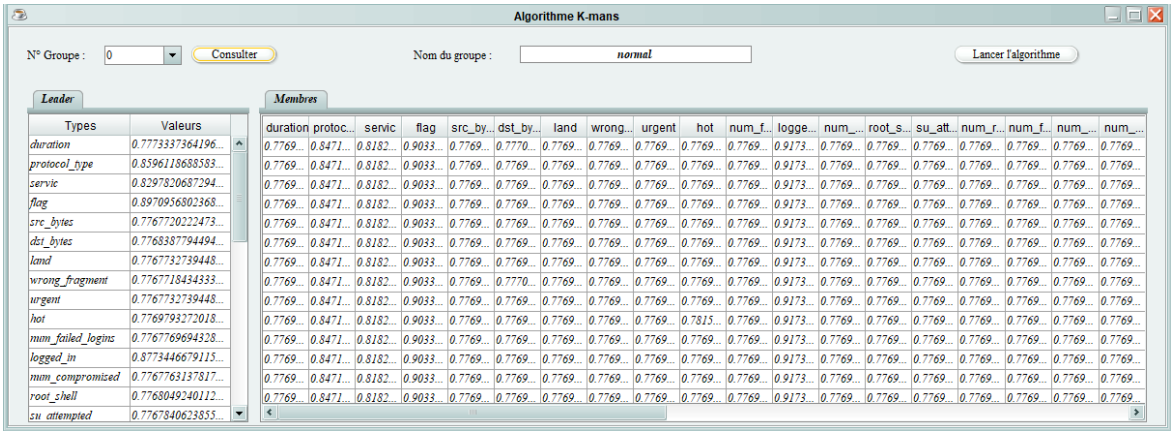

FIGURE 4.13 – Algorithme K-ppv

— **Paramètre de l'algorithme MLP**

Le nombre de neurones de la couche d'entrée ainsi que ceux de la couche de sortie de notre MLP sont constants par contre le nombre de neurones de la couche cachée ainsi que les autres paramètres d'apprentissage (nombre d'itérations, intervalle .....) sont variables. Effectivement notre interface permet à l'utilisateur de choisir les valeurs désirées.

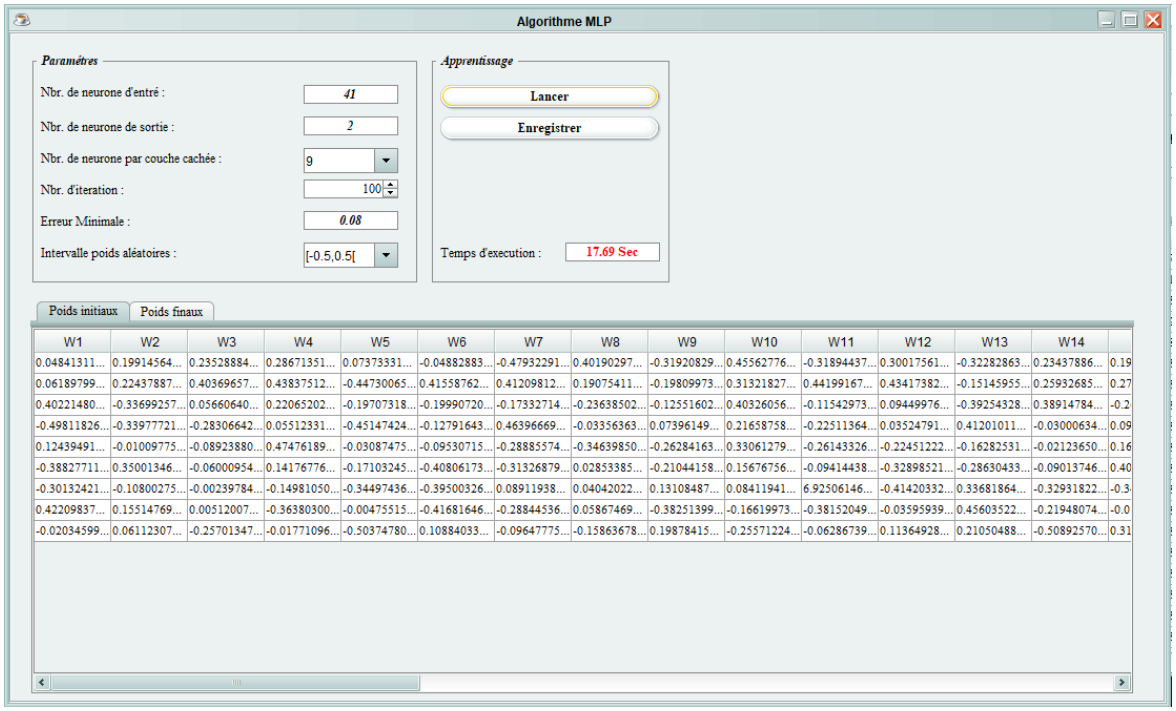

FIGURE 4.14 – Algorithme MLP(poids finaux Wij)

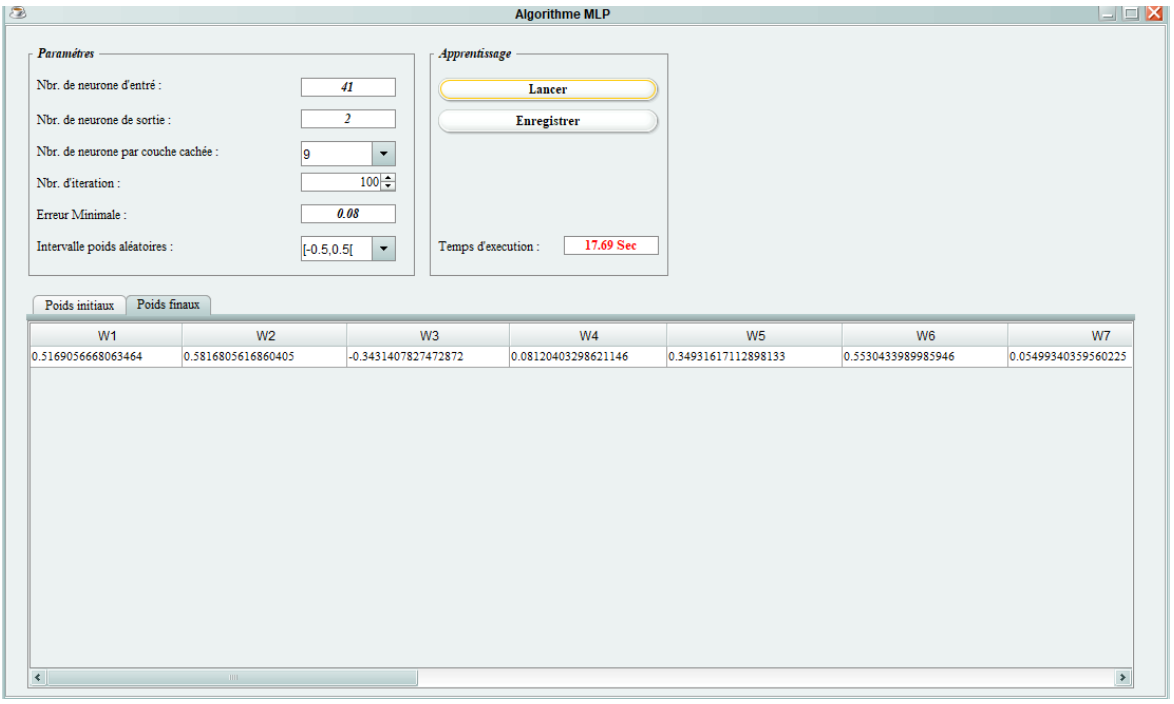

FIGURE 4.15 – Algorithme MLP(poids finaux Ajk)

2. **Machine contenant l'agent attaquant**

Lors du démarrage de l'agent attaquant plusieurs agents réactifs sont lancés afin de charger et d'envoyer la base vers l'agent serveur par la suite les etape suivantes sont réalisée :

— choix du pourcentage de paquets a envoyé.
— détermination du nombre de paquets a envoyés vers l'agent serveur muni de l'adresse ip de la machine destinataire.
— choix du temps séparant les paquets : nombre de paquets envoyé par seconde
— chargement de la base de Test.
— envoie de paquets vers le serveur et l'attente de réponses (accepte /refusé).

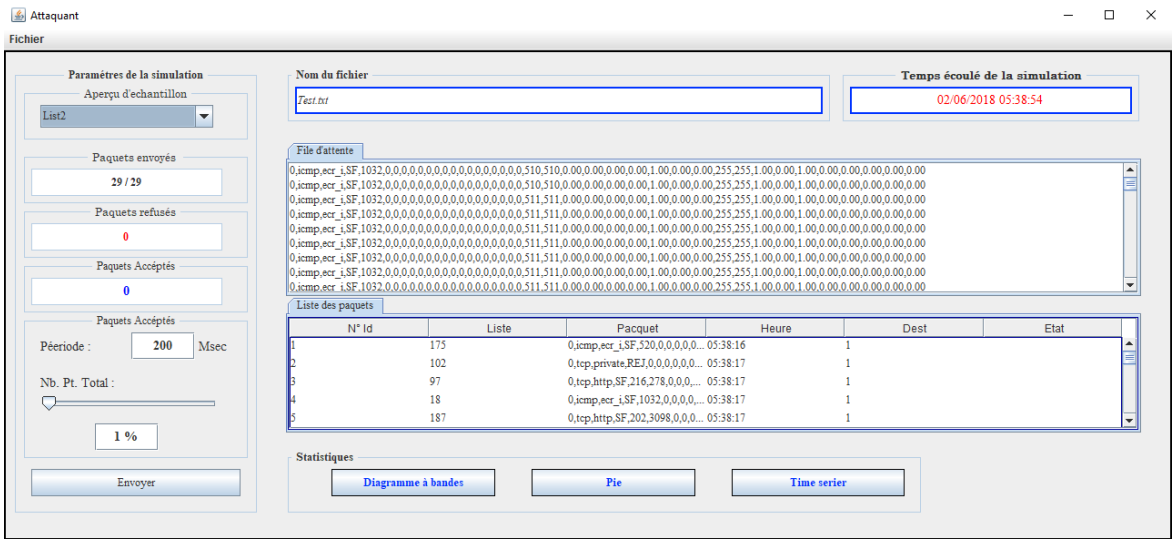

FIGURE 4.16 – Interface Agent attaquant

4.3.3 calcul des performances

Pour mesurer la qualité de la performance du modèle de détection d'intrusions, le résultat de ce dernier sera comparé avec les données réelles (données marquées) où pour l'ensemble de données NSL-KDD toutes les données ont été étiquetées, c'est-à-dire que la classe de chaque instance est connue. Chaque instance est qualifiée de normale ou d'anomalie.

		Classe détectée (prédite)	
		Normale	**Attaque**
Classe réelle	**Normale**	Vrai négatif **TN**(True Negative)	Faux positif **FP** (False Positive)
	Attaque	Faux négatif **FN** (False Negative)	Vrai positif **TP** (True Positive

FIGURE 4.17 – matrice de confusion

Les paramètres utilisés pour mesurer la performance do notre SDI sont les suivant : [M.BOUROUH 2017]

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

Les mesures d'évaluation

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

$$prcision = \frac{TP}{TP + FP} \times 100\% \tag{4.5}$$

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 :

$$Rappel = \frac{TP}{FN + TP} \times 100\% \tag{4.6}$$

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.

$$FP = \frac{FP}{TN + FP} \times 100\% \tag{4.7}$$

le taux de réussite (Accuracy) : Indique la façon dont la technique de détection est correcte. C'est une métrique qui traduit également le rapport entre les détections correctes et les détections totales obtenues.

$$Accuracy = (\frac{TP + TN}{FP + FN} + TP + TN) \times 100\% \tag{4.8}$$

Mesure F(Moyenne harmonique) : La moyenne harmonique F combine le rappel et la précision en un nombre compris entre 0 et 1.

$$Mesure\_F = \frac{2}{\frac{1}{R} + \frac{1}{P}} \times 100\% \tag{4.9}$$

Où R et P sont respectivement le rappel et la précision.

— **Interface Serveur (Performances)**

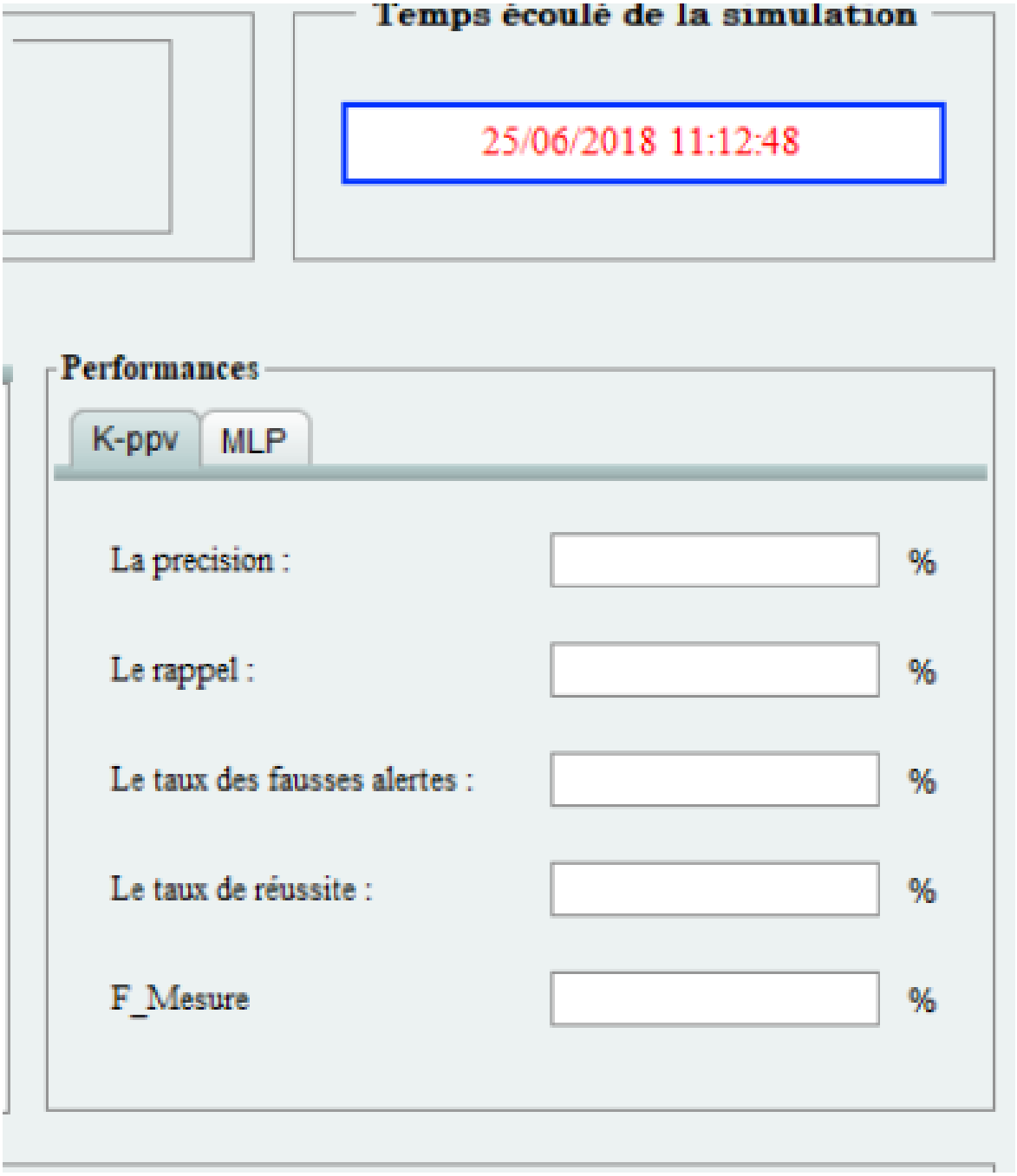

FIGURE 4.18 – Interface Agent serveur (Performances)

4.3.4 Comparaison et discussion des résultats

Paramètres de test

Nous avons effectué plusieurs test dans notre expérimentions en utilisons des parties de la BDD [KDD99], ce choix par partie est du au grand nombres de paquets inclus.

On a utilisé trois tranches de la BDD d'apprentissage (5%, 10%, 20%). Les expérimentations en utilisent les deux algorithmes K-ppv et MLP.

Les paramètres de test pour l'algorithme K-ppv sont standard (nombre de classes et leurs leadeurs fixe).

Pour la part de MLP les paramètres de test sont comme suit :

Paramètres	**Valeur**
Nombre neurone d'entrées	41
Nombre neurone sortie	2
Nombre Couche cachée	1
Erreur	0.00001
Nombre itérations	1000
Intervalle	[-0.5, 0.5]
Mutation	0.001

FIGURE 4.19 – paramètres d'apprentissage MLP

Résultats Obtenu

1. Premier test Algorithme K-ppv

BDD%	NB Attributs	Taux de réussite%	Le Rappel %	La précision%	Les fausses Alarmes %	F_Mesure %
5%	41	90.75	93.58	93.28	15.91	93.43
10%		91.78	94.35	95.61	20.29	94.98
20%		93.57	95.78	93.25	9.42	94.50

FIGURE 4.20 – Evaluation des résultats obtenus avec algorithme K-ppv avec SMA

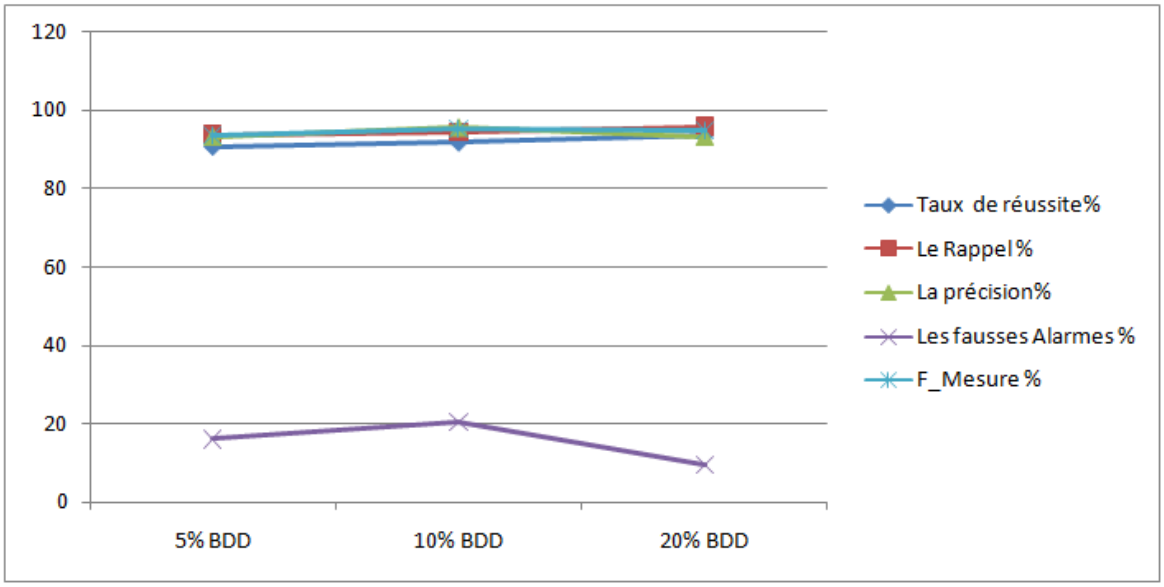

FIGURE 4.21 – Evaluation des résultats obtenus avec algorithme K-ppv avec SMA

2. **Deuxième test MLP**

BDD%	NB Attributs	Taux de réussite%	Le Rappel %	La précision%	Les fausses Alarmes %	F_Mesure %
5%	41	88.10	92.25	90.94	21.72	91.59
10%		92.56	91.82	99.08	3.97	95.31
20%		86.34	97.23	82.28	28.49	89.14

FIGURE 4.22 – Evaluation des résultats obtenus avec algorithme K-ppv avec SMA

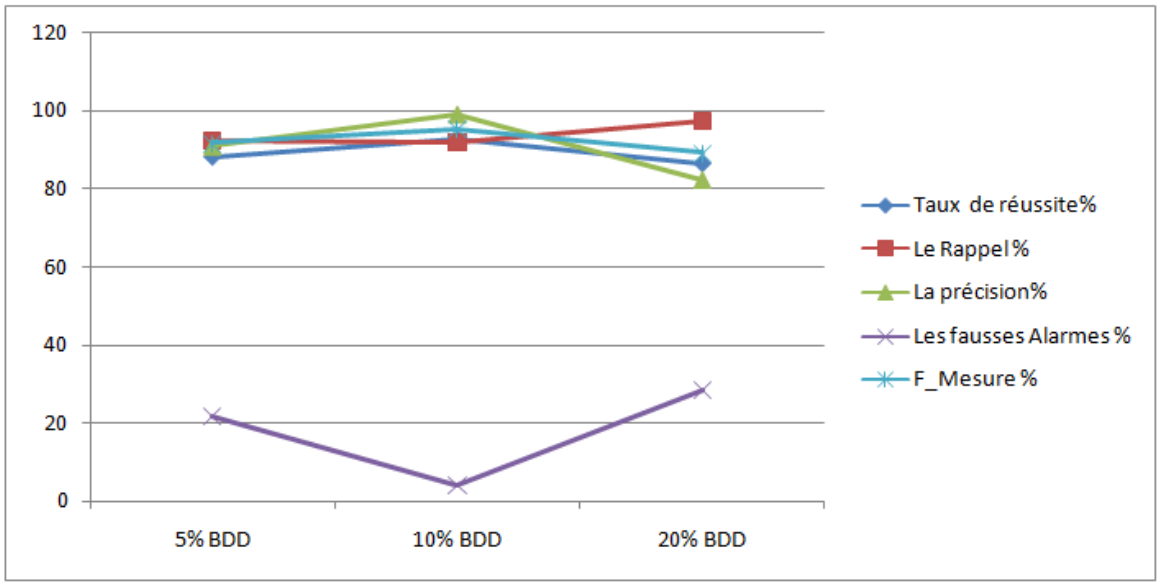

FIGURE 4.23 – Evaluation des résultats obtenus avec MLP et SMA

4.4 Analyse et comparaison des résultats

Les résultats présentés ci-dessus montrent que les performances de notre modèle de détection d'intrusions satisfaisants, il détecte les intrusions avec un taux de réussite, rappel, précision arrivant respectivement jusqu'à 93.57%, 97.23%, 95.61% et un taux de fausses alarmes ne dépassant pas 28.49%.

BDD%	Taux de réussite (accuracy)			Le rappel			La précision		
	K-ppv	MLP	différence	K-ppv	MLP	différence	K-ppv	MLP	différence
5%	90.75	88.10	-2.65%	93.58	92.25	-1.33%	93.28	90.94	-2.34%
10%	91.78	92.56	+0.78%	94.35	91.82	-2.53%	95.61	99.08	+3.47%
20%	93.57	86.34	-7.23%	95.78	97.23	+1.45%	93.25	82.28	-10.97%

FIGURE 4.24 – Comparaison des mesures d'évaluation entre K-ppv et MLP Avec SMA

Les résultats montrent également que l'algorithme K-ppv est plus performant que le MLP avec une couche cachée.

4.5 comparaison des résultats avec un autre travail [M BOUROUH ; 2017]

BDD	Approche	Taux de réussite	Précision	Rappel
TEST 1	SMA MLP	**86.34 %**	**82.28%**	**97.23%**
	MLP	**83.42 %**	**96.55%**	**73.50%**
TEST 2	MLP	**82.75%**	**82.15%**	**71.69%**
TEST 3	MLP	**81.61%**	**94.74%**	**71.67 %**
TEST 4	MLP	**84.47%**	**96.30%**	**75.62%**
TEST 5	MLP	**82.15%**	**93.54%**	73.73%

FIGURE 4.25 – comparaison des résultats avec un autre travail [M BOUROUH ; 2017]

4.6 Exemple de statistique de la simulation du logiciel

— statistiques Agents Analyseur

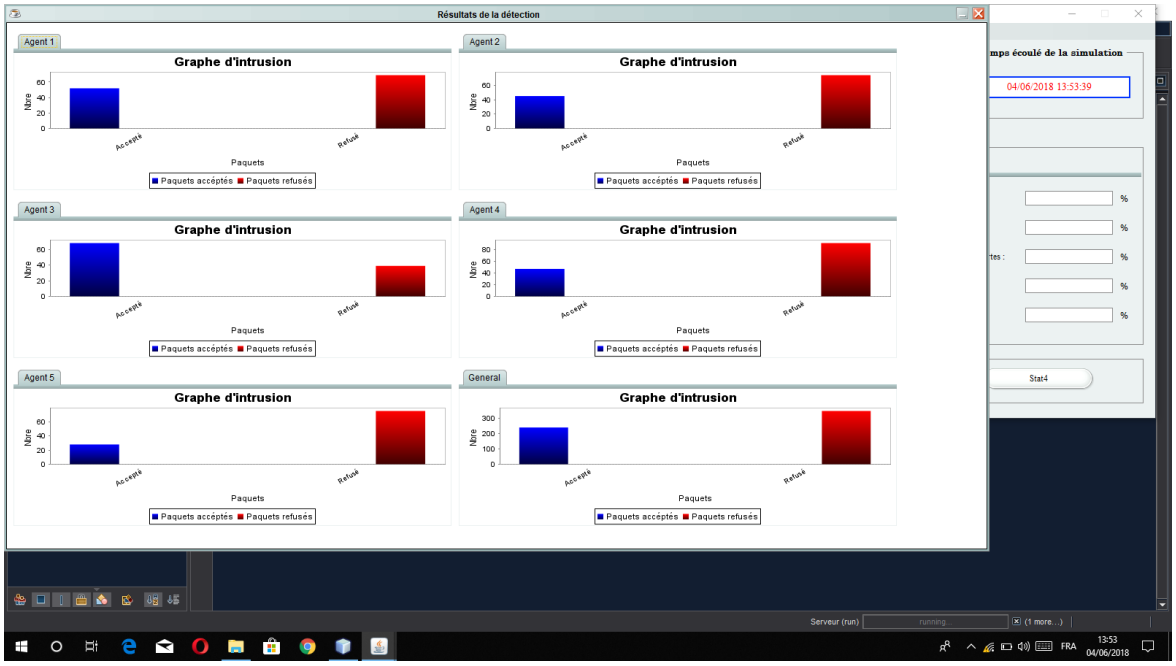

FIGURE 4.26 – statistiques Agents Analyseur et Serveur

— **statistiques Agent Attaquant**

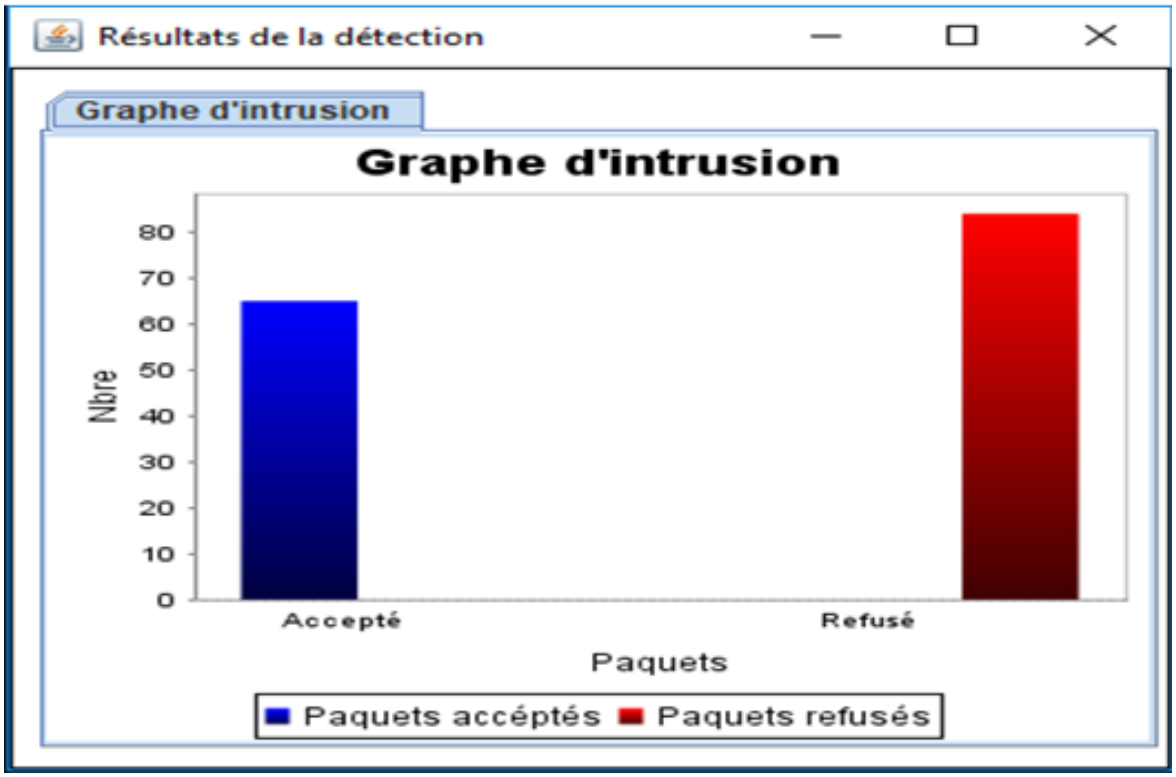

FIGURE 4.27 – statistiques Agent Attaquant

4.7 Conclusion

Dans ce dernier chapitre nous avons présenté nos deux approches proposées, la première approche est une hybridation entre la méthode kppv et les SMAs tandis que la deuxième approche est réalisée via le MLP et les SMAs. Nos deux approche ont testé et validé via la BDD KDD 99[KDD 99] qui a été utilisé pour l'apprentissage et test.

Dans un premier temps nous avons détaillé les différents étapes nécessaire à la mise en œuvre de nos approches, puis par la suite nous avons présentez les résultats obtenus après plusieurs essai et avec des parties différentes de la BDD exprimée en pourcentage.

Les résultats ont obtenue sont satisfaisant et ont prouvé l'efficacité des deux approches, notant toute fois que les résultats du MLP sont meilleur que ceux de l'algorithme K-ppv.

Conclusion générale

Conclusion générale

Les attaques informatiques augmentent de jour en jour, cette augmentation présente un problème majeur pour les réseaux informatiques. En outre la détection en temps réel est devenue primordiale, raison pour laquelle différentes méthodes ont été mises en place.

Dans la première partie de notre travail nous avons introduit les notions de sécurité informatique et attaques puis par la suite nous avons présenté un état de l'art sur les Système de Détection d'Intrusion SDIs. Notre objectif alors est la conception d'un SDI afin de distinguer les anomalies représentant ainsi des attaques.

Pour la réalisation de notre SDI nous avons opté pour les System Multi-Agent en tenant compte des caractéristiques avantageuses qu'ils présentent. Notre approche se repose sur un nouveau paradigme de SMA basée sur l'algorithme K-ppv et le réseau de neurones du type MLP.

L'algorithme K-ppv est une méthode qui a largement montré son efficacité dans le domaine de la classification, quant au MLP il est très approprié pour concevoir un SDI due principalement à sa capacité d'apprentissage.

Pour tester et évaluer notre travail nous avons choisi de travailler avec la base de données KDD99, qui a été conçu spécialement pour l'évaluation des SDIs. Les différentes expérimentations ont donné des résultats satisfaisant en matière de détection où nos approches parviennent même à détecter de nouvelle attaque et en un temps raisonnable.

En conclusion la plupart des objectifs du thème ont été atteint tels que la méthode K-ppv avec SMA qui parvient même à distinguer le type de l'attaque en plus de la détection de cette dernière. Mais il reste des perspectives et des perfectionnements à réaliser dans le futur.

Comme perspective nous proposons la réalisation d'un modèle de classification via un MLP capable de distinguer les types d'attaques au lieu de les classer en deux classe normal et anormal ainsi qu'une minimisation de la BDD (KDD99) en effectuant une sélection des meilleurs attributs dans les vecteurs représentant les paquets de la BDD.

Bibliographie

Bibliographie

[Abdelhalim Zaidi, 2011]Recherche et détection des patterns d'attaques dans les réseaux IP à haut débits ;Abdelhalim ZAIDI ;Janvier 2011.

[Adriana Leite, et al 2017]A hybrid and learning agent architecture for network intrusion detection ;Adriana Leite, Rosario Girardi ;2017.

[Ahmim , 2014]Ahmim, Ahmed, and Nacira Ghoualmi–Zine. "A new adaptive intrusion detection system based on the intersection of two different classifiers." International Journal of Security and Networks 9.3 (2014) : 125-132.

[Ali K, 2001] ; Ali KARTIT ; Une nouvelle approche de détection d'intrusions et étude des problèmes liés au déploiement de politiques de sécurité dans les réseaux informatiques ;Soutenue le : 05/11/2011.

[Arlabosse, 2004]. Arlabosse, F., Gleizes, M. P., & Occello, M. (2004). Méthodes de Conception de Systèmes Multi-agents.

[ARSÈNE SABAS 2001] Sabas, A. (2001). Systèmes multi-agents : une analyse comparative des méthodologies de développement : vers la convergence des méthodologies de développement et la standardisation des plateformes SMA (Doctoral dissertation, Université du Québec à Trois-Rivières).

[B. Gosselin 1996]. Gosselin, B. (1996). Application de réseaux de neurones artificiels à la reconnaissance automatique de caractères manuscrits. Faculté Polytechnique de Mons.

[bellifemine,1999] Bellifemine, F., Poggi, A., & Rimassa, G. (1999, April). JADE–A FIPA-compliant agent framework. In Proceedings of PAAM (Vol. 99, No. 97-108, p. 33).

[Berrani, SA et al 2002]. Berrani, SA., Amsaleg, L., & Gros, P. « Recherche par similarités dans les bases de données multidimensionnelles : panorama des techniques d'indexation.» Ingénierie des systèmes d'information (RSTI série ISI-NIS), 7(5-6), pp 65-90.2002.

[C. Kahn et al, 1998] C. Kahn, D. Bolinger, and D. Schackenberg. Communication in the common intrusion detection framework v 0.7. http ://www.isi.edu/brian/cidf/drafts/communication.txt, Jun. 1998.

[Chaib-Draa, 89] Chaib-Draa, B., Moulin, B., Mandiau, R., & Millot, P. (1992). Trends in distributed artificial intelligence. Artificial Intelligence Review, 6(1).

[Cohen & Levesque, 88 ; 90]. Levesque, H. J., Cohen, P. R., & Nunes, J. H. (1990, July). On acting together. In AAAI .

[Cole et al, 2005] Cole , E .; Krutz,R .; Conley J.(2005) ;Network Security Bible , Wiley Publishing Inc , ISBN 13 :978-0-7645-7397-2.

[D. Denning 1987] D. Denning «An intrusion detection models », IEEE, transaction on software engineering 13(2) : 222-232, 1987.

[Debar et al, 2000] Magnus Almgren, Herve Debar, and Marc Dacier. A lightweight tool for detecting web server attacks. In Proceedings of the Network and Distributed System Security Symposium (NDSS'2000), pages 157–170, San Diego, CA, February 2000.

[Demazeau et al 90] Y. Demazeau. "From Interactions to Collective Behaviour in Agent-Based Systems". In Proceedings of the First European Conference on Cognitive Science (ECCS'95), Saint Malo, France.

[Demazeau et al 91] J-P Briot, Y Demazeau, "Introduction aux agents : Principes et architecture des systemes multi-agents", Collection IC2, Hermes.

[Drogoul 2005] A. Drogoul, "Systemes multi-agents", Projet MIRIAD, Rapport technique,OASIS/LIP6, Universite Paris 6, 2005.

[Duro & Hoarau, 2004] O. Duro & W.Hoarau, " Systeme Multi-Agents", 2004.

[Étienne Duris, 2004]NT Réseaux ;IDS et IPS ; Nicolas Baudoin ;Ingénieurs2000 ; Marion Karle ;2003/2004 ; IR3

[Ferber 995] J. Ferber. "Les Systemes Multi-Agents : Vers une intelligence collective". Inter-Editions, 1995.

[Fikes, 82], Gasser, L. (1992). An overview of DAI. Distributed Artificial Intelligence : Theory and Praxis, 9(9-29), 28.

[Franklin 96] Franklin, S., & Graesser, A. (1996, August). Is it an Agent, or just a Program ? : A Taxonomy for Autonomous Agents. In International Workshop on Agent Theories, Architectures, and Languages (pp. 21-35). Springer, Berlin, Heidelberg.

[G. DREYFUS 1998] G. DREYFUS "les réseaux de neurones" Mécanique Industriel et Matériaux, n51,septembre 1998.

[Haystack Labs 1997] Haystack Labs 1997 ;stalker(1997) ,available from the company's ;Website at http ://www.haystack.com/stalk.htm.

[Helman et Liepins, 1993] Paul Helman and Gunar E. Liepins. Statistical foundations of audit trail analysis for the detection of computer misuse. IEEE Transactions on Software Engineering, 19(9) :886–901, September 1993.

[Holland 1975] Holland, J. H. (1975). Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence.

[Imane Méziane Tani , 2007]. Stratégie de rendez-vous dans les systèmes multi-agents Imane

Méziane Tani ;Université Abou bekr Belkaid - Ingénieur en informatique 2007.

[J. Zimmermann et al , 2002]. Ahmim, Ahmed, and Nacira Ghoualmi–Zine. "A new adaptive intrusion detection system based on the intersection of two different classifiers." International Journal of Security and Networks 9.3 (2014) : 125-132.

[J.Greensmith, J. Feyereisl , 2008] Greensmith, J., Feyereisl, J., & Aickelin, U. (2008). The dca : Some comparison. Evolutionary Intelligence, 1(2), 85-112.

[James Canady ,1998] Cannady,J. , "Artificial Neural Network s for Misuse Detection," Proc eedings,National Information Systems Security Conference (NISSC '98), October, Arlington ,VA, pp . 443 -456. 1 998.

[Jamont ;2005] Jamont, J. P. (2005). DIAMOND : Une approche pour la conception de systèmes multi-agents embarqués (Doctoral dissertation, Institut National Polytechnique de Grenoble-INPG).

[Javitz et al., 1993] Javitz, H. S., Valdes, A., & NRaD, C. (1993). The NIDES statistical component : Description and justification. Contract, 39(92-C), 0015.

[Jennings et al 2000] M.Wooldridge, N. R. Jennings et D. Kinny, "The Gaia Methodology for Agent-Oriented Analysis and Design". Journal of Autonomous Agents and Multi-Agent Systems, 3(3) :285– 312, 2000.

[Jonathan Krier, 2006] ; David BURGERMEISTER, Jonathan KRIER, Système de détection d'intrusion,2006. (http ://dbprog.developpez.com) ;

[kayacik, 2005]. Kayacik, H. G., Zincir-Heywood, A. N., & Heywood, M. I. (2005, October). Selecting features for intrusion detection : A feature relevance analysis on KDD 99 intrusion detection datasets. In Proceedings of the third annual conference on privacy, security and trust.

[KEN 1998]. Kendall, K. A Database of computer Attacks for the evaluation of intrusion Detection Systems. Thèse de Master, institute de technologie Massachusetts.1998.

[Krier, 2006]. David Burgermeister, Jonathan Krier . Les systèmes de détection d'intrusions http ://dbprog.developpez.com 2006.

[Kummar et spafford 1994] Kumar, S., & Spafford, E. H. (1994). A pattern matching model for misuse intrusion detection.

[Lekagning et al 2016] Lekagning, B. H. D., & Tindo, G. (2016). VERS UNE NOUVELLE ARCHITECTURE DE DETECTION D'INTRUSION RESEAUX A BASE DE RESEAUX NEURONAUX.

[M. Sebring et al 1988] Sebring, M. M. (1988). Expert systems in intrusion detection : A case study. In Proc. 11th National Computer Security Conference, Baltimore, Maryland, Oct. 1988 (pp. 74-81).

[M.BOUROUH 2017] M BOUROUH ; Détection d'intrusions à base des réseaux de neurones et algorithmes génétiques ;2017.

[Maes 95] Shardanand, U., & Maes, P. (1995, May). Social information filtering : algorithms for automating “word of mouth”. In Proceedings of the SIGCHI conference on Human factors in computing systems ACM Press/Addison-Wesley Publishing Co.

[Maxime Beaudoin , 2001] Maxime Beaudoin ; Donnees malicieuses : theorie et analyse d'attaques.1 ; 25 aout 2001

[Mé et Alanour, 1996] Mé,L. ;Alanou,V,(1996).Detection d'intrusion dans un système informatique :methode et outils , TSI Journal, VOL.96N4,pp.429-450.

[Minsky 85] Minsky, M. (1985). Why intelligent aliens will be intelligible. In Extraterrestrials. Science and Alien Intelligence (pp. 117-128).

[Mustapha HADD, 1999]Mustapha HADD , « Classification de la population en catégories socio-économiques : méthodologie et application pratique » , mémoire de master, Institut national de statistiques et d'économie appliquée - Ingénieur d'État Option : Démographie 1999.

[N. Habra et al 1992] N. Habra, B. L. Charlier, A. Mounji, and I. Mathieu, ASAX : Software architecture and rule- based language for universal audit trail analysis," in European Symposium on Research in Computer Security (ESORICS), 1992, pp.435450. [Online]. Available citeseer.ist.psu.edu/habra92asax.html.

[N. Nuansri et al 1999] Nuansri, N., Singh, S., & Dillon, T. S. (1999). A process state-transition analysis and its application to intrusion detection. In Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual (pp. 378-387). IEEE.

[N.KAbachi 1996] Ouzrout, Y., Kabachi, N., & Vincent, L. (1996, April). Une société d'Agents pour la prise de Décision dans les organisations productives. In 4ème journées francophones IAD & SMA.

[NedaAfzali et al 2014] Seresht, N. A., & Azmi, R. (2014). MAIS-IDS : A distributed intrusion detection system using multi-agent AIS approach. Engineering Applications of Artificial Intelligence, 35, 286-298.

[Oumiloud et al, 2013] Classification non supervisée : Application de k-means 2013-2014.

[P. G. Neumann 1990]. Teresa F. Lunt, Ann Tamaru, Fred Gilham, R. Jagannathan, Caveh Jalali, Harold S. Javitz, Alfonso Valdes, and Peter G. Neumann. A real-time intrusion-detection expert system. Technical report, SRI International, June 1990.

[P. Porras et al, 1998] Porras, Phillip A., and Alfonso Valdes. "Live Traffic Analysis of TCP/IP Gateways." NDSS. 1998.

[Philip, 1998] A. Phillip, Porras and Alfonso Valdes. Live traffic analysis of tcp/ip gateways. Proc. ISOC Symposium on Network and Distributed System Security (NDSS98). (San Diego, CA, March 98), Internet Society.

[Porras and Kemmerer 1992] Porras and Kemmerer ;Penetration State transition analysis : a rule based intrusion detection approach the Eight Annual Computer Security application

conference 1992.

[R. Feiertag et al, 1999]R. Feiertag, C. Kahn, P. Porras, D. Schackenberg, S. Staniford-Chen, and B.Tung.A common intrusion specification language. Site web :http ://www.isi.edu/brian/cidf/drafts/la Jun. 1999.

[Reichgelt, 90] Major, N., & Reichgelt, H. (1990). ALTO : An automated laddering tool. IOS Press.

[Ricordel 2001] Ricordel, P. M. (2001). Programmation orientée multi-agents : développement et déploiement de systèmes multi-agents voyelles (Doctoral dissertation, Grenoble INPG).

[Russell 97], Barkley, R. A. (1997). Behavioral inhibition, sustained attention, and executive functions : constructing a unifying theory of ADHD. Psychological bulletin, 121(1), 65.

[Shoham 93]. Shoham, Y. (1993). Agent-oriented programming. Artificial intelligence, 60(1), 51-92.

[SLIMANI Ahmed, 2011] SLIMANI Ahmed, Application des systèmes immunitaires artificiels à la détection d'intrusion, USTO-MB : 2011.

[STO 2000] Stolfo,S, Fan, W, lee,W, Prodromidis, A et Chan, P.cost-based Modeling for Fraud and intrusion Detection :Results from the JAM project, Proceeding of DARPA Information Survivability Conference and Exposition ; IEEE computer Press .pp.130-144. 2000.

[Tarfa Hamed et al 2018] Hamed, T., Ernst, J. B., & Kremer, S. C. (2018). A Survey and Taxonomy of Classifiers of Intrusion Detection Systems. In Computer and Network Security Essentials (pp. 21-39). Springer, Cham.

[Vaccaro and liepins ,1989] Vaccaro, H. S., & Liepins, G. E. (1989, May). Detection of anomalous computer session activity. In Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on (pp. 280-289). IEEE.

[Wathiq Laftah Al-Yaseen et al 2016] Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system Wathiq Laftah Al-Yaseen a , b „ Zulaiha Ali Othman , Mohd Zakree Ahmad Nazri ;2016.

[Weiss 99] Weiss, Gerhard, ed. Multiagent systems : a modern approach to distributed artificial intelligence. MIT press, 1999.

[Wespi et al 2000] Wespi, A., Dacier, M., & Debar, H. (2000, October). Intrusion detection using variable-length audit trail patterns. In International Workshop on Recent Advances in Intrusion Detection (pp. 110-129). Springer, Berlin, Heidelberg.

[Wooldridge et al 95] Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents : Theory and practice. The knowledge engineering review, 10(2), 115-152.

[Yassine Maleh et al 2015] A Global Hybrid Intrusion Detection System for Wireless Sensor Networks Yassine Maleh*a, Abdellah Ezzatib , Youssef Qasmaouic, Mohamed Mbidac.

[Yousef Abuadlla et al 2005]. Yousef Abuadlla && all "Flow-Based Anomaly Intrusion Detection System Using Two Neural Network Stage", Computer Science and Information systems 11(2) : 601-622 : 2012.

Résumé

Avec l'évolution de l'informatique, l'informations et les réseaux .la vulnérabilités aux intrusions a augmenté .Dans le but de faire face a ce phénomène plusieurs travaux et recherche ont vu le jour.

L'objet de ce mémoire est la réalisation d'un system de détection d'intrusion (SDI) basés sur l'intelligence artificielle .l'étude de l'algorithme K-ppv (K plus proche voisins) et les réseaux de neurones artificielles (inspiré des neurones humains) nous a permis de faire face a la majorité des intrusions connu.

Dans le cadre de ce travail nous proposons un nouveau paradigme de system Multi-Agents en utilisant la base de données KDD99 .L'algorithme L'algorithme analyse les connexions réseaux afin de les classer comme normale ou anormales. Les résultats expérimentaux montrent que le système est performant avec un taux de reconnaissance satisfaisant.

Mots clefs :

SDI , Système multi-agent, réseau neuronal , K-ppv , sécurité informatique.

Abstract

With the evolution of infotmatic, information and networks have become increasingly vulnerable. In order to cope with this phenomenon, several works and research have emerged.

The purpose of this dissertation is the realization of an intrusion detection system (SDI) based on artificial intelligence. The study of the K-ppv algorithm (K nearest neighbors) and artificial neural networks (inspired by human neurons) allowed us to deal with the majority of known intrusions.

As part of this work we propose a new pardigma of Multi-Agent system using the KDD99 database. Algorithm The algorithm analyzes network connections in order to classify them as normal or abnormal. Experimental results show that the system performs well with a satisfactory recognition rate.

Keywords :

IDS, multi-agent system, neural network, K-ppv, computer security.