N^o d'ordre :

République Algérienne Démocratique & Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Faculté des Sciences Exactes
Université Djillali Liabès
Sidi Bel Abbès

Thèse de doctorat en sciences informatique

Option : Informatique

Présentée par : Abdelkader KHOBZAOUI

Contribution des techniques de datamining dans l'amélioration des systèmes de détection d'intrusion dans les réseaux informatiques

Année universitaire 2016-2017

Soutenue le ../../.... devant le Jury composé de :

M. LEHIRECHE Ahmed	Professeur à l'UDL	Président
M. BENSLIMANE Sidi Mohamed	Professeur à l'ESI de Sidi Bel Abbès	Examinateur
M. FARAOUN Kamel Mohamed	Professeur à l'UDL	Examinateur
M. RAHMOUN Abdellatif	Professeur à l'ESI de Sidi Bel Abbès	Examinateur
Mme ZAOUI Lynda	Maître de conférences **"A"** à l'USTO	Examinateur
M. YOUSFATE Abderrahmane	Professeur à l'UDL	Directeur de Thèse

Remerciements

En premier lieu, je tiens à remercier, grandement, Monsieur **Abderrahmane Yousfate**, pour avoir accepté d'encadrer ce travail doctoral, pour ses multiples conseils, pour ses cours et leçons et pour sa bonne humeur et son bon humour tout au long de ce travail.

Mes remerciements s'adressent également à Monsieur **Ahmed Lehireche** pour avoir accepté de présider le jury de cette thèse et à tous les autres membres du jury : Madame **Lynda Laoui** de l'UMCO, Monsieur **Abdellatif Rahmoun** de l'ESI de Sidi Bel Abbès, Monsieur **Kamel Mohamed Faraoun** de l'université Djillali Liabés et Monsieur **Sidi Mohamed Benslimane** de l'ESI de Sidi Bel Abbès pour avoir accepté d'évaluer mon travail de thèse.

Je tiens à exprimer mes plus vifs remerciements et gratitudes, à Monsieur **Boucif Amar Bensaber** et à Monsieur **M'hamed Mesfioui**, professeurs à l'université du Québec à Trois-Rivières pour leurs Générosité aussi bien scientifique que sociale. Et à travers ces deux formidables personnes je tiens, également, à exprimer toutes mes salutations aux membres du laboratoire de Mathématiques et Statistiques appliquées dont j'ai eu l'honneur et le plaisir de côtoyer durant mes différents séjours scientifiques à l'université du Québec.

Il m'est impossible d'oublier d'exprimer mes vifs remerciements à Monsieur **Cazi Abderrahim** l'heureux gentil bonhomme pour sa générosité, son extraordinaire humour et pour tous les inoubliables moments qu'on a passer ensemble.

Mes remerciements vont aussi à tous les membres de ma famille pour leur soutien quotidien indéfectible et leur patience manifestée à l'égard de nom absentéisme récurrent tout au long de ce travail.

Enfin, ces remerciements ne seraient pas complets sans mentionner tous mes amis qui, avec cette question récurrente, "quand est-ce que tu la soutiens cette thèse ?", bien qu'angoissante en période fréquente de doutes, m'ont permis de ne jamais dévier de mon objectif final.

Abdelkader Khobzaoui

ملخص

أن الاستعمال المكثف لشبكات الإعلام الآلي، و الأنترنيت على وجه الخصوص، جعل من معظم الأنظمة المعلوماتية أهدافا لهجمات متطورة باستمرار. وعليه فإن كل مؤسسة مرتبطة بالشبكة العنكبوتية يمكن ان تكون ضحية هجمات أو اعتداءات إن عاجلا أو آجلا. عواقب هذه الهجمات عادة ما تكون جد وخيمة و قد ترهن استمرار المؤسسة. و من أجل حماية المؤسسات و الأنظمة من هذه الأخطار تم تطوير طرق عديدة لمواجهة الهجمات المختلفة التي تهدد أمن الشبكات سواءا على مستوى البيانات أو الموارد. من أهم هذه الاليات ندكر : تقنيات التعمية، اليات التوثيق و مراقبة الدخول او الولوج للشبكات و كذلك الجدران النارية. لكن هذه الاليات الوقائية، ورغم جديتها، ليست قادرة على كشف أو إحباط الانواع الجديدة (غير معروفة\معرفة مسبقا) من الهجمات الإلكترونية المتطورة والمعدلة باستمرار لاستغلال نقاط ضعف أنظمة الإعلام الآلي الناتجة عن أخطاء في التصاميم والإنجازات المتعلقة بها(أنظمة الإعلام الآلي) . أمام هذا الوضع، اصبح من الضروري إيجاد طرق و أليات جديدة أكثر نجاعة قادرة على مراقبة و تتبع مختلف أنشطة مستعملي أنظمة الإعلام الآلي و كذا التطبيقات البرمجية التي تشغل لحسابهم من أجل الكشف عن محاولات الاختراق التي يمكن ان تنفذ عن بعد و من داخل الشبكات.

أفظت الدراسات و البحوث الى بعث مقاربة دفاعية جديدة عرفت تحت اسم كشف الاختراقات أو كشف التسلل. وكان J. P. Andersson اول من نشر مقالا علميا في هذا الموضوع سنة 1980. هذه المقاربة تعنى بتحديد الأنشطة العادية و المسموح بها أو تلك التي تعتبر تهديدا للنظام والتمييز بينها. و لقد عرفها Bace & Mell على انها عملية رصد وتحليل الأحداث الواردة في نظام الإعلام الآلي من أجل الكشف عن أدلة وجود مشكلة أمنية. و عليه فإن عملية كشف الاختراقات توفر معلومات تخص محاولات اختراق الأنظمة سواءا كانت ناجحة أم فاشلة و ذلك عن طريق تحليل كل المعلومات التي تم تجميعها ضمن ملفات التدقيق الأمني الخاصة بالبرمجيات، أنظمة التشغيل و الشبكات. و لما كان حجم هذه الملفات جد كبير و في تزايد مستمر نتيجة الاستعمال المفرط لأجهزة الكمبيوتر ضمن شبكات عالمية ذات تدفق عالي فأن ادماج اليات وخوارزميات تنقيب البيانات ضمن عملية كشف الاختراقات يعتبر حلا طبيعيا جد فعال لما توفره من امكانية تحليل قواعد البيانات الضخمة و استنباط المعلومات و العلاقات و الأنماط الغير ظاهرة او المخفية. فتقنيات التعلم الآلي تحت الإشراف تمكن من بناء نماذج دقيقة انطلاقا من المعطيات الخاصة بالهجمات التي تعرض لها النظام سابقا. أما تقنيات التعلم الآلي بدون إشراف فتوفر امكانية تحديد الأنشطة الخبيثة. فعلا لقد اتجهت الدراسات الحديثة إلى بناء نظم ذكية لكشف الاختراقات المعروفة وغير المعروفة، مستندة على مفاهيم واليات وخوارزميات تنقيب البيانات و نتج عن هذه الدارسات انظمة كشف الاختراقات اثبتت جدارتها وتفوقها على الأنظمة الكلاسيكية. و في نفس نسق هذه الدراسات نحاول ضمن هذه الأطروحة استعراض اهم الأعمال العلمية المنجزة في إطار توظيف اليات وخوارزميات تنقيب البيانات في عملية تطوير انظمة كشف الاختراقات اكثر ذكاء و فعالية. كما سنعمد الى تطوير مقاربة كشف الاختراقات تعتمد على إحدى هذه الأليات والخوارزميات.

الكلمات المفتاحية: كشف الاختراقات، نظام كشف الاختراقات، تنقيب البيانات، تصنيف، تصنيف آلي.

Résumé

L'utilisation continue des réseaux informatiques et du web dans la société d'aujourd'hui a fait que les ressources de la majorité des systèmes informatiques sont devenus *fortiori* des cibles attrayantes d'attaques de plus en plus sophistiquées. De ce fait, tôt ou tard, toute entreprise connectée à internet peut se trouver victime d'une agression électronique à n'importe quel moment et les conséquences d'une telle attaque peuvent être catastrophiques. Les efforts de recherches et de développement consentis en matière de lutte contre de telles menaces ont abouti à un nombre considérable d'outils et de moyens pour éviter, ou repousser dans le temps, les différents types d'attaques. Parmi les plus classiques, on trouve les mécanismes d'authentification, de contrôle d'accès, les protocoles cryptographiques ou encore les pare-feux (Firewalls). Toutefois ces outils, de nature préventive, souffrent d'un nombre considérable d'inévitables vulnérabilités. Ces outils ne sont pas en mesure de faire face, efficacement, aux différentes attaques qui sont continuellement sophistiquées, diversifiées et adaptées à exploiter les faiblesses des systèmes informatiques dûs souvent à des conceptions négligentes ou à des erreurs d'implémentation. Ainsi le développement d'une nouvelle composante ou approche de sécurité capable de surveiller les activités des applications et des utilisateurs d'un système informatique s'impose afin de détecter ou identifier toute sorte d'intrusion. En effet, c'est la prétention de l'approches "réactive" dite "Détection d'intrusion" introduite, initialement par J. P. Andersson en 1980. La détection d'intrusion, comme son non l'indique, consiste à repérer des activités anormales ou suspectes. Bace & Mell la définissent comme étant le processus de surveillance et d'analyse des événement occurrents au sein d'un système informatique dans le but de détecter l'évidence d'un problème de sécurité. Ainsi, par le biais de l'analyse des différents fichiers d'audit de sécurité, la détection d'intrusion permet aussi bien d'avoir une connaissance sur les tentatives d'intrusion réussies que sur celles qui auraient échoués. Ces fichiers d'audit, générés soit par les applications soit par les systèmes d'exploitation ou encore les périphériques réseaux, deviennent de plus en plus volumineux vu l'utilisation accrue des ordinateurs, notamment au sein des réseaux dont le débit ne cesse d'accroître. De ce fait, l'intégration des méthodes de fouille des données (Data mining) dans la détection d'intrusion semble être la solution la plus **naturelle** pour explorer cette importante masse de données afin d'extraire des caractéristiques, des relation et/ou des règles permettant de détecter le maximum d'attaques possibles au moment opportun. En effet, l'utilisation des techniques de fouille des données dans la sécurité des systèmes informatiques a suscité, au cours des trois dernières décennies, un intérêt considérable de la part de la communauté des chercheurs et des professionnels de l'informatique et de la fouille des données.

La fouille des données peut contribuer à l'amélioration des performances des systèmes de détection d'intrusion soit par la construction de modèle précis à partir de l'historique des attaques perpétuées dans le passé en utilisant des techniques d'apprentissage supervisé ou par l'identification des activités malveillantes en utilisant des techniques d'apprentissage non supervisé. Dans le cadre de cette thèse nous nous intéressons à la détection d'intrusion comme approche pour faire face aux différentes activités malveillantes pouvant corrompre la sécurité des systèmes informatiques et plus particulièrement nous mettrons l'accent sur le rôle de la fouille des données dans la promotion et le développement des systèmes de détection d'intrusion.

Mots Clés : Détection d'intrusion, Systèmes de détection d'intrusion, Datamining, Classification, Clustering.

Abstract

As consequence of the widespread use of Internet and computer networks experienced during the last decade, the computer systems of most firms and organisations become target of many increasingly sophisticated attacks. Research and development efforts conducted in computer security field have provided a considerable number of tools and mechanisms to prevent different types of threats and attacks. Typical examples of these tools are authentication mechanisms, access control, cryptographic protocols and firewalls. However these preventive tools suffer from a considerable number of inevitable vulnerabilities are not able to effectively ward off a various attacks that are increasingly sophisticated, diversified and adapted to exploit system's weaknesses often caused by careless design and implementation flaws. This created the need for a new security component that monitors users and applications activities in order to detect eventual misuses and/or anomalies in a computer network. in fact, this was the pretension of a new reactive approach called "intrusion detection" introduced by J. P. Andersson in 1980.

Actually, Intrusion detection is a complementary tool to conventional security mechanisms and, as its name suggests, consists to identify abnormal or suspicious activities in network computers. Bace & Mell defined it as the process of monitoring and analysing events occurring in a computer system in order to detect evidence of a security problem. So, intrusion detection provides a knowledge on successful intrusion attempts as well as those who have failed through the analysis of different security audit files. These audit files, generated either by applications, peripheral networks or operating systems, become increasingly bulky sight the increased use of the computer in particular within the networks whose flow does not cease to increasing. Thus, the integration of data mining methods in intrusion detection seems to be the most natural solution for exploring this large data sets in order to extract features, relationship and/or rules to detect attacks in an opportune moment. Indeed, the use of data mining techniques in computer security has attracted over the last three decades, considerable interest from community of researchers and professionals from computer sciences and data mining.

Data mining contributes to the improvement of the intrusion detection systems performances either by the construction of precise model starting from the history of the attacks perpetuated in the past using techniques of supervised learning or by the identification of malicious activities using unsupervised learning techniques. In this Thesis, we are interested in intrusion detection as approach to deal with malicious activities able to corrupt the computer network security and more particularly we will stress the role of the data mining in intrusion detection systems development and promotion.

Keywords: Intrusion detection, Intrusion detection systems, Data mining, Classification, clustering.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES ALGORITHMES

PRÉAMBULE

Motivations

Initialement l'Internet et sa gamme de protocoles étaient conçus pour assurer une communication adaptée et un échange aisé et sûr de données entre des communautés de recherches. A cette époque, l'environnement qu'offrait cette nouvelle technologie était comparable à un environnement collégial "peuplé"[332] par des personnes dotées de bonnes intentions, se connaissant et se faisant confiance. L'échange libre et ouvert d'informations était leur but commun. Mais avec l'émergence de l'utilisation des réseaux TCP/IP, cette communauté créatrice de ce monde "merveilleux" a été rejointe par d'autres personnes avec des éthiques et comportements différents. Parmi les utilisateurs on ne comptait plus que des utilisateurs sérieux, mais également des utilisateurs malveillants qui n'hésitaient pas, dans certaines situations, à raser un site ou détruire physiquement une station pour effacer leurs traces.

La protection des données ainsi que leurs supports de stockage et de transmission devient plus que nécessaire. Elle est vitale pour certaines entreprises et organisations. Il est connu que si une entreprise arrive à perdre 60% de son système d'information, elle a de forte chance de disparaître dans les six mois qui suivront l'incident. Actuellement, des millions de citoyens ordinaires utilisent les réseaux dans leur vie quotidienne pour des opérations aussi ordinaires que banales telles que : les opérations ou transactions bancaires, des achats, des paiements de factures d'électricité, d'eau et de gaz, des envoies de courriers etc. ; ce qui fait que des détails personnels peuvent être facilement acquis et utilisés pour nuire à l'existence du propriétaire. La sécurité informatique est devenue une partie importante dans la sécurité physique et personnelle des individus et des entreprises.

Il n'est pas surprenant de constater que le nombre d'incidents de sécurité continue de monter(Fig. 1), causant ainsi des pertes financières de plus en plus importantes(Fig.2). Le sondage Global Security de 2014 rapport que le nombre d'incidents détectés a atteint un total de 42,8 millions, enregistrant ainsi un bond de 48% par rapport à 2013 et causant 34% de perte financières de plus par rapport à 2013. "Les menaces prennent des formes de plus en plus diverses : virus, vers informatiques, programmes malicieux, sabotage et usurpation d'identité", synthétise Ted DeZabala[281], un porte-parole de Deloitte & Touche [1].

La sécurité a affaire à des adversaires intelligents, prédestinés et parfois bien placés. Ainsi rendre un réseau plus sécurisé implique beaucoup plus de réflexion que de le maintenir exempt des erreurs de programmation[332]. La sécurité est un processus continu dans l'espace et le temps et nécessite beaucoup de savoir, de savoir faire et de savoir être.

La sécurité informatique tente d'assurer un contrôle d'accès efficace, au différentes ressources d'un réseau ou système informatique d'une part, et une transmission sûre des données d'autre part. Il est à distinguer entre service de sécurité et mécanisme de sécurité[332] :

1. Un service de sécurité est la performance d'un ensemble de fonctions et d'actions fournissant une qualité ou un avantage particulier pour une entité(utilisateur ou client) comme

1. Un des leaders mondiaux de l'audit et des services professionnels. www.deloitte.com

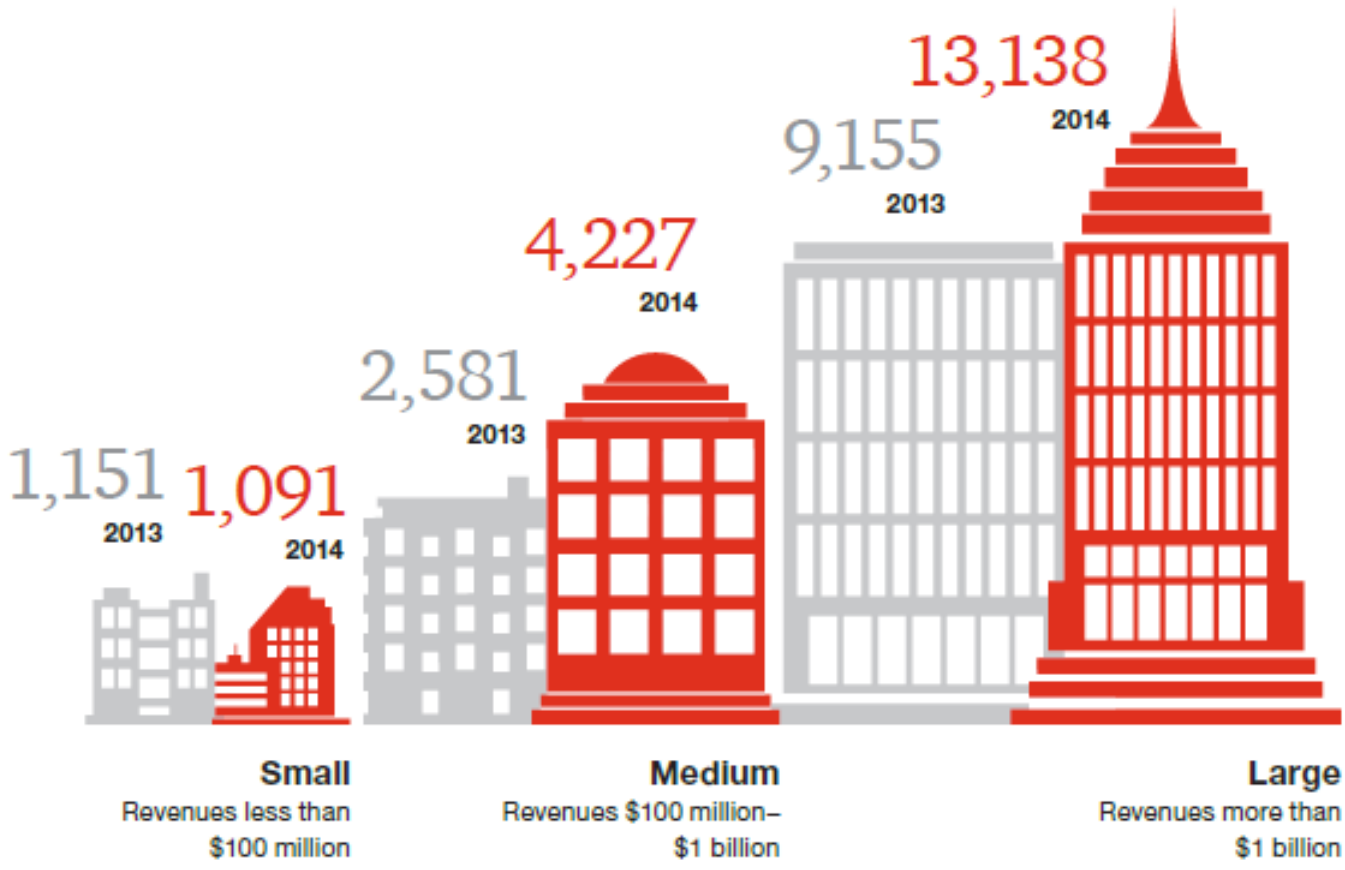


FIGURE 1 – Nombre d'incidents enregistrés 2013-2014

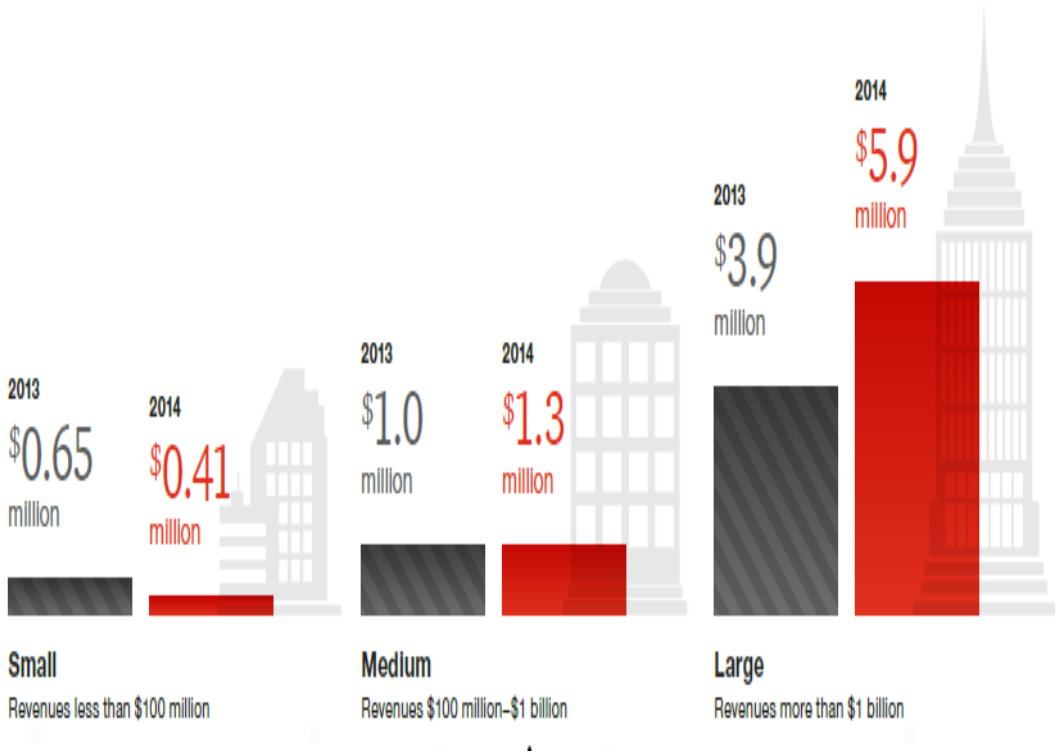


FIGURE 2 – Perte financière moyenne 2013-2014

spécifier par la police de sécurité.

2. Un mécanisme de sécurité peut être utiliser pour assurer ou fournir un (ou plusieurs) service de sécurité.

Par exemple, l'authentification d'utilisateur, comme service de sécurité, peut être implémenté avec des mots de passe ou par biométrie. Il existe plusieurs algorithmes de cryptage qui peuvent être utilisés pour assurer la confidentialité des données. Dans les deux cas(Service et mécanisme de sécurité), on doit distinguer entre spécification et implémentation. La spécification identifie ce qu'on a besoin, alors que l'implémentation le fournisse.

Sous sa forme la plus simple, la sécurité consiste à s'assurer que des fouineurs ne peuvent ni lire ou dans le pire des cas, ni modifier des informations ou messages destinés à d'autres et à interdire à des personnes non autorisées d'accéder à des services : ou plus clairement, elle peut être définie comme étant l'ensemble de méthodes techniques et outils chargés de protéger les ressources d'un système informatique afin d'assurer, selon le modèle de référence d'**OSI** [2][189, 190] les cinq services suivant :

1. **La confidentialité** première préoccupation des militaires, semble être la qualité la plus importante d'un système sûr. Elle consiste à s'assurer que l'information privée ou confidentielle ne soit pas interceptée, visualisée ou copié par des utilisateurs non autorisés[389]. Un autre aspect de la confidentialité est la protection du flot de trafic contre l'analyse. Cela requiert qu'un attaquant ne puisse observer ou relever, sur un équipement de communication, les caractéristiques d'un trafic, tel que : les sources et destinations, les fréquences, longueurs ou autres.
2. **L'authentification** est le mécanisme permettant de certifier qu'un sujet (utilisateur ou un programme s'exécutant pour le compte d'un utilisateur) est bien ce qu'il prétend être[389]. Ce mécanisme est essentiel pour permettre la définition des droits d'accès des différents sujets et leur mise en œuvre. Typiquement, l'authentification des utilisateurs est effectuée au cours du processus d'ouverture de session quand un utilisateur envoie les informations sur son identité(généralement un nom d'utilisateur et un mot de passe).
3. **L'intégrité**, dite aussi l'authentification de données, évite la corruption, l'altération et la destruction des données dans le réseau de manière non autorisée. La corruption de l'intégrité des données peut avoir plusieurs sources[389] :
   — Les bogues logiciels ou actions malveillantes de la par des utilisateurs.
   — L'infection virale du système informatique, les chevaux de Troie.
   — Les panes matérielles causées par l'usure, des accidents ou catastrophes naturelles
   — Les erreurs de saisie, de stockage ou de transmission réseau.
   A fin de minimiser ces menaces d'intégrité, les procédures suivantes doivent être implémentées[389] :
   — La sauvegarde régulière des données importantes dans des endroits sûr.
   — L'utilisation des listes d'accès pour contrôler les autorisations d'accès aux données.
   — Maintenance curative et préventive du matériel.
   — L'utilisation des signatures numériques pour s'assurer que les données n'ont pas étaient altérées pendant leur transmission et stockage.
   — L'ajout de code dans les applications pour valider leurs entrées.
4. **La non répudiation** traite des signatures et permet d'éliminer le risque qu'un émetteur ou récepteur puisse nier avoir envoyé ou reçu un message alors que réellement cela été le cas. A titre d'exemple, lors d'une transaction commerciale électronique, le service de la non répudiation oblige le client à ne pas démentir le fait qu'il a adressé une requête d'achat au fournisseur. Et au même titre oblige le fournisseur à ne pas démentir le fait

2. Open Systems Interconnection

que le client lui a adressé une requête d'achat. On distinguera alors un service de non répudiation à l'origine et un service de non répudiation à la délivrance

5. **Le contrôle d'accès**, comme service de sécurité, offre des outils permettant d'empêcher l'usage non autorisé des ressources. Ce service est communément implémenté en utilisant les listes de contrôle (Control lists ACLs) spécifiant une liste de protections appliquées à un objet tel qu'un fichier, un répertoire ou un processus. Comme il peut être géré grâce à des gestionnaire de police de sécurité tel que : Cisco Secure Policy Manager (CSPM) sur Cisco firewalls, virtual private network (VPN) gate-ways et les systèmes de détection d'intrusion, ou Group Policy sur les plateformes Microsoft Windows[389].

Il convient de noter, que certains de ces services de sécurité se chevauchent et peuvent être mutuellement exclusifs, à titre d'exemple une confidentialité trop forte entraîne une perte de disponibilité. Aussi ces cinq services de sécurité son complétés par d'autres mesures tel que :

1. **La disponibilité** des systèmes et des données dans le cadre de l'usage prévu. est une exigence destinée à assurer que le système fonctionne correctement et que le service n'est pas refusé aux utilisateurs autorisés. Elle consiste à protéger le réseau contre les attaques intentionnelles ou accidentelles, tel que l'effacement non autorisé de données ou tout autre déni de service ou d'accès aux données d'une part et d'empêcher l'utilisation du système ou des données à des fins non autorisées.
2. **Le Confinement**, ce principe complémentaire à la confidentialité s'inscrit dans le même but du bon usage des informations. Le confinement garantit qu'un sujet n'arrive pas à divulguer volontairement le contenu des objets auxquels il a accès à quelqu'un qui n'a pas le droit d'y accéder.
3. **Le secret du flux** empêche tout utilisateur non autorisé d'avoir la possibilité d'analyser les flux des données à travers le réseau. Tout accès illégal, même en lecture, à un flux de données permet à l'utilisateur de déduire des informations utiles et qui peuvent, ultérieurement, servir ses intentions malveillantes. La taille des messages échangés, leurs sources et leurs destinations, ainsi que la fréquence des communications entre les utilisateurs sont des exemples de données à préserver pour prévenir le secret des flux dans le réseau et le rendre plus sûr.

De plus, le modèle de référence d'**OSI** définie un ensemble de mécanismes de sécurité pour implémenter les services de sécurité mentionnés ci-haut :

1. Chiffrement ;
2. mécanismes de signature numérique ;
3. mécanismes de contrôle d'accès
4. Mécanismes d'intégrité des données ;
5. Mécanismes d'échange d'authentification ;
6. Mécanismes de contrôle de flux ;
7. Mécanismes de contrôle de routage
8. Mécanismes de Notarisation.

En complément de ces mécanismes, spécifiques, de sécurité, le modèle d'**OSI** énumère, également, les cinq mécanismes pervasifs suivants :

1. Fonctionnalités éprouvé (Trusted functionality) ;
2. Étiquettes de sécurité (Security labels) ;
3. Détection d'événements.

4. Audit de sécurité.
5. Mécanisme de récupération(recovery).

La mise en œuvre de la sécurité implique, selon [230] l'emploi d'un ensemble d'actions qui peuvent être implémentées séparément ou combinées et se repartissent en trois classes de mesures :

La prévention : Elle vise à réduire la probabilité d'apparition d'un incident de sécurité. Elle englobe un ensemble de mécanismes mis en place à l'intérieur ou à l'extérieur du système. Ces mécanismes consistent à concevoir, implémenter et configurer le système assez correctement pour que les attaques ne puissent pas avoir lieu, ou du moins, soient sévèrement gênées par l'utilisation des techniques classiques de sécurité. De ces techniques nous nous en énumérons l'identification, l'authentification, le contrôle d'accès physique et logique ainsi que des techniques cryptographiques. L'administrateur réseaux doit recourir aussi à des outils de détection de failles telles la vulnérabilité des mots de passe, attributions de permissions et d'autorisations risquées (nœuds fantômes sur le réseau, installation non contrôlée de logiciels, etc.).

L'évitement : On l'appelle aussi avortement. C'est une mesure anti-attaque qui consiste à rendre l'information circulant dans le réseau informatique incompréhensive ou illisible par un pré-traitement(chiffrement) à l'origine avant sa transmission sur les supports de communication. Ainsi, toute erreur ou modification peut être détectée au niveau du récepteur lors du processus de reconstruction de l'information(déchiffrement)(Fig.3).

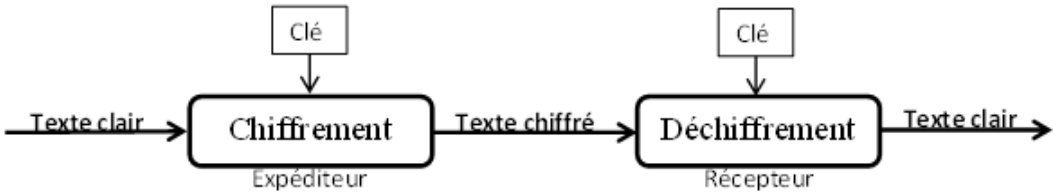


FIGURE 3 – Chiffrement et déchiffrement de données

La détection d'attaques : Cette classe de mesures s'intéresse à la recherche d'éléments indiquant que le système sous surveillance est, ou a été, victime d'une activité malveillante. Un mécanisme de détection d'attaque doit signaler toute action suspecte ou comportement anormal soit en temps réel, cas idéal, soit, en différé, générer des rapports et des alertes post-exploit. Il doit lancer, également, des procédures de récupération et de remise en fonctionnement du système après un crash. La détection d'attaque couvre un ensemble de techniques telles celles de défection, de préemption, de dissuasion ou encore de détection d'intrusion. Cette dernière semble être le moyen le plus prometteur de cette classe.

En effet, la détection d'intrusion est devenue un élément indispensable de toute architecture de sécurité informatique[302] et a suscité un grand intérêt de la part des chercheurs et des développeurs en matière de sécurité informatique. Ainsi de nombreuses approches et techniques (voir à titre d'exemples [42, 94, 129, 357, 379]) ont été proposées pour "construire" des systèmes de détection d'intrusion.

Ces technique sont issues de plusieurs disciplines scientifiques telles la statistique, l'intelligence artificielle, l'apprentissage automatique, ... Récemment, des chercheurs et des fournisseurs ont exploré la possibilité d'utiliser des techniques de datamining dans la détection d'intrusion, dans l'espoir d'améliorer davantage les performances des systèmes de détection d'intrusion. Ainsi, des techniques telles l'analyse multivariée, K-means, réseaux bayésiens naïfs, réseaux de neurones, séparateurs à vaste marge ou *support vecteur machine* (SVM), arbres de décision,

systèmes immunitaires, algorithmes évolutionnaires, ... ont été essentiellement utilisées seules ou combinées entre elles et/ou avec des techniques de réduction de données et de sélection d'attributs telle l'analyse en composant principale [121, 421], l'analyse discriminante linéaire ou l'analyse discriminante générale [369] afin d'éliminer la redondance et les bruits dans les données, d'une part, et réduire le temps de réponse des systèmes de détection d'intrusion, d'autre part.

Il est à noter qu'aucune des approches proposées n'a la prétention d'être la parfaite solution pour le problème de détection d'intrusion mais chacune d'elle présente des avantages et des inconvénients. Cependant, les approches de détection reposant sur l'hybridation de deux ou plusieurs techniques semblent être plus performantes.

Objectifs et contributions

Le principal objectif de cette thèse est, d'une part, de montrer l'intérêt d'employer des techniques et des méthodes issues de la fouille de données dans le but de rendre les systèmes de détection d'intrusion plus performants aussi bien en matière de précision qu'en temps de réponse. Et, d'autre part, de développer une approche de détection d'intrusion à base d'une ou plusieurs techniques de fouille de données. Ainsi dans le cadre de cette thèse nous sommes parvenus à réaliser les tâches suivantes :

— Dresser un état de l'art sur les différentes techniques de la fouille des données ayant été utilisées dans la détection d'intrusion.

— Tester et comparer différents techniques et algorithmes du datamining telles : K-means, K-plus proches voisins, les SVM, les réseaux de neurones, les réseaux bayésiens, etc. L'implémentation de ces techniques a été essentiellement faite sous l'environnement de développement statistique **R**.

— Réaliser un snifer, pour la capture et l'analyse des paquets réseau en utilisant la bibliothèque **libpcap** [3] sous linux.

Les efforts consentis dans le cadre de cette thèse ont aboutis aux production scientifiques suivantes :

— A. Khobzaoui & A.Yousfate, Contribution du Data mining aux performances des Systèmes de détection d'intrusion, CIIA'06 Saida, 2006.

— A. Khobzaoui & A.Yousfate, Data mining approach for Intrusion detection, Jetic'07 Bechar, 2007.

— A. Khobzaoui, M. Mesfioui, A.Yousfate, B. A. Bensaber, On Copulas-based Classification Method for Intrusion Detection, 5th IFIP International Conference on Computer Science and its application(CIIA'2015), Springer International Publishing, 2015.

— A. Khobzaoui & A.Yousfate, Intrusion Detection with Multi-Connected Representation, Inernational Journal of Computer Network and Information Security, Vol8(1), pp. 35-42, 2016.

— Articles soumis :

  A. Khobzaoui & A. Yousfate, A topological test based Classification for Network Intrusion Detection, Malaysian Journal of Computer Science

3. `https://sourceforge.net/projects/libpcap/`

Organisation de la thèse

Le reste de ce manuscrit est organisé, en deux parties, comme suit :

La première partie est constituée des trois premiers chapitres. Le quatrième et le cinquième chapitre constituerons la seconde partie consacrée à la présentation de nous deux principales contribution ayant aboutis à des publications d'envergure internationale.

1. Le premier chapitre sera consacré à la détection d'intrusion dans les réseaux informatiques. On commencera par définir en quoi consiste cette activité, quels sont ses outils et ces techniques ainsi que les domaines impliqués. Comme on présentera ses qualités et ses limites.

2. Le deuxième chapitre sera consacré à l'introduction du datamining, on y présentera l'architecture globale de son processus, sa méthodologie et ses tâches ainsi que ses principales algorithmes.

3. Le troisième chapitre, sera consacrée à la contribution du datamining dans l'amélioration des systèmes de détection d'intrusion. On y présentera :

   — Les besoins architecturaux.

   — Les différents défis.

   — Le processus de datamining appliqué à la détection d'intrusion.

   — Un état de l'art sur les différentes techniques de la fouille des données ayant été utilisées dans la détection d'intrusion clôtura ce chapitre.

4. Le quatrième présentera notre première contribution[216], ayant consisté à utiliser les copules empiriques comme outil de modélisation de la structure de l'éventuelle dépendance dans un classifieur probabiliste supervisé de trafic réseaux. Cette approche, permet d'atténuer la malédiction de la dimensionnalité et de traiter les données dans toutes les situations même si la variance n'existe pas. Comme elle considère les éventuelles dépendances non linéaires entre les attributs décrivant les ensembles de données utilisés pour l'apprentissage et le test du classifieur ainsi construit.

5. Le Cinquième chapitre, sera consacré à la présentation d'une deuxième contribution[217] basée sur la constatation, faite lors de nos différentes expérimentations sur différentes techniques de classification et de clustering, suivante :" La majorité des approches de détection d'intrusion ayant été proposées dans la littérature considèrent que les comportements aussi bien normal aussi bien qu'intrusifs représentés dans un espace topologique sont implicitement connexes. Cette hypothèse n'est pas évidente, une simple projection 2D, issue d'une analyse aux composantes principales sur l'ensemble de données de DARPA[91], nous a révélé que certaines représentations peuvent être non connexes". Ce qui a entrainé des imperfections dans les systèmes de détection d'abus d'utilisation et d'anomalie conçus sous cette hypothèse de connexité.

6. La conclusion générale et les perspectives clôtureront cette thèse.

Première partie

Contexte

CHAPITRE 1

DÉTECTION D'INTRUSION

1.1 Introduction

Nous appelons **intrusion** toute violation de la sécurité logique d'un système informatique[279]. Ces tentatives de subversion s'appuient sur divers types de faiblesses(Fig.1.1) pouvant être classifiées en quatre catégories[250] :

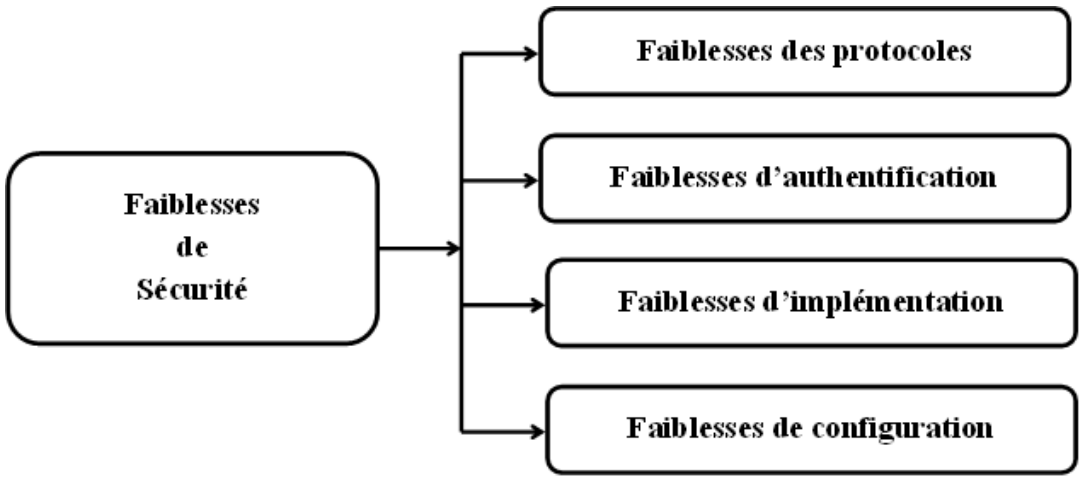


FIGURE 1.1 – Typologie des faiblesses de sécurité[250]

1. **Faiblesses des protocoles :** Les protocoles réseau n'ont pas été, initialement, conçus pour tenir compte des problèmes de sécurité. Ni le protocole IP ni SNMP, par exemples, ne comportent de couche sécurité et s'expose à diverses attaques, tel que les attaques par fragmentation, déni de service,... D'autres formes d'attaques exploitent des bogues ou de mauvaises implémentations des piles TCP/IP dans les systèmes réseau.
2. **Faiblesses d'authentification :** Les protocoles réseau n'ont prévu aucun mécanisme d'authentification véritable et subissent des attaques qui s'appuient sur ces faiblesses d'authentification comme les attaques de type spoofing. Généralement, les pirates tentent de s'infiltrer dans un réseau informatique d'une itératives en utilisant des comptes génériques, standardisés tels admin, toor, sybase, solaris, linux, etc., associés à des mots de passe identiques au nom du compte. Quant aux mots de passe des constructeurs, il suffit de se rendre sur le site `http://www.google.fr` et de rechercher "default password" pour se faire une idée du laxisme ambiant[250].

3. **Faiblesses d'implémentation ou bogues :** Les faiblesses d'implémentation ou des bogues des programmes (systèmes d'exploitation, application de routage,...) exposent les réseaux à de nombreux types d'attaques très sophistiquées tel que les attaques de type SYN flooding et ping-of-death.

4. **Mauvaises configuration :** Une mauvaise configuration des équipements et logiciels de gestion ou d'administration réseau est à l'origine de plusieurs attaques. Par exemple un firewall mal configuré laisse passer du trafic non autorisé par la politique de sécurité. La configuration des équipements réseau est critique et doit suivre des règles strictes d'implémentation afin d'éviter que le réseau ne soit compromis.

On peut lutter contre de telles attaques de deux façons :

1. La première, consiste à construire des systèmes complètement sécurisés, on peut exiger, par exemple, des utilisateur de s'identifier et s'authentifier, comme nous avons la possibilité de protéger les données par des méthodes cryptographiques et des mécanismes très serrés de contrôle d'accès. Mais en réalité cela n'est pas possible pour les raisons suivantes :

   — En pratique, il n'est pas possible de construire un système complètement sécurisé. Miller[285] présente un rapport contraignant sur des bogues dans les programmes de grande diffusion et les logiciels d'exploitation qui semble indiquer, d'une part, qu'un logiciel libre de bogues est toujours un rêve et personne ne semble vouloir faire l'effort d'essayer de développer un tel logiciel d'autre part.

   — Le traitement de toute transaction venant à un système sécurisé (s'il existe) sera trop lent.

   — les méthodes cryptographiques ont leurs propres problèmes. Les mots de passe peuvent être crackés, les utilisateurs peuvent perdre leurs mots de passe, et des cryptosystemes entiers peuvent être cassés.

   — Même un système véritablement sécurisé est vulnérable aux attaques pouvant être menées par ses utilisateurs légitimes.

   — L'efficacité des systèmes est inversement proportionnelle à la dureté des mécanisme de contrôle d'accès.

   Puisque, ce n'est pas demain qu'on va se débarrasser des systèmes ayant des vulnérabilités, on aimerais bien être en mesure de détecter aussitôt que possible (de préférence en temps réel) les différentes attaques qui peuvent surgir et de prendre une mesure appropriée au moment opportun. C'est essentiellement le rôle d'un système de détection d'intrusion.

2. Une autre manière de détecter les intrusions consiste à "fouiller" manuellement dans les fichiers d'audit produits par les systèmes d'exploitation à fin de trouver des traces d'éventuelles attaques. Or la taille astronomique de ces fichiers rend quasiment impossible une telle analyse. De plus, ces fichiers sont la première cible de toute attaque puis que l'intrus termine son exploit par une phase de nettoyage qui, généralement, consiste à détruire tous simplement ces fichiers. Là aussi, les systèmes de détection d'intrusion s'imposent comme une inévitable solution.

1.2 Principe

Les systèmes informatiques qui n'ont pas été agressés pressentent les caractéristiques suivantes[52] :

1. Les actions effectuées par les utilisateurs et les processus sont, généralement, conformes à un modèle statistiquement prévisible. Un utilisateur qui n'utilise la machine que pour du traitement du texte est peu susceptible d'exécuter une fonction d'entretien système
2. Les actions des utilisateurs et des processus du système ne comporte aucune séquence de commandes susceptibles de compromettre la politique de sécurité du système. En théorie, de tel séquence est exclue. Dans la pratique, seulement, des séquences connues peuvent être détectées
3. Les actions des utilisateurs et processus du système se conforment un à ensemble de spécifications décrivant les actions qu'un processus ou utilisateur pourra exécuter (ou ne pas exécuter).

Denning[105] présume qu'un système sous attaque n'arrive pas à satisfaire au mois une de ces caractéristiques. Pour une bonne compréhension de ces caractéristiques, considérant l'exemple suivant :

Un intrus qui tente à pénétrer un système informatique par une porte arrière (back door), procédera, en premier lieu, à la modification d'un fichier de configuration système. Et s'il arrive à s'introduire en tant qu'utilisateur normale (sans privilège) il doit acquérir des privilèges système lui permettant de modifier ces fichiers. Or un utilisateur normal ne demande pas, habituellement, de privilèges système ce qui contre dit la première caractéristique. Les techniques employées pour acquérir ces privilèges peuvent impliquer des séquences de commandes conçues pour violer la politique de sécurité du système (caractéristique 2). Pour effectuer des modifications sur les fichiers système, un utilisateur doit nécessairement violer les spécifications d'actions permises préétablies (caractéristique 3). Si l'intrus modifie un fichier d'un utilisateur, les processus s'exécutant au compte de cette utilisateur auront un comportement anormale, par exemple ils établissent des connexions avec des sites dont l'accès ne lui été pas possible auparavant ou exécutent des commandes que l'utilisateur n'effectuait pas (caractéristique 1). Ces commandes peuvent violer la police de sécurité en gagnant des privilèges systèmes (caractéristique 2).

L'exploitation d'une vulnérabilité d'un système informatique nécessite, selon l'hypothèse de Denning, l'utilisation anormale de commandes ou d'instructions. Ainsi, une violation de sécurité peut être détectée en cherchant des anomalies. Denning considéré les anomalies comme étant des déviation d'une activité habituelle (détection d'anomalie), l'exécution des actions qui mènent à l'effraction (détection d'abus), et des actions contradictoires avec les spécifications associées aux programmes privilégiés (détection basée spécification). Aussi elle a définit pour un système de détection d'intrusion les caractéristique suivantes :

1. Le système de détection d'intrusion doit être en mesure de détecter une grande variété d'intrusions qu'elles soient internes ou externes, préalablement connues ou inconnues. Ceci suggère un mécanisme d'apprentissage et/ou d'adaptation aux nouveaux types d'attaques ou à des changement dans l'activité d'un utilisateur.
2. Le système de détection doit détecter les intrusions d'une façon opportune. " opportun " ne signifie pas, forcement, en temps réel. La détection d'une intrusion après une courte durée de son exécution est en générale suffisante. La détection d'une intrusion ayant lieu une année auparavant est inutile.
3. Le système de détection d'intrusion doit présenter l'analyse sous un format simple et facile à comprendre, idéalement, un signal lumineux vert en cas d'absence d'intrusion et

qui devient rouge en cas de détection d'une intrusion. Malheureusement, en pratique, ce n'est pas aussi simple, les systèmes de détection d'intrusion présentent au chargé de la sécurité des données plus complexes et ce dernier détermine quelle action entreprendre. Puisque les mécanismes de détection d'intrusion peuvent surveiller beaucoup de systèmes (pas simplement un), l'interface utilisateur est d'importance critique.

4. Le système de détection d'intrusion doit être précis. Il ne doit pas identifier une action légitimes comme étant une anomalie ou un abus d'utilisation. Un faux positif se produit quand un système de détection d'intrusion rapporte une attaque, alors qu'aucune attaque n'est en cours. Les faux positifs réduisent la confiance en exactitude des résultats et l'effort qui doit être fourni. Cependant, les faux négatifs (se produisant quand un système de détection d'intrusion ne rapporte pas une attaque en cours) sont plus mauvais, parce que le but d'un système de détection d'intrusion est de rapporter des attaques. Ces deux types d'erreurs doivent être, impérativement, réduits au minimum

1.3 Approches de détection d'intrusion

Les méthodes de détection définissent la philosophie sur laquelle l'analyseur est construit. Depuis le rapport séminal de J.P. Anderson[28], plusieurs approches de détection d'intrusion ont été suggérées. Plusieurs schémas de classification de ces approches ont été proposés dans la littérature spécialisée. La plus populaire de ces classifications [98, 100, 45] consiste à classer les approches de détection en deux grand classes : la détection d'anomalie et la détection d'abus d'utilisation. Une classification, qui est considérée ici, également aussi connue que la première, les classe en trois catégories d'approches [52, 317] à savoir : Détection d'abus d'utilisation, détection d'anomalie, détection par spécification. La performance de ces approches est mesurée en terme de : Faux négatif et de faux positif. Le faux positif désigne la situation dans laquelle le système de détection d'intrusion signale une activité normale comme étant une intrusion. Alors que le faux négative décrit le fait qu'une intrusion est reportée comme étant une activité normale. La détection d'abus d'utilisation tente de coder les connaissances sur les intrusions

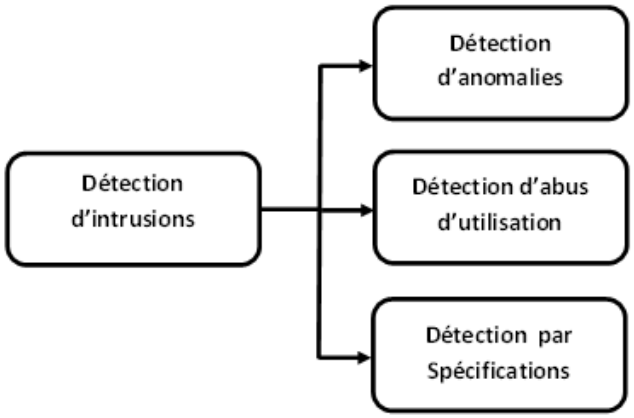


Figure 1.2 – Approches de détection d'intrusion

connues sous forme de signatures spécifiques. La détection d'anomalie et la détection par spécification établissent un modèle à partir de flux de données observés sous les conditions normales sans la présence d'aucune intrusion. Dans la détection par spécification, les experts en matière de sécurité informatique prédéfinissent les différents comportements autorisés du système. Tout événements ne coïncidant pas avec les spécifications est reporté comme intrusion.

1.3.1 Détection d'abus d'utilisation

Est l'approche la plus basique et la plus ancienne. Elle repose sur le concept de bibliothèque de signatures d'attaques et consiste à surveiller (monitoring) le trafic réseau à la recherche des empreintes (signatures) d'attaques connues et répertoriées dans une base de connaissances(signatures) sous forme de règles. Les données d'audit collectées par le système de détection d'intrusion sont comparées avec le contenu de la base de signatures. Si une correspondance est trouvée, une alerte est générée(Fig.1.3). Dans le cas échéant, toute intrusion sera considérée comme une action légitime.

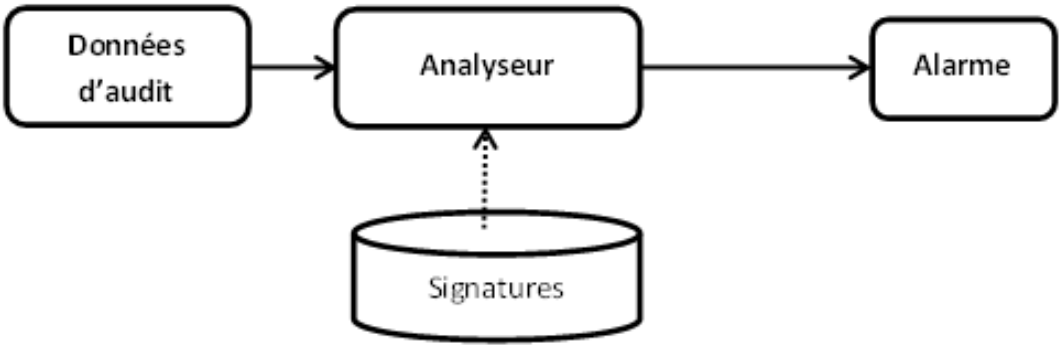


FIGURE 1.3 – Diagramme d'un système de détection d'abus d'utilisation[230].

Les signatures d'attaques sont établies par des experts de sécurité familiarisés avec les vulnérabilités des systèmes et les attaques ou menaces connues. Les systèmes de détection, basés sur cette approche, se caractérisent les uns des autres par la façon de représenter les signatures d'attaques et les mécanismes utilisés pour vérifier les occurrences de ces signatures dans les données d'audit. Ils utilisent, généralement, des systèmes experts pour analyser les données d'audit. Les systèmes de détection d'abus produisent un faible taux de faux positifs[230]. Cela est du au fait que les langages de description d'attaques permettent, habituellement, de modéliser les attaques, à un niveau très fin, de tel façon que seulement quelques activités légales coïncident avec une entrée de la base de signatures. par contre, ils sont incapables de détecter de nouvelles attaques(non décrites dans la base de signatures) ou même des variantes d'attaques connues. MIDAS[357], Wisdom and Sense (W & S) [395], NADIR[191], NIDES[262],furent les premiers systèmes de détection utilisant cette approches.

1.3.2 Détection d'anomalie

Ces modèles "comportementaux" sont apparus bien plus tard que les systèmes à base de signatures. Initialement proposés par JP. ANDERSON[28] puis repris et étendus par D.E. DENNING[105], ces modèles se basent sur l'hypothèse selon laquelle l'exploitation d'une vulnérabilité du système implique un usage anormal de celui-ci. Une intrusion est donc identifiable en tant que déviation par rapport au comportement habituel d'un utilisateur.

La détection d'anomalie suppose que tout comportement inattendu est l'évidence d'une intrusion. Elle analyse un ensemble de caractéristiques du système et compare leur comportement à un ensemble de valeurs prévues. Dans le cas où les statistiques calculées ne concordent pas avec les mesures prévues une tentative d'intrusion est signalée(Fig. 1.4).

Implicite est la croyance qu'un certain ensemble de métrique peut caractériser le comportement prévu d'un utilisateur ou d'un processus. Il est a noter qu'il existe plusieurs approches et techniques pour construire ou décrire un comportement normale de l'utilisateur :

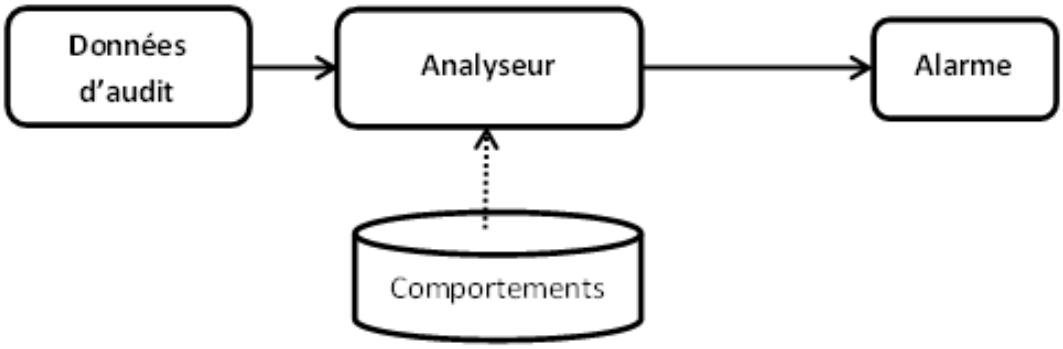


FIGURE 1.4 – Diagramme d'un systéme de détection d'anomalie[230].

— Observation de seuils : Il s'agit de fixer le comportement normal d'un utilisateur par la donnée de seuils à certaines mesures (par exemple, le nombre maximum de mots de passe erronés). Comme il est difficile de caractériser un comportement intrusif en terme de seuil, cette méthode peut entraîner beaucoup de fausses alarmes et d'événements malveillants non détectés.

— Approche bayésienne : les réseaux Bayésiens mettent l'accent sur les relations de causalité existantes. Dans les situations où la connaissance de l'ensemble des relations entre les phénomènes est incomplète, il devient nécessaire de les décrire de manière probabiliste[64]. Les indications obtenues progressivement sur l'état du système modélisé influent sur la confiance que l'on accorde à une hypothèse donnée.

— Profilage d'utilisateurs[53] : On établit des profils individuels du travail des usagers, auxquels ils sont censés adhérer ensuite. Au fur et à mesure que l'utilisateur change ses activités, son profil de travail attendu se met à jour. Il reste cependant difficile de déterminer un profil pour un utilisateur irrégulier ou très dynamique.

— Profilage de groupes : Pour réduire le nombre de profil à gérer, on classe les utilisateurs par groupe. Chaque groupe est caractérisé par genre de travail commun. Un profil de groupe est calculé en fonction de l'historique des activités du groupe entier. On vérifie que les individus du groupe travaillent en conformité et ne dévient pas par rapport à ce qu'a été défini comme profil de groupe. Mais il est parfois pas évident de trouver le groupe le plus approprié à une personne. D'ailleurs, il est parfois nécessaire de créer un groupe pour un seul individu.

— Profilage d'utilisation de ressources : Il s'agit d'observer l'utilisation de certaines ressources comme les processeurs, les ports de communication, les comptes, les applications, les mémoires de masse, la mémoire vive sur de longues périodes, on vérifie et on compare par rapport à ce qui a été observé par le passé. On peut aussi observer les changements dans l'utilisation des protocoles réseau, rechercher les ports qui voient leur trafic augmenter anormalement. L'expérience a montré qu'il est difficile d'interpréter les écarts par rapport au profil normal.

— Profilage de programmes exécutables : Les virus, les chevaux de Troie et autres programmes malveillant peuvent être démasqués profilant la façon dont les objets du système comme les fichiers ou les imprimantes sont utilisés. Donc, le profilage de programmes exécutables stipule qu'on observe l'utilisation des ressources du système par les programmes exécutables. Ce profilage peut se faire par type d'exécutable. On peut par exemple détecter le fait qu'un serveur d'impression se mette à attendre des connexions sur des ports autres que ceux qu'il utilise d'habitude.

— Profilage statistique : Denning a défini un modèle statistique de comportement utilisateur dans [105]. Ce modèle statistique permet de déterminer, au vue de n observations $x_1, ..., x_n$ faites sur une variable x, si la valeur x_{n+1} de l'observation $(n+1)$ est normale ou non. Explicitement, un profil est constitué d'un ensemble de variables représentant une quantité accumulée d'événements (nombre de fois qu'une commande système particulière à été exécutée par un utilisateur, nombre de quantum de temps CPU occupé par un programme, etc.) pendant une certaine période de temps.

— Graphes : Certaines approches comportementales[79] utilisent des modèles à base de graphes pour mettre en évidence des propriétés et des relations entre ces propriétés. L'intérêt de cette approche est qu'elle permet de traiter plus facilement des événements rares. On parle d'apprentissage des comportements normaux dans le cas d'un système informatique ouvert et de spécification des comportements normaux dans le cas d'un système informatique fermé.

1.3.3 Détection par spécification

La détection d'anomalie tente de détecter des états peu connus ou insolites, la détection de malveillance ou d'abus d'utilisation, quant à elle, consiste à identifier les mauvais états résultant de l'exécution d'une séquence d'actions par un utilisateur ou un processus. Or l'approche de détection d'intrusion par spécification tente de déterminer si une séquence d'instructions viole les spécifications décrivant le comportement qu'une application ou le système exploitation devrait avoir[52].

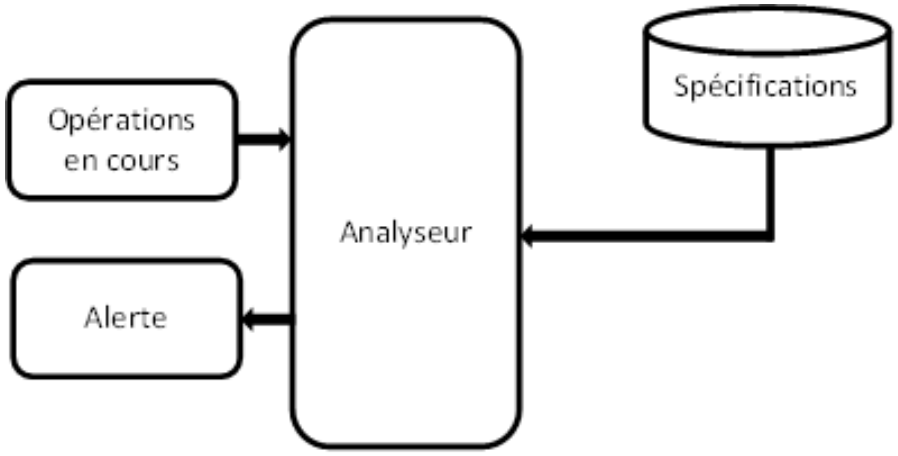


FIGURE 1.5 – Diagramme d'un systéme de détection par spécification.

Ces spécifications, faites manuellement[358] lors d'une phase d'apprentissage, concernent les applications, les protocoles et le système d'exploitation et se présentent sous forme d'un ensembles de contraintes. Comme le développement de spécifications détaillées est une tâche qui s'avère très difficile est consomme beaucoup de temps, seulement les programmes qui sont susceptibles de changer d'une manière quelconque l'état de protection du système nécessitent une spécification et une vérification[52]. Par exemple, l'éditeur de police de windows NT change les configurations relatives à la sécurité donc on doit lui associer des spécifications d'exécution.

Deux modèles de spécification ont été proposés. Le premier consiste à spécifier séparément la police pour le système et pour chaque application. Cette spécification, généralement faite à l'aide d'un langage de spécification de police, établie une liste de tous les appels systèmes autorisés pour une application. Ainsi chaque application se verra attribuer une police indépendante des

autres applications. L'autre crée, pour toute application, un modèle d'exécution en se basant sur le code source ou binaire et toute déviation de ce modèle sera considérée conne une intrusion.

La détection d'intrusion par spécification devrait avoir la précision de la détection d'abus d'utilisation et également la capacité de détecter de nouveaux types d'attaques comme la détection d'anomalie. Cependant, d'une part, la détection par spécifications exigent un bon niveau de compétence technique : en effet, une bonne connaissance des opérations effectuées par les applications est nécessaire car une telle connaissance doit être traduite en spécifications du comportement dans un format compréhensible par le système de détection intrusion.

1.3.4 Comparaison des approches de détection

1.3.4.1 Détection d'abus vs Détection d'anomalies

Considérons dans un premier temps la différence entre la détection d'abus et la détection d'anomalie sur le plan de connaissance, configuration, données générées, exactitude.

1. **Connaissance :** Un système de détection d'intrusion utilisant l'approche de détection des malveillances doit " connaître " toutes les signatures possibles. Il doit identifier les détails d'une attaque aussi bien que son modèle à un niveau d'abstraction élevé qui caractérise la classe de l'attaque. De son côté un système se basant sur les modèles comportementaux doit disposer d'une complète connaissance sur les différents comportements probables du système pour être en mesure de détecter toutes les attaques. En réalité cela n'est pas possible et représente une situation idéale.
2. **Configuration :** En général, un système de détection utilisant une base de signatures exige un effort de configuration moins que celui exigé par un système de détection basé sur les modèles comportementaux. Cependant il nécessité plus de données, d'analyse et de mise à jour. Par contre, les systèmes de détection basés sur les modèles comportementaux sont plus difficiles à configurer par ce qu'il demande une définition compréhensive des comportements connus et probables du système. En général, un support automatique est fourni mais nécessite beaucoup de temps dans son développement et les données qu'il utilise doivent être claires.
3. **Données Générées (reported data) :** Les systèmes de détection d'intrusion utilisant une base de signatures produisent des conclusions basées sur " pattern matching ". Alors que les conclusions des systèmes de détection basés sur les models comportementaux sont basées sur des corrélations statistiques entre les profils actuels et probables.
4. **L'exactitude des signatures :** Les profils, décrivant les comportements, non correctement spécifiés produisent, potentiellement, un nombre élevé de "faux positifs" et de "faux négatifs".

Afin de contourner les inconvénients et de tirer profits des avantages de chacune des approches, certaines systèmes de détection hybrides utilisent une combinaison des modèles comportementaux (détections des anomalies) et de la détections de malveillance.

1.3.4.2 Détection d'abus vs Détection basée spécification

La distinction entre la détection par spécification et la détection d'abus mérite également d'être considérée. La première approche détecte les violations des spécifications par programme, et prétend implicitement que si tous les programmes adhèrent à leurs spécifications, la politique du site ne sera pas violée. La deuxième ne fait pas de telles prétentions, au lieu de cela elle se concentre sur la politique globale du site. Supposez qu'un attaquant pourrait attaquer

un système de telle manière qu'aucun programme n'ait a violé ses spécifications mais l'effet combiné de l'exécution des programmes pendant l'attaque ait violé la politique du site. La détection d'intrusion d'abus pourrait détecter l'attaque (selon la perfection des règles). La détection d'intrusion d'anomalie pourrait également détecter l'attaque (selon la caractérisation du comportement prévu). Cependant, la détection d'intrusion par spécification ne détecterait pas cette attaque. Essentiellement, si les spécifications d'un programme sont sa " politique de sécurité, " la détection par spécification pourrait être vue comme étant une simple forme locale (per-programme) de détection d'abus.

Il convient de signaler l'existence de l'approche hybride suggérée par plusieurs recherches pionniers dans le domaine de la détection d'intrusion[262, 194]. En fait, certains systèmes utilisent une combinaison de l'approche comportementale et de l'approche par scénarios pour remédier aux inconvénients de chacune. Ce type de systèmes utilisent des approches basées sur la détection de signatures pour détecter les attaques connues et des approches basées sur la détection des anomalies afin de détecter des attaques nouvelles ou inconnues.

[387, 106, 312, 432, 187] présentent des travaux typiques plus récents de recherches portant sur les systèmes hybrides de détection d'intrusion. A titre d'exemple, à chaque compte utilisateur sera attribué un profil qui lui permettra d'accéder à certaines ressources sensibles, toutefois il est d'usage de vérifier que des attaques connues n'aient pas pour cible ces ressources. Inversement, utiliser un enregistrement portant le mot "alarme" ne caractérise aucune signature d'attaque, mais il est utile de savoir que cela était déjà arrivé.

1.4 Systèmes de Détection d'intrusion(SDI)

On appelle Systèmes de Détection d'intrusion un mécanisme écoutant le trafic réseau de manière furtive afin de repérer des activités anormales ou suspectes et permettant ainsi d'avoir une action de prévention sur les risques d'intrusion. Un système de détection d'intrusion n'est en aucun cas une mesure de sécurité autonome mais un complément indispensable aux mécanismes de sécurité préventifs.

1.4.1 L'architecture d'un système de détection d'intrusion

A un niveau macroscopique, un système de détection d'intrusion est constitué, essentiellement de trois composantes comme illustrer dans la figure 1.6.

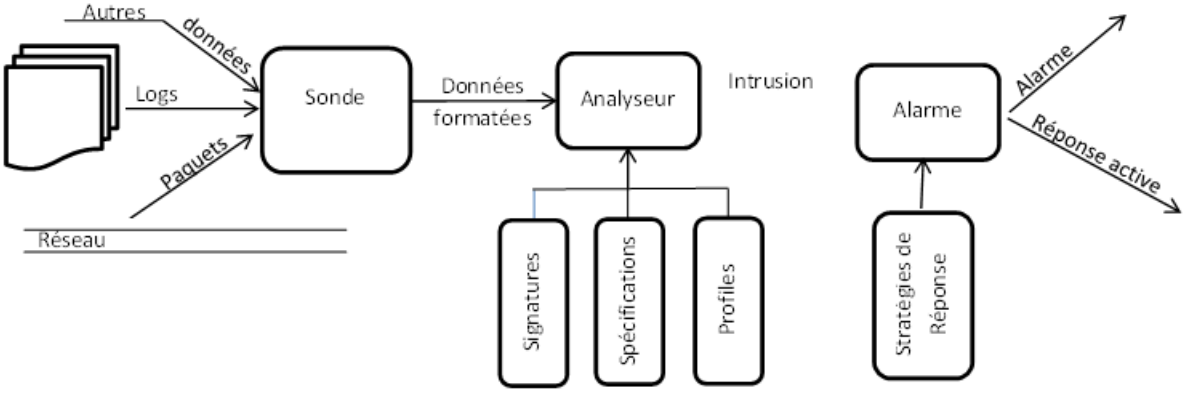


FIGURE 1.6 – Architecture d'un système de détection d'intrusion

— Détecteur (Sonde) : Le détecteur peut être réalisé sous forme de logiciel seulement ou d'une combinaison de Matériel/Logiciel. Il est responsable de la collecte des informations sur l'usage du système et de leur transmission, d'une manière sûre, vers l'analyseur. Comme indiqué sur la figure 1.6, les données collectées peuvent être issues des fichier journaux des systèmes d'exploitation et des applications ou issues des fichiers d'audit réseaux. Il est à signaler que ces informations, qui nécessitent par fois des opérations de filtrage, son stockées sous forme d'enregistrement d'audit dans un format spécial (format du système de détection). Le principal défi d'un système de détection d'intrusion reste la sécurisation de communication des "rapports" d'activité générés par les sondes. Pour cela on a recours généralement à des techniques cryptographiques.

— L'analyseur : L'analyseur, récupère les informations d'audit générées par une ou plusieurs sondes ou d'un autre analyseur. Après une phase de prétraitement qui consiste à réduire la quantité de données collectée en éliminant les données inutiles ou redondantes, il procède à l'analyse de ces données à la recherche des traces d'une éventuelle intrusion (achevée ou en cours). Pour effectuer son analyse, il emploie soit une seule technique ou approche de détection soit une combinaison d'une ou plusieurs approches. L'utilisation d'une combinaison de techniques est l'issue la plus utilisée car elle accentue l'efficacité du système de détection.
L'analyseur peut être implémenté sur le systèmes qu'il surveille ou un système séparé. Un système séparé empêche un attaquant ayant réussi son exploit d'effacer ou de modifier l'information de détection d'une part et d'autre part il emploie peu de ressources du système surveillé.

— Système d'alerte : Le système d'alerte reçoit l'information de l'analyseur, et, en cas d'une intrusion détectée, prend la mesure appropriée. Dans certains cas, il se contente, simplement, d'aviser le chargé de sécurité par un simple message ou un beep. Dans d'autres cas, il peut prendre une certaine mesure pour répondre à l'attaque. La réponse peut prendre plusieurs formes : coupure de la connexion, filtrage de paquets, ordonner le ou les détecteurs de collecter plus de données...

  Le système d'alerte peut, également, disposer d'une interface graphique qui permet de visualiser l'état du système surveillé et de contrôler son comportement.

En plus de ces trois composantes, un système de détection d'intrusion peut éventuellement contenir un "pot de miel" qui n'est autre qu'un sous-système conçu et configuré pour qu'il soit visible et accessible par un intrus. Lors de toute attaque détectée, l'intrus est dirigé vers ce sous-système afin de récolter suffisamment d'informations pour le contourner. Un système de détection d'intrusion dispose, aussi, d'une ou plusieurs bases de données et/ou de connaissances. Ces bases contiennent les informations de configuration et les règles relatives aux modèles de détections dictant le comportement du système de détection lors de son fonctionnement.

1.4.2 Taxonomie des systèmes de détection d'intrusion

Suit au premier modèle statistique proposé par Denning[105], plusieurs systèmes de détection d'intrusion ont été proposés. Cette diversité a donnée naissance à plusieurs schémas de classification pour ces systèmes selon divers critères. Selon [98] les critères suivantes caractériserait au mieux un système de détection d'intrusion (Fig. 1.7) :

1. Les méthodes ou approches de détection (section 1.3) définissent la philosophie sur laquelle l'analyseur est construit. Selon ce critère les systèmes de détection peuvent être répertoriés en trois classes : Système à base de :

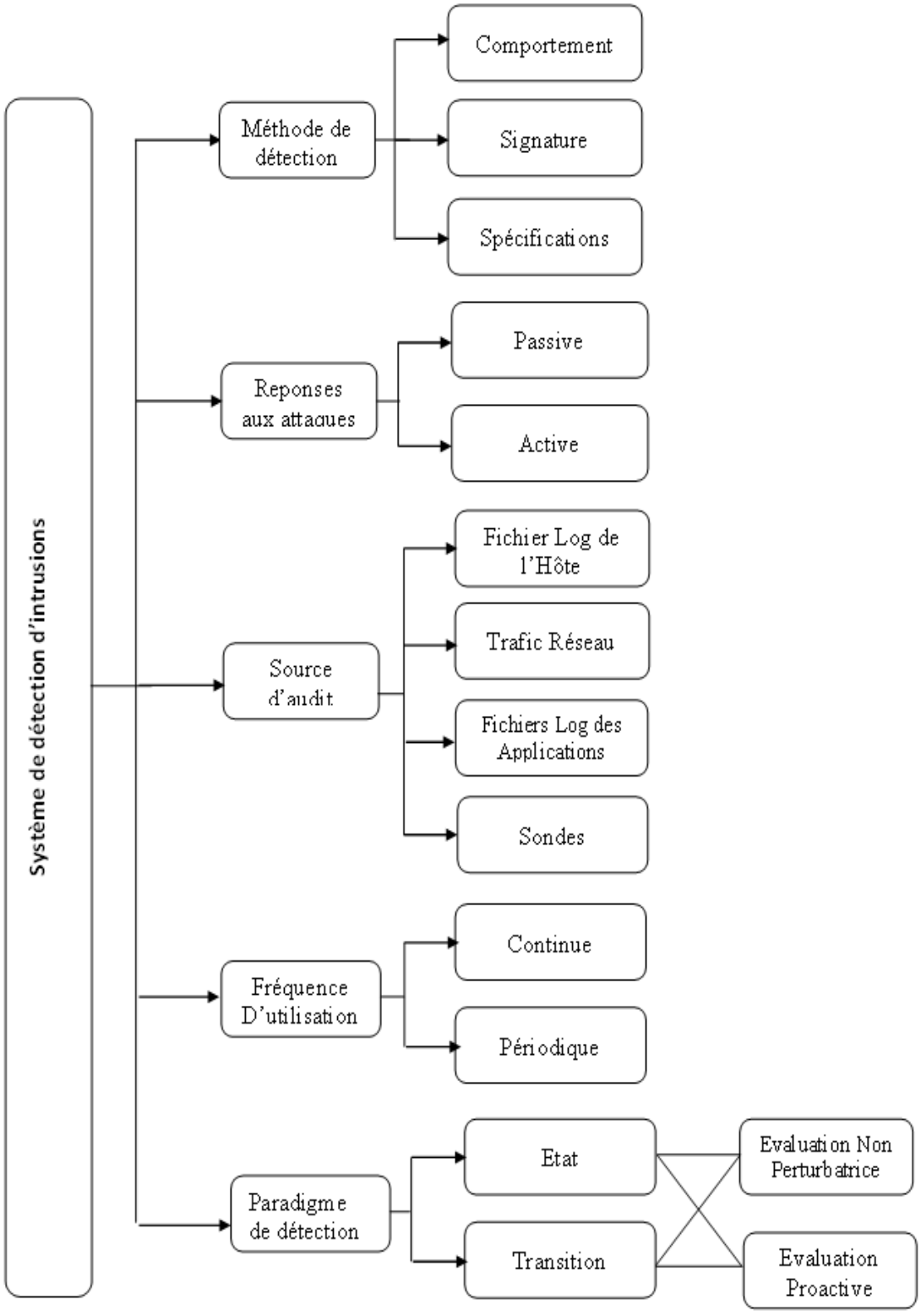


FIGURE 1.7 – Classification des Systèmes de détection d'Intrusion

— Détection d'abus d'utilisation,
— Détection d'anomalies
— De spécifications

2. Selon ses possibilités intrinsèques et sa configuration, un systèmes de détection d'intrusion est dit passif s'il se contente seulement d'émettre une alerte à destination de

l'administrateur quand une intrusion est détectée. Dans le cas ou des mesure pro-actives sont prises, comme par exemple, terminaison d'une connexion, suspension du processus incriminé, reprogrammation du pare-feu afin qu'il bloque le trafic réseau provenant de la source malveillante suspectée, il sera dit actif.

3. Fréquence d'utilisation : Ce critère distingue entre les systèmes qui analysent les données en temps réel et ceux qui le font en mode différé ou périodiquement.
4. Le paradigme de détection décrit le mécanisme de détection employé par le système de détection d'intrusion. Les systèmes de détections utilisent essentiellement deux type de moteurs de détection. Le premier s'appuie sur les informations ou données décrivant les différentes transitions ayant lieu dans le système comme par exemple : l'exécution de certains programmes ou certaines séquences d'instructions, l'arrivée de certains paquets,... Le deuxième évalue l'état de certaines parties du système comme l'intégrité des programmes stockés, les utilisateurs privilégiées. Dans les deux cas la collecte des information et/ou données se fait soit par une interrogation directe du système soit en écoutant passivement les événements. L'évaluation de l'état ou la transition peut être non perturbatrice ou pro-active. L'évaluation non perturbatrice consiste à évaluer les vulnérabilités des versions des applications ou des bannières, puis les comparer avec une des vulnérabilités connues stockées dans une table. si la version de l'application est dans la base, le système est considéré comme étant dans un état vulnérable, sinon il sera marqué comme étant à l'état sécurisé[98].

   L'évaluation pro-active effectue l'analyse en déclenchant explicitement des événements sur l'environnement pour déterminer les états ou créer des transitions[98].
5. La source des données d'audit indique l'endroit duquel les détecteurs (sondes) collectent leurs données. Dans ce cadre on distingue trois types de systèmes de détection d'intrusion(Fig. 1.8) :

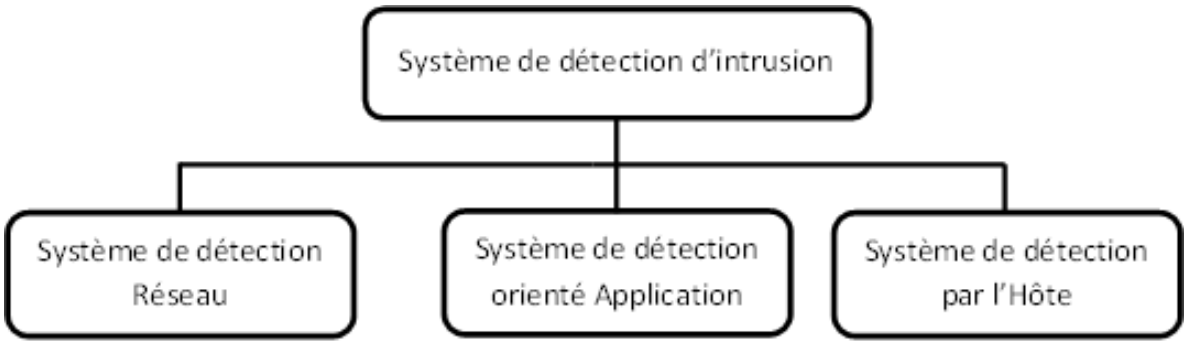


FIGURE 1.8 – Types de Système de détection d'intrusion

a) Les systèmes de détection orientée application : détectent les attaques visant une application spécifique. L'information d'audit nécessaire à la détection d'intrusion est, habituellement, obtenue en utilisant l'appel système "syslog" ou en dotant l'application, elle même, par des mécanismes spécifiques d'audit. Ajouter des mécanismes d'audit à une application existante exige la modification de l'application de sorte qu'elle produise l'information d'audit en réponse aux événements appropriés de sécurité. Ceci peut être accompli par différentes manières :

— En modifiant directement le code source de l'application pour inclure le code d'audit. Cette approche exige que le code d'application soit disponible et modifiable.

— Interposer le code responsable d'extraire l'information d'audit dans les interfaces utilisées par l'application tels les appels systèmes ou les bibliothèques standards (C standards library). Mais avec cette approche, des applications autre que celles surveillées seront affectées en terme de performances (deviendront plus lentes lors de leur exécution).

— Utiliser les "hooks extension" [1] fournies par l'application elle même pour implémenter l'ensemble des fonctionnalités d'audit, cette approche offre plus de flexibilités, mais les application n'offre pas toute cette possibilité.

b) Le système de détection par l'hôte opèrent au niveau d'une machine et analysent les données d'audit générées par son système d'exploitation à la recherche des éventuelles intrusions. Les sources de données d'audit peuvent être :

— **Le système d'information :** Les systèmes d'exploitation rendent disponible, au profit des processus, dans l'espace d'utilisateur des informations relatives à leurs fonctionnements internes et à leur sécurité. Ces informations sont collectées et traitées par des programmes tel que ps, vmstat, netstat. Elles sont, le plus souvent, très complètes et très fiables car elles sont produites par le noyau. Malheureusement, peu de systèmes d'exploitation sont dotées de mécanismes collectant, systématiquement et sans interruption cette information.

— **Service de syslog :** Le syslog est un service d'audit fourni par beaucoup de systèmes d'exploitation de type UNIX. Il permet à des programmeurs d'indiquer un message textuel décrivant un événement à enregistrer. L'information additionnelle, comme le moment où l'événement s'est produit et l'hôte sur lequel le programme fonctionne, est automatiquement ajoutée. En raison de leur simplicité, des événements de syslog sont employés intensivement par des applications. Cependant, les applications enregistrent habituellement des informations utiles pour le débogage qui ne sont pas nécessairement utiles pour la détection d'intrusion. En outre, un format spécifique d'audit n'est pas imposé par le service mais change selon le programme qui les a générées. Ainsi, il peut être difficile d'extraire des données d'audit à partir des log. En conclusion, les fichiers logs peuvent être facilement pollués par des messages crées par l'intrus.

Un système de détection d'intrusion par l'hôte a les avantages suivants :
— Il détecte les tentatives de contourner les systèmes de détection d'intrusion réseaux tel que les attaques par fragmentation des paquets et "time live" attaque.
— Il détecte des attaques masquées avec des techniques de chiffrement
— Il permet de vérifier si une attaque a réussi réellement. (un système de détection réseau détecte l'attaque mais il ne peut pas déterminer si l'attaque a réellement réussie)
— Il ne demande pas de matériel spécialisé

D'un autre coté il doit être installé sur chacun des postes à surveiller et doit être compatible avec plusieurs systèmes d'exploitation.

c) Système de détection réseau s'installent sur le réseau ou sur un segment du réseau et surveillent, en temps réel, le trafic des paquets à la recherche d'une signature d'attaque.
Par conséquent, dans de tels systèmes de détection les détecteurs sont des sniffers réseau. L'analyse du contenu des paquets de réseau peut être effectuée à différents niveaux

1. Une interface fournie sous forme de package permettant l'insertion de code personnalisé, offrant des fonctionnalités additionnelles, dans une application.

de sophistication. Elle peut, par exemple, concerner les entêtes des paquets circulant dans le réseau ou elle inclue la totalité de son contenu, comme elle peut exploiter la connaissance sur les protocoles utilisés lors d'une connexion. Des niveau d'analyse plus élevés supportent des analyses plus sophistiquées mais elles sont très lentes et exigent plus de ressources.

Les systèmes de détection réseaux sont très attrayants parce qu'ils sont facile à déployer et en n'ont presque pas d'impact sur les hôtes surveillés. Un système de détection d'intrusion réseau a deux avantages : le premier avantage est son émission d'alerte en temps réel, ce qui donne aux administrateurs la possibilité d'arrêter ou de contenir une intrusion avant qu'elle ne réussisse. C'est particulièrement valable pour les attaques de type DOS qui doivent être traitées immédiatement pour atténuer les dommages. D'autre part les systèmes de détection d'intrusion réseau sont capables de stocker des informations sur la session, ce qui aide les administrateurs à déterminer les vulnérabilités présentent dans leur système. Le type et nature de l'attaque lancée par un intrus contre un système indique ou détermine les vulnérabilités existant sur le réseau.

Les systèmes de détection d'intrusion réseau son indépendant des systèmes d'exploitation, mais nécessitent un n œud dédié à la surveillance, d'une carte réseau fonctionnant en mode "promiscuous" et une connexion sécurisée entre les différentes sondes et la console maîtresse.

La classification des systèmes de détection d'intrusion en systèmes orientés réseau et à ceux orientés systèmes se rapporte à la manière dont sont collectées les informations par les systèmes de détection et non pas à la façon dont à lieu leurs traitement. Pour garantir cette distinction, nous introduisons le terme collecte de données orientée système ou hôte et collecte de données orientée réseau. Ainsi une autre classification s'impose. Elle consiste à distinguer des systèmes de détection distribués et ceux centralisés. Dans les systèmes de détection d'intrusion distribués, les données sont collectées et traitées au niveau de multiples hôtes contrairement aux systèmes centralisés dont lesquels les données peuvent être collectées d'une manière distribuée mais traitées d'une façon centralisée. En général, on croit que la collecte des données orientée hôte est meilleure que celle orientée réseau pour les raisons suivantes :

— Les techniques, orientée hôte, de collecte de données permettent d'avoir des données reflétant exactement se qui se passe au niveau de l'hôte. Tandis qu'un moniteur réseau pourrait potentiellement manquer des paquets.
— Les mécanismes de collecte de données, orientés réseau, sont, selon Ptacek et Newsham [321], sujets à des attaques d'insertion et d'évasion

1.4.3 Réponse aux incidents

Idéalement, une tentative d'intrusion doit être détectée, identifiée et interrompue avant qu'elle ne réussisse. Ceci implique, typiquement, une étroite surveillance du système et d'agir, de façon manuelle ou automatique, au moment opportun pour défaire l'attaque. Typiquement, la réponse à une attaque consiste en premier lieux à ramener le système à un état de conformité avec sa politique de sécurité initialement définie. Le scénario de réponse à une attaque est le suivant :

En premier lieu et avant qu'une attaque ne soit détectée, des procédures et mécanismes de détection et de réponse doivent être établis. Une fois l'attaque détectée, il faut l'identifier et l'isoler afin de limiter les éventuels dommages. Après vient l'étape d'éradication de l'attaque qui consiste à bloquer l'attaque en cours et prévoir toute future attaque similaire. Après que le

système compromis soit ramené à un état sûr et conforme à la politique de sécurité, les problèmes ou vulnérabilités du système doivent être identifiés et corrigés. Et en fin une poursuite judiciaire ou administrative doit être prise à l'encontre de l'intrus une fois identifié.

L'isolement de l'attaque consiste à Contenir ou confiner une attaque signifie limiter l'accès de l'intrus aux ressources de système et de réduire son champs d'action autant que possible. L'intrus peut être, dans un premier temps, passivement surveillé afin de recueillir des informations sur l'attaque et probablement sur son but. Par exemple, il pourrait être intéressant de connaître le type de système d'exploitation utilisé par l'intrus. Un moniteur passif peut examiner les en-têtes des paquets TCP/'IP entrants au système surveillé et produire une signature pouvant être comparée aux signatures connues des systèmes d'exploitation pour identifier le type du système ayant générer ces paquets. Pour cela l'intrus est généralement dirigé vers une cible séduisante communément appelée pot de miel. Clifford Stoll [2], par exemple, quand il a détecter, pendant l'été 1986, la présence d'un intrus [3] dans le système informatique de "Lawrence Berkeley Laboratory", il s'est contenté de le surveiller pour un moment afin d'identifier son but et de le localiser. Il à réaliser que l'intrus opérait de l'extérieur des Etats Unis et qu'il recherchait des documents concernant l'armement nucléaire. Pour le localiser complètement, les autorités des affaires étrangères ont exigé une connexion assez longue. Pour cela Stoll a créé un fichier volumineux contenant les mots clés que l'intrus a utilisés. En découvrant le fichier, l'intrus croyant qu'il a atteint son but, c'est mis à le télécharger. La durée de téléchargement a été suffisamment longue pour qu'il soit localisé et arrêté

L'éradication d'une attaque signifie arrêter ou bloquer l'attaque. L'approche habituelle consiste à refuser complètement l'accès au système par la terminaison de la connexion ou la suspension des processus incriminés. Un aspect important de l'éradication doit s'assurer que l'attaque ne reprend pas immédiatement. Une méthode commune d'implémenter cette approche consiste à mettre en place un mécanisme de contrôle d'accès, local ou distant, aux ressources système. Ces mécanismes de contrôle d'accès sont habituellement intégrés dans le noyau du système d'exploitation à fin de rendre leur contournement plus difficile. Ils attendent certains évènements pour réagir tel qu'un appel système, probablement avec des configurations ou paramètres privilégiés. Quand l'évènement se produit, ces mécanismes prennent le contrôle et exécutent des actions spécifiées. Ils peuvent se contenter de journaliser l'évènement, de refuser l'accès (en retournant un code d'erreur au processus appelant) ou génèrent et traitent des données auxiliaires tel que le nombre d'appels système.

La poursuite ou la contre attaque prend deux formes. La première implique des mécanismes légaux, tels que des plaintes juridiques, criminelles et civiles. Ceci exige des preuves matérielles pour que les autorités juridiques puissent établir que l'attaque était vraie (en d'autres termes, que le site n'a pas inventé l'incident). Toutefois les exigences des lois changent d'une communauté à une autre avec le temps. La deuxième forme est une attaque technique, dans laquelle le but est d'identifier l'agresseur, de l'endommager assez sérieusement pour arrêter l'attaque courante et pour décourager des futures attaques. Cette approche a plusieurs conséquences importantes qui doivent être considérées :

1. la contre attaque peut nuire d'autre partie innocente. L'attaquant peut personnifier un autre site. Dans ce cas-ci, la contre attaque peut endommager une partie complètement innocente, au lieu des attaquants originaux. Alternativement, les agresseurs auraient dû quitter le site d'où l'attaque a été lancée. Attaquer ce site ne résout pas le problème. Il élimine simplement une base d'où de futures attaques pourraient être lancées.
2. La contre attaque peut avoir des effets secondaires. Par exemple, si elle consiste à inonder

2. Administrateur-système du Lawrence Berkeley Laboratory.
3. Markus Heiss un pirate allemand.

une cible spécifique, l'inondation pourrait bloquer des segments réseau que d'autres parties auront besoin de traverser, ce qui les endommagerait.

3. La contre attaque est antithétique à l'utilisation partagée d'un réseau. La raison d'être des réseaux et le partage des données et des ressources et la fourniture des voies de communication. Indépendamment de la raison, une attaque contre un réseau le rend moins utilisable parce qu'elle absorbe ses ressources. Par conséquent, les sites doivent être protégés en limitant le partage et la communication au delà de ce qui est nécessaire pour leur exploitation sûre.
4. La contre attaque peut être légalement recevable. Et du point de vue juridique il est très raisonnable de réserver à une contre attaque le même traitement réservé à une attaque, particulièrement si d'autres parties innocentes sont endommagées par la contre attaque.

Dans des circonstances exceptionnelles, la contre attaque peut être appropriée pour gêner l'éventuel intrus et lui compliquer la tâche, néanmoins elle s'est révélée peut efficace et trop onéreuses ; de ce fait il est fortement conseillé de l'éviter, le plus possible, aux profits des issues judiciaires (civil ou criminel)

1.4.4 Techniques contre SDI

La plus part des intrus sont conscients de l'existence des systèmes de détection d'intrusion et leur réservent des attaques évasives tel que : l'inondation, la fragmentation, le cryptage et l'obscurcissement.

1. **L'inondation** : L'efficacité des systèmes de détection d'intrusion, à capturer le trafic malveillant, à l'analyser afin de détecter des éventuels attaques, dépend de la disponibilité de la mémoire et de la puissance des processeurs. Ainsi l'attaque par inondation consiste à tenter de priver le système de détection de ces ressources en inondant le réseau par un trafic bruité. Ainsi l'IDS se trouvera entrain d'analyser, inutilement, un grand volume de trafic.
2. **La fragmentation** : La fragmentation est une technique évasive très commune, elle profite du fait que les différents réseaux autorisent un maximum variable d'unités de transition (MTU), et consiste à fragmenter ses paquets sur plusieurs unités qui une fois rassemblées au niveau de l'hôte cible causent une attaque. Pour plus de complexité et d'efficacité, cette technique peut être combinée avec la technique de flooding.

   Pour que les systèmes de détection puissent détecter l'attaque, ils doivent être en mesure de stocker les unités fragmentées pour pouvoir reconstituer le paquet initial. Cela exige d'importantes capacités mémoire et un temps de traitement très considérable.
3. **Le cryptage** : Le cryptage des paquets rend anodins les systèmes de détection d'intrusion. C'es pourquoi les pirates font, le plus souvent appel à des mécanismes de chiffrement, tels que SSL ou SSH pour dissimuler leurs attaques.
4. **L'obscurcissement** : Cette technique est incontestablement, la plus utilisée. Elle consiste à utiliser une codification non standard pour représenter les données dans le but de masquer l'attaque. Cela peut être réalisé soit par des insertion des caractères spéciaux tel que le retour chariot, le caractère de tabulation,... ou l'utilisation d'un uni-code différent du code ASCII et indépendant de tout langage, programme ou plate forme. Par exemple l'intrus peut tromper l' IDS en remplaçant le caractère "/" dans une sollicitation de page web par le caractère uni-code c1. [4]

4. Le fait de dissimuler une attaque utilisant des caractères Unicode, hexadécimal ou de contrôle pour cacher une attaque contre un système de détection d'intrusion est communément appelé **obfuscation**

1.4.5 Interopérabilité des systèmes de détection d'intrusion

La détection d'intrusion a suscité ces dernières années une très grande attention et son déploiement au sein des entreprises, se fait d'une manière très rapide. Pour faire face à cette demande grandissante, une pléthore de produits de détection d'intrusion, en version commerciale aussi bien qu'en "open source" a été lancée (environ 130 ont été répertoriés dans [71] en janvier 2003). Or le problème crucial de ces produits fermés et incompatibles entre eux, réside dans le fait qu'aucun de ces systèmes détection n'est capable, à lui seul, de faire face à toutes les attaques et qu'on ne disposait pas d'un moyen standard pour les faire communiquer et collaborer. Dans ce contexte, plusieurs efforts ont été consentis afin de résoudre ces problèmes et de définir ainsi des outils d'interopérabilité entre systèmes de détection d'intrusion. Parmi les aboutissements de ces efforts, on cite :

Le CIDF(Common Intrusion Detection Framework)[316] fut le résultat d'un projet de recherche entamé, en 1997, par DARPA [5]. L'objectif du CIDF est de permettre, d'une part l'interopérabilité des différents systèmes impliqués dans la surveillance du système informatiques et la ré-utilisation de leurs composants. Cette plate forme est composée de quatre types de composants : Des générateurs d'événements (E-boxes), des analyseurs(A-boxes), une bases de données(D-boxes) et d'un mécanisme de réponse(R-boxes). Les mécanismes de contre-mesures sont également représentés sous forme de C-box(Fig. 1.9).

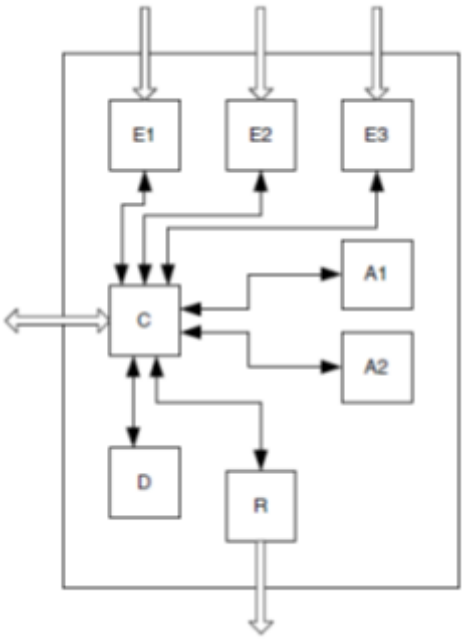


FIGURE 1.9 – Architecture de CIDF

— Le générateur d'événement, constitué d'un ensemble de composants dits E-boxes fonctionnant en tant que "senseurs terminaux", fournit, au système de détection d'intrusion, des informations pertinentes sur le système surveillé.
— L'analyseur, un ensembles de A-boxes explore les données provenant des E-boxes dans le but de détecter une attaque.
— La base de données(système de stockage), ou "D-box", permet de conserver les trace des événements que le moteur d'analyse a considéré comme étant hostiles.

5. Defense Advanced Research Projects Agency

— Le mécanisme de réponse(R-boxes) constitue l'unité de réactions ayant pour objectif de réagir au nom d'autres composants de CIDF. Une réponse peut être, par exemple, : l'arrêt d'un processus, la ré-initialisation des connexions...

Le module de contre-mesures ("C-Box"), présent dans le cas où la détection se fait en temps réel, vise, si cela est possible, à contrer une attaque en cours.

La communication entre ces composants se fait en utilisant des objets dits (GIDOs) "Généralized Intrusion Detection Objects". Ces objets sont représentés via un format standard commun défini à l'aide de CISL [6] un langage appartenant à la famille Lisp. Le projet fut abandonné en 1999 et ce modèle n'a été implanté par aucun produit. Néanmoins, ses idées on été reprises par l' Intrusion Detection Working Group (IDWG) un groupe de travail co-dirigé par ses coordinateurs.

L' IDMEF(Intrusion Detection Message Exchange Format) [99] : a été proposé, en 1999, par le groupe de travail de détection d'intrusion au sein de l'Internet Engineering Task Force (IETF) pour être un format standard d'interopérabilité et d'échange de rapports d'incidents entre les systèmes de détection d'intrusion, de prévention d'intrusion et de collecte d'informations de sécurité et les applications qui doivent interagir avec eux. En effet, grâce à un langage commun, il devenait possible de centraliser les divers messages de détection d'intrusion provenant de multiples sondes en un seul et même module central de traitement des incidents(manager). Ce dernier se retrouve au cœur du fonctionnement des systèmes de détection d'intrusion hybrides tel que **Prelude-IDS** [7] et permet d'enregistrer, corréler et/ou présenter les informations issues des sondes. IDMEF définit deux classes(types) de messages : messages d'alerte et messages Heartbeats.

— Les messages d'alertes sont générés, d'une manière asynchrone, par un analyseur lors de la détection d'un évènement présent dans sa configuration. Un message d'alerte peut être relié, selon l'analyseur, à un simple évènement, ou à plusieurs évènements détectés.

— Un message Heartbeat permet à un analyseur de signaler son état au manager. La réception d'un tel message par le manager permet de vérifier que l'analyseur est bien dans un état de fonctionnement. Cependant, si à une ou plusieurs reprises (selon la configuration), le manager ne reçoit pas de tels messages, alors l'analyseur est supposé défaillant.

D'une manière évidente, il devrait être implémenté au niveau du canal entre la sonde et le moniteur auquel elle envoie les alertes, toutefois d'autres emplacements sont possibles :

— Au niveau du système de gestion de la base de données devant stocker les résultats emmenant de différents systèmes de détection d'intrusion ce qui permettrait une analyse globale de ces résultats au lieu d'une analyse séparée de chaque jeux.

— Au sein du système de corrélation d'événements devant recevoir des alertes de différents systèmes de détection d'intrusion. Ce la rendrait possible une cross-correlation plus sophistiquée d'alertes provenant de plusieurs systèmes de détection au lieu de se contenter d'une simple corrélation limitée aux alertes produites par un seul système de détection.

— Au niveau de l'interface graphique devant afficher les alertes produites par différents systèmes de détection. Un format standard d'échange de données pourra, non seulement,

6. Common Intrusion Specification Language

7. Prelude-IDS (http ://www.prelude-technologies.com), est un système de détection d'intrusion hybride,Distribué sous licence GPL, conçu pour être modulaire, distribué, souple, et résistant aux attaques. Sa modularité permet notamment de lui rajouter facilement de nouveaux types de détecteurs d'intrusion, d'analyseurs de logs et d'un mécanisme de corrélation, le tout au format et à la norme IDMEF, bien que de nombreux autres formats de logs sont compatibles.

faciliter considérablement cette tâche d'affichage, mais aussi permettra la communication d'information sur ces alertes.

En fin, il est a noter que le XML a été retenu par IDWG pour implémenter l'IDMEF vu sa popularité et sa flexibilité mais aussi pour les raisons suivantes :

- Donne la possibilité de définir un langage spécifique pour la détection d'intrusion ainsi que celle d'étendre ce langage pour des révisions ultérieures
- XML est un support substantiellement disponible, ainsi que ses outils de traitement
- Disponibilité de documents et des APIs pour analyser et valider XML pour plusieurs langages tel que Java, C/C++. L'accès répondu à ces outils rend l'adoption de l'IDMEF plus facile, rapide et beaucoup plus conviviale.
- XML projette de devenir un standard reconnu mondialement.
- XML peut supporter le filtrage et l'agrégation une fois associer à XSL
- XML et libre.

1.4.6 Critères de choix d'un SDI :

Aujourd'hui les systèmes de détection d'intrusion sont réellement devenus indispensables lors de la mise en place d'une infrastructure de sécurité opérationnelle. Ils s'intègrent donc toujours dans un contexte et une architecture qui imposent des contraintes pouvant être très diverses. C'est pourquoi il n'existe pas de grille d'évaluation unique pour ce type d'outil. Pourtant un certain nombre de critères peuvent être dégagés ; ceux ci devront nécessairement être pondérés en fonction du contexte de l'étude.

1. **Fiabilité :** Un détecteur d'intrusion doit être fiable ; les alertes qu'il génère doivent être justifiées et aucune intrusion ne doit pouvoir lui échapper. Un système de détection d'intrusion générant trop de fausses alertes sera à coup sûr désactivé par l'administrateur et un autre ne détectant rien sera rapidement considéré comme inutile.
2. **Réactivité :** Un système de détection d'intrusion doit être capable de détecter les nouveaux types d'attaques le plus rapidement possible. Pour cela il doit rester constamment à jour. Des capacités de mise à jour automatique sont pour ainsi dire indispensables.
3. **Facilité de mise en œuvre et adaptabilité :** Un système de détection d'intrusion doit être facile à mettre en œuvre et doit pouvoir surtout s'adapter au contexte dans lequel il doit opérer ; il est inutile d'avoir un système de détection d'intrusion émettant des alertes tous les 10 secondes si les ressources nécessaires à une réaction ne sont pas disponible pour agir dans les mêmes contraintes de temps.
4. **Performance :** la mise en place d'un système de détection d'intrusion ne doit en aucun cas affecter les performance des systèmes surveillés. De plus, il faut toujours avoir la certitude que le système de détection d'intrusion a la capacité de traiter toute l'information à sa disposition (par exemple un système de détection d'intrusion réseau doit être capable de traiter l'ensemble du flux pouvant se présenter à un instant donné sans jamais dropper de paquets) car dans le cas contraire il devient trivial de masquer les attaques en augmentant la quantité d'information.
5. **Multi canal :** Un bon système de détection d'intrusion doit pouvoir utiliser plusieurs canaux d'alerte (email, téléphone, fax...) afin de pouvoir garantir que les alertes seront effectivement émises.
6. **Information :** Le système de détection d'intrusion doit donner un maximum d'information sur l'attaque détectée afin de préparer la réaction.
7. **Classification :** il doit être aisé de hiérarchiser la gravité des attaques détectées afin d'adapter le mode d'alerte.

1.5 Limites et avenir des systèmes de détection d'intrusion

Un système de détection d'intrusion est appelé à faire face à des problèmes d'ordre scalaire ou fonctionnel pouvant affecter directement ses performances(Fig. 1.10).

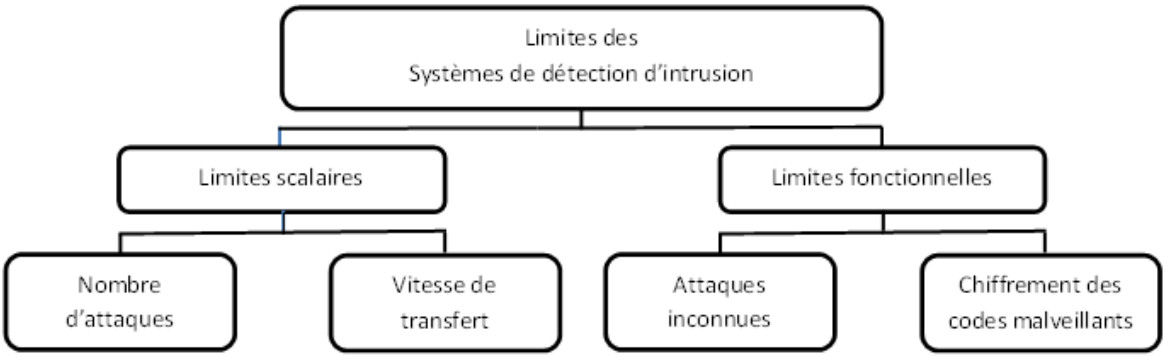


FIGURE 1.10 – Les limites du système de détection d'intrusion

— Les problèmes d'ordre scalaire sont liés aux capacités calculatoires du système de détection et sont disproportionnelles par rapport aux besoins d'analyse exigés par l'environnement où il est installé. Nous citons, par exemple, le nombre des vulnérabilités et des failles qui ne cesse d'augmenter et qui pénalise les capacités d'analyse de système.
— Les problèmes d'ordre fonctionnel sont liés aux mécanismes de détection utilisés par le système et sont souvent engendré par des services mal utilisés ou bien non traités par le système. Par exemple, nous citons l'utilisation des techniques de chiffrement pour cacher des codes malveillants.

Par conséquent, l'amélioration des performances de système de détection est devenue un souci majeur pour la communauté de recherche du domaine. Ces améliorations sont souvent liées aux environnements de travail, tels que les systèmes d'information, les utilisateurs, et les infrastructures technologiques qui vont avec. Les travaux de recherche dans ce domaine ont ciblé dans un premier temps, l'amélioration des mécanismes de détection, qui représentent le noyau du système de détection d'intrusion. Plusieurs algorithmes ont été proposés, ces derniers utilisent plusieurs approches : le pattern matching, les systèmes experts, les automates, les algorithmes génétiques, la fouille de données, les modèles statistiques et l'apprentissage automatique. D'autres travaux ont ciblé l'optimisation des implémentations des systèmes de détection sur des plateformes matérielles. L'idée dans ce type d'approche, est de profiter des puissances de calculs offerts par les dispositifs électroniques dédiés aux calculs intensifs. On trouve des approches basées sur les cartes programmables à base des FPGA(Field-Programmable Gate Array) et des approches qui utilisent les processeurs graphique (GPU :Graphics Processing Unit) ou bien les processeurs réseau.

1.6 Conclusion

La détection d'intrusion dans les réseaux ne vient pas concurrencer les mécanismes de sécurité traditionnels mais, au contraire, les compléter. Même si on ne peut pas atteindre la sécurité absolue, on veut au moins pouvoir détecter l'intrusion afin d'y remédier. Néanmoins comme tous les outils techniques, les systèmes de détection d'intrusion ont des limites que seule une

analyse humaine peut compenser. Un peu comme les Firewalls, les détecteurs d'intrusion deviennent chaque jour meilleurs grâce à l'expérience acquise avec le temps mais ils deviennent aussi de plus en plus sensibles aux erreurs de configuration et de paramétrage. Par conséquent, il est plus que fondamental de former correctement les personnes chargées de la mise en œuvre et de l'exploitation des systèmes de détection d'intrusion. Malheureusement, il semble que c'est encore là où aujourd'hui encore subsiste la plus grande partie de la difficulté.

Cette technologie n'est pas encore arrivée à maturité et les outils existants ne sont pas toujours à la hauteur des besoins. Plusieurs problèmes et défis persistent tel que :

1. La difficulté de définir des frontières entre comportement normal et anormal
2. La difficulté d'explorer et analyser de grandes masses de données générées par les différentes sondes.
3. La nécessité d'une adaptation et mise à jour très fréquentes pour des environnements en évolution continue.
4. La distribution des données très déséquilibrée.

Pour faire face à de tels défis, des techniques issues de la fouille des données ont été utilisées. Cette nouvelle tendance à donné naissance à de nouveaux modèles de détection d'intrusion, dits de deuxième génération, plus performants en sens de précision et de rapidité.

CHAPITRE 2

LA FOUILLE DE DONNÉES

2.1 Introduction

Les sciences et technologies modernes se basent sur le modèle dit "du premier principe" pour décrire les systèmes physiques, biologiques et sociaux. Une telle approche débute avec un modèle scientifique de base, tel que les lois Newtoniennes de mouvement ou les équations de Maxwell en électromagnétisme, puis construisent sur ce modèle une variété d'applications en construction mécanique ou électrique. Dans cette approche, des données expérimentales sont utilisées, d'une part, pour vérifier le modèle du "premier principe" sous-jacent et d'autre part pour estimer quelques paramètres difficiles, ou parfois impossible, à mesurer directement.

Toutefois, dans de nombreux domaines, le premier principe sous-jacent n'est pas connu ou le système sous étude et trop complexe pour qu'il soit modélisé (formalisé) mathématiquement. De tels systèmes génèrent, grâce à l'utilisation des ordinateurs de plus en plus puissants, de grande quantité d'informations pouvant être utilisées, en absence de modèle du premier principe, pour déterminer un modèle en estimant les relations qui peuvent exister entre les variables du système(dépendance entrée/sortie). Actuellement, on assiste à une migration de la modélisation et d'analyse classiques basées sur le modèle du premier principe vers une analyse basée directement sur les données.

Nous nous sommes progressivement habitués au énormes volumes de données qui remplissent nos ordinateurs, nos réseaux et notre vie. Des agences gouvernementales, des institutions scientifiques et commerciales possèdent des énormes ressources de collecte et de stockage de données. En réalité, seulement une petite quantité de ces données sera utilisée, car, dans la plus part des cas, le volume des données est simplement trop important pour qu'il soit gérable ou encore la structure des données est trop complexe pour qu'elle soit analysée efficacement. Ceci est du au fait que l'effort initial consenti pour la construction des base de données(Data Set) était focalisé sur des problèmes de stockage efficace et a, en quelque sorte, négligé la manière dont ses données sont éventuellement utilisées et analysées. La nécessité de comprendre de grandes et complexes bases de données riches en informations est, pratiquement, commune dans tous les domaines scientifiques, technologiques et économiques. Dans le monde des affaires, par exemple, les données relatives au entreprises et aux clients sont reconnues comme étant un capital stratégique. Ainsi la capacité d'extraire la connaissance utile cachée dans ces données et d'agir sur cette connaissance devient de plus en plus importante dans le monde concurrentiel d'aujourd'hui. Le processus entier d'application d'une méthodologie, basée sur l'exploitation des ordinateurs, pour découvrir la connaissance à partir des données est communément connu sou le nom de Data Mining[208].

Le Concept de fouille des données(Datamining) apparaît en 1989 sous un premier nom de "Knowledge discovry in databases", avant qu' apparaisse 1991 pour la première fois le terme de datamining ou "minage/fouille des données ". La littérature spécialisée, fournit différentes définitions pour le datamining et on est loin d'un consensus universel. Certains auteurs le présentent comme étant une simple forme des statistiques enrichie avec la théorie de l'apprentissage alors que d'autres voient qu'il est un nouveau concept et le qualifient de révolutionnaire.

Le datamining trouve ses racines dans une variété de disciplines dont les plus importantes sont les statistiques, les technologies des bases de données, l'apprentissage automatique et la théorie de contrôle(Fig. 2.1)...

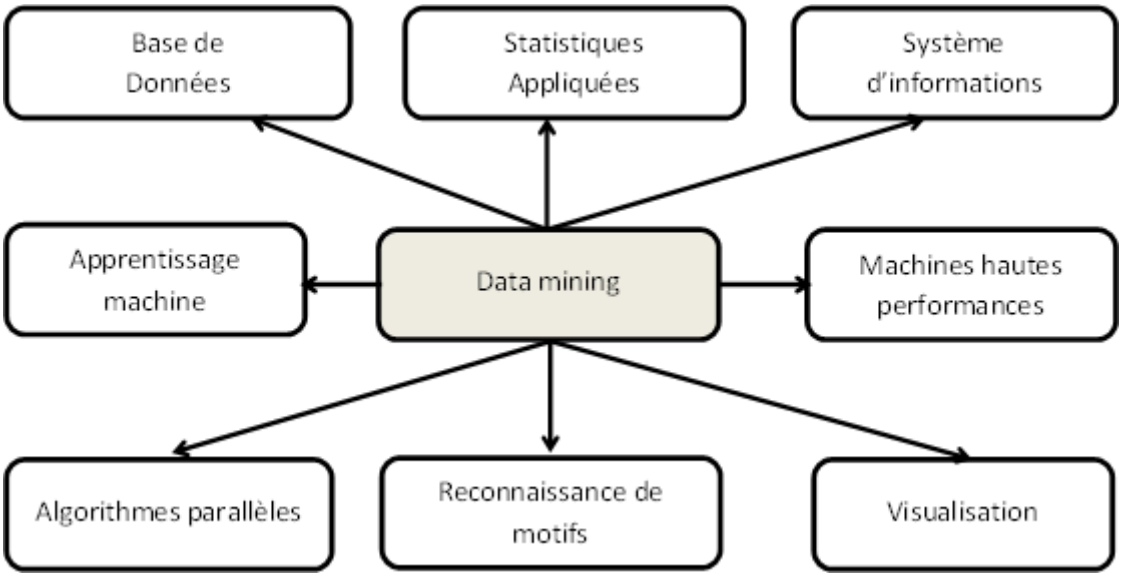


FIGURE 2.1 – Le data mining comme une confluance de multiple diciplines

Selon le groupe Gartner [1], "Le Datamining est le processus de découvrir de nouvelles corrélations, modèles et tendances significatifs par le tamisage de grandes quantités de données stockées dans des entrepôts, en utilisant des technologies d'identification de modèle aussi bien que des techniques statistiques et mathématiques". Quant à Peter Cabena et Al[66] le définissent comme étant "un champ interdisciplinaire regroupant des techniques de l'apprentissage, de l'identification de modèle, des statistiques, des bases de données, et de la visualisation pour aborder la question de l'extraction de l'information à partir de bases de données volumineuses.

De son côté, Ussama Fayyad[134] le définisse comme étant "le processus non trivial d'identification de modèles valides, nouveaux, potentiellement utiles, et compréhensibles dans les données".

Ainsi, le data mining est une démarche ou un assortiment de procédés permettant l'extraction des information ou des relations et des faits, à la fois, nouveaux et significatifs à partir d'une abondante quantité de données.

2.2 Processus du Datamining

La découverte de modelé n'est qu'une étape du processus d'extraction de connaissances à partir des données. En effet le projet **CRoss Industry Standard Process for Data Mining**[413] le défini comme étant un processus composé de six étapes(Fig. 2.2) :

1. The Gartner Group, www.gartner.com.

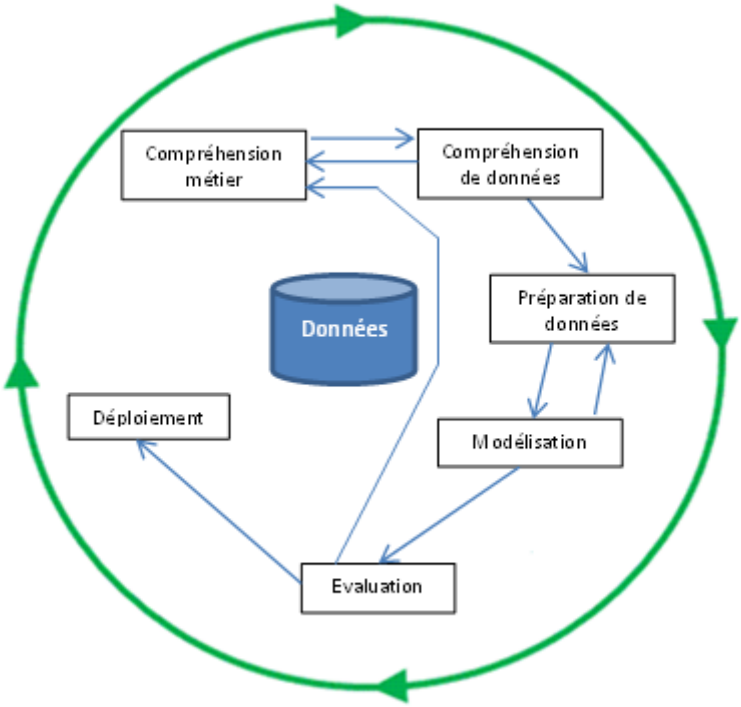


FIGURE 2.2 – Le cycle de vie du CRISP-DM
Le cycle de vie du CRISP-DM[413]

1. **La compréhension du business (Business understanding) :** Cette phase initiale porte sur la compréhension des objectifs et des exigences du projet. Durant cette étape, sont définis habituellement, un ensemble de variables pour les dépendances inconnues, et dans la mesure du possible, une forme générale de cette dépendance comme une première hypothèse. Pour un simple problème, il peut y avoir plusieurs hypothèses, d'où la nécessité d'une étroite collaboration entre expert du domaine et spécialiste en datamining. Cette collaboration n'est pas seulement requise uniquement pour cette étape mais doit continuer pendant tout le processus du Datamining.
2. **La compréhension des données (Data understanding) :** Dans cette étape on s'intéressera essentiellement à la manière dont les données sont générées et collectées. La génération des données peut se faire soit sous contrôle d'un expert (designed experiment approach), ou sans l'intervention de celui-ci (observational opproach). Un échantillonnage aléatoire de données, est admis dans la plupart des applications de datamining. Typiquement, la distribution d'échantillonnage est soit complètement inconnue, après que des données soient collectées, ou partiellement et implicitement donnée dans la procédure de collecte des données. Cependant il est très important, de comprendre comment la collecte de données affecte sa distribution théorique, une telle connaissance a priori peut être très utile pour la modélisation et pour l'interprétation finale des résultats. En outre, il est important de s'assurer que les données utilisées pour estimer un modèle et celles utilisées plus tard pour l'examiner et l'appliquer viennent de la même distribution d'échantillonnage inconnue. Dans le cas contraire le modèle estimé ne peut être employé avec succès dans une application finale.
3. **La préparation des données(Data preparation) :** Les données, habituellement collectées à partir des bases de données existantes, des datawerhous ou de datamarts, ne doivent pas être utilisées directement dans leur état brut du fait qu'elles comportent, en général, un taux très considérable de bruit, de manque et de données inconsistantes.

Mais doivent être, au préalable, soumises à une étape de pré-traitement qui consiste à :

(a) Nettoyer les données du bruit et des valeurs aberrantes, inconsistantes ou incohérentes.

(b) Transformer et intégrer les données : l'intégration des données revient à émerger les données issues de plusieurs sources dans un seul entrepôt de données cohérent (datawarehousse ou cube de données).

La transformation englobe des opérations de normalisation, de lissage (clustering, régression), de généralisation et d'agrégation des données. La normalisation, par exemple, peut améliorer l'exactitude et l'efficacité des algorithmes d'extraction comportant des mesures de distance.

(c) Réduction des données : Revient à réduire la taille des données à fin d'obtenir une représentation réduite, mais sans perte d'intégrité, des bases de données originales. Les analyses effectuées sur ces bases de données réduites doivent fournir les mêmes résultats (ou presque) que ceux obtenus en opérant directement sur les bases de données originales. Parmi les stratégies de réduction de données on cite :
— Agrégation du cube de données
— Réduction de dimensions
— Compression de données
— Discrétisation et génération hiérarchie des concepts

4. **La modèlisation (Modeling) :**
Cette étape porte sur le choix et l'implémentation de la technique de datamining appropriée et le calibrage de ses paramètres.
5. **L'évaluation (Evaluation) :**
à ce niveau le(s) modèle(s) sont évalué(s) et les étapes suivies pour la construction du modèle sont réévaluées pour s'assurer que le projet respecte les objectifs préalablement définis.
6. **Déploiement (Deployment) :**
La création du modèle ne représente pas la fin du projet. Même si le but initial du projet est d'augmenter les connaissances de données, les connaissances acquises ont besoin d'être organisées et présentées d'une manière utilisable par l'utilisateur final.

2.3 Architecture d'un système de datamining

L'architecture d'un système de datamining typique peut avoir les composants principaux suivants (figure 2.3) :

— Une base de données, un entrepôt de données, ou tout autre dépôt d'information : Ensemble de bases de données, d'entrepôts de données, de feuilles de diffusion, ou d'autres genres de dépôts de l'information. L'étape de pré-traitement est exécutée sur les données.
— Un serveur de base de données ou de datawarehouse : responsable de chercher les données appropriées, basé sur la demande de l'utilisateur.
— la base de connaissance : Englobe la connaissance du domaine qui est employée pour guider la recherche, ou pour évaluer l'utilité (interestingness) des modèles résultants. Une telle connaissance peut inclure des hiérarchies de concept, employées pour organiser des attributs ou des valeurs d'attribut en différents niveaux d'abstraction. Des connaissances, telle que la croyance d'utilisateur qui peut être employée pour évaluer l'utilité d'un modèle en se basant sur son imprévisibilité (unexpectedness), peuvent également être incluses en plus des contraintes (additional interestingness constraints), des seuils et des meta-données décrivant, par des données issues de multiples sources hétérogènes.

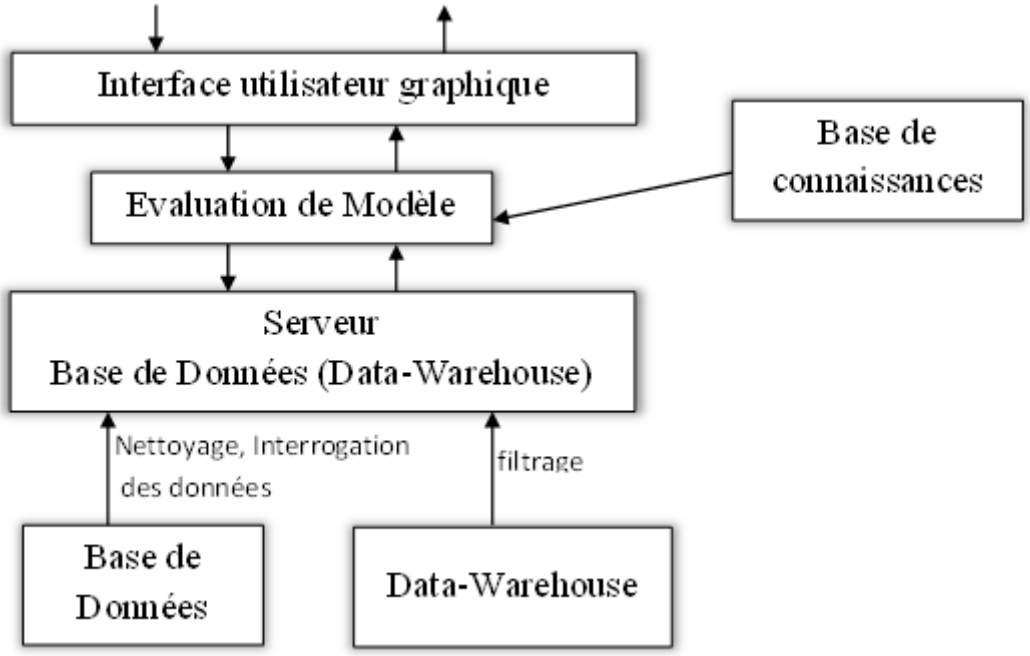


FIGURE 2.3 – Architecture d'un système de datamining

— Moteur d'extraction de données : Il est essentiel au système de datamining et se compose idéalement d'un ensemble de modules fonctionnels pour des tâches telles que l'analyse de caractérisation, d'analyse d'association, de classification, d'évolution et de déviation.
— Module d'évaluation de modèle : Ce composant utilise typiquement des mesures d'interestingness et inter-agit avec le module d'extraction de données afin de focaliser la recherche vers les modèles intéressants. Il peut accéder à des seuils d'interestingness stockés dans la base de connaissance. Alternativement, le module d'évaluation de modèle peut être intégré avec le module d'extraction, selon la méthode d'extraction de données employée. Pour l'exploitation efficace de données, il est fortement recommandé de pousser l'évaluation de l'interestingness du modèle aussi profondément que possible dans le processus d'extraction afin de ne confiner la recherche seulement qu'aux modèles intéressants.
— Interface utilisateur graphique : Assure la communication entre le système de datamining et l'utilisateur. Et permet à ce dernier d'interagir avec le système en spécifiant une question ou une tâche de datamining fournissant des informations pour guider la recherche, et en effectuant un datamining exploratoire de données basé sur les résultats d'extraction de données intermédiaires. En plus, ce composant permet à l'utilisateur de parcourir les données du datawherehous et leurs structures, d'évaluer les modèles extraits, et à visualiser les modèles dans différentes formes.

2.4 Les Tâches du Datamining

Les techniques de data mining sont essentiellement des techniques de découverte de motifs et de corrélation dans un ensemble de données [74]. Certaines techniques, tel que les règles d'associations[8], sont propres au datamining mais la plus part sont issues d'autres domaines tel que les statistiques, l'apprentissage automatique,... Les tâches du data mining peuvent être classées selon différents critères et visions. par exemple en supervisées et non-supervisées selon que l'intervention de l'utilisateur soit exigée ou pas. en descriptives et prédictives selon la nature prédictive ou descriptive du modèle résultant. ou encore en transparente et opaque selon

à ce que les détailles de l'algorithme utilisé(plusieurs algorithmes peuvent être utilisés) soient visibles à l'utilisateur ou non. pour M. H., Dunham[118] la catégorisation des tâches de data maining en tâches descriptives et prédictives est naturelle(Figure 2.4). Les méthodes descriptives et prédicatives peuvent être combinées au sein d'une même application du datamining. Par exemple, le clustering peut être appliqué à un ensemble de données à fin d'identifier les différents clusters naturels qui le composent. Chaque instance de données, lui sera ainsi assignée une étiquette définissant sans appartenance à un groupe. Dans une deuxième phase, un modèle de classification, utilisant les données étiquetées comme donnes d'apprentissage, peut être développé à fin de prédire l'étiquette d'une nouvelle instance de données.

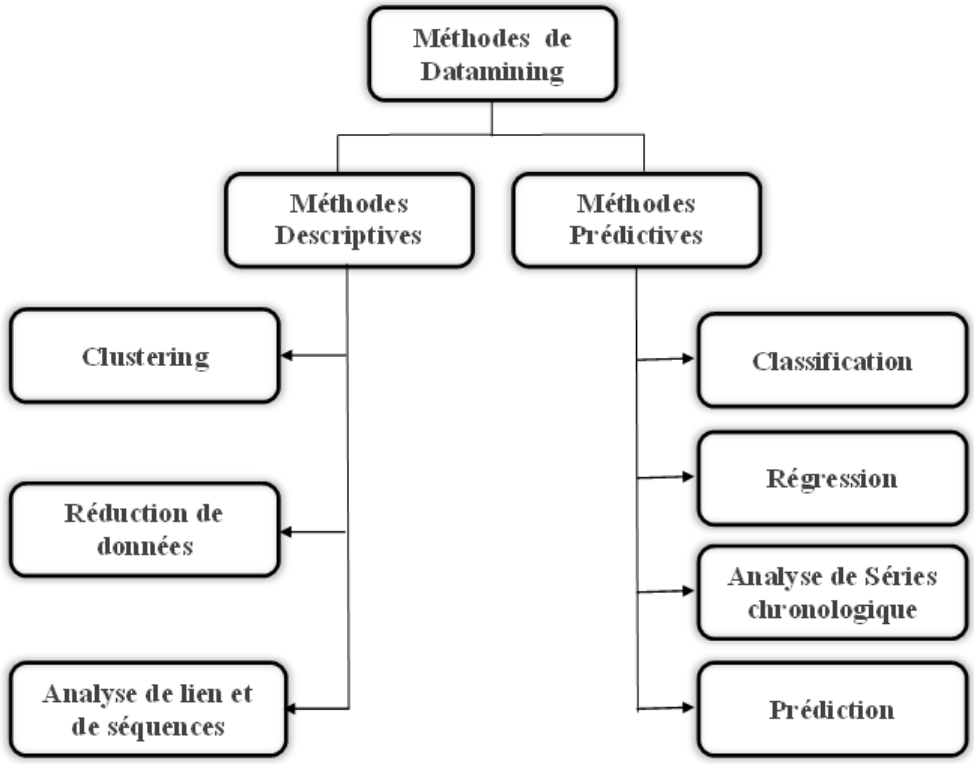


FIGURE 2.4 – Les tâches de datamining

2.4.1 Méthodes descriptives

Les méthodes descriptives consistent à explorer un jeu de données afin de faire ressortir les régularités et les tendances cachées par le volume des données. Cette catégorie de tâches regroupe : Les méthodes de réduction de données, le clustering, l'analyse des liens et l'analyse de séquences.

2.4.1.1 Réduction de données

La réduction des données consiste à réduire, par agrégation d'informations, le volume des données explorées par les autres techniques de datamining[130]. La réduction doit être faite de telle façon à ce que la perte d'information soit aussi faible que possible et à ce que les résultats des méthodes appliquées aux données réduites soit aussi similaire que possible à ceux obtenus en utilisant les données brutes. La réduction de données concerne à la fois la réduction du nombre, ou la taille,des données et du nombre d'attributs ou variables.

La réduction du nombre de données permet d'éliminer les instances de données redondantes et/ou celles ayant une faible influence sur l'étude et dans certains cas, les instances de données en conflits. Cette opération est très liée à la méthode d'analyse choisie. Elle peut être effectuée dans la phase d'acquisition des données ou même dans la phase d'extraction des connaissances avant de passer à l'analyse proprement dite. Les instances de données similaire peuvent être regrouper ensemble par une technique de clustering où chaque cluster est remplacé par son centroïde. Aussi, la réduction de la taille des données peut être effectuée par les techniques d'échantillonnage et d'agrégation de cube de données[169].

La réduction du nombre d'attributs ou de dimensionnalité, Motivée par fondamentalement par la malédiction de dimensionnalité[253], consiste à choisir ou à extraire, à partir d'un ensemble d'attributs, un sous-ensemble optimal d'attributs pertinents selon un critère de performance. La notion de pertinence d'un sous-ensemble d'attributs dépend des objectifs et des critères du système. Selon Dash et Al[96], la sélection d'attributs est un processus à quatre étapes illustrées dans la figure 2.5.

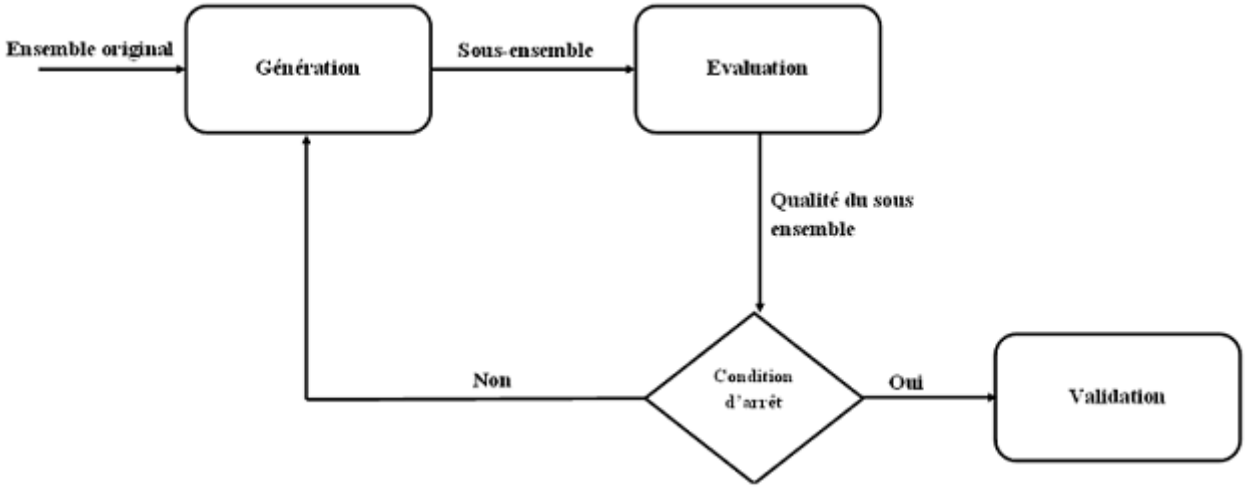


FIGURE 2.5 – Sélection d'attribut

Et peut se faire selon trois stratégies : La première consiste à définir au préalable le nombre p d'attributs à sélectionner puis appliquer un algorithme de sélection pour trouver le meilleur sous-ensemble de p attributs. Dans la deuxième stratégie, On définit le plus petit sous-ensemble d'attributs dont la performance est plus grande ou égale à un seuil prédéfini. La troisième stratégie, quant à elle, cherche à trouver un compromis entre l'amélioration de la performance, l'erreur de classification par exemple, et la réduction de la taille du sous-ensemble d'attributs. Le but de cette stratégie consiste à sélectionner un sous-ensemble d'attributs qui optimise les deux objectifs en même temps. Selon l'organisation de processus de recherche, les algorithmes de sélection d'attributs peuvent être regroupé en quatre catégories[8, 377] comme illustré par la figure 2.6.

— Filtres : Dans cette catégories de méthodes, la sélection des attributs est indépendante de l'algorithme d'apprentissage et se base, uniquement, sur les propriétés intrinsèques de l'ensemble des données(Fig. 2.7). Elle propose un sous ensemble de variables satisfaisant pour expliquer la structure cachée des données indépendant de l'algorithme d'apprentissage utilisé.
Ces méthodes, qui s'adaptent bien aux problème à haute dimensions(nombre d'attributs large), emploient des mesures statistiques pour attribuer à chaque attributs un score, définissant le degré de sa pertinence. Le sort, conserver ou rejeter, des attributs sera défini à la base de ce score. Parmi les mesures utilisés dans la littérature comme score

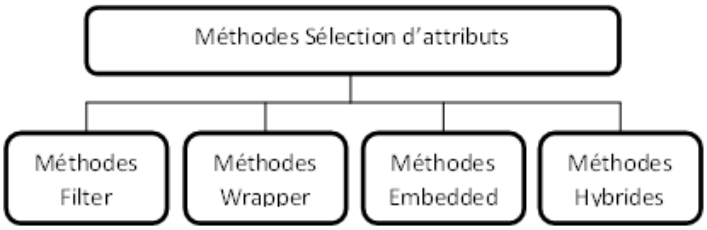


FIGURE 2.6 – Catégories de méthodes de sélection d'attribut

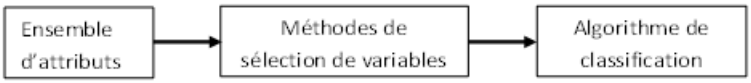


FIGURE 2.7 – Selection d'attributs par les méthodes Filtres

on cite :

— Le critère de Fisher qui mesure le degré de séparabilité des classes à l'aide d'une caractéristique donnée et est défini par :

$$F(i) = \frac{\sum_{k=1}^{K} p_k \cdot (\mu_k^i - \mu^i)^2}{\sum_{k=1}^{K} p_k \cdot (\sigma_k^i)^2}$$

Où p_k, μ_k^i et σ_i sont respectivement l'effectif, la moyenne et l'écart type du $i^{\text{ème}}$ attribut au sein de la classe k. μ^i est la moyenne globale du $i^{\text{ème}}$ attribut.

— L'entropie d'une valeur particulière de l'attribut discret est mesuré comme suit :

$$E = -\sum_{i=1}^{k} p_i \cdot log(p_i)$$

— SNR(Signal-to-Noise Ratio coefficient) est un score qui mesure le pouvoir discriminatoire d'un attribut entre deux classes. D'une manière similaire au critère de Fisher, cette méthode classe les attributs en calculant le rapport entre la valeur absolue de la différence des moyennes des classes et la moyenne des écart-types des classes. Le SNR pour un attribut, dans un problème de classification binaire, est donné par :

$$SNR(i) = \frac{2 \cdot |\mu_1^i - \mu_2^i|}{\sigma_1^i + \sigma_2^i}.$$

Où μ_j^i et σ_j^i; $j = 1, 2.$ représentent respectivement la moyenne et l'écart type du $i^{\text{ème}}$ attribut au sein de la classe j

D'autre mesure ou approches d'évaluation des méthodes de sélection d'attributs à base de filtres on été proposées dans la littérature tel que le test de χ^2, les méthodes à base d'information mutuelle et la méthode **Relief** et ces variantes[8].

— Wrapper : Les méthodes wrapper, appelées aussi les méthodes enveloppantes, ont été introduite par John & Kohavi[202] en 1994. Ces méthodes consistent à sélectionner des sous ensembles d'attributs candidats et d'évaluer leurs degré de pertinence avec un algorithme de classification appliqué à un ensemble de données d'apprentissage d'une maniéré itérative(Fig. 2.8).

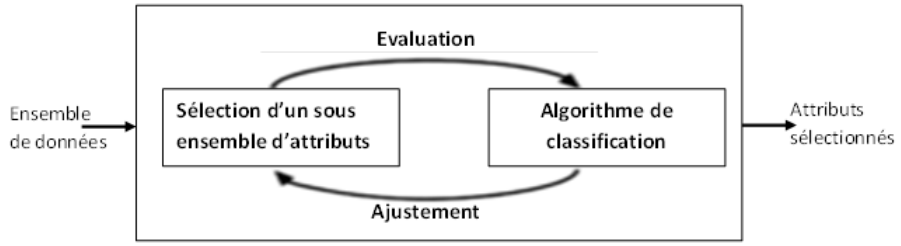


FIGURE 2.8 – Principe des méthodes Wrappers

Selon Liu et al.[252], Étant donné un inducteur $\mathcal{I}$ et un ensemble de données d'apprentissage $\mathcal{D}$ décrit par les attributs $A_1, A_2, \ldots, A_n$, le sous ensemble $\mathbf{X_{op}}$ optimal d'attributs est celui qui maximise la précision du classificateur induit $\mathcal{C} = \mathcal{I}(\mathcal{D})$. La précision de la prédiction est typiquement considérée comme le plus important indicateur de la pertinence des attributs qui peut être, généralement, vérifiée par un mécanisme de validation croisée. Ces méthodes sont caractérisées par leurs simplicité conceptuelle mais ne permettent pas de comprendre les éventuelles relations de dépendances conditionnelles pouvant exister entre les attributs d'autre par la procédure de sélection dépend étroitement de l'algorithme de classification utilisé et est de complexité temporaire très importante, voir non raisonnable si le nombre des attributs est très grand.

— Méthodes Embedded : Les méthodes Embedded intègrent directement la sélection dans le processus de l'apprentissage à un coût inférieur au méthodes wrappers et permettent de capturer les dépendances entre attributs. De plus, ces méthodes, considèrent non seulement les relations entre un attribut donné en entrée et celui donné en sortie, mais cherchent également les relations locales. Les arbres de décision sont l'illustration la plus emblématique. Mais, en réalité, sont classées dans cette catégorie toutes les techniques, tel que les support vecteur machine (SVM), qui consiste à évaluer l'importance d'une variable en cohérence avec le critère utilisé pour évaluer la pertinence globale du modèle. Bien que ces méthodes soient largement utilisées en apprentissage, elles ont été relativement peu étudiées dans le contexte de la sélection de variables en apprentissage d'ordonnancement.

— Les méthodes hybrides sont, en réalité, une combinaison des deux premières méthodes(Fig. 2.9).

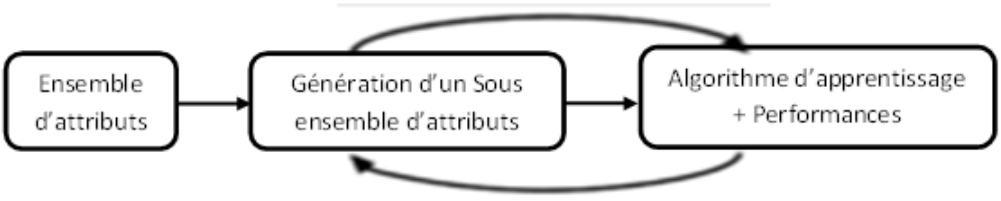


FIGURE 2.9 – Principe des méthodes hybrides[314]

La première étape, de ce type de méthodes, est généralement basée sur des méthodes à filtre pour réduire le nombre d'attributs. Ensuite, une méthode Wrapper est utilisée pour sélectionner le nombre d'attributs désiré à partir de l'ensemble réduit des attributs obtenu lors de la première étape. La complexité temporaire des méthode hybride, est similaire à celles des méthodes à filtre.

L'analyse aux composantes principale(ACP)[203] est l'une des techniques linéaires les plus utilisées dans le cadre de réduction de dimensions. L'ACP consiste à trouver des combinaisons linéaires orthogonales des attributs. Ces Combinaisons, dites Composantes principales, sont

en théorie, indépendants les unes des autres et peuvent être classées par ordre d'importance. Partant d'un ensemble de vecteurs $x_i,\ i = 1\ldots, N$, de dimension d, la méthode consiste à chercher les axes de projection orthogonaux suivant lesquels la variance est maximisée. L'approximation optimale, au sens de l'erreur quadratique moyenne, d'un vecteur x_i par un vecteur t_i de dimension $q < d$ est donnée par :

$$\hat{t}_i = W_q^t(y_i - \mu)$$

Où μ est la moyenne des x_i et W_q est la matrice de projection composée des q premiers vecteurs propres de la matrice de covariance Σ correspondant aux q plus grandes valeurs propres triées dans un ordre descendant $(\lambda_i)_{i=1,\ldots,q}$. l'erreur quadratique de l'approximation est donnée par la somme des valeurs propres écartées :

$$e^2 = \sum_{k=q+1}^{d} \lambda_k$$

q est choisi tel que $\dfrac{\sum_{k=1}^{q} \lambda_k}{\sum_{k=1}^{d} \lambda_k} \geq p$. Où $p \in \{0, 1\}$ un seuil choisie arbitrairement.

Un autre critère, du choix de p, est basé sur la règle de Kaiser selon laquelle sont retenues seulement les composantes ayant des valeurs propres plus grandes que la variance moyenne. Si la matrice de corrélation est utilisée, les composantes ayant des valeurs propres supérieurs à 1 son retenues[130]. Dans [201], L. Jimenez et D. Landgrebe comptent deux majeurs inconvénients pour cette approche. Le premier est l'absence d'un modèle génératif des données et d'une densité de probabilité associée (i.e., vraisemblance). Le second est son caractère global qui suppose implicitement que la distribution des données est un hyperellipsoïde caractérisé par sa moyenne et sa matrice de covariance globale. Ainsi, elle peut provoquer la perte définitive d'informations caractérisant d'éventuelles structures locales des données. A fin de contrarier ses problème des versions robustes d'ACP ont été proposées(voir [130] pour plus de détailles).

L'analyse discriminante de Fisher aussi appelée analyse factorielle discriminante est une autre méthode, linéaire, de sélection d'attributs proposée par Fisher en 1936[140]. Cette méthode, applicable lorsque les classes des individus sont connues, consiste, étant données un ensemble de n données $x_1, x_2, \ldots, x_n$ de dimension d reparties en k classes $C_1, C_2, \ldots, C_k$ différentes, à choisir entre les combinaisons linéaires des variables celles qui maximise l'homogénéité de chaque classe. Autrement dit, cette méthode cherche un espace vectoriel de faible dimension qui maximise la variance inter-classe donnée par :

$$S_W = \sum_{i=1}^{k} \sum_{x \in C_i} (x - \mu_i)(x - \mu_i)^t$$

et minimisant la variance intra-classe définie par : $S_B = \sum_{i=1}^{k} n_i(\mu_i - \mu)(\mu_i - \mu)^t$
Le calcule de la projection est ainsi obtenu en maximisant le critère de Fisher :

$$J(W) = \frac{|W^t S_B W|}{|W^t S_W W|}$$

.

Ainsi, une direction w qui maximise J satisfait :

$$S_B W = \lambda S_W W$$

Et si S_W n'est pas singulière, on se ramène à un problème de valeur propres

$$S_W^{-1} S_B W = \lambda W$$

On n'a qu'à chercher les valeurs et les vecteurs propres de $S_W^{-1} S_B$. Et puisque le rang de S_B est égale à $k-1$ au maximum on ne pourra donc extraire que $k-1$ nouveaux attributs.

Dans la catégorie de méthodes heuristiques de sélection d'attributs, on compte la sélection séquentielles croissante(Sequential Forward Selection-SFS) (Algorithme 1) proposée par Marill et Green en 1963[272], et la Sélection séquentielle arrière(Sequential Backward Selection-SBS)(Algorithme 2) proposée par Whitney en 1971[409].

Algorithme 1 : sélection séquentielles croissante

Entrées :
$F = \{f_1, f_2, \ldots, f_n\}$;
M : taille de l'ensemble finale;
Output : $E = \{f_{s1}, f_{s2}, \ldots, f_{sm}\}$;

```
1 début
2     E ← Φ;
3     pour i = 1 à m faire
4         pour j = 1 à |F| faire
5             Évaluer f_j ∪ E;
6         f_max ← meilleure f_j ;
7         E ← E ∪ f_max;
8         F ← F/f_max;
9     retourner E;
```

Algorithme 2 : Sélection séquentielle arrière

Entrées :
$F = \{f_1, f_2, \ldots, f_n\}$;
M : taille de l'ensemble finale;
Output : $E = \{f_{s1}, f_{s2}, \ldots, f_{sm}\}$;

```
1 début
2     E ← F;
3     pour i = 1 à (n − m) faire
4         pour j = 1 à |E| faire
5             Évaluer E/f_j;
6         f_min ← la plus mauvaise f_j ;
7         E ← E/f_min;
8     retourner F;
```

La SFS est l'algorithme de sélection d'attribut le plus simple et probablement le plus rapide. Cet algorithme commence avec un ensemble vide de variables(déjà sélectionnés) noté E. A chaque itération l'attribut f_j, parmi ceux non sélectionnés, optimisant un certain critère d'évaluation, est ajouté à E. Ce processus est ainsi réitéré jusqu'au critère d'arrêt, qui correspond généralement soit à la performance qui n'augmente plus ou à l'atteinte du maximal d'itération. La SBS se base sur le même principe, à la différence que la méthode commence avec tout

l'ensemble des attributs. À chaque itération, on retire de l'ensemble des attributs déjà sélectionnés qui donne la moins bonne performance. Cette méthode permet de trouver un meilleur sous-ensemble de caractéristiques, car elle prend en considération les interactions de chaque attribut avec un ensemble plus large. Cependant, la méthode SBS pose des difficultés en termes de performance calculatoire et est très difficile à appliquer pour des systèmes à très grande dimension. En 1978, Kittler[220] a proposé des généralisation des méthodes SFS et SBS. Dans ces nouvelles versions dite GSFS et GSBS Il est possible d'ajouter ou de retirer itérativement les attributs par groupe. Ces méthodes permettent d'améliorer légèrement la performance des méthodes initiales, mais ils conservent toujours les mêmes problèmes que les méthodes de base. Dans [228], Kudo recommande de les utiliser si l'on désire que la dimension du sous-espace soit très faible (GSFS) ou très proche de la dimension initiale de l'espace d'attributs (GSBS).

Les algorithmes génétiques sont particulièrement intéressants pour le problème de sélection d'attributs par le fait qu'ils sont très efficaces en matière de recherche non-linéaire. Leur application en sélection d'attributs a surtout été réalisée en optimisation mono-objectif à l'aide de méthodes wrapper notamment avec les algorithmes de classification tel que l'algorithme des plus proche voisins (K-Nearest-Neighbor), les réseau de neurones et tables de décisions euclidiennes. Dans une première étape, plusieurs sous ensemble d'attributs sont générés aléatoirement pour former, ainsi, la population initiale. itérativement, ont fait évoluer la population courante en calculant la fitness de chaque individus et en appliquant les opérateurs de crossover et du mutation selon une certaine probabilité prédéfinie. Dans chaque itération, les chromosomes à forte fitness ont plus de chance d'être sélectionnés pour la production. La fonction de fitness permet de faire évoluer la population vers des individus de meilleur qualité. Le sous ensemble optimale est obtenu après un nombre, suffisamment grand, de générations. Le codage des individus peut se faire selon deux façons[204]. La première considère que l'espace de recherche est représenté par toutes les partition de l'ensemble d'attributs. Une solution est représenté par une chaine binaire de longueur fixe dont le $i^{\text{ème}}$ bit indique si le $i^{\text{ème}}$ attribut est sélectionné ou nom. Dans la seconde, la taille de chaque individu est égale au nombre d'attributs et est codé par l'index des attributs sélectionnés et des zéros. Les attributs peuvent être représentés plusieurs fois. Cette redondance est susceptible de ralentir la perte de diversité. Les opérateurs génétiques utilisés sont une variante du crossover uniforme, un opérateur de mutation et un opérateur, Delete future, qui consiste à enlever un attribut sélectionné aléatoirement à partir d'un individu et de tous ses copies s'il est dupliqué plusieurs fois. [253] couvre, d'une manière claire, concise et cohérente les concepts clés, et les principes de de base de la sélection d'attributs ainsi qu'un état de l'art des différents algorithmes utilisés dans le domaine.

2.4.1.2 Analyse des liens

L'analyse de lien est une approche descriptive d'exploration des données et consiste à identifier les relations et les corrélations dans une importante masse de données. On distingue deux principales techniques dans cette approche à savoir l'extraction de règles d'association et la découverte de séquences. Les règles d'association ont suscité beaucoup d'attention en matière de fouille des données et ont été appliquées dans divers domaines tel que : Le marketing, La médecine, l'assurance médicale, la détection de fraudes... Initialement introduites par Agrawal et Al [11] pour capturer et représenter les éventuelle relations et corrélations pouvant exister entre un ensemble d'attributs. Les règles d'association décrivent les motif fréquents dans une grande masse de données et peuvent être représenter, formellement, comme suit[40]

$$\left(\bigwedge_{i=1}^{m} A_i = v_i\right) \rightarrow \left(\bigwedge_{i=m+1}^{n} A_i = v_i\right) [s, c] \tag{2.1}$$

où les (A_i, v_i) avec $i = 1, \cdots, n$, représente les paires attribut-valeur. La conjonction figurant

au coté gauche de l'équation 2.1 est dite "antécédent" et celle du coté droit est dite "conséquent". Les deux paramètre s et c sont dit respectivement le support et la confidence de la règle. Le support est défini comme étant le pourcentage de transactions satisfaisant l'antécédent et définit la portée de la règle, et est utilisée pour éliminer les règles de faible importance. La confidence c, définit la précision de la règle, elle mesure la pertinence de l'inférence faite par une règle et exprime une estimation de la probabilité conditionnelle du conséquent sachant l'antécédent. En effet, une valeur élevée du support implique que la règle est statistiquement significative de même, une valeur élevée de la confidence caractérise le fait qu'une règle, dont le côté à gauche est prédictif de son côté droit, est forte. Évidemment, les règles fortes et statistiquement significatives sont les plus attrayantes. La génération des règles d'association se fait en deux phases. En premier lieu, les itemset fréquents ayant un support supérieur à un seuil minimum sont identifiés à partir des données d'audit. Les itemset, ainsi obtenus, sont utilisés pour générer les règles d'association dont la confidence est supérieur à un seuil minimum prédéfini. Dans la littérature, on trouve plusieurs algorithmes et méthodes d'extraction des règles. Apriori(Algorithme 8) proposé par Agrawal et Al en 1993 [11] est le plus connus et appliqué.

Algorithme 3 : L'algorithme Apriori

1 **début**
2 Trouver tous les itemset de longueur 1 satisfaisant ayant un support supérieur au seuil minimum;
3 Générer, iterativement, les ensembles d'items de longueur k (k-itemset) par la combinaison de deux itemset de longueur k-1;
4 Écarter tout k-itemset peu fréquent qui contient un sous ensemble de longueur k-1;
5 Trouver tous les k-itemset ayant un support supérieur au seuil minimum;
6 Générer tout les sous-ensembles non vide d'un itemset fréquent générés en 2;
7 Extraire, pour chaque sous-ensemble non vide généré en 3 les règles correspondantes;
8 **fin**

Une règle d'association traduit seulement une occurrence conjointe entre deux ensembles d'items. Elle n'exprime ni une relation de causalité ni une relation d'ordre. La découverte de relations entre items dans une période de temps fait l'objet de l'analyse de séquences[169]. Cette dernière est utilisée pour déterminer les motifs séquentiels dans un ensemble de données[118].

Le concept de motifs séquentiels[376] constitue une extension à celui des règles d'association en intégrant diverses contraintes temporelles. L'extraction de tels motifs consiste à construire des ensembles d'items, couramment associés sur une période de temps bien spécifiée. La tâche d'extraction de motifs séquentiels met en évidence des associations inter-transactions, contrairement à celle des règles d'association qui extrait des combinaisons intra-transactions. Par exemple, des motifs séquentiels peuvent montrer que "60% des gens qui achètent une télévision, achèteront un magnétoscope dans les deux années qui suivent". Ce problème, posé à l'origine dans un contexte de marketing, intéresse à présent des domaines aussi variés que les télécommunications (détection de fraudes), la finance, ou encore la médecine (identification des symptômes précédant les maladies). Le processus de découverte de motifs séquentiel peut être scinder en cinq étapes :

1. Phase de trie : lors de cette étape l'ensemble des transactions est, implicitement, converti en une base de séquences. Les séquence, ainsi trouvées, doivent être triées dans un ordre chronologique afin d'optimiser les traitements.
2. Phase de détermination des motifs : Cette phase consiste à déterminer l'ensemble L de

tous les motifs de séquences de taille 1 (séquences Larges) dont le support est supérieur à un certain seuil. Les motifs sont, ici, des séquences composées uniquement d'un seul événement. Le support d'un motif est définit comme étant le rapport entre le nombre de séquence supportant ce motif et le nombre total des séquences.

3. Phase de transformation : Cette phase consiste à éliminer les éléments des événement n'apparaissant pas dans au moins un événement de l'ensemble, L, des séquences Large trouvé lors de la phase précédentes. Aussi, l'événement se verra attribué un identifiant unique.
4. Phase de détermination des séquence : Les séquences désirées sont extraites de l'ensemble L selon un algorithme d'extraction de séquences.
5. Phase d'élimination des motifs non maximums : Dans cette phase, consiste à éliminer de l'ensemble, L, les séquences non maximales. Dans certain algorithme d'extraction de motifs, cette phase est combinée avec la phase précédente à fin de réduire le temps de calcul des séquences non maximales.

Plusieurs algorithmes ont été proposés pour la découverte de motifs séquentiels. Les pionniers furent AprioriAll, AprioriSome[13] et DynamicSome[12] développés par Agrawal et Srikant. AprioriAll et AprioriSome ont été améliorés, par la suite, en intégrant la possibilité de spécifier une fenêtre d'événement ainsi qu'un intervalle ce qui a donné naissance à l'algorithme GSP(Generalized Sequential Pattern). Voir [266] pour une concise taxonomie des algorithmes d'extraction de motifs séquentiels.

2.4.1.3 Clusetring

Le clustering est une méthode descriptive d'apprentissage non supervisé qui consiste à grouper un ensemble d'objets dans des sous-groupes de sorte que les objets similaires sont groupés dans un même sous-groupe(cluster), tandis que les objets différents sont groupés dans différentes classes [167]. Il est à noter, qu'à la différence de la classification, où les objets sont assignés à des classes prédéfinies, Les classes dans le cas du clustering sont aussi à définir. Ainsi, les objets sont organisés en une représentation efficace qui caractérise l'échantillon étudié. Formellement, la structure de clustering est représentée comme une collection de sous-ensemble $C = C_1, C_2, \cdots, C_k$ d'un ensemble S tel que $S = \cup_{i=1}^{k} C_i$ avec $C_i \cap C_j = \Phi$ pour $i \neq j$. Le clustering de l'ensemble S revient à définir une fonction d'affectation $f : S \to [0,1]^k$, $x \to f(x)$. définie comme suit[167] :

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{pmatrix}$$

Où $f_i(x) \in [0,1]$ pour $i = 1, 2, \ldots, k$ et $x \in S$, et

$$\sum_{i=1}^{k} f_i(x) = 1 \quad \forall x \in S.$$

Si $\forall x \in S$, $f_i(x) \in \{0,1\}$ alors le clustering représenté par f est dit exclusif(Hard) autrement il est dit flou(Fuzzy). Chaque cluster C_k est caractérisé par :

— Son centre de gravité $\mu_k = \dfrac{1}{|C_k|} \sum\limits_{x_i \in C_k} x_i$.

— Son inertie $J_k = \sum\limits_{x_i \in C_k} d(x_i, \mu_k)$ où $d(x_i, \mu_k)$ désigne la distance entre un élément x_i et μ_k. L'inertie d'un cluster mesure la concentration des points du cluster autour du centre

de gravité. Plus cette inertie est faible, plus petite est la dispersion des points autour du centre de gravité.

— Sa matrice de variance-covariance : $\Sigma_k = \sum_{x_i \in C_k} (x_i - \mu_k)(x_i - \mu_k)'$

Si on désigne par μ le centre de gravité du nuage de points de l'ensemble S alors le terme

$$J = \sum_k n_k^2 d^2(\mu_k, \mu)$$

désignera l'inertie inter-cluster qui mesure "l'éloignement" des centres des clusters entre eux. Plus cette inertie est grande, plus les clusters sont bien séparés. Une bonne méthode de clustering produit des clusters dont les éléments ont, d'une part, une forte similarité intra-classe et une faible similarité inter-classe, d'autre part(Fig. 2.10).

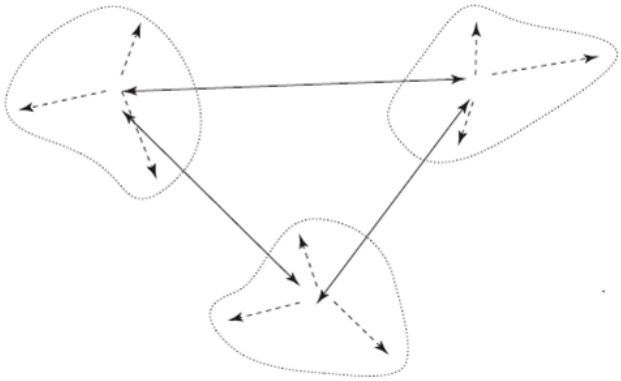


FIGURE 2.10 – L'inertie, Inter et Intra clusters.

Aussi, la qualité d'un clustering dépend de la mesure de similarité utilisée et de sa capacité à trouver des motifs intéressants. Globalement le processus de clustering invoque cinq étapes[166]. La première, dite présentation de motifs, consiste à déterminer le nombre et le type des attributs par des méthodes d'extraction et sélection d'attributs. La deuxième consiste à définir une mesure de distance appropriée au domaine d'études. Dans ce cadre on compte plusieurs distances, dont les plus connues, sont :

— La distance euclidienne définie par $d_{Euc}(x, y) = \sum_{i=1}^{p} (x_i - y_i)^2$ qui est cas un particulier de la distance de Minkowski donnée par : $d_{Mink}(x, y) = \left(\sum_{i=1}^{p} |x_i - y_i|^q \right)^{\frac{1}{q}}$.

— La distance de Manhattan donnée par : $d_{Man}(x, y) = \sum_{i=1}^{p} |x_i - y_i|$

— La distance de Mahanalobis définie par : $d_{Mah}(x, y) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)}$

Où x et y sont deux instances de données de dimension p. x_i et y_i sont respectivement la $i^{\text{ème}}$ composante de x et de y et Σ^{-1} est la matrice inverse de la variance-covariance. $(x-y)^T$ dénote le transposé de (x-y).

La distance euclidienne est communément utilisée pour évaluer la proximité dans des espace à deux ou trois dimensions et est bien adaptée aux ensembles de données contenant des clusters compacts ou isolés. Avec la distance de Minkowski les attributs à large valeurs ont tendance à avoir une nette domination sur les autres. Ce problème peut être surmonté, par exemple, par :

— Normalisation ou mise à échelle des données par une transformation de type

$$x_i = \frac{x_i - x^i_{min}}{x^i_{max} - x^i_{min}}$$

Où x^i_{min} et x^i_{max} représente, respectivement la valeur minimale et maximale du $i^{\text{ème}}$ attribut. et x_i une valeur prise par le $i^{\text{ême}}$ attribut.

— Standardisation des données (avoir des données centrées-réduites) par la transformation

$$x_i = \frac{x_i - \mu_i}{\sigma_i}$$

Où μ_i et σ_i sont respectivement la moyenne et l'ecart type du $i^{\text{ême}}$ attribut.

— Toute autre schéma de pondération.

Une corrélation linéaire entre attributs peut aussi fausser une mesure de distance. Cette distorsion peut être atténuée par un nettoyage de données ou en utilisant la distance de Mahanalobis.

La troisième étape, dite étape de clustering, consiste à appliquer un algorithme de regroupement à fin de regrouper les instances de données dans un nombre significatif de clusters. L'appartenance d'une donnée à un cluster peut être fixe ou floue. Dans le premier cas, la donnée est affectée à un et un seul cluster, alors que dans le deuxième cas une donnée peut appartenir à deux ou plusieurs clusters avec une certaine probabilité. Dans la quatrième étape, dite étape d'abstraction de données, un ou plusieurs prototypes (ou données représentatives) des clusters sont extraits de sort qu'il soit facile de comprendre le clustering résultant. Par exemple, un cluster peut être représenté par son centroïde. A l'étape finale, l'efficacité de l'algorithme de clustering est évaluée. Cette évaluation peut être interne, externe ou relative. Dans une évaluation interne, on tente de déterminer si la structure est intrinsèquement appropriée aux données. Dans l'évaluation externe, la structure obtenue est comparée à la structure apriori des données. Et en fin, dans l'évaluation relative, On applique un test pour mesurer le mérite relatif de deux structures. Bien qu'aucune définition formelle du concept de cluster ne soit donnée dans la littérature, on y trouve plusieurs définitions, opérationnelles. Par exemple, selon [166] :

Block[54] suggère qu'un cluster est un groupe de points satisfaisant un ensemble de critères plausibles tel que :

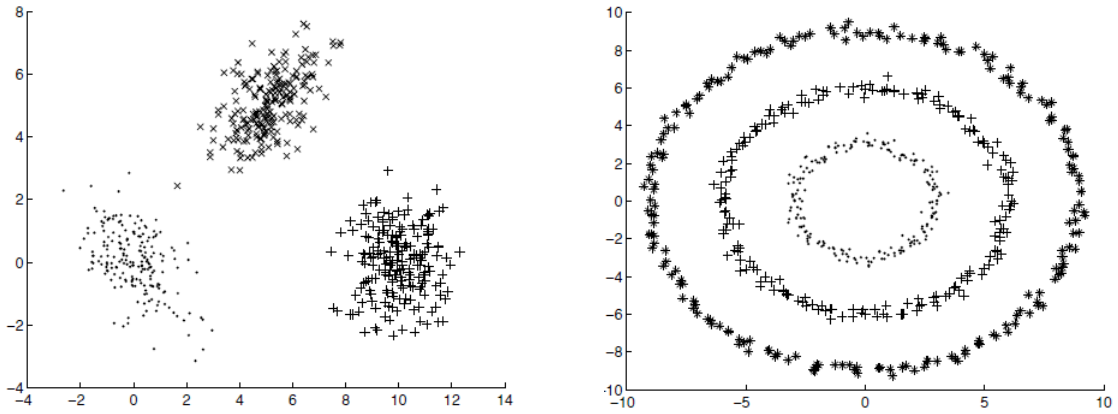
— Partager des propriétés étroitement liées ;
— Montrer des petites distances mutuelles ;
— Avoir des relations avec au moins un point du groupe ;
— Peut être clairement distinguable du reste des points de l'ensemble de données.

Carmichael et al.[68] suggèrent qu'un ensemble de données forment un cluster si leur distribution satisfait les condition suivantes :(a) Des régions continues et relativement denses existent dans l'espace de données et (b) des régions continues et relativement vides existent dans l'espace de données.

Lorr[258] affirme qu'il existe deux type de cluster pour les données numériques à savoir des cluster compacts et des clusters chaînés. Un cluster compact(Fig. 2.11(a)) est formé d'un ensemble de points ayant une haute similarité mutuelle. Alors qu'un cluster chaîné(Fig. 2.11(b)) est formé par un ensemble de points dans lequel chaque deux points sont atteignables par un chemin.

Un cluster compact peut être représenté par un seul centre, alors qu'un cluster chainé est, usuellement, représenté par plusieurs centres.

De nombreux algorithmes de clustering ont été proposés et il est difficile de fournir une nette catégorisation à cause du chevauchement qui existe entre les catégories. Une catégorie peut partager certaines caractéristiques avec d'autres catégories. Néanmoins, il est utile de présenter une image relativement organisée des méthodes de clustering[169]. Globalement, les méthodes de clustering peuvent être classées dans les catégories suivantes[128] Les méthodes



(a) Ensemble de données avec trois clusters compacts

(b) Ensemble de données avec trois clusters chaînés

FIGURE 2.11 – Types de clusters

.

de partitionnement, les méthodes hiérarchiques, les méthodes à base de modèle, les méthodes à base de densité et en fin les méthodes à base de grilles(Fig. 2.12).

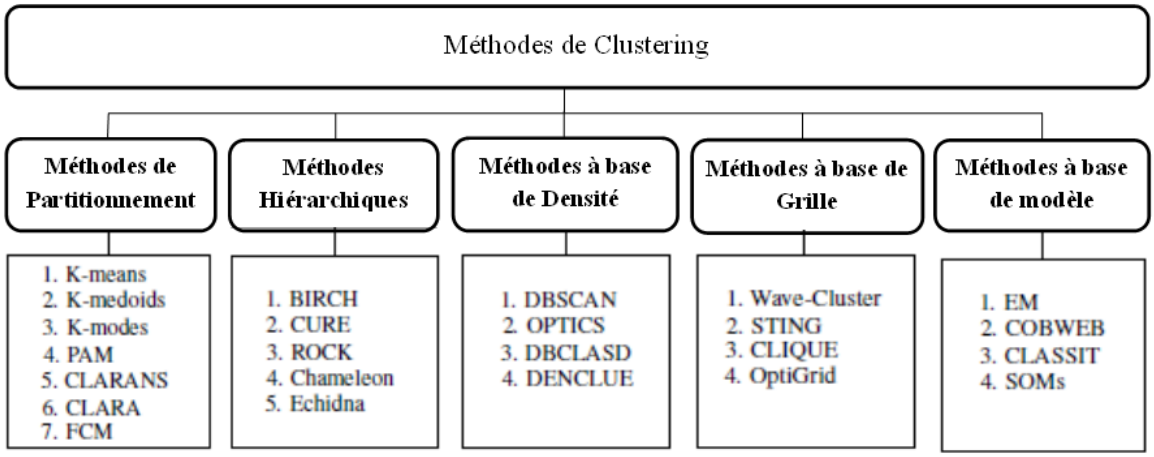


FIGURE 2.12 – Méthodes de clustering

— Étant donné un ensemble de n objets, une méthode de partitionnement consiste à construire k sous-ensembles ou partition où chaque partition, représente un cluster. Le nombre de cluster k; $k \leq n$ peut être spécifié ou non[192]. Les clusters issus d'une méthode partitionnelle sont tel que : (a) chaque cluster contient au moins un objet et (b) chaque objet appartient à un et un seul cluster. Cette dernière exigence est délaissée dans certaine technique de partitionnement flou. Étant donnés le nombre de clusters k et une mesure de similarité, Les méthodes de partitionnement construisent une première partition puis appliquent itérativement une technique de ré-localisation qui tente d'améliorer le partitionnement en déplaçant des objets d'un cluster à un entre. Lors de ce processus un critère de partitionnement ou une fonction objectif, basée sur une mesure de similarité, est à optimisé. Plusieurs méthodes de partitionnement ont été proposées, dont les plus connues sont : K-means et k-medoïds, k-modes , c-means, PAM, CLARA[169]...

— Les méthodes de clustering hiérarchique produisent une décomposition hiérarchique d'un ensemble de données. Cette décomposition peut être agglomerative ou divisive(Fig. 2.13).

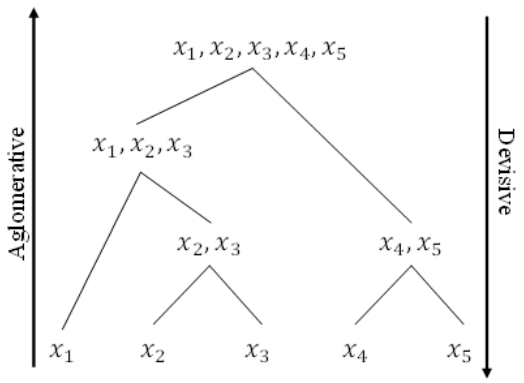


FIGURE 2.13 – Méthodes de clustering Hiérarchique

Dans le premier cas[169], l'approche de clustering, dite ascendante(Algorithme 4), Les n éléments à classer sont mis, chacun dans un groupe. Puis, à chaque itération, Les deux groupes les plus proches l'un de l'autre, selon une mesure de dissimilarité, dite aussi indice d'agrégation, sont, successivement, fusionnés ensemble jusqu'à obtenir un seul cluster contenant tous les n éléments ou qu'un critère d'arrêt soit satisfait.

Algorithme 4 : Clustering hiérarchique Agglomératif

Entrées :

un ensemble $D = \{x_1, x_2, \cdots, x_n\}$ de n observations.

$d_c(\cdot, \cdot)$ Mesure de distance pour deux clusters.

```
1 début
2     C ← {{x_i}|x_i ∈ D; i = 1, ··· , n};
3     Calculer la matrice de distances entre groupes;
4     tant que (|C| > 1) et Critère d'arrêt non satisfait faire
5         Trouver (p, q) tel que : (p, q) = argmin_{{C_i,C_j}∈C; C_i≠C_j} d_c(C_i, C_j)
6         C ← (C\{C_p, C_q}) ∪ {C_p ∪ C_q}
7         si (critère d'arrêt satisfait) alors
8             retourner C ;
9         Mettre à jour la matrice de distance entre clusters;
```

Comme on rassemble d'abord les éléments les plus proches, la première itération a un indice d'agrégation faible, mais celui-ci va croître d'itération en itération. Inversement, les méthodes divisives, ou descendantes(Algorithme 5), commencent avec un seul cluster contenant tous les n éléments à classer. Puis, ittérativement, un cluster est partitionné en un certain nombre de clusters plus petits. Cette ittération s'arrête lorsqu'un critère d'arrêt soit satisfait(usuellement le nombre k de clusters désirés)[128] ou chaque cluster est assez cohérent[169] : chaque cluster contient un seul élément ou les élément de chaque cluster sont suffisamment similaire les uns aux autres.

Algorithme 5 : Clustering hiérarchique Descendant

Entrées :

un ensemble $D = \{x_1, x_2, \cdots, x_n\}$ de n observations.

$d_c(\cdot, \cdot)$ Mesure de distance pour deux clusters.

```
1  début
2      C ← {D};
3      tant que ( ∃C_x : (C_x ∈ C ∧ |C_x| > 1) ) faire
4          Trouver (p, q) tel que :
           (p, q) = argmax_{{C_i,C_j}; C_i∪C_j=C_x ∧ C_i∩C_j=Φ} d_c(C_i, C_j)
5          C ← (C\{C_x}) ∪ {C_p, C_q}
6          si (critère d'arrêt satisfait) alors
7              retourner C ;
8          fin
9      fin
10 fin
```

Le défi des méthodes divisives réside dans la façon par laquelle un large cluster est partitionné. Pour un cluster contenant n éléments, il est possible de définir $2^{n-1} - 1$ partitions exclusives. Quand n est grand, Le temps nécessaire pour examiner toute ces possibilités devient non raisonnable. Pour remédier à ce problème, les méthodes divisibles font, généralement, recours à des techniques heuristiques. Or ces dernières peuvent conduire à des partitionnements erronées. En raison de ce défi, les méthodes agglomératives sont plus appréciées et utilisées que les méthodes divisives[169].

Qu'il s'agit d'une méthode agglomérative ou divisive, il existe de multiples critères pour déterminer la distance entre deux clusters arbitraires C_i et C_j :

— Lien simple(Single linkage) défini la distance entre deux clusters C_i et C_j comme étant le minimum des distances entre les éléments de C_i et ceux de C_j et est défini par :

$$dist_{min}(C_i, C_j) = \min_{x \in C_i, y \in C_j} \{d(x, y)\}$$

Le lien simple a tendance à produire des clusters long et minces ce qui conduit, parfois, à ce que les données hétérogènes soient groupées ensemble.

— Lien Complet(Complete link) considère, comme distance entre clusters, la distance maximale entre les élément de C_i et ceux de C_j et est défini par :

$$dist_{max}(C_i, C_j) = \max_{x \in C_i, y \in C_j} \{d(x, y)\}$$

Le lien complet à tendance à former des clusters, sphériques, plus compact.

— Le lien moyen consiste à calculer la moyenne des distances entre les éléments des deux clusters et est défini par :

$$dist_{avg}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)$$

Où $d(O_1, O_2)$ dénote la distance entre les deux objets O_1 et O_2.

Les clusters obtenus, par un lien moyen, ont tendance a avoir des inerties intra-clusters(within-cluster variability) égales.

— Méthode du centre : La distance entre deux clusters C_i et C_j est définie comme étant la distance entre leurs centres de gravité respectifs $\bar{c_i}$ et $\bar{c_j}$.

— Méthode de Ward (Incremental Sum of Squares) : La distance entre deux clusters C_i et C_j est définie comme étant l'incrément dans le critère de l'erreur quadratique entre cluster :

$$\Delta(i,j) = E_{C_i \cup C_j} - E_{C_i} - E_{C_j}$$

$E_{C_k} = \sum_{x \in C_k} d^2(x, \bar{c_k})$ est la somme des carrés des distances euclidiennes entre chaque élément $x \in C_k$ et $\bar{c_k}$(le centre de C_k). Il semble que :

$\Delta(i,j) = |C_i| \cdot |C_j| \frac{d^2(\bar{c_i}, \bar{c_j})}{|C_i| + |C_j|}$. La distance pondérée entre les centre de gravité.

Lorsque appliquées à des données bien structurées, toutes ces méthodes de calcule de distances entre clusters donnent presque le même résultats et peuvent conduire à des résultats complètement différents dans le cas d'une structure complexe ou cachée[287].

Communément, Une structure d'arbre, dite **dendrogramme** est utilisée pour représenter le processus du clustering hiérarchique. Le dendrogramme montre, pas à pas, la façon dont les objets sont groupés ensemble, dans les approches agglomératives ou partitionnés dans les approches divisives(Fig.2.14).

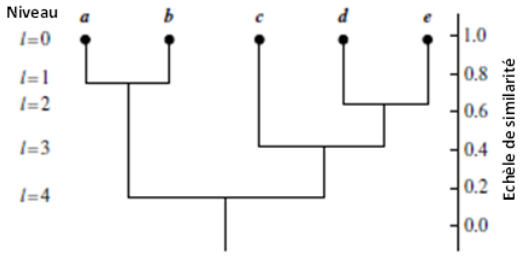


FIGURE 2.14 – Endrogramme représentant le clustering hiérarchique de l'ensemble $\{a, b, c, d, e\}$ [169]

Il est à noter que le fusionnement et la partition des clusters lors du clustering hiérarchique sont des opérations non réversibles(ne peuvent être défaites). Bien que cette rigidité est utile, puisqu'elle conduit à un coût de calcul réduit[169], elle est considéré comme le majeur inconvénient de ces méthodes[128]. Aussi, d'une part le choix d'une bonne mesure de similarité est souvent loin d'être trivial surtout en présence des données aberrantes et/ou manquantes. et d'autre par la bonne décision de fusionner ou de partitionner un cluster est, dans la plus part des méthodes de clustering hiérarchique cherchée localement par conséquent la hiérarchie des clusters obtenue peut être floue. A fin de surmonter certains de ces problèmes des méthodes de clustering probabiliste ou à base de modèle ont été proposées.

— Les méthodes à base de modèle tentent à optimiser l'adéquation des données à certains modèles mathématiques et se basent sur l'hypothèse que les données sont générées à partir d'une mixture finie de distributions de probabilités. La figure 2.15 présente un ensemble de données issues d'une mixture de trois distributions gaussiennes.

De ce fait l'homogénéité des groupes ne se base plus sur des considérations géométriques mais s'appuie sur l'analyse de la distribution de probabilité des données considérées. Et se traduit par le fait que les éléments qui sont dans un même groupe sont issues d'une

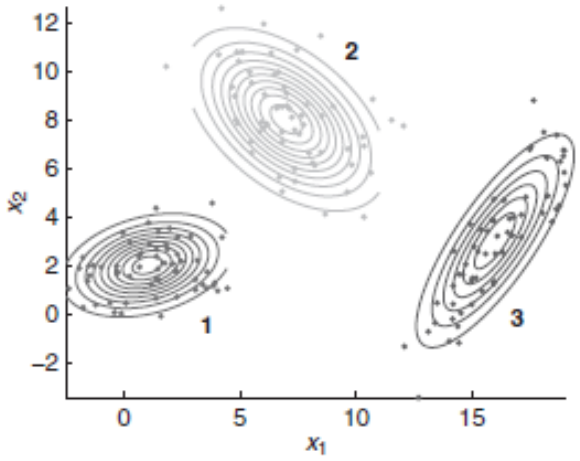


FIGURE 2.15 – Densité de probabilité de deux clusters générant un troisième par interaction

même distribution. Formellement, un modèle de mélange peut être défini comme suit : Soit un ensemble de n observations $D = \{x_1, x_2, \ldots, x_n\}$ supposées des réalisation de n variables aléatoires $X_1, X_2, \ldots, X_n (X_i \in \mathcal{R}^p;\ i = 1, 2, \ldots, n)$ dont chacune est issue d'une distribution propre à l'échantillon auquel appartient l'observation associée. Si on note par K le nombre total de ces distributions et on introduit une variable aléatoire Z qui va servir de label pour chaque observation, chaque X_i se verra associé un vecteur de dimension K, noté $Z_i = \{Z_{i1}, \ldots, Z_{iK}\}$ tel que :

$$Z_{ik} = \begin{cases} 1 & \text{si la } i^{\text{me}} \text{ observation est issue de la } k^{\text{éme}} \text{ distribution} \\ 0 & \text{sinon.} \end{cases}$$

Les variables aléatoires Z_i ont pour distribution une loi multi-nomiale de paramètres les probabilités a priori : $Z_i \sim \mathcal{M}(1; \pi_1, \ldots, \pi_K)$; où $\pi_k = P(Z_{ik} = 1); i = 1, 2, \ldots, K$ représente la probabilité qu'une observation x_i prise au hasard provient de la $k^{\text{éme}}$ distribution. On a évidemment $\sum_{k=1}^{K} \pi_k = 1$. La distribution de X_i sachant que l'observation i est issue de la $k^{\text{éme}}$ distribution est notée :

$$X_i | Z_{ik} = 1 \sim f_k(x_i)$$

où f_k est la distribution de probabilité de la $k^{\text{ème}}$ composante. En générale, ces f_k sont supposées des formes paramétriques i.e $f_k(.) \equiv f(., \theta_k)$ dont la forme fonctionnelle est connue et θ_k est le vecteur de paramètres inconnus de la distribution f. Par exemple, si on suppose que les données sont issues d'une mixture d'une distribution gaussienne, le vecteur paramètre θ_k sera composé du vecteur moyen μ_k et la matrice de covariance Σ_k et la densité aura la forme :

$$f(x; \mu_k, \Sigma_k) = \frac{exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k)\right)}{(2\pi)^{\frac{p}{2}} |\Sigma_k|^{\frac{1}{2}}}$$

La loi du coupe (X_i, Z_i) est donnée par

$$f(x_i, z_i) = P(z_i) f(x_i | z_i) = \pi_{z_i} f(x_i | z_t)$$

Où $\pi_{z_i} = P(Z_i = z_i) (\pi_{z_i} = \pi_k \, si \, z_{ik} = 1 \, \text{par exemple})$, est la densité de X_i conditionnellement à $Z_i = z_i$.

La distribution de X_i peut donc s'écrire sous la forme :

$$f(x_i;\theta) = P(Z_{i1}=1)\cdot f(x_i;\theta_1) + \ldots + P(Z_{iK}=1)\cdot f(x_i;\theta_K)$$

$$= \sum_{k=1}^{K} P(Z_{ik}=1)\cdot f(x_i;\theta_k)$$

$$= \sum_{k=1}^{K} \pi_k f(x_i;\theta_k).$$

Où $\theta = (\pi_1,\ldots,\pi_K,\theta_1,\ldots,\theta_K)$ représente le vecteur de paramètre du modèle.

Le problème de clustering consiste à prédire les labels $z_1,\ldots,z_n$ des observations $x_1,\ldots,x_n$ sur la seule connaissance des valeurs prises par les p variables explicatives sachant que l'on ne connaît pas nécessairement le nombre K des composantes(distributions) du mélange. Évidemment le vecteur $z = (z_1, z_2, \ldots, z_n)$ défini une partition $C = \{C_1, C_2, \ldots, C_K\}$ des données $x_1, x_2, \ldots, x_n$. Si z est observée, les clusters devront être connus et les données de chaque cluster C_k sont supposées provenir d'une distribution de densité gaussienne $\phi(\cdot;\,\alpha_k)$ avec $\alpha_k = (\mu_k, \sigma_k)$ pour $k = 1, 2, \ldots, K$. Par conséquent, la probabilité conditionnelle sur z aurait une forme permettant une inférence facile. Malheureusement, z n'est en générale pas connu et doit être estimé. Le vecteur z peut être estimé conjointement avec K, le nombre total des composante, et $\theta = (\pi_1, \pi_2, \ldots, \pi_K, \theta_1, \theta_2, \ldots, \theta_K)$. Cette estimation peut se faire par plusieurs approches. Dont la plus ancienne est la méthodes des moments, proposée par Pearson[309] pour estimer les cinq paramètres $(\mu_1, \mu_2, \sigma_1, \sigma_2, \pi)$ d'un modèle de mixture gaussienne unidimensionnelle, qui nécessite la résolution d'une equation polynômiale de $9^{\text{éme}}$ degré. On compte, aussi, des techniques graphiques, des méthodes Bayésiennes et les méthodes de maximisation de vraisemblance. Une concise description de certaines méthodes d'estimation des paramètres du modèle de mixture finie est donnée dans [127]. L'approche maximisation de vraisemblance est la plus utilisée et consiste, à maximiser la vraisemblance qui peut s'écrire, sous l'hypothèse de l'indépendance des observations, comme suit :

$$\begin{aligned}\mathscr{L}(x_1, x_2, \ldots, x_n;\theta) &= \prod_{i=1}^{n} f(x_i|\theta). \\ &= \prod_{i=1}^{n} \left\{ \sum_{k=1}^{K} \pi_k \cdot f(x_i;\theta_k) \right\}.\end{aligned}$$

Dans le cas particulier d'une loi discrète, cela se ramène à :

$$\mathscr{L}(x_1, x_2, \ldots, x_n;\theta) = P_\theta(X_1 = x_1, \ldots X_n = x_n) = \prod_{i=1}^{n} P_\theta(X_i = x_i)$$

Il s'agit, donc, de trouver un estimateur $\hat{\theta}$ tel que :

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \mathscr{L}(x_1, x_2, \ldots, x_n;\theta)$$

Cette estimation peut être aborder par la procédure classique de différentiation de L par rapport à θ. Ce la revient à trouver un θ_0 solution du système non linéaire des équations normales :

$$\frac{\partial \mathscr{L}}{\partial \theta_i} = 0.$$

puis d'examiner les dérivées de 2^{ieme} ordre pour vérifier que le θ_0 est bien un point maximum et non pas autre point stationnaire. L'intérêt de cette méthode réside dans le fait que sous des conditions générales peu restrictives, l'estimateur obtenu possède les propriétés suivantes :

— il est asymptotiquement non biaisé ;
— il a asymptotiquement la plus faible variance ;
— il suit asymptotiquement une distribution normale.

Usuellement les équations normales ne sont pas explicitement et analytiquement solvables et la fonction de vraisemblance, pour un mélange de distributions est souvent non bornée($\nexists M$tel que $L(.;\theta) \leq M$) tous comme $\frac{\partial \mathscr{L}}{\partial \theta_i}$. Pour contourner ces problèmes, on fait souvent recours à des algorithmes itératifs de recherche de maximum d'une fonction. L'algorithme espérance-maximisation(Expectation-Maximisation (EM)), proposé par Dempster et al en 1977[103], est, usuellement, utilisé pour estimer les paramètres des densités de probabilité de la mixture finie. Pour obtenir un estimateurs de maximum de vraisemblance, l'algorithme EM utilise la distribution conjoint d'une observation x et la variable latente z inconnue qui indique l'appartenance à chaque cluster. Le couple (x, z) représente une donnée complète et le log-vraisemblance des données complètes et dit log-vraisemblance complet ou vraisemblance de classification et est donné par :

$$\begin{aligned}\mathscr{L}_C(\theta_1, \theta_2, \ldots, \theta_K; z_1, z_2, \ldots, z_n|D) &= \prod_{i=1}^{n} f(x_i; z_i|\theta). \\ &= \sum_{i=1}^{n} log\left(\sum_{k=1}^{K} z_{ik}\pi_k f(x_i;\theta_k)\right) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} log\left(\pi_k f(x_i;\theta_k)\right)\end{aligned}$$

La probabilité $P(z_{ik} = 1|x_i;\theta)$ est définie comme étant :

$$\tau_{ik}(\theta) = \frac{\pi_k f(x_i;\theta_k)}{f(x_i;\theta)}$$

Pour alléger la notation $\tau_{ik}(\theta)$ sera noté τ_{ik} lorsque le paramètre θ est clair à partir du contexte. Les deux vraisemblances, régulière et complète, sont liées comme suite :

$$\begin{aligned}\mathscr{L}_C(\theta_1, \theta_2, \ldots, \theta_K; z_1, z_2, \ldots, z_n|D) &= \sum_{i,k} z_{ik} log\left(\pi_k f(x_i;\theta_k)\right) \\ &= \sum_{i,k} z_{ik} log\left(\tau_{ik} f(x_i;\theta)\right) \\ &= \sum_{i,k} z_{ik} log(\tau_{ik}) + \sum_{i,k} z_{ik} log\left(f(x_i;\theta)\right) \\ &= \sum_{i,k} z_{ik} log(\tau_{ik}) + \sum_{i=1}^{K} log\left(f(x_i;\theta)\right) \\ &= \sum_{i,k} z_{ik} log(\tau_{ik}) + \mathscr{L}(x_1, x_2, \ldots, x_n;\theta)\end{aligned}$$

Où $\sum\limits_{i,k} z_{ik} log(\tau_{ik})$ peut être reformulé comme suit :

$$\begin{aligned}\sum_{i,k} z_{ik} log(\tau_{ik}) &= \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} log(P(z_{ik} = 1|x_i;\theta)) \\ &= \sum_{i=1}^{n} log\left(P(z_{ik} = 1|x_i;\theta)\right) \\ &= log\left(P(Z|X;\theta)\right)\end{aligned}$$

de ce fait la relation entre $\mathscr{L}$ et $\mathscr{L}_C$ peut être réécrite comme suit :

$$\mathscr{L}(x_1, x_2, \ldots, x_n; \theta) = \mathscr{L}_C(\theta_1, \theta_2, \ldots, \theta_K; z_1, z_2, \ldots, z_n|D) - log\left(P(Z|X;\theta)\right)$$

Or le log-vraisemblance ne peut être évalué que si les z_{ik} sont connus, cependant, il est possible d'estimer la valeur de log-vraisemblance en considérant l'espérance conditionnelle à une valeur courante de θ, on aura donc :

$$\begin{aligned}\mathscr{L}(x_1, x_2, \ldots, x_n; \theta) = &\overbrace{\mathbb{E}_{Z\sim P(\cdot|X;\theta^t)} + \left[\mathscr{L}_C(\theta_1, \theta_2, \ldots, \theta_K; z_1, z_2, \ldots, z_n|D)\right]}^{Q(\theta,\theta^{(t)})} \\ &+ \underbrace{\mathbb{E}_{Z\sim P(\cdot|X;\theta^t)}\left[-log(P(Z|X;\theta))\right]}_{H(\theta,\theta^{(t)})}\end{aligned}$$

$Q(\theta, \theta^{(t)})$ est l'espérance conditionnelle de la vraisemblance complète et $H(\theta, \theta^{(t)})$ l'entropie. En définissant l'incrément du log-vraisemblance comme suit :

$$\Delta\mathscr{L} = \mathscr{L}(\theta^{(t+1)}; X) - \mathscr{L}(\theta^{(t)}; X)$$

$\theta^{(t+1)} = \text{argmax}_\theta\, Q(\theta^{(t)})$ augmente aussi le log-vraisemblance :

$$\Delta\mathscr{L} = \underbrace{\left[Q(\theta^{(t+1)}, \theta^{(t)}) - Q(\theta^{(t)}, \theta^{(t)})\right]}_{\geq 0 \text{ par la définition de léteration t+1}} - \underbrace{\left[H(\theta^{(t+1)}, \theta^{(t)}) - H(\theta^{(t)}, \theta^{(t)})\right]}_{\leq 0 \text{ par l'énegalité de Jensen}}$$

alors, Par conséquence, il est possible de maximiser la vraisemblance en optimisant $Q(\theta, \theta^{(t)})$ dit vraisemblance pondérée. Pour le problème de mélange de loi, on a

$$\begin{aligned}Q(\theta, \theta^{'}) &= \mathbb{E}_{P(Z|X;\theta')} + \left[\mathscr{L}_C(\theta_1, \theta_2, \ldots, \theta_K; z_1, z_2, \ldots, z_n|D)\right] \\ &= \sum_{i,k} P(z_{ik} = 1|x_i; \theta^{'}) log\left(\pi_k f(x_i; \theta_k)\right) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}(\theta^{'}) log\left(\pi_k f(x_i; \theta_k)\right)\end{aligned}$$

On peut donc définir l'algorithme EM de la manière suivante :

Algorithme 6 : L'algorithme EM

Entrées : $D = \{x_1, x_2, \ldots, x_n\}$ n observations suivant une loi $f(x, \theta)$;
Output : θ maximisant le log-vraisemblance.;

```
début
    t ← 0;
    θ ← θ^0 Valeurs choisies aléatoirement;
    tant que (pas de Convergence) faire
        Étape E- Évaluation de l'espérance : Q(θ, θ^t);
        Étape M- Maximisation :
            θ ← argmax_θ (Q(θ, θ^t));
        t ← t + 1;
    fin
    retourner θ^t;
fin
```

A fin d'illustrer le fonctionnement de l'algorithme EM considérons le modèle gaussien avec une matrice de covariance Σ commune et des moyennes μ_k différentes. La densité du mélange est donc :

$$\begin{aligned} f(x_i, \theta) &= \sum_{k=1}^{K} \pi_k f(x_i; \theta_k) \\ &= \sum_{k=1}^{K} \pi_k \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x_i - \mu_k)^T \Sigma^{-1}(x_i - \mu_k)\right) \end{aligned}$$

A l'étape E-step, les probabilités postérieures τ_{ik} sont calculées avec la valeur courante de $\theta^{(t)}$, puis l'étape $M - step$ maximise $Q(\theta, \theta^{(t)})$ dont la forme est la suivante :

$$\begin{aligned} Q(\theta, \theta^{(t)}) &= \sum_{i,k} \tau_{ik} log(\pi_k) - \sum_{i,k} \tau_{ik} log\left((2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}\right) - \frac{1}{2} \sum_{i,k} \tau_{ik}(x_i - \mu_k)^T \Sigma^{-1}(x_i - \mu_k) \\ &= \sum_{i,k} \tau_k log(\pi_k) - \underbrace{\frac{np}{2} log(2\pi)}_{\text{term constant}} - \frac{n}{2} log(|\Sigma|) - \frac{1}{2} \sum_{i,k} \tau_{ik}(x_i - \mu_k)^T \Sigma^{-1}(x_i - \mu_k) \\ &= \sum_{i,k} \tau_k log(\pi_k) - \frac{n}{2} log(|\Sigma|) - \sum_{i,k} \tau_{ik} \left(\frac{1}{2}(x_i - \mu_k)^T \Sigma^{-1}(x_i - \mu_k)\right). \end{aligned}$$

Où $t_k = \sum_{i=1}^{K} \tau_{ik}$. L'étape M-step qui maximise cette expression par rapport à θ, applique les mises-ajour, suivantes, définissant θ^{t+1}

$$\begin{aligned} \pi_k^{(t+1)} &= \frac{t_k}{n} \\ \mu_k^{(t+1)} &= \frac{\sum_i \tau_{ik} x_i}{\tau_k} \\ \Sigma^{(t+1)} &= \frac{1}{n} \sum_k W_k \\ W_k &= \sum_i \tau_{ik}(x_i - \mu_k)(x_i - \mu_k)^T \end{aligned}$$

Il est à noter que les valeurs optimales pour π_k, μ_k, et Σ sont obtenues on annulant les dérivées partielles du Lagrangien :

$$\mathcal{L}(\theta) = Q(\theta, \theta') + \lambda \left(\sum_k \pi_k - 1\right)$$

avec

$$\begin{aligned} Q(\theta, \theta') &= \max_{\theta} \sum_{i,k} \tau_{ik}(\theta') log\left(\pi_k f(x_i, \theta_k)\right) \\ &= \sum_k log\left(\pi_k \sum_i \tau_{ik}\right) - \frac{np}{2} log(2\pi) - \frac{n}{2} log(|\Sigma|) - \frac{1}{2} \sum_{i,k} \tau_{ik}(x_i - \mu_k)^T \Sigma^{-1}(x_i - \mu_k). \end{aligned}$$

qui doit être maximisé sujet à $\sum \pi_k = 1$.

— La plus part des méthodes de clustering à base de distance entre objets ne permettent de trouver que les clusters de forme sphérique et présentent des difficultés à découvrir les clusters de forme arbitraire[168]. A fin de palier à ce problème, des méthodes à base de densité ont été développées dans le cadre de l'apprentissage non supervisé. Dans ces méthodes, non paramétriques, un cluster est considéré comme étant une région, de forme quelconque, contigüe à haute densité. Les clusters sont séparés les uns des autres par des régions contigües de faible densité. Les objet se trouvant dans des régions de séparation sont généralement considérés comme objets aberrants ou bruit. La figure 2.16 présente une illustration d'un clustering à base de densité montrant trois groupes distinguables indiqués par des triangles, des points et des rectangles où la dissimilarité entre les points est donnée par la distance euclidienne.

FIGURE 2.16 – Clustering à base de densité[84]

Il est à noter que ces méthodes n'exigent pas que le nombre de clusters soit fournie en entrée et ne font aucune hypothèse sur la densité sous-jacentes ou sur la variance intra-groupes pouvant existée dans les données. En conséquence les clusters ne sont pas nécessairement constitués par des points ayant une haute similarité intra-groupe, mesurée par une distance et peuvent avoir des formes arbitraire et sont souvent dits groupes naturels. D'autre part, ces méthodes reposent sur les concepts fondamentaux de densité, de voisinage d'un objet, de point noyau, de point limite, d'accessibilité et connectivité. Étant donnée un objet x_i d'un ensemble X, le $\epsilon-$voisinage de x_i(Fig. 2.17) noté $V_\epsilon(x_i)$ est l'ensemble d'objets de X distants d'au plus ϵ de x_i :

$$V_\epsilon(x_i) = \{y \in X | d(x_i, y) \leq \epsilon\}$$

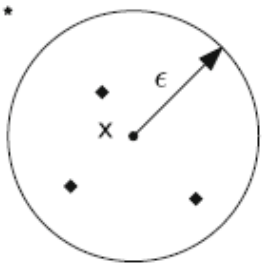


FIGURE 2.17 – $\epsilon-$voisinage

Où $d(\cdot, \cdots)$ représente une certaine fonction de distance. On distingue deux types d'objets ou de points. A savoir les points se trouvant à l'intérieur d'un cluster et ceux qui se trouve au frontières. Les premiers, dits points noyaux comptent dans leurs $\epsilon-$voisinage au-moins $MinPts$(seuil prédéfini) points. Le $\epsilon-$voisinage des seconds, dits points limites, à tendance à contenir un nombre de point nettement inférieur à $MinPts$.

Un point limite doit appartenir à un $\epsilon-$voisinage d'un point noyau. Un point qui n'est ni un point noyau ni un point limite est dit point aberrant(bruit)(Fig. 2.18).

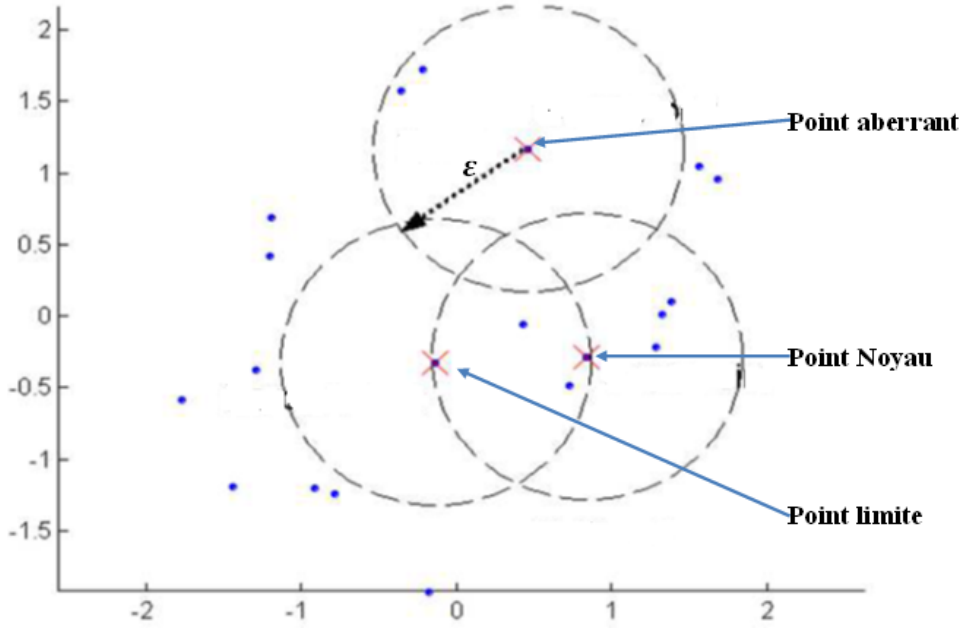
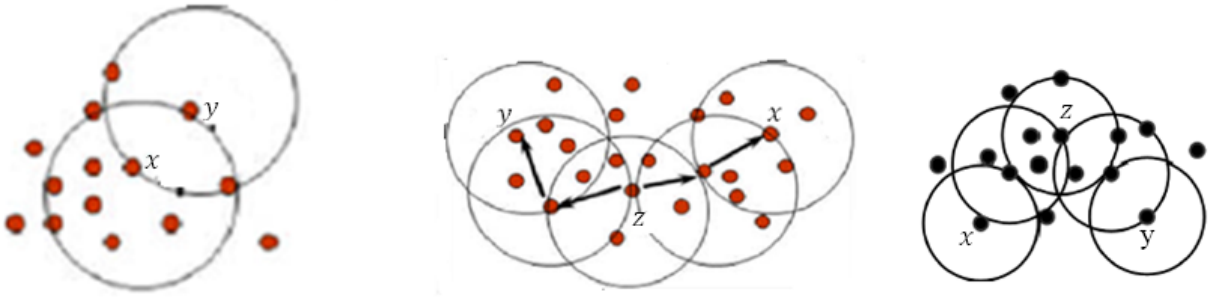


FIGURE 2.18 – types de points. $\epsilon = 1$ et $MinPts = 4$

On dit qu'un point y est "directement densité-accessible" à partir d'un point noyau x si $y \in V_\epsilon(x)$(Fig. 2.19(a)). Ainsi y est directement accessible à partir de x si le $\epsilon-$voisinage de x contient au moins $MinPts$ points dont y. La densité-accessibilité est donnée par la closure transitive de la "directe densité-accessibilité"(Fig. 2.19(b)). Autrement dit y est dit "densité accessible" à partir de x s'il existe une chaine $x_1, x_2, \ldots, x_m$ dans X tell que $x_1 = y$ et $x_m = x$ et $\forall i = 1, \ldots, m-1$ avec x_{i+1} est directement accessible par x_i. Le fait qu'un objet y soit densité-accessible à partir de l'objet x est généralement noté par $y >_D x$. La relation "densité-accessible" est :



(a) Directement densité-accessiblelité (b) Densité-accessibilité (c) Densité-connectivité

FIGURE 2.19 – Densité-accessibilité et Densité-connectivité

1. transitive puisque la densité-accessibilité est donnée par la closure transitive de la "directe densité-accessibilité".
2. symétrique pour les objets noyaux, puisque une chaine de x à y peut être inversée si y est un noyau.
3. réflexive pour les objet noyaux.

Deux points x et y sont dit "densité-connectés" s'il existe un troisième point z à partir du quel x et y sont "directement densité accessible"(Fig. 2.19(c)). La relation "densité accessible" est symétrique, réflexive pour les noyaux, et non transitive. Mais si x est densité-connecté à y via z_1 et y est densité-connecté à t via z_2 alors x est densité-connecté à t si et seulement si z_1 est densité-accessible à partir de z_2 ou z_2 est densité-accessible à partir de z_1.

Un ensemble densité-connecté est défini comme étant un ensemble d'objet densité-connectés. Il est à noter qu'un point limite est toujours densité-accessible à partir d'un point noyau et il est possible qu'il ne le soit pas à partir d'un autre point limite.
Un point aberrant est un point non-noyau qui n'est jamais densité-accessible à partir d'un autre point[323].

Un sous ensemble $C \subseteq X$ est dit cluster si pour tout $x \in C$ et $y \in C$ les deux conditions, de maximalité et de connectivité, suivantes sont vérifiées[167] :

— $\forall x, y \in X$ si $x \in C$ et y est densité-accessible à partir de x alors $y \in C$.
— $\forall x, y \in C$, x et y sont densité-connecté.

L'ensemble des objets densité-accessibles à partir d'un objet noyau constitue un densité-cluster. La construction d'un tel cluster revient à rechercher un objet noyau puis à agglomérer autour de ce noyau, tous les objets densité-accessibles par ce noyau. Une décomposition d'un ensemble à base de densité(Density based decomposition - DBD) est définie par les conditions suivantes[347] :

1. $DBD = \{C_1, \ldots, C_k, N\}; k \geq 0$.
2. $C_1 \cup, \ldots \cup C_k \cup N = X$.
3. $\forall i \leq k$: C_i est un ensemble densité-connecté dans X. Tous les éléments de la décomposition, sauf un, sont densité-connectés
4. S'il existe C tel que C est densité-connecté dans X alors il existe un $i \leq k$ tel que $C = C_i$. Tous les ensemble densité-connectés de X doivent être dans la décomposition.
5. $N = X \diagup (C_1, \ldots, C_k)$. L'ensemble N contient les objets qui n'appartiennent à aucun ensemble densite-connecté de la décomposition. Cette ensemble n'est pas densité connecté est peut être vide.

Cette décomposition est caractérisée par le fait qu'il ne peut avoir de chevauchement entre deux clusters qu'au niveau des objets se trouvant aux frontières des deux clusters. Si $C_1,\ C_2 \in DBD$ alors $\forall x \in C_1 \cap C_2$ on a x n'est pas un objet noyau(Fig. 2.20).

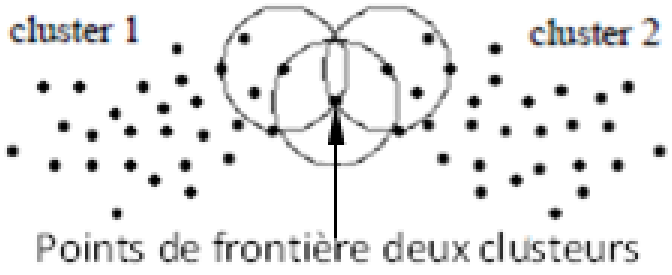


FIGURE 2.20 – Chevauchement de deux clusters

Plusieurs algorithmes de regroupement à base de densité ont été proposés dans la littérature. Globalement ces algorithme se distinguent par :

1. La façon dont la densité est estimée,
2. La façon dont la notion de connectivité est définie
3. La façon dont le processus de découverte des composantes connexes du graphe induit est implémenté.
4. La natures des structures de données utilisées.

Selon [84], le regroupement à base de densité a été probablement introduit pour la première fois par Wishart en 1969. Son algorithme[414] consistait en six étapes décrites comme suit(Algorithme 7) :

Algorithme 7 : Algorithme de Wishart de clustering à base de densité

1 Choisir une distance seuil r et une densité seuil k;

2 Calculer la matrice de distance triangulaire;

3 Évaluer les densités k_i pour chaque point(donnée) définie comme étant le nombre de point distant d'au plus r du point i;

4 Enlever les point ayant $k_i < k$;

5 Grouper les points denses restant($k_i > k$) par un simple lien et calculer les centres(modes);

6 Réaffecter chaque point non dense à un cluster approprié selon un certain critère.

En 1975, Hatigan a proposé la notion de **density countour cluster** à un niveau ϵ qui est une généralisation du concept de densité-cluster. Étant donnés un ensemble $X \subset \mathcal{R}^p$ et $f(x)$ la densité à chaque point x, Un cluster "densité contour" à un nivaux ϵ est le sous ensemble C satisfaisant

$$f(x) \geq \epsilon \, \forall x \in C.$$

La densité à l'intérieure de C est supérieure à ϵ, mais elle est inférieure à ϵ, quelque part, sur chaque chemin connectant $x \in C$ à un point y, se trouvant à l'extérieur de C. Le cluster C est conforme à l'exigence, informelle, qu'un cluster doit être une région à haute densité entourée par une région à faible densité[172]. La figure 2.21 montre un exemple d'un cluster densité-contour, C, d'un niveau 3. Toutes les densités à l'intérieur de C sont supérieures à 3. Et quelque densités sur chaque chemin entre un $x \in C$ et un y à l'extérieur de C sont inférieures à 3.

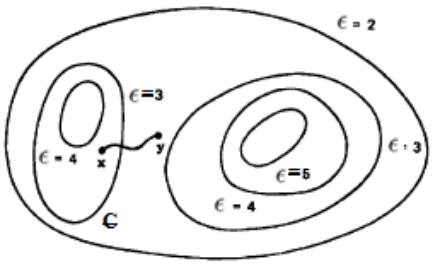


FIGURE 2.21 – Clusteurs densité-contour[172].

Tout comme Wishart, Artigan propose une version de clustering à lien simple avec laquelle il construit en ensemble connecté maximal de densité supérieure à un seuil donné. En 1996, Ester et Al proposèrent [126] leur algorithme baptisé DBSCAN(Density-Based

Spatial Clustering of Applications with Noise). Cet algorithme choisit, aléatoirement un objet x est cherche tous les objets densité-accessibles à partir de cet objet. Si le nombre des objets ainsi trouvés est supérieure au seuil MinPts, ces objets sont regroupés ensemble pour former un cluster. Dans le cas contraire (x est un point limite), l'algorithme choisit un entre objet(Algorithme 8).

Algorithme 8 : DBSCAN

Entrées :

ϵ Rayon de voisinage ;

MinPts Densité seuil ;

$X = \{x_1, x_1, \ldots, x_n\}$ ensemble de n données à classer;

Output : $\mathscr{C} = \{C_1, C_2, \ldots, C_k\}$ ensemble de clusters;

```
1  début
2      C ← Φ;
3      i ← 0;
4      tant que (|X| > 0) faire
5          Choisir aléatoirement un point x;
6          X ← X/x;
7          Grouper dans une classe C_i tout les point densité-accessible à
             partir de x;
8          si (|C_i| > MinPts) alors
9              C_i ← C_i ∪ {x} ;
10             C ← C ∪ {C_i} ;
11             X ← X/C_i;
12             i ← i + 1 ;
13 retourner C;
```

DBSCAN n'est en mesure de regrouper que les objets point et se base essentiellement sur la notion de voisinage à base de distance. Afin de remédier à ces carences, Sander, dans [348],a proposé une généralisation de la notion de clustering à base de densité introduite par [126] comme suit : En premier lieu il proposa de remplacer la notion de voisinage à base de distance par le voisinage à base d'un prédicat binaire qui doit être symétrique et réflexif, noté $NPred(p, q)$ où p,q sont deux objets de l'ensemble considéré X, ainsi le $NPred$-voisinage d'un objet p est défini comme étant l'ensemble des objets de X satisfaisant le prédicat $NPred$:

$$V_{NPred}(p) = \{q \in X | NPred(p, q)\}.$$

Deuxièmement, au lieu de se contenter de compter le nombre des objets d'un voisinage, il propose d'utiliser prédicat unaire plus générale, noté MinWeight, pour décider si un cluster est dense ou pas. On dira qu'un ensemble $V \subseteq X$ d'objets a un poids minimum si $MinWeight(V)$ est vérifié. Et finalement, il propose de considérer non seulement les objets "point" mais aussi les objets spatiaux étendus tel que les polygones pour lesquels, les prédicats $NPred$ et $MinWeight$ semblent être, respectivement, plus naturels que l'ϵ-voisinage et la cardinalité minimale $MinPts$. Ainsi DBSCAN fut généralisé par GDBSCAN. [31, 51, 304, 360] présentent des surveilles et des comparaisons des plus connus algorithmes de clustering à base de densité.

— Toutes les méthodes de clustering classiques, calculent des similarité à base de distance, le plus souvent Euclidienne, entre les centroïdes des clusters puis tentent à regrouper les données selon leurs indices de disimilarité. Les méthodes de clustering à base de grille (Grid-based methods), Initialement introduit par Warnekar et Krishna[407], pouvant être hiérarchique ou par partition, quant à elles suivent une approche axée plutôt sur l'espace spatial des données et consistent à organiser cet espace sous forme d' une structure de grille multi-niveaux. L'espace des données est partitionné en un nombre fini de cellules, indépendantes de la distribution des données à regrouper en utilisant des informations statistiques sur données. Chaque cellule non vide est pondérée par le nombre de données qu'elle contienne. Et peut être un cube, une région, un hyper-rectangle et est caractérisée par :

1. Le nombre de points qu'elle contienne.
2. La moyenne de toutes les valeurs d'attributs dans la cellule.
3. La variance de toutes les valeurs d'attributs dans la cellule.
4. Le minimum et le maximum des valeurs d'attributs dans la cellule.
5. La distribution que suivent les objets de cette cellule. Les types, potentiels, de distributions sont : La loi uniforme, la loi normale et la loi exponentielle.

Le regroupement est effectué par l'agrégation itérative des cellules denses adjacentes pour former des clusters. Ces derniers sont former selon deux approches. La première consiste à calculer la densité de chaque cellule, puis fusionner des cellules adjacentes de façon à ce que le bloc résultant soit suffisamment dense et uniforme. La deuxième consiste à détecter les limites entre zones à haute densité et celles à faible densité, puis reconstituent des clusters à la base de ces limites. Le grand avantage des méthodes de clustering à base de grille est la réduction significative de la complexité temporaire, notamment pour les très grandes masses de données vue que dans la plus part des applications le nombre de cellule est nettement inférieur au nombre des données. selon [161], Typiquement, un algorithme de clustering à base de grille se compose des étapes basiques suivante(Algorithme 9) :

Algorithme 9 : Algorithme de clustering à base de grille typique

1 Création de la structure de grille :Diviser l'espace des données en un nombre fini de cellule;
2 Calculer la densité pour chaque cellule;
3 Trier le cellule selon leurs densité;
4 Identifier les centre des clusters;
5 Explorer le voisinage des cellules.

Bien que les algorithmes de clustering à base de grille se sont révélés plus rapides que ceux basés sur la distance et ceux à base de densité, Ils sont, globalement, sensibles aux défis suivants[8]

1. La non-uniformité. L'utilisation d'une seul grille uniforme inflexible peut s'avérer insuffisant pour atteindre la qualité ou l'efficacité désirées pour le regroupement des données à distribution fortement irrégulière.
2. Localité : En présence des variations locales, dans la forme et la densité des données, l'efficacité du regroupement sera limité par la taille prédéfinie des cellules, les cellules frontières et le seuil de densité des cellules significatives.
3. La dimensionalité : La performance du regroupement dépend de la taille des structures des grilles qui est proportionnelle à la dimension de l'espace des données. Dans le cas de forte dimension, la taille de la grille devient très importante et le filtrage des bruits

et la sélection des attributs pertinents deviennent très difficiles. En conséquence, les approches à base de grille pourraient ne pas être scalables.

Pour faire face au problème de non-uniformité des algorithmes adaptatifs, tel que AMR(Adaptive Mesh Refinement Clustering)[249] et MAFIA(Merging of Adaptive Finite IntervAls)[157], recouvrant l'espace des données par des grilles flexibles à multi-résolution, ont été proposées. L'algorithme de construction d'une grille flexible[224] commence par un hyper-cube contenant toutes les données et à chaque étape, la cellule contenant le plus grand nombre de données est divisée en deux sous-cellules par un hyper-plan aléatoires ayant des directions uniforme jusqu'à à ce qu'un nombre M, de cellules non vides soit atteint. Ainsi, à la fin de cette itération, les données sont condensées en un ensemble de polyèdres pondérés. A chaque itération, les données son codées par un mot binaire (codeword) qui correspond à une $H-$représentation des cellules. Chaque données appartienne à une cellule particulière. Le processus de division se présente naturellement sous forme d'un arbre binaire comme illustré par la figure 2.22.

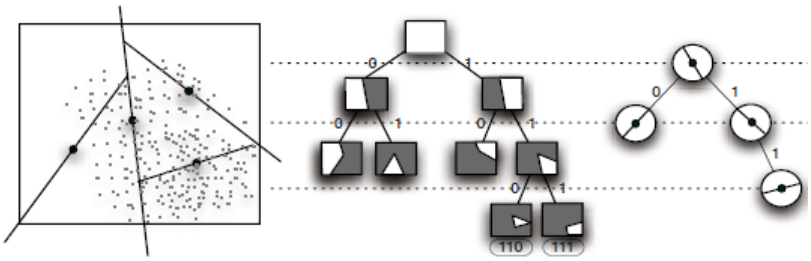


FIGURE 2.22 – Dommaine de données, l'Arbre de cellules et l'arbre des hyper-plan[224].

Une grille flexible est une réalisation particulière d'un processus stochastique construit iterativement durant le raffinement des cellules et sa résolution est automatiquement adaptée à la densité locale. Les parties les plus fine de la grille se trouvent dans les régions à forte population. Le résumé obtenu a de petites cellules à haute densité dans les régions denses et de grandes cellules à faible densité dans les régions peu peuplées.

Il est à rappeler qu'un hyperplan dans $\mathcal{R}^d$ est l'ensemble

$$H = \{x \in \mathcal{R}^d | < u \cdot x >= t \in \mathcal{R}\}$$

où u est un vecteur non nul de $\mathcal{R}^d$, t est un scalaire($t \in \mathcal{R}$) et $< \cdot >$ dénote le produit scalaire. u est dit vecteur direction de H et t son décalage(offset). Un hyperplan uniforme aléatoire peut être obtenu en prenant un vecteur, gaussien à d dimensions, aléatoire normalisé à 1 comme vecteur direction. Un polyèdre $P \subset \mathcal{R}^d$ est l'intersection d'un nombre fini de demi-espaces fermés :

$$P = \{x \in \mathcal{R}^d | < a^i \cdot x >\leq b_i, 1 \leq i \leq m\} \text{ pour certains } a^i \in \mathcal{R}^d, b_i \in \mathcal{R}.$$

Un polyèdre admet deux représentations[435] :

1. Une $H-$représentation le décrivant comme solution d'un système d'inégalités :

$$P = P(A, z) = \{x \in \mathcal{R}^d : Ax \leq z\} \text{ pour certaine } A \in \mathcal{R}^{m \times d}, z \in \mathcal{R}^m.$$

Où $Ax \leq z$ représente une réécriture simplifiée d'un système d'inégalités : $a_1 x \leq z_1, \ldots, a_m x \leq z_m$, où les $a_1, a_2, \ldots, a_m$ sont les lignes de A et les $z_1, z_2, \ldots, z_m$ sont les composantes de z.

2. Une $V-$représentation qui décrit un polyèdre comme étant une somme de Minskowski : $P = cone(U) + conv(V)$ où cone(U) et conv(V) sont deux ensemble générés, respectivement, par une combinaison canonique et convexe des vecteurs des deux ensembles finis U et V.

Algorithme 10 : Algorithme de construction d'une grille flexible[224]

Entrées :

$X = \{x_1, x_2, \ldots, x_n\}$;

D : Hyper-rectangle contenant X ;

M : taille désirée (nombre de cellules);

Output :

$S = \{(S_1, p_1), \ldots, (S_M, p_M\}$ un ensemble de polyèdre;

début
  $S_0 \leftarrow \{(D, 1, []\}$ région initiale contenant toutes les données;
  $T \leftarrow$ Arbre binaire d'hyperplan;
  $NR \leftarrow$ liste vide (NR : neighborhood relation between the cells);
  tant que $(|S_0| < M)$ **faire**
    $(C, p, \omega) \leftarrow$ cellule de S_0 avec un p maximal et un codeword ω ;
    $H_{split} \leftarrow$ un hyperplan aléatoire passant par le centre de C ;
    Ajouter H_{split} à T au noeud d'index ω ;
    $\{(C_1, p_1, \omega_1), (C_2, p_2, \omega_2)\} \leftarrow$ les cellules résultants de la division de C avec l'hyperplan H_{split};
    Remplacer (C, p, ω) dans S_0 par les éléments non vides de $\{(C_1, p_1, \omega_1), (C_2, p_2, \omega_2)\}$;
    Mettre à jour la relation de voisinage NR des nouvelles cellules remplaçants C
  fin
  Extraire S à partir de S ;
fin

En ce qui concerne le problème de localité, des algorithmes de décalage d'axes (axis-shifting) ont été introduits. Ces algorithmes adoptent des stratégie de partitionnement basées sur le décalage des axes pour identifier les régions de haute densité. Par exemple dans [251] les auteurs ont proposé $ACICA^+$(Axis-shifted Crossover-Imaged Clustering Algorithm) un algorithme(Algorithme 11) qui consiste à décaler les coordonnées des axes, dans chaque dimension, par un pas égale à un demi de la largeur d'une cellule créant ainsi une nouvelle structure de grille. Ce décalage, permettant la reconnaissance des régions fortement dense adjacentes à des cellules à faible densité, est considéré comme un ajustement dynamique de la taille des cellule originales et la "densité seuil" des cellules significatives. Les expériences ont montré que cet algorithme est moins influencé par la taille des cellules que les autres algorithmes à base de grille. Globalement, $ACICA^+$ suit les étape suivantes(Algorithme 11) :

Algorithme 11 : Algorithme$ACICA^+$ [251]

1 Construction de la première structure de grille ;
2 Identification des cellules Significatives ;
3 Transformation de la grille ;
4 Identification des cellules Significatives ;
5 Génération de la nouvelle structure et un pré-clusterng.;
6 Génération du clustering final. ;

Pour remédier au problème de la "malédiction de dimensionnalité" plusieurs algorithmes ont été proposés parmi les quels ont cite CLIQUE et OptiGrid et leurs variantes. CLIQUE(Clustering In QUEst), proposé par Agrawal et al[14], est un algorithme combinant l'approche de clustering à base de densité avec celle à base de grille et consiste à sélectionner, automatiquement, des sous-espaces appropriés au lieu de considérer l'espace des attributs tout entier. OptiGrid(Density-Based Optimal Grid Partitioning) à été proposé par Hinneburg et Al[177] pour remédier à plusieurs aspect de la malédiction de dimensionalité, tel que le filtrage de bruit, la scalabilité du processus de construction et la sélection des attributs les plus pertinents, en optimisant la fonction de densité.

Une revue des algorithmes classiques de clustering à base de grille et notamment ceux qui ont été proposés pour remédier aux défis de la non uniformité, de localité et de dimensionalité est présentée dans [8].

2.4.2 Les Méthodes prédictives

Les méthodes prédictives visent à prédire la valeur d'un attribut particulier, dit attribut cible, en se basant sur un ensemble d'attributs dits explicatifs. L'attribut ou la variable cible peut être discrète(classification) comme elle peut être continue(régression). Issues de plusieurs disciplines tel que les statistiques, l'apprentissage automatique, l'intelligence artificielle les méthodes prédictives fournissent, aux décideurs et analystes, un moyen de prédire des événements et/ou des comportements futures via un ensemble d'algorithmes appliqués à des ensembles de données pertinentes. Cette catégorie de tâches englobe les méthodes de : classification, prédiction, régression et d'analyse des séries temporaires.

2.4.2.1 Classification

La classification, dite aussi classement dans la littérature française, vise la prédiction d'une valeur catégorique(étiquette ou classe) $y \in \{1, \ldots, C\}$ où C est le nombre de classes. Si $C = 2$ la classification est dite binaire, y est généralement supposée appartenir à $\{0, 1\}$. Dans le cas où $C > 2$ la classification est multi-classes. Si les étiquettes de classes ne sont pas mutuellement exclusif, par exemple une personne peut être classée comme grande et forte), On parle de classification multi-étiquettes. Mais il serait préférable de considérer ce cas comme un problème de prédiction de plusieurs étiquettes de classes binaires liées(modèle à plusieurs sorties)[296]. Selon Han et Al.[169], la classification consiste à construire, à partir d'un ensemble de données d'apprentissage, un modèle(ou une fonction), qui décrit et distingue la classe d'un objet ou d'un concept. Cette affectation est faite à partir des caractéristiques explicatives de l'objet et se fait généraliser par une formule, un algorithme ou un ensemble de règles pour classifier de nouveaux objets. La classification est un processus à deux étapes : apprentissage et classification(Fig. 2.23).

Dans l'étape d'apprentissage, un classificateur est construit en analysant un jeu de données d'apprentissage formé des couples (x, y) où x est une instance de donnée et y est l'étiquette de

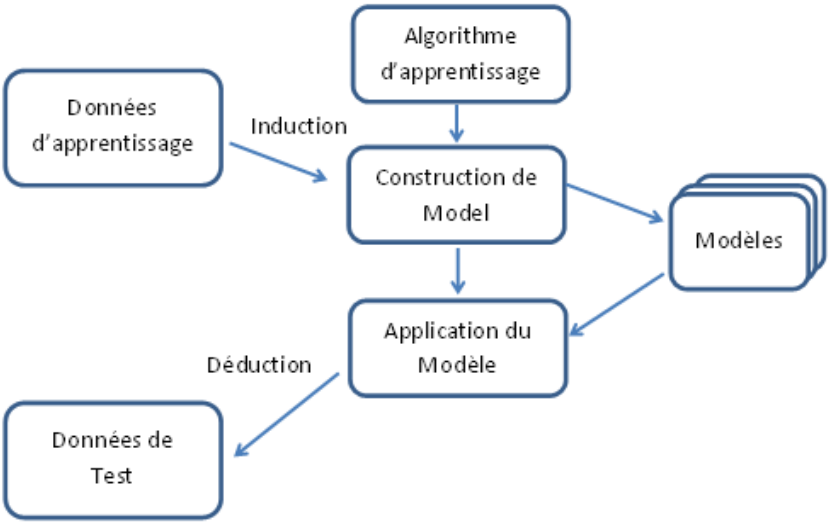


FIGURE 2.23 – Modèle de classification

la classe qu'il lui est associée. Dans la seconde étape, le Classificateur ainsi construit est utilisé pour prédire la classe d'une nouvelle instance de donnée.

Plusieurs algorithmes de classification ont été proposés dans la littérature. Chacun de des ces algorithmes se distingue par la manière dont les relations sont extraites à partir des données connues dites données d'apprentissage. Typiquement, ces algorithmes peuvent être catégorisés en deux classes : Génératifs et discriminatifs[196].

Les algorithmes génératifs, ou informatifs, produisent un modèle de densité de probabilité, qui peut avoir une structure graphique, sur toutes les variables, pouvant être discrètes ou continues et également multidimensionnelles. Ce modèle de densité est, par la suite, manipulé via des marginalisation standard et des d'opérations de conditionnement et peut être utilisé pour la classification ainsi que la régression. Les modèles génératifs couvrent, généralement, les distributions et les mixture exponentielles et peuvent être : des modèles gaussiennes, bayésiens, des mixtures gaussiennes multinomials, des modèles de Markov cachés, des champs aléatoires de Markov, des réseaux de croyances sigmoïdes et des réseaux bayésiens. Le classificateur bayésien[67] et le plus connu des modèle génératifs. Il consiste à assigné un objet x à une classe C_i selon une probabilité postérieure $P(C_j|x)$ pour $j = 1, \ldots, K$. si $P(C_i|x) \geq P(C_j|x)\ \forall j \neq i$.. Le classificateur bayésien est particulièrement adapté aux données à haute dimensionalité. Et malgré sa simplicité il fournit très souvent des performances comparables à certaines méthodes de classification plus sophistiquées tel que les arbres de décision et les réseaux de neurones.

Les approches discriminantes, quant à elles, ne font aucune tentative explicite pour modéliser la distribution sous-adjacente aux données. Les algorithmes de cette catégorie, s'intéressent uniquement à l'optimisation d'une fonction objective reliant les entrées aux sorties désirées(une classe discrète ou un scalaire). Seule les frontières de classification résultantes sont ajustées. Les exemples des méthodes discriminantes les plus populaires comprennent entre autre, la régression logistique, les machines à vecteurs de support et les réseaux de neurones traditionnels.

Les méthodes les plus utilisés pour la classification des données sont des arbres de décision, les méthodes basées sur des règles, des méthodes probabilistes, les machines à vecteurs de support(SVM), les méthodes d'instance, et les réseaux de neurones. Chacune de ces méthodes sont succinctement discutées dans [8].

2.4.2.2 Régression

La régression statistique une technique de l'apprentissage supervisé, qui permet d'étudier et de mesurer, au sein d'un ensemble de données, la relation mathématique existant entre une variable continue de sortie et une ou plusieurs variables d'entrées. La variable estimée est appelée variable dépendante et la ou les variables qui expliquent ses variations sont appelées variables indépendantes. Selon que la variable dépendante dépend d'une ou de plusieurs variables indépendantes, on parle de régression simple ou multiple. Si la relation entre les variables est linéaire, la régression est dite linéaire(Voir [86] pour plus de détaille). Formellement, le problème de régression consiste à trouver, étant donnée un ensemble d'entrainement $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \subset \mathfrak{R}^m \times \mathfrak{R}$, une fonction $\hat{f} : \mathfrak{R}^m \to \mathfrak{R}$ vérifiant $\hat{f} \approx y_i$; $\forall i = 1, \cdots, n$. En réalité, une telle fonction est difficile à trouver, on recherche plutôt une fonction solution de

$$Min \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$$

.

Dans le cas où la fonction $\hat{f}$ est linéaire c-a-d $\hat{f} = \langle \omega, x \rangle + b$ où ω est un vecteur de $\mathfrak{R}^m$ et b est un scalaire, le problème revient à trouver un hyperplan caractérisé par ω^* et b^* tel que :

$$(\omega^*, b^*) = argmin_{(\omega,b)} \sum_{i=1}^{n} (y_i - \langle \omega, x_i \rangle - b)^2.$$

En régression linéaire multiple, qui n'est qu'une extension de la régression linéaire, on a

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_n x_{ni} + \epsilon_i \qquad i = 1, \cdots, n$$

qui peut être représentée matriciellement comme suit :

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} \cdots x_{1m} \\ \vdots & \vdots \ddots \vdots \\ 1 & x_{n1} \cdots x_{nm} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

soit d'une maniéré compacte

$$Y = X\beta + \epsilon$$

Où Y et X sont respectivement une matrice $n \times 1$ de sortie et une matrice $n \times (m+1)$ d'entrées d'entrainement. β, ϵ sont respectivement une matrice $(m+1) \times 1$ de coefficients et une matrice $n \times 1$ d'erreurs d'estimation qui exprime l'information manquante dans l'expression linéaire des valeur y_i à partir des x_i. L'erreur SSE(Sum of Squares of Error) est calculé comme suit[169] :

$$SSE = (Y - X\beta)(Y - X\beta)$$

et après optimisation

$$\frac{\partial SSE}{\partial \beta} = 0 \Rightarrow (X'X)\beta = X'Y$$

d'où

$$\beta = (X'X)^{-1}(X'Y)$$

Il est à noter que si la taille de l'ensemble d'apprentissage est très élevée, le calcul de β devient pénible. Un problème de régression non linéaire est généralement transformé à un problèmes de régression linéaire à l'aide de transformation de variables[169].

2.4.2.3 Analyse des séries chronologique

Une série temporelle ou série chronologique est définie comme étant un ensemble d'observation enregistrées ou collectées à intervalles réguliers au cours du temps[63]. Par exemple l'ensemble des mesures, tel que des températures, des pression ou la position d'un objet..., envoyées par un capteur à des intervalles temporelles réguliers constitue un exemple typique d'une série chronologique. Sont enregistrées, dans une base de données, les mesures prises par le capteur ainsi que le temps réel où ces mesures étaient faites. La Figure 2.24 montre un cas typique d'une série temporelle.

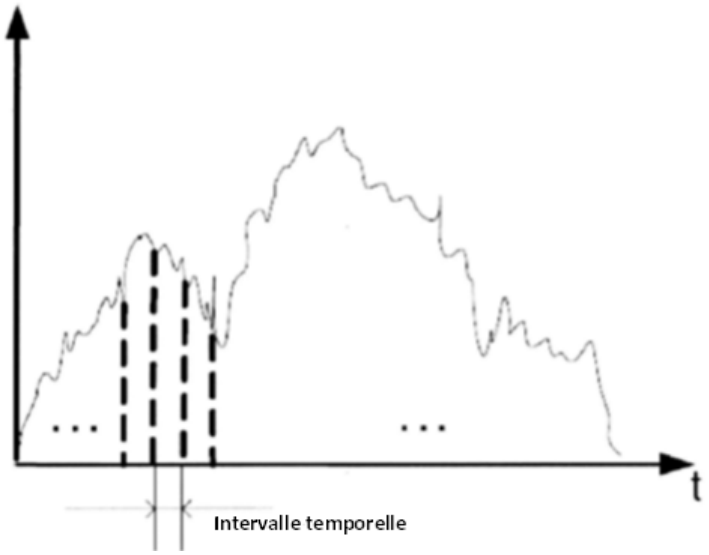


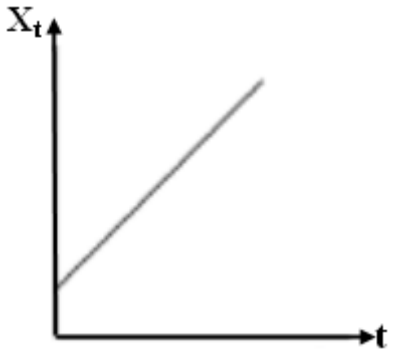
FIGURE 2.24 – série chronologique

Cet ensemble d'observations, qui sont souvent interpolées pour former des valeurs à intervalles réguliers avant que la séries soit analysée, constitue une famille de variables aléatoires réelles généralement notées $(X_t)_{t\in\Theta}$ ou $(X_t, t \in \Theta)$. Θ est dit espace des temps et peut être discret ou continu. Dans le cas discret $\Theta \subset \mathcal{Z}$, le plus souvent on a $\Theta = \mathcal{Z}$, les dates d'observations sont le plus souvent équidistantes(Mensuels, trimestriels, ect) et sont indexées par des entiers : $t = 1, 2, \cdots, T$ où T représente le nombre d'observations. Dans le cas où Θ est continu, l'indice de temps est à valeurs dans un intervalle de $\mathfrak{R}$ et on dispose d'une, au moins potentiellement, infinité d'observations issues d'un processus $(X_t)_{t\in\Theta}$ dit à temps continu et où Θ est un intervalle de $\mathfrak{R}$.

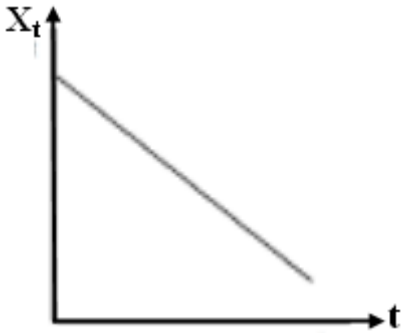
En générale, une série chronologique peut être décomposée en quatre composantes[5], selon le processus illustré par la figure 2.25, chacune exprimant un aspect particulier du mouvement de ses valeurs :

FIGURE 2.25 – Processus de décomposition d'une série chronologique

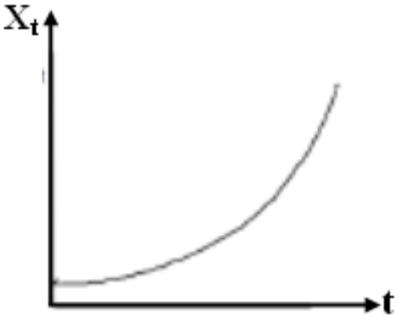
— La tendance(f_i, $1 \leq 1 \leq n$) représente une composante linéaire ou, le plus souvent, non linéaire(Fig. 2.26) qui capte l'évolution, à long terme, du phénomène et ne se répète pas ou au moins ne se répète pas dans l'intervalle de temps dans le quel sont capturées les observations.



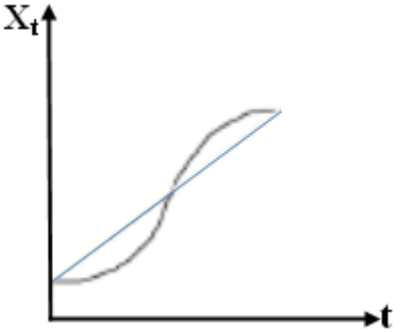
(a) Tendence linéaire croissante



(b) Tendence linéaire décroissante



(c) Tendance non linéaire croissante



(d) Tendance non linéaire croissante

FIGURE 2.26 – Tendance linéaire et non linéaire

C'est une fonction à variation lente et peut être estimée sous forme paramétrique, polynômiale, logarithmique, exponentielle, etc ou par des opérations de lissage. Formellement, on parle de tendance lorsque la série $(X_t)_{t\in\Theta}$ peut s'écrire sous forme d'une combinaison linéaire de m fonctions de temps :

$$X_t = \sum_{j=1}^{m} \alpha_j f_j(t) + \epsilon_t \qquad 1 \leq t \leq n.$$

Lorsque

$$X_t = \alpha t + \beta + \epsilon_t$$

la tendance est dite linéaire ($m = 1$ et $f(t) = \alpha t + \beta$). Une tendance polynômiale se traduira par

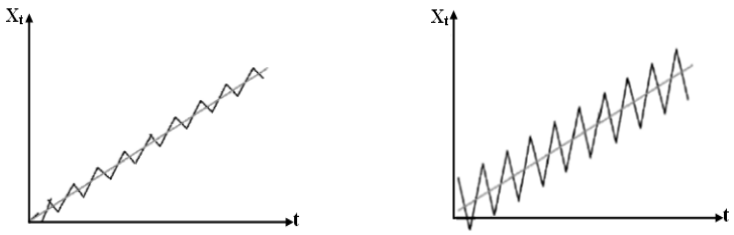
$$X_t = \alpha_1 t^p + \alpha_{p-1} + \cdots + \alpha_{p+1} + \epsilon_t$$

. ϵ_t est un résidu où ne figure plus la tendance et qui, de ce fait, a une allure relativement homogène dans le temps. Il est à noter qu'il est possible de définir des tendances logarithmiques, exponentielles etc... La tendance sera dite multiplicative si la séries s'écrit

$$X_t = f_t \times \epsilon_t \quad t = 1, 2, \ldots$$

Où f_t peut prendre l'une des formes évoquées plus haut(linéaire, polynômiale).
La tendance d'une série chronologique peut être déterminer soit une régression linéaire ou par la méthode des moyens mobiles[277]. La première méthode consisté à calculer les coefficients d'une droite qui représente la tendance. Dans la deuxième méthode on calcule la moyenne des valeur qui entourent chaque valeur et à remplacer la valeur par cette moyenne.

— La saisonnalité, ou variation saisonnière(s_i, $1 \leq 1 \leq n$) correspond à un phénomène qui se répète à des intervalles systématiques au fil du temps. La saisonnalité ou variation saisonnière permet donc de distinguer à l'intérieur d'une même période une répétition stable dans le temps d'effets positif ou négatifs qui se composent sur l'ensemble de la période(Fig.2.27).



(a) (b)

FIGURE 2.27 – Variation saisonnière

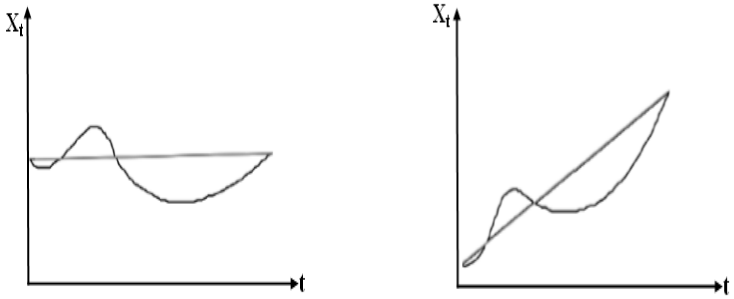
La variation saisonnière exprime le fait que la série ; $(X_t)_{t\in\Theta}$ peut se décomposée en :

$$X_t = s_t + \epsilon_t \qquad 1 \leq t \leq n.$$

Où s_t est périodique, c'est à dire, $s_{t+T} = s_t$, $\forall t \geq 1$ où T est la période et ϵ_t est un résidu non-périodique et sans tendance. Le terme s_t est de moyenne nulle est déterminé en considérant, après calcule de la tendance, la différence $X_t - f_t$.

Tendance et saisonnalité peuvent coexister dans les données réelles. Par exemple, les ventes d'une entreprise peuvent croître rapidement au cours des années, mais maintiennent un motif saisonnier consistent (par exemple, jusqu'à 25% des ventes annuelles de chaque année sont faites en Décembre, alors que seulement 4% sont faites en Août).

— Les variations cycliques(c_i, $1 \leq 1 \leq n$) correspond à des fluctuations irrégulières à long terme(plusieurs années), en générale de faible intensité mais de nature aléatoire(Fig. 2.28).



(a) (b)

FIGURE 2.28 – Les variations cycliques

— les variations irrégulières ou aléatoires(e_i, $1 \leq 1 \leq n$), causées par des influences imprévisibles qui ne sont ni régulières ni répétées dans des motifs particuliers. On ne dispose pas de techniques statistiques pour mesurer ces fluctuations aléatoires dans une série chronologique.

L'analyse d'une série chronologique consiste à faire une description mathématique de ses éléments. Cela reviendrait à estimer ou prédire, séparément ces quatre composantes puis de les combiner[5] additivement :

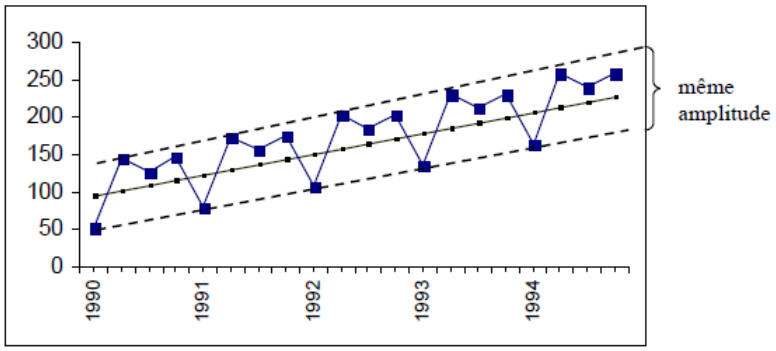
$$X_i = f_i + s_i + c_i + e_i$$

ou multiplicativement

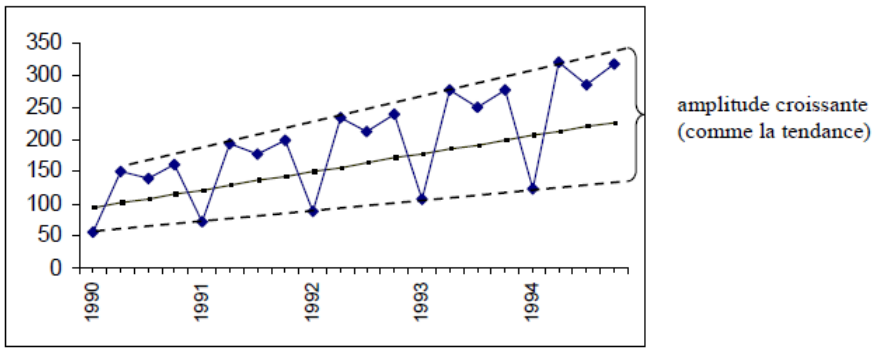
$$X_i = f_t \times s_i \times c_i \times e_i$$

.

Le modèle additif, plus approprié lorsque l'amplitude des variations est constante autour de la tendance(Fig. 2.29(a)), est basé sur l'hypothèse que les quatre composantes sont indépendantes l'une de l'autre. Tandis que le modèle multiplicatif assume que les quatre composantes de la série chronologique ne sont pas nécessairement indépendante et l'une influe sur l'autre. Ce dernier modèle est plus approprié lorsque l'amplitude des variations semble être proportionnelle au niveau de la série(Fig. 2.29(b)).



(a) Les deux droites tracées sont à peu près paralléles entre elles.



(b) Les deux droites tracées ne sont pas paralléles entre elles.

FIGURE 2.29 – Modèles additif et multiplicatif d'une série chronologique

Ces deux modèles de décomposition peuvent être combinés ensemble pour former un modèle mixte. On peut supposer par exemple que la composante saisonnière agit de façon multiplicative alors que les fluctuations irrégulières sont additives :

$$X_i = f_i(1 + s_i) + c_i = f_i + f_i s_i + c_i \qquad 1 \leq i \leq n.$$

Une alternative à l'utilisation du modèle multiplicatif, consiste à transformer d'abord les données jusqu'à ce que la variation de la série semble être stable dans le temps, puis d'utiliser un modèle additif. Notons que le passage du modèle additif au modèle multiplicatif peut se faire aisément via une transformation logarithmique vu que $X_t = T_t \times I_t \times C_t \times E_t$ est équivalent à $log(X_t) = log(T_t) + log(I_t) + log(C_t) + log(E_t)$.

Il existe une grande variété de séries chronologiques qui peuvent être classés en plusieurs catégories selon de différents points de vue[219]. Ainsi, une serie chronologique peut être continue ou discrète selon à ce que les données sont collectées d'une maniéré continue avec un dispositif analogique, par exemple, ou observées à des intervalles de temps. Comme elle peut être uni-variée si une observation est collectée à un instant donnée ou multivariée si elle est obtenue en enregistrant deux ou plusieurs phénomènes en même temps. Les valeurs des observations recueillies sont représentées, dans ce cas, en tant que vecteurs. La distinction entre séries uni-variée et multivariée est, par nature difficile. Aussi, une série chronologique peut être stationnaire ou non stationnaire. La stationnarité, comme propriété, est liée à la valeur moyenne et à la variance des observations[303]. Ces deux mesures devraient être constantes dans le temps. La variance entre deux observations X_t et X_{t-d} doit dépendre uniquement de la distance entre ces deux observations et ne change pas dans le temps :

$$\begin{aligned} &Ex_t = \mu, && t = 1, 2, \ldots \\ &Var(x_t) = E(x_t - \mu)^2 = K_0, && t = 1, 1, \ldots \\ &Cov(x_t, x_{t-d} = E(x_t - \mu)(x_{t-d} - \mu) = K_d. \end{aligned}$$

Avec $d = \ldots, -2, -1, 1, 2, \ldots$, et où μ, K_0 et K_d sont des constantes. Statistiquement parlé, une série est dite stationnaire si les distributions conjointes $X(t)$ et $X(t - \tau)$ dépendent de τ seulement. La séries chronologique sera dite non-stationnaire si sa structure stochastique change avec le temps. Si la distribution de la série suit une loi normale, elle est dite gaussienne sinon, elle sera dite non-gaussienne. Et en fin une série chronologique exprimée en tant que sortie d'un modèle linéaire est appelée une série temporelle linéaire et est définie formellement par le modèle linéaire[303] :

$$y(t) = \sum_{j=-\infty}^{\infty} \alpha_j x(t - j)$$

où les coefficients α sont tel que :

$$\sum_{j=-\infty}^{\infty} |\alpha| < \infty$$

Une série chronologique linéaire est générée par un processus stationnaire de deuxième ordre en utilisant la technique **World's decomposition** pour éliminer sa composante déterministe. La sortie d'un modèle non linéaire est appelé une série temporelle non linéaire. Certaines séries non linéaires sont représentées par des séries bi-linéaires modelées comme suit[303] :

$$x_t = z_t + \sum_{i=1}^{p} a_i x_{t-i} + \sum_{j=1}^{q} b_j z_{t-j} + \sum_{i=1}^{r} \sum_{j=1}^{s} C_{ij} x_{t-i} z_{t-j}$$

L'analyse des séries chronologique traite le problème d'identification des caractéristique basiques ainsi que la découverte de la structure interne des séries chronologiques. Globalement, lors d'une analyse des séries chronologiques, on est amené à lisser la série, la modéliser et à la prédire.

Dans une première étape, on nettoie la série des éventuels effets systématiques et des effets occasionnels puis, on impute des valeurs aux dates où manque l'observation de la série.

Notons que les valeurs atypiques susceptibles de fausser la modélisation peuvent parfois être traitées comme des valeurs manquantes. Une fois, la séries nettoyée, on passe à l'étape descriptive qui consiste à dégager des éléments de synthèse, en général sous forme de nouvelles séries, qui résument au mieux la grandeur observée ou d'exhiber certaines caractéristiques. Il est à noter qu'une représentation graphique des observations constitue une étape primordiale dans le processus de l'analyse d'une série chronologique. Les points $(X_t, t)\,; t = 1, \ldots, T$ sont projetés dans un système d'axes orthogonaux et joints chronologiquement par des segments de droite pour facilité la visualisation. Cette représentation a pour but de dégager, d'une part, la tendance du phénomène et les périodes de stabilité, d'autre part. Par fois la représentation graphique nécessite des transformations des valeurs de la série. La modélisation d'une série chronologique consiste à capturer la structure stochastique de la série à fin de déterminer l'ensemble, correcte, des paramètres à estimer. Pour cela en a recours à deux types de modèles :

— Un modèle sans variable explicative dans lequel ont explique le présent de la série par son propre passé ;

$$X_t = g(X_{t-1}, X_{t-2}, \cdots) + u_t$$

C'est la solution quand on ne dispose pas d'information sur les variables susceptibles d'agir sur la variable d'étude. Si les erreurs u_t sont de moyenne nulle, et de variance constante et non corrélées deux à deux, on se trouve, alors, en présence d'un modèle auto-régressif.

— Un modèles avec variables explicatives :

$$y_t = g(X_t) + u_t$$

X_t est un vecteur de variables explicatives pouvant contenir des valeurs retardées de y_t, et u_t une erreur présentant généralement une auto-corrélation.

La prédiction d'une série chronologique consiste à estimer le comportement future de la série sur la base des corrélations entre les variables au cours de temps par l'extraction des informations sur les observations passées et courantes. La prédiction peut être basée soit sur un modèle soit construite sans ajustement préalable d'un modèle, cas du lissage exponentiel et de ses généralisations.

2.4.2.4 Prédiction

Le but de la prédiction est de déterminer un résultat future et non pas un comportement courant comme est le cas dans la classification. le résultat d'une prédiction peut être une valeur catégorique ou continue. Toutes les méthodes de classification et d'estimation peuvent être, également, utilisées dans certaines circonstances appropriées pour la prédiction. Cela inclut les méthodes statistiques traditionnelles d'estimation de points et d'intervalles de confiance, de corrélation et de régression linéaire simple et multiple aussi bien que les méthodes d'extraction de connaissances et de fouille des données tel que les réseaux de neurones, les arbres de décision, l'algorithme des k-plus proches voisins[237].

2.5 La méthodologie du Datamining

Pour mener à bien une action de datamining, il existe deux possibilités méthodologiques : le test d'hypothèse et/ou la découverte de connaissances (dirigée ou non).

2.5.1 Test d'hypothèse

C'est une approche descendante qui consiste à formuler, sur la base des résultats d'une analyse statistique préalable ou sur de l'expérience et de l'imagination, des hypothèses qui seront validées par des données existantes et éventuellement par des études parallèles. Cette approche débute par une phase essentielle qui consiste à collecter des "bonnes" idées auprès des groupes de travail préalablement formés. Ces groupes de travail constitués des différents intervenants et concernés par l'application permettent à chacun de réagir aux idées des autres en fonction de son niveau d'expertise et aboutissent à une vue multi angles du problème considéré et génèrent de nouvelles questions pouvant engendrer de très bonnes idées. Une fois les hypothèses formuler il faut procéder à la sélection des données qui devront permettre de valider ou d'invalider ces hypothèses. Par la suite il faut localiser les différentes sources de données qui le plus souvent sont des sources externes. Les données ainsi collectées passeront par un processus de pré-traitement qui consiste à les nettoyer des redondances et des valeurs aberrantes, et à compléter ou éliminer le manque qui pourrait apparaître. Puis, de les fusionner dans une seule base de données (DataWarehouse) cohérentes et intègres. Cela devrait aboutir sur un modèle informatique qui devrait être testé et évalué sur des données réelles. Les résultats doivent être analysés et interprétés pour voir comment les hypothèses se vérifient. Cela exige à la fois des connaissances analytiques et spécifiques au domaine ce qui justifie la nécessité des groupes pluridisciplinaires.

2.5.2 La découverte de connaissance

La découverte de connaissance, quant à elle, est une approche ascendante et est la technique de Datamining la plus significative et la plus utilisée. Elle consiste à extraire une information pertinente et inconnue à partir d'un ensemble de données. On distingue la découverte de connaissances dirigée et non dirigées.

2.5.2.1 Découverte de connaissances non dirigées

Aussi appelée apprentissage non supervisé, la découverte non dirigée est historiquement la vocation des produits de datamining. Son but est la découverte des différentes structures significatives enfouies dans une base de données et fournie un ensemble de connaissances le plus souvent exprimées sous forme de règles auxquelles un indicateur de confiance est associé. Cet indicateur permet de quantifier la fiabilité de la règle, mais reste en générale insuffisant et dépend de la taille de l'échantillon. Par exemple si une règle qui ne concerne qu'un ou deux individus même avec un facteur de confiance égal à 100% devra être rejetée ; car non suffisamment significative (sauf si ce n'est qu'une vérification d'une règle déjà validée). Le processus de la découverte de la connaissance non dirigée commence par une étape d'identification des ressources de données disponibles et de s'assurer de leur "bonne" qualité. Il va donc falloir identifier les données, les localiser, et identifier les formats et les codages,... Un travail organisationnel et logistique important est nécessaire pour disposer des données sous une forme utile en vue de la découverte de connaissances. Puis les données sont vérifiées, transcodées, transformées et enfin regroupées dans une base de données avec un format propice à l'exploration par une application de datamining. Très souvent des champs supplémentaires, issus de résultats de calculs ou transformations depuis des champs existants, sont rajoutés. Ceci est particulièrement vrai si l'on cherche des relations entre champs ou pour suivre des évolutions dans le temps. L'ensemble des données ainsi obtenu est scindé en trois sous ensembles distincts : des ensemble d'apprentissage, de tests et d'évaluations.

L'ensemble d'apprentissage est utilisé pour construire le modèle initial. C'est depuis cet ensemble que le système va calculer ses différents paramètres. (Par exemple l'utilisation des

techniques descriptives évolutives en statistique tels la construction des classes de référence, l'ACP, l'AFD,...). ne fois les paramètres calculés, il faut vérifier comment ils se comportent sur l'ensemble de tests. Celui-ci va permettre d'ajuster les valeurs trouvées à l'étape précédente et les rendre moins sensibles à la modification de l'ensemble d'apprentissage. Enfin, les paramètres seront testés sur l'ensemble d'évaluation. Si les résultats obtenus sont proches de ceux attendus, on pourra alors valider le système. Dans le cas contraire, il faudra analyser les raisons de cette différence. Pour mesurer la validité des résultats obtenus, on utilisera les outils statistiques traditionnels.

L'étape suivante consiste à identifier la ou les techniques à mettre en œuvre et réaliser le programme. Le modèle informatique ainsi obtenu est évalué. La plus grande difficulté est de déterminer le volume d'apprentissage optimal. Pour ce faire, il faut tester les données connues et inconnues. Si les données connues sont trop importantes, on risque de trouver des paramètres d'estimation très précis sur cette population mais qui donneront des valeurs très médiocres sur une population inconnue. Le résultat sera similaire si le volume d'apprentissage est trop faible. Il faut donc trouver un compromis, comme illustré sur la figure qui suit.

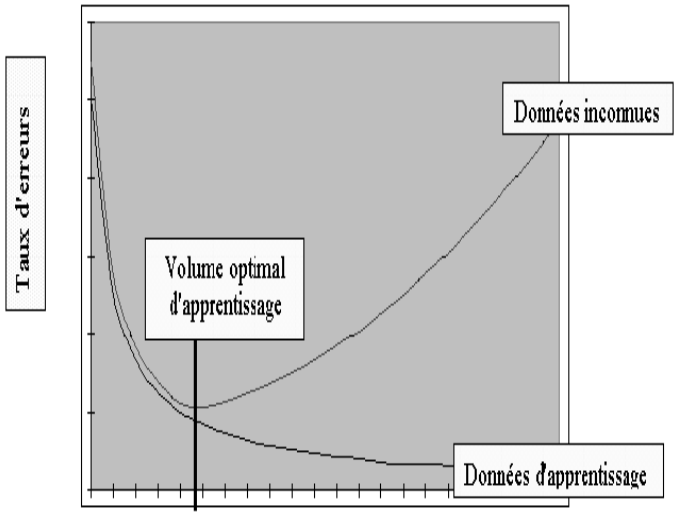


FIGURE 2.30 – Détermination du volume optimal d'apprentissage

Lorsque le modèle ou l'apprentissage est effectué, il faut l'appliquer à de nouvelles données. Cette étape permet au système d'appliquer ses connaissances à ces données. Cette phase d'application du modèle à de nouvelles données est suivie par une étape d'identification des cibles potentielles pour la découverte de connaissances dirigée. L'utilisateur va exploiter les conclusions et les connaissances. Cela débouche sur de nouvelles interrogations qui se traduisent généralement par une approche de découverte de connaissances dirigée. Dans la dernière étape dans le processus de découverte de connaissances non dirigé les nouvelles connaissances générées aux étapes précédentes permettent de générer de nouvelles hypothèses, qu'il faut retravailler. Nous entrons ici dans le cadre du test d'hypothèses.

2.5.2.2 Découverte de connaissances dirigées

La découverte de connaissances dirigée consiste à expliquer un (ou plusieurs) champ en fonctions d'un ou plusieurs autres champs. Le champ cible est spécifié par l'utilisateur. La connaissance extraite peut être une fonction du temps où un événement du passé explique une valeur actuelle. De point de vue méthodologique, elle à le même principe que celui de la découverte de connaissances non dirigée. La seule différence est que dans le cas précédent,

les connaissances générées débouchent soit vers un test d'hypothèse, soit vers un processus de découverte de connaissances dirigée.

2.6 Conclusion

Le datamining n'est pas une simple technique mais plutôt une collection de techniques qui nous permet d'atteindre ou d'extraire des informations pertinentes cachées dans de grandes masses de données. Il est un processus itératif et interactif à plusieurs étapes dont le but de générer une "intelligence". Son succès n'est pas déterminé par la performance ou l'exactitude du modèle mais par sa valeur. Le datamining effectif n'exige pas uniquement une claire compréhension des issues invoquées mais nécessite aussi une préparation excessive des quantités de données - identifier les variables importantes, nettoyer, coder et analyser les données. Sans préparation de données appropriée, le datamining est apte à produire des informations inutiles ou erronées.

Le datamining a émergé comme un outil stratégique entre les mains des décideurs pour gagner l'intelligence utile à leurs entreprises. Une telle intelligence les aides à améliorer leur productivité dans plusieurs secteurs critiques comprenant le marketing, le développement de produit et les services à la clientèle. L'industrie des télécommunications, et le secteur banquier et financier sont les premiers "consommateurs" des méthodologies ayant enregistré des résultats impressionnants. En conséquence, les applications de datamining continuent à se développer. Ce développement est lié à l'intégration des nouvelles théories issues des disciplines sur lesquelles le data mining s'appuie.

CHAPITRE 3

DATA MINING ET DÉTECTION D'INTRUSION

3.1 Introduction

L'utilisation omniprésente des ordinateurs et des réseaux informatiques dans la société d'aujourd'hui a attribué à la sécurité des réseaux informatiques une priorité primordiale. Ainsi la détection d'intrusion, bien que récente, est devenue une composante essentielle de toute architecture de sécurité, certes elle n'est pas suffisante en elle même pour garantir une sécurité totale du réseau mais combinée à d'autre éléments de sécurité tel que les firewalls, la cryptographie, le contrôle d'accès,... peut maintenir un très bon niveau de sécurité.

Dans le but de rendre les systèmes de détection d'intrusion plus performant et plus résistants aux différentes menaces, les professionnels de sécurité issus des institutions académiques, gouvernementales et industrielle ont engagé des efforts de recherches et de développements très conséquents. Ainsi, depuis les travaux d'Anderson[27, 28] portant sur la surveillance et l'audit des systèmes informatiques, puis ceux de Dening[105] portant sur la modélisation de la détection d'intrusion, une large variété de techniques et d'approches ont été proposées pour construire des systèmes de détection d'intrusion plus performants. Ces systèmes, dits de première génération, peuvent être, globalement, classés en deux catégories : Les systèmes à base de règles et les systèmes à base de techniques statistiques.

1. Les systèmes à base de règles et les systèmes à base de techniques statistiques. Les systèmes de détection d'intrusion à base de règles étaient implémentés sous forme de systèmes experts et ont étaient utilisés aussi bien pour la détection des anomalies [395, 115] que pour la détection des abus d'utilisation[105, 262, 231]. Dans le cadre de la détection des abus d'utilisation, tout événement, tiré à partir des données d'audit, lié à une attaque ou intrusion connue est traduit en terme des règles de type **if-then-else**[40]. Alors que dans le cas de la détection d'anomalie, c'est le comportement normal des utilisateurs et des programmes qui est décrit en terme de règles. Ces règles sont créées de deux façons : La première consiste à générer les règles à partir de la police de sécurité pré-établie. Les règles ainsi obtenues décrivent le comportement attendu des utilisateurs ou des programmes. Dans la deuxième, le profil normal est décrit par un ensemble de règles générées à partir des données d'audit lors d'une phase d'apprentissage. Ce Type de règles constitue le moyen le plus naturel pour modéliser le savoir et le savoir faire des experts humains en matière de description des différentes attaques connues et du comportement normal des utilisateurs. Mais le fait que le processus de leurs génération est manuel d'une part et nécessite, pour construire des modèles convenables, un grand volume de données, dont la collecte s'avère une tâche très difficile notamment pour les réseaux à trafic intense d'autre part, rend la construction de ce type de systèmes très

difficile et couteuse.

2. Les systèmes à base de techniques statistiques mesurent le comportement d'un utilisateur ou du système d'exploitation avec un certain nombre de variables prélevées à intervalles réguliers de temps (minute, heure, jour,...)[40]. Le nombre de login et de logout effectués par un utilisateur, des mesures sur la session (temps de début, temps de fin), le nombre de fichiers et de dossiers utilisés durant une périodes de temps, l'utilisation d'espace de stockage, mémoire utilisée, temps CPU consommé... en sont des exemples typiques de ces variables. Ces mesures sont utilisées pour construire des profils qui représentent le comportement normal d'un sujet pouvant être, selon Denning[105] un utilisateur, un processus ou le système lui même. Des seuils et des intervalles sont associés à chacune de ces variables[300]. Lors du processus de détection, les variables du profil normal sont comparées à celles collectées durant les sessions en cours et toute déviation significative est considérée comme une intrusion, de ce fait une alerte doit être émise à l'intention de l'administrateur. Ces profils doivent être conçus pour consommer peu d'espace mémoire pour stocker leurs états internes et d'être facile à mettre à jour, car chaque profils et appelé à être mis à jour chaque fois où les données d'audit changent.

La majorité des systèmes à base de règles et/ou à base de techniques statistiques ont été conçus d'une manière manuelle et Ad-hoc et dépendent fortement de l'intuition et de l'expérience des experts en sécurité informatique en matière de collecte et d'analyse des données d'audit, de leur habilité à construire le dit profil normale, à comprendre et à coder les signatures des attaques connues. De ce fait chacun de ces systèmes de détection d'intrusion était conçu pour des clients et des environnements spécifiques et ne peuvent être facilement déployés sur d'autres environnement même ayant des polices de sécurité semblable à leur environnement cible. Et comme les systèmes informatiques(Hôts et réseaux) deviennent de plus en plus complexes ces systèmes ne sont pas en mesure de détecter de nouvelles attaques. La détection de nouvelles attaques était la première prétention des systèmes de détection d'anomalie. Mais en réalité ces systèmes sont caractérisés par un taux de faux positif considérablement élevé et ne sont utilisés qu'à des fin académiques ou dans des laboratoires de recherches[65]. De plus, on ne disposait d'aucun standard pour l'expérimentation et l'évaluation des performances de ces systèmes de détection d'intrusion[231] vu l'absence d'un format unifié des données d'audit permettant la comparaison de l'efficacité de ses systèmes par rapport à un scénario commun d'attaque.

Pour améliorer les performances de ces systèmes de détection d'intrusion qualifiés de traditionnels et de classiques, des chercheurs, informaticiens et spécialistes en datamining, ont proposé des approches de détection reposant sur des techniques issues de la fouille des données. Ces approches tentent en premier lieu à supprimer les éléments Ad-hoc et manuels de la conception des systèmes de détection d'intrusion par l'automatisation de processus de construction du modèle de détection d'une part et à remédier aux problèmes liés à l'exploitation des grandes masses de données d'autre part. Lee et Stolfo[240] furent les premiers ayant proposé une architecture de détection d'intrusion à base de de techniques de datamining(Fig. 3.1).

Dans cette architecture, des données brutes sont collectées en format ASCII puis structurées sous forme d'enregistrements décrivant des connexions à l'aide d'un ensemble d'attributs tel que la durée de connexion, le service invoqué,... Ses enregistrements sont soumis à un processus itératif d'extraction de motifs fréquents (règles d'associations) à fin d'extraire les motifs d'activités.

Après cette première tentative, un grand nombre d'algorithmes de datamining on été appliqués à la détection d'intrusion et à la détection d'anomalie en particulier.

L'intégration des techniques de datamining dans la détection d'intrusion semble être une solution naturelle pour construire des modèles de détection plus précis et plus performants pour

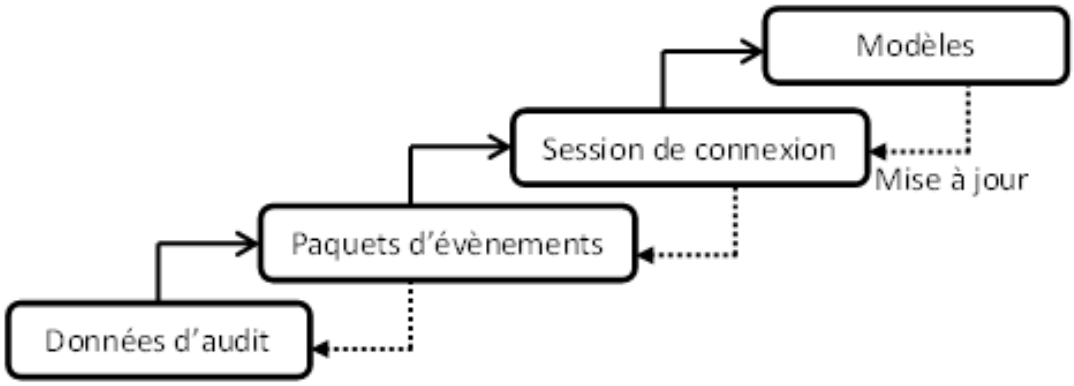


FIGURE 3.1 – Architecture de détection d'intrusion à base de datamining

les raison suivantes :

— Ces techniques, issues de plusieurs domaines tel que les statistiques, l'intelligence artificielle, et l'apprentissage automatique, peuvent extraire automatiquement des relations et des similarités non triviales cachées dans de grandes masses de données. Ces relations et similarités permettent la découverte et la prédictions des motifs et des comportements des utilisateurs et des applications.

— Les techniques de réduction de données et de sélection d'attributs les plus pertinentes permettent de réduire, considérablement, délais et besoins en ressources calculatoire(charge CPU et mémoire de stockage) nécessaires pou l'exploration des grandes masse de données.

— Les résultats fournis par ces techniques peuvent être utilisés dans des systèmes de prise de décision automatique ou évalués par des expert humains.

3.2 Les défis

Dans [116] les auteurs classent les défis de la fouille des données en terme de cyber sécurité en quatre domaines d'applications : La modélisation à grande échelle des réseaux, la détection de l'intrusion, le Dynamisme du réseau, la préservation de la vie privée dans la fouille des données.

La modélisation des réseaux n'est pas une tâche facile, plusieurs mesures associées au graphes sont difficiles à calculer pour les réseau sous-jacents, comme il est difficile de construire un modèle explicatif d'un réseau vu le degrés de précision exigé pour l'apprentissage et la prédiction : Des réseaux réalistes sont simulés, à différentes échelles, afin de tester des algorithmes de défense, et des anomalies non conformes au modèle et qui représentent potentiellement des menaces sont détectées. Un modèle de réseau peut être extrait partiellement au profit d'une analyse avancée et un réseau peut être construit d'une maniérer réel et significatives mais peut ne pas suivre l'hypothèse des variables aléatoires indépendantes identiquement distribuées. Les défis persiste dans le calcul des mesures associées au graphe dans le modèle de réseau. Des exemples de tels modèles sont la dynamique du réseau de télécommunication, les réseaux courrier électronique par les quels les virus se propagent et les hyperliens entre sites web. La plus grande distance entre deux nœud du graphes(diamètre du graphe) est un exemple typique des mesures de graphe qu'on est amenées à calculer. La difficulté des calculs exige le recours aux techniques et modèles de la fouille des données qui peuvent découvrir la nature réelle des données en faisant appel à de simple modèles.

La détection d'intrusion à base de techniques de datamining souffre essentiellement des problèmes suivants :

— Le volume important des données hétérogènes à traiter,

— Le changement dynamique des menaces,

— La difficulté de distinguer les comportements normaux des anormaux.

Ces défis nécessitent le recours aux méthodes qui peuvent agréger l'information dynamiquement et localement afin de détecter les attaques multi-étapes et de prévoir les menaces potentielles et rares sur la base de l'analyse de comportement des données et des événements du réseau. Les méthodes les plus employées, dans ce contexte, utilisent des modelés statistiques ou ceux à base de règles pour détecter les menaces en temps réel en utilisant une détection adaptative avec la modélisation des données temporaires et manquantes. L'échantillonnage des données au sein des réseaux à grande échelle doit être adaptatif aux incertitudes de l'évolution physique des réseaux, des codes et des comportements malveillants. Une modélisation adaptative et dynamique s'avère nécessaire pour l'évolution des structures et des caractéristiques des données.

Pour pouvoir prédire les futures menaces ou attaques sur la base de l'évolution des codes malveillants on est appelés à faire recours à de nouvelles méthodes de fouille des données. Or comme la structure détaillée du réseau est inconnu et vu que les hôtes sont touchés à différents degrés, la connaissance sur l'évolution des menaces est limitées.

Les techniques de fouille des données jouent un rôle critique dans la détection des intrusions dans les systèmes informatiques. Mais peuvent être, également, utilisées d'une manières à compromettre le principe de la préservation de la vie privée dans le data mining (PPDM Privacy Preservin Data Mining) qui vise à protéger les données privées d'être divulguées, volées ou mal utilisées par des utilisateurs malveillants.

3.3 Besoins architecturaux

Le datamining est lui même un processus consommant beaucoup de ressources informatiques. Un déploiement efficace d'un système de détection utilisant les techniques de datamining exigent une architecture et une infrastructure adaptatives et scalables capables de supporter : le stockage des données d'audit, leurs traitement, la génération et la distribution de modèles, aussi bien que l'interaction avec les autres éléments préexistants dans l'infrastructure globale de sécurité. La figure 3.2 présente une approche modulaire àcette architecture en utilisant un modèle proposé par Tomas Abraham[2]. Selon les applications et les environnements spécifiques, des modules ou des éléments particuliers (des agents d'apprentissage et de détection) devraient être ajoutés dans leur contexte, complétant la technique du datamining choisie.

1. Des détecteurs : avec des performances optimales en matière de gestion des volumes de données et le leur vitesse d'acquisition.
2. Base de données : Puisque on est appelé àstocker un énorme volume de données, àmaintenir ces données régulièrement àjour, et àfournir des réponses àdes requêtes complexes dans un laps de temps acceptable, il est très recommander de choisir un système de gestion de base de données très performant. L'utilisation des processeurs massivement parallèles est une solution très bien appréciée.
3. Espace de stockage nécessaire pour manipuler les données concernées par le processus de datamining (résultats intermédiaires et finals, fichiers temporels, etc..).
4. Des voies de transmission entre les différents détecteurs et analyseurs qui seraient capables de supporter un volume grandissant de trafic. Le transport de données doit être sécurisé.

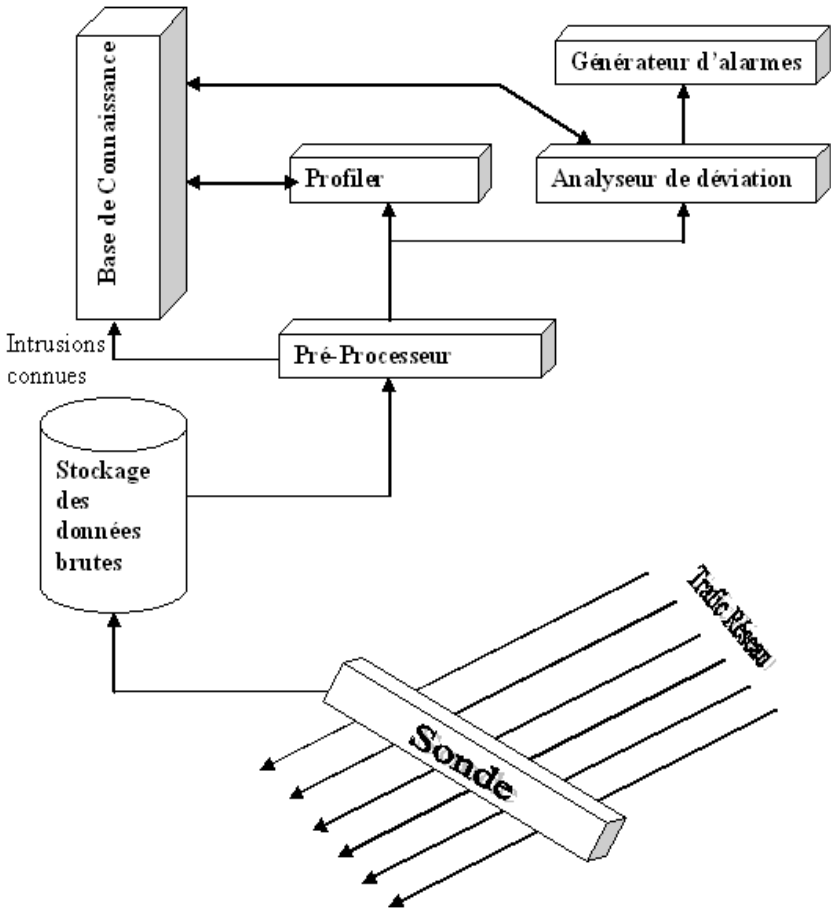


FIGURE 3.2 – Architecture modulaire pour un système de détection d'intrusion utilisant les techniques de datamining

5. Une puissance de calcule : Les outils de datamining demandent de considérable puissance de traitement et de mémorisation. Il a été constaté que l'application du datamining dans la détection d'intrusion nécessitait au moins quatre fois plus de mémoire et de puissance CPU que les autres domaines.
6. Des produits datamining pour analyser le volume d'informations afin d'extraire des connaissances de haut niveau à partir des données filtrées et composées.

D'autre part et en prenant en considération que l'intervention humaine est exigée dans le processus, il est souhaitable que le personnel impliqué ait la capacité d'interagir avec la base de données et les applications de datamining. De plus, la plupart des techniques de détection d'intrusion nécessitent des ensembles de données d'entraînement et de tests.

L'application des techniques de datamining dans la détection d'intrusion nécessite trois sortes de compétences : sécurité réseaux, datamining et développement des applications base de données. Naturellement, de solides connaissances en réseaux et en détection d'intrusion son nécessaires, mais aussi l'habilité d'aborder des grands problèmes abstraits. Les analystes devraient avoir de solides connaissances en statistiques et en apprentissage automatique, mais également en réseaux informatiques. Les concepteurs de base de données auront besoin de bonnes qualifications dans la conception efficace de base de données, et en datawarehousing.

3.4 Processus de datamining pour la détection d'intrusion

Le processus de datamining recherche les modèles cachés relatifs àdes intrusions précédemment non détectés pour aider à développer de nouveaux modèles en créant de nouvelles connaissances àpartir des données d'audit(Fig. 3.3).

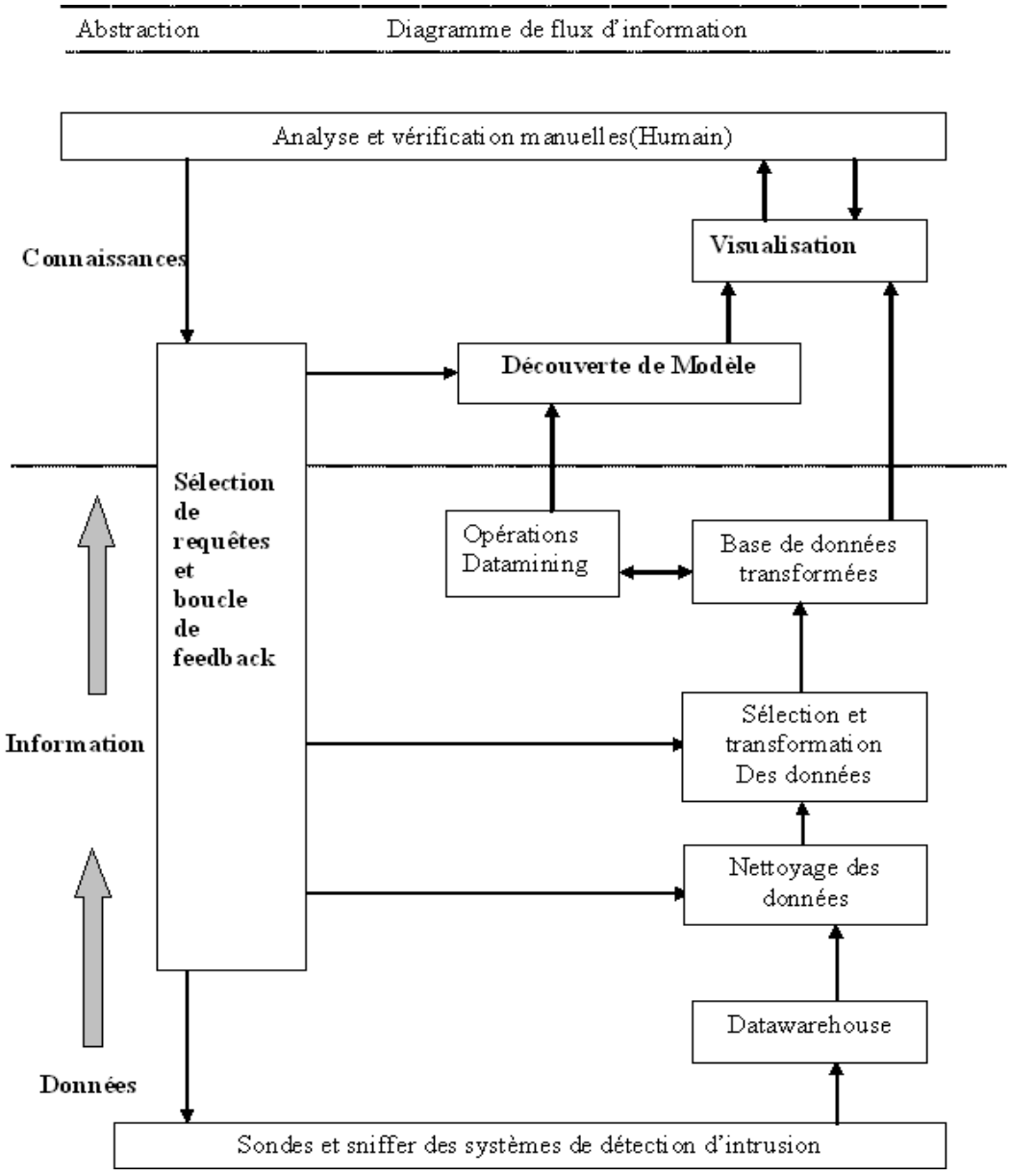


FIGURE 3.3 – Datamining pour la détection d'intrusion

Les données brutes issues des systèmes de gestion de réseau et des systèmes de détection d'intrusion sont collectées et classées dans un entrepôt de données. Puis soumises à un processus de nettoyage qui consiste à supprimer les données erronées, incomplètes et aberrantes, à vérifier la cohérence globale des données. L'étape suivante consiste à sélectionner les sous ensembles de données sur lesquels des opération de datamining seront exécutées. Cette étape, dite étape de sélection et transformation de données, peut être manuelle ou automatique. Les opérations de datamining sont exécutées sur un petit ensemble de données puis étendues sur l'ensemble de données entières. Le modèle émergeant sera validé. Des nouvelles connaissances de détection

sont extraites de l'ensemble de données puis des modèles plus raffiné sont développées.

Ces Modèles tentent de prédire des futures événements en se basant sur l'historique. En outre les analystes exigent des outils de visualisation pour soutenir les processus d'identification de modèle.

Le processus de datamining, en sa totalité, est raffiné en ajustant des paramètres , des ensembles de données, et des associations. Bien que la détection d'intrusion est intuitivement un processus à temps réel par nécessité[65], il est très préférable de l'exécuter en mode temp-différé (off-time), pour les raisons suivantes :

1. Un système de détection opérant en mode temps-réel est appelé à collecter les données d'audit et de les analyser "en même temps" ce qui consomme beaucoup de ressources et dégrade les performances du réseau.
2. De plus, la complexité des algorithmes de datamining générant les profils à partir des ensembles de données, est usuellement d'ordre $\mathcal{O}(n^3)$[368] et deviendra plus tractable, selon [45], dans un environnement off-line si les contraintes pseudo temps réel n'existent pas.
3. En mode off-line il est possible de transférer, durant les heures creuses, touts les journaux d'audit des différents hôtes appartenant au réseau surveillé vers un poste central pour corrélation est analyse.
4. En mode off-line, il est généralement admit que toute les connections sont déjà achevées de ce fait nous disposons du luxe de calculer toutes la caractéristiques et de vérifier les règles de détection une par une.

Aussi, les systèmes de détection d'intrusion à temps réel sont sujets à plusieurs attaques tel que l'inondation par des paquets fragmentés dans le temps. Lors de telle attaque, le système de détection d'intrusion consacrera ses ressources (CPU et mémoire) et son temps pour la collecte de ces paquets à fin de reconstruire ce faut trafic et de l'analyser par la suite. Durant ce temps, l'intrus peut facilement passer inaperçu. Encore, certains systèmes de détection d'intrusion commencent à ignorer les paquets IP dés qu'ils sont inondés avec une très grande cadence. L'environnement off-line est sensiblement moins vulnérable à de telles menaces notamment en présence d'un firewall qui filtre le trafic entrant dans le réseau. Compte tenus des exigences calculatoires élevées des techniques du datamaing, notamment en détection d'intrusion, **Brugger**[65] conclut qu'un traitement en "off-line" constitue une partie standard dans une architecture de sécurité.

3.5 La détection d'intrusion à base de Dataming

La fouille des données peut contribuer à l'amélioration des performances des systèmes de détection d'intrusion soit par la construction de modèle précis à partir de l'historique des attaques perpétuées dans le passé en utilisant des techniques d'apprentissage supervisé ou par l'identification des activités malveillantes en utilisant des techniques d'apprentissage non supervisé. Dans le cas de la détection d'abus d'utilisation, chaque séquence ou ensemble de séquences caractérisant une attaque est traduit en termes de motifs exprimant des signatures d'attaques. Une fois ces derniers identifiés et corrélés, des mesures de similarités pouvant exister entre ces motifs et ceux collectés ou identifiés à partir des données d'audit lors des périodes de surveillance, doivent être calculées à fin de détecter et identifier des éventuelles attaques. Typiquement, la détection d'abus d'utilisation se fait en cinq étapes[116] comme illustré dans la figure 3.4. Dans la première phase, des données d'audit issues des fichiers log, trafic réseau et registres systèmes sont collectées puis soumis, dans une deuxième phase, à un processus de pré-traitement qui

consiste à appliquer des opérations de réduction de bruis, de normalisation et d'extraction et sélection des caractéristiques. Ces données restructurées seront utilisées, dans une troisième étape, pour construire des modèles d'apprentissage d'intrusions tel que des systèmes experts basés sur la connaissance sur les codes malveillants et les vulnérabilités connues. Ces derniers peuvent être construits soit par des experts du domaine ou automatiquement par des systèmes d'apprentissage intelligents. Les modèles de classification ainsi construits sont utilisés dans la phase de surveillance pour détecter tout type d'abus d'utilisation connu. Une fois une action malveillante détectée, une décision doit être prise soit automatiquement par le système de détection ou manuellement par le chargé de la sécurité.

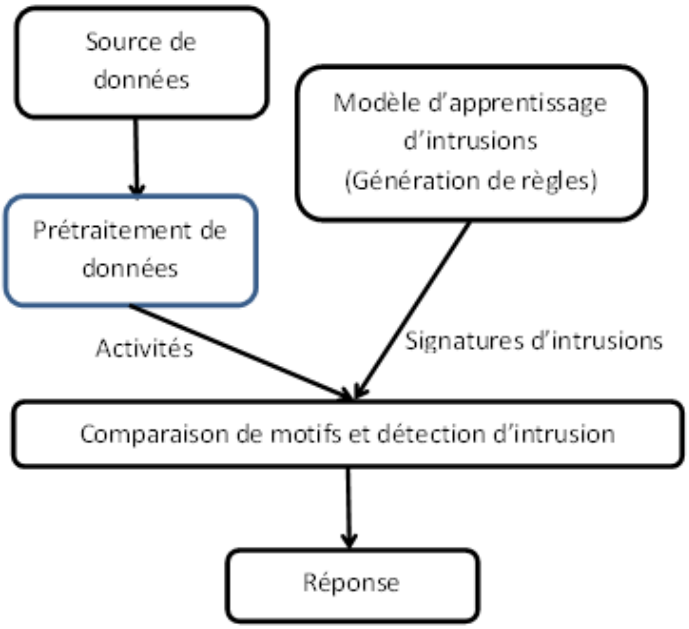


FIGURE 3.4 – Workkflow de détection d'abus d'utilisation

La détection d'anomalie quant à elle se base sur l'hypothèse que l'exploitation d'une faille du système nécessite une utilisation anormale de ce dernier[105], et consiste donc à apprendre les comportements normaux des sujets(au sens de Denning[105]) en observant le système pendant une période données dite phase d'apprentissage. Ces comportements sont représentés en utilisant des moyens statistiques comme les fréquences, moyennes, écart-types, etc. Ces profiles ou modèles sont stockés dans la base de données du système de détection et sont comparés, lors des périodes de surveillance, avec les comportementprésents des sujets. Toute déviation significative entre ces comportements donnera lieu à une alerte. En général, la base des comportements est mise à jour périodiquement pour prendre en compte les évolutions possibles des comportements des sujets. La détection d'anomalie et aussi un processus à cinq étape[116]. Les deux premières semblables à celles de la détection d'abus d'utilisation consistent à collecter les données d'audit des différentes sources de données(systèmes d'exploitation, applications, trafic réseau...) puis appliquer des opérations de nettoyage, de réduction de données et de dimensions et en fin des opérations d'extraction et de sélection de caractéristiques(Fig. 3.5).

Données et informations ainsi obtenues sont utilisées dans une troisième étape pour construire un profil pour les comportements normaux. Dans la phase de détection, les comportements capturés sont comparés à ce profil normal et toute déviation est reportée comme une intrusion. En réalité, la construction de ce profil normal n'est pas une tâche facile. Les données d'apprentissage équilibrées sont difficile à obtenir dans un réseau réel. De plus, tout changement dans l'environnement réseau, ou des services entraine un changement dans les motifs du trafic

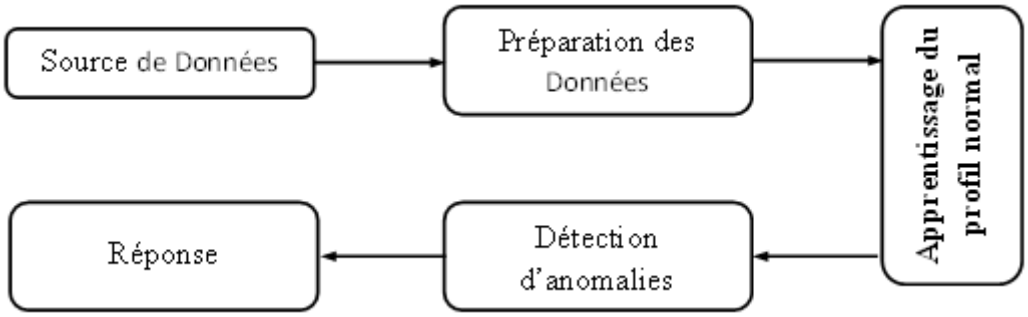


FIGURE 3.5 – Workkflow de détection d'anomalie.

normal. A fin de surmonter ces inconvénients propres au méthodes d'apprentissage supervisé des techniques semi-supervisées et non supervisées sont fréquemment employées.

Les techniques d'analyse de liens, de classification et de clustering sont les principales techniques de la fouille des données ayant été fortement utilisées dans la détection d'intrusion.

— **Classification** : Le processus de détection d'intrusion peut être vu comme un pure problème de classification qui consiste à scinder le trafic réseau en au-moins deux catégories : Trafic normale ou malveillant. Dans la phase d'apprentissage, une quantité, suffisante de données d'audit est collectée. Chaque instance de donnée recueillie est étiquetée comme étant normale ou anormale. Par la suite un algorithme de classification sera sélectionné à fin de construire un classificateur. Ce dernier devra être en mesure de prédire la classe ou la nature d'une nouvelle donnée. Bien que les techniques de classification peuvent être utilisées aussi bien pour la détection des anomalies que pour la détection des abus d'utilisation elles sont le plus souvent utilisées pour la détection des abus d'utilisation[239]. Le problème principale pour une telle approche est comment le système pourra-t-il apprendre la nature des activités à fin de ne pas confondre une activité non hostile avec une activité malveillante et vice versa. Plusieurs systèmes de détection d'intrusion à base d'algorithmes de classification ont été proposés dans la littérature. Certains de ces systèmes utilisent des techniques simples tel que les réseaux de neurones, les machines à vecteurs de support(SVM), les technique floues, ect. D'autre, par contre utilise des combinaisons de plusieurs techniques. Les techniques de classification les plus appliquées à la détection d'intrusion sont : Les arbres de décision, les algorithmes génétiques, les techniques de la logique floue, les technique d'immunologie et les réseaux de neurones.

— **Clustering** : Le clustering offre un moyen très efficace pour identifier les modèles cachés, particulièrement dans un contexte de détection d'intrusion, vue sa capacité à trouver de nouvelles attaques. Et selon Labib et Nemuri[234] Les techniques de clustering sont plus adaptées à un traitement temps-réel des données et offrent de très bonnes performance. De plus, ces techniques ne nécessitent aucun jeu de données pour l'apprentissage. Les techniques de clustering ont été appliquées avec abondance. Par exemple Portnoy et al[320], Eskin et al[125], et Chan et al[73] ont appliqué une largeur fixe et l'algorithme du k-plus proches voisins(k-nearst neighbor) aux fichiers log de connexions pour rechercher les activités malveillantes qui représentes des anomalies dans un trafic réseau. Marin et Al[273] ont également employé une approche semblable utilisant la quantification de vecteur d'apprentissage (LVQ), qui est conçue pour trouver la frontière optimale de Bayes entre les classes, employant l'algorithme du k-moyennes(k-means) pour déterminer le positionnement initial du vecteur. Malheureusement, cette approche ignore des malveillances très importantes, tel que les dénis de services, et l'écoute réseau massif, qui

doivent avoir leur propre cluster. Pour contourner ce problème, Chan et Al (2003) ont introduit une distance et une densité de clusters après avoir constater que les attaques appartenaient souvent à des clusters éloignés. Staniford et al[378] ont utilisé l'approche de recuit simulé(simulted annealing) pour grouper les évènements (paquets anormaux) de telle façon que les attaques coordonnées de balayage de ports(ports scans) soient groupées ensemble. Ils réussirent, ainsi, à ramener la forme polynomiale du temps d'exécution à une forme linéaire.

— **Analyse de liens** : Les règles d'association se sont avérées un outil simple et raisonnablement efficace pour distinguer un trafic normal d'un trafic malveillant. Le système de détection est un ensemble de règles d'association et de modèles d'épisodes fréquentes[239] qui peuvent être utilisées pour extraire la connaissance nécessaire sur la nature des données d'audit. Cette approche a deux caractéristiques attrayantes :

  — Les règles générées sont faciles à comprendre et par conséquent aisément vérifiables par un analyste.

  — Plusieurs ensembles de règles peuvent être produits et employées avec un méta-classificateur.

  Dans la détection des abus d'utilisation les règles sont vues comme des scénarios décrivant les attaques réseau. Le mécanisme de détection reportera une intrusion potentielle si une activité d'un utilisateur ou un programme s'avère compatible avec une règles pré-établie pour détecter une menace(Fig. 3.6).

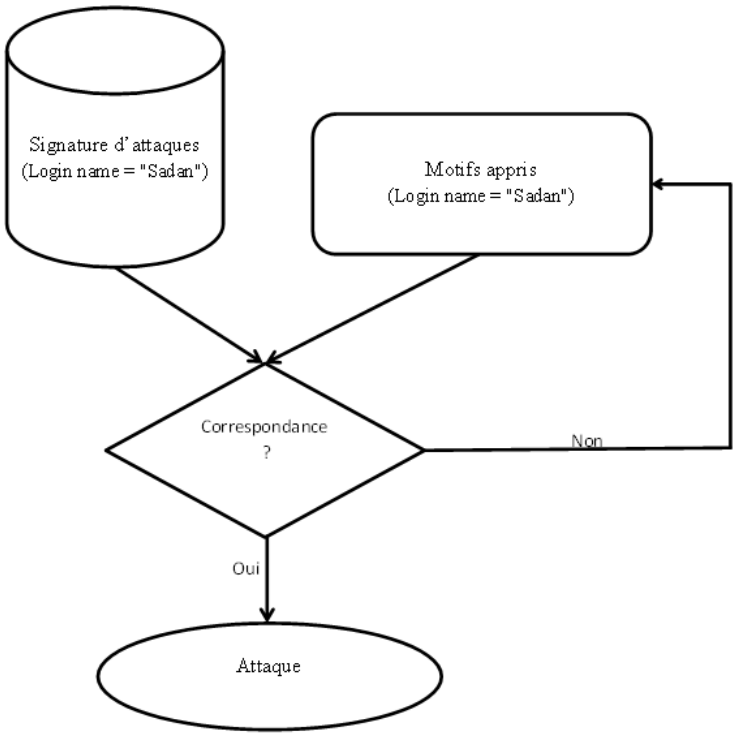


FIGURE 3.6 – Détection d'abus d'utilisation à base de règle

En détection d'anomalie les règles sont utilisées pour décrire les profiles normaux des utilisateurs, des programmes et des autres ressources. Le mécanisme de détection identifie une attaques potentielle si un programme ou un utilisateur effectue une action en contradiction avec une règle pré-établie.

La construction d'un modèle de détection utilisant les règles d'association se fait en deux étapes. La première consiste à explorer les données d'audit à fin d'extraire les mo-

tifs consistant décrivant les comportements des utilisateurs et les différents programmes. Dans la seconde étape, ces motifs sont utilisés pour construire (entrainer) des Classificateurs capable de détecter des anomalies. Un grand nombre de systèmes de détection d'intrusion utilisant des techniques à base de règle ont été proposés dans la littérature :

— RIPPER[83], est le plus populaire de ces système connu d'être rapide et stable ayant générer un ensemble de règles d'association et des motifs fréquents concises lui permettant de classifier le trafic réseau correctement.

— NSM(Network Security Monitor)[175] un système de détection à base de règle d'association fut le premier système de détection utilisant directement le trafic réseau comme une source primaire de données.

— JAM "Java Agent for Meta-learning"[381] est un système de détection d'abus utilisation développé à l'université de la Colombie utilise, lors d'une phase d'apprentissage, des règles d'association pour décrire les relations entre attributs des données d'audit et les épisodes fréquentes pour modéliser les motifs séquentiels décrivant les différents événements d'audit. Règles d'association et motif séquentiels sont utilisés pour construire des modèles décrivant les comportements intrusifs.

— Dans[241] Lee et al ont proposé de décrire une connexion à l'aide d'un ensemble d'attributs tel que la durée de connexion, service,... Puis des règles d'association ont été extraite à partir de l'ensemble de données ainsi obtenu pour construire des modèles de détection d'intrusion.

— ADAM("Audit Data Analysis and Mining")[41] est un système de détection d'anomalie à temps réel doté d'un module capable de distinguer, parmi les événements suspects, les fausses alertes des attaques réelles. Lors d'une phase d'apprentissage, ADAM construit un profil normal à partir d'un jeu de données ne contenant aucune attaque et le décrit à l'aide d'un ensemble de règles d'association. Lors de la phase de détection, il explore les donnes d'audit recueille lors d'un intervalle de δ secondes(δ étant un paramètre) à fin d'extraire de nouvelles règles d'association n'apparaissant pas dans le profil normal. Si le support d'une tel règle d'association dépasse un certain seuil prédéfini elle est considérée comme suspecte et est traitée par le module de classification, précédemment entrainé, qui décidera de la classer soit comme attaque connue, attaque inconnue ou comme fausse alerte. seuls les attaques réelles sont présentées à l'opérateur. Pour remédier au problème des règles d'association redondantes ou non pertinentes qui peuvent être générées, ADAM opère comme suit :

1. Les deux côtés d'une règle d'association son combinés, de façon à rendre une règle d'association sous la forme $\left(\bigwedge_{i=1}^{m} A_i = v_i\right)$ au lieu de celle représenté par l'équation 2.1 donnée dans la section 8 et rappelée ici

$$\left(\bigwedge_{i=1}^{m} A_i = v_i\right) \rightarrow \left(\bigwedge_{i=m+1}^{n} A_i = v_i\right) [s, c] .$$

   Il est à noter que le concept de confidence est obsolète pour ce type de règle d'association.

2. Seul les règles d'association qui comptent l'adresse de l'hôte source(SourceIP) et l'adresse (DestIP) ou numero du port(DestPort) de l'hôte de destination parmi leurs attributs sont autorisées, ie la règle de forme $\left(\bigwedge_{i=1}^{m} A_i = v_i\right)$ doit satisfaire

$$\exists k, l; (A_k = SourceIP \wedge A_l \in \{DestIP, DestPort\}) .$$

ADAM a été amélioré de deux façons :

1. Les règles d'association décrivant le profil normal ont été cataloguées en fonction de temps (heur de jour, jour de la semaine)[245] ce qui a permis de raffiner d'avantage le profil normal en spécifiant ses variations durant différentes périodes de temps.
2. l'introduction des règles d'association multi-niveau a permis la détection des attaques coordonnées et distribuées[42].

De plus, des techniques complémentaire d'extraction et de visualisation peuvent être utilisées pour améliorer d'avantage les performances des systèmes de détection d'intrusion et réduire les besoins en puissance calculatoire et en capacité de stockage [239, 320].

3.6 Quelques Techniques de Datamining appliquées à la détection d'intrusion

Dans cette section nous présentons les techniques de datamining qui ont été le plus largement utilisées dans la détection d'intrusion. Pour plus de détail sur ces techniques nous invitons les lecteurs intéressés à consulté les ouvrages en datamining telque [8, 9, 66, 169, 208, 237, 270]. Une étude approfondie sur le rôle du datamining dans la détection d'intrusion et proposée dans [34]. Cette étude concerne 75 différents papiers ayant proposé une approche de détection d'intrusion. A l'issue de cette étude, les auteurs concluent que 67% des papiers se sont intéressés au problème de détection d'anomalie et 23% se sont intéressés à la détection d'anomalie et d'abus d'utilisation et seulement 10% concernait la détection d'abus d'utilisation.

3.6.1 Les réseaux Bayésiens

Les réseaux bayésiens, proposés par Pearl[308], sont un formalisme de raisonnement probabiliste basé, conjointement, sur la théorie des graphes et sur la théorie des probabilités et en particulier sur le théorème d'inversion des probabilités introduit par le révérend Thomas Bayes[46] et qui a été, par la suit, approfondi par Laplace[236] en introduisant les probabilités des causes des événements. La théorie des graphes fournit les outils appropriés pour la description et l'exploitation graphique des relations de dépendance ou d'indépendance entre les variables. La théorie des probabilités, quant à elle, apporte un formalisme permettant la quantification des relations de dépendance en associant à chaque variable une loi de probabilité. Ainsi, un réseau bayésien est constitué de deux composantes. une graphique et l'autre quantitative. La composante graphique, dite aussi qualitative, permet de représenter d'une manière très simple la connaissance sous forme d'un graphe orienté acyclique. La composante quantitative(numérique) offre un moyen de quantifier l'incertitude des relations d'influence entre les variables. Formellement, étant donné un ensemble de variables aléatoires $X = \{X_1, \ldots, X_n\}$, un réseau bayésien $\mathcal{B} = \langle \mathcal{G}, \Theta \rangle$, où Θ est un ensemble de paramètres, est défini par[297] :

1. Un graphe orienté acyclique $\mathcal{G} = (X, E)$ représentant la composante graphique de $\mathcal{B}$. X dénote l'ensemble des nœuds où chaque nœud représente une variable aléatoire. A chaque variable aléatoire X_i est associée une table de probabilité locale $\theta_i \in \Theta$ représentant les probabilités des valeurs de X_i sachant toutes les valeurs possibles de ses parents. E est l'ensemble des arcs. Chaque arc entre deux nœuds de $\mathcal{G}$ traduit une relation de dépendance directe entre les deux variables aléatoires associées à ces deux nœuds. Il est à noté que le graphe $\mathcal{G}$ peut prendre plusieurs formes(Fig. 3.7) : chaîne, arbre, poly-arbre, graphe orienté acyclique avec boucle.

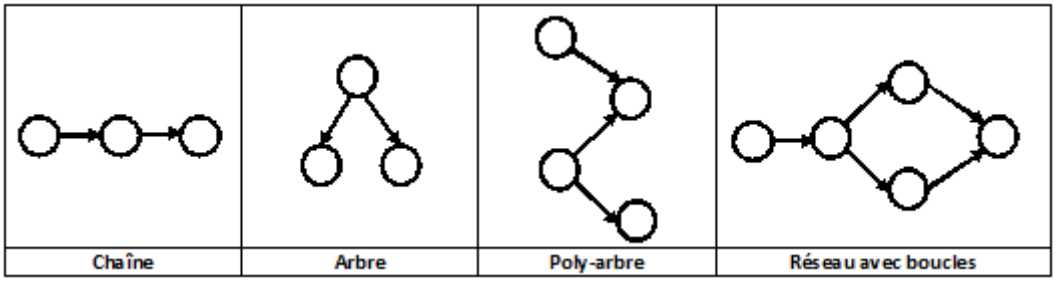


FIGURE 3.7 – Différentes structures de réseaux bayésiens

2. Un espace probabilisé fini (Ω, Z, P) tel que : $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Pa(X_i))$ où $Pa(X_i)$ est l'ensemble des parents du nœud X_i dans $\mathcal{G}$.

La probabilité conditionnelle d'une valeur prise par une variable aléatoire X_i sachant la valeur des autres variables X_j peut être calculée par le théorème de bayes comme suit[46]

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

Les distributions de probabilité locales doivent satisfaire las conditions de normalisation suivantes :

1. Si X_i est un nœud sans parents alors la distribution locale associée à X_i doit satisfaire la condition suivante : $\forall x_i \in \mathcal{D}_{X_i}, \sum_{x_i} P(x_i) = 1$
2. Si X_i a des parents, alors la distribution conditionnelle associée à X_i doit satisfaire :

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

Intuitivement, l'inférence dans un réseau Bayésien consiste à propager une ou plusieurs informations certaines au sein du graphe, pour en déduire comment sont modifiées les croyances concernant les autres nœuds. Le fait d'inférer les valeurs possibles des racines(causes) en observant les nœuds feuilles est dit diagnostic ou explication. La prédiction revient à observer les causes et à inférer les nœuds feuilles. La construction d'un réseau bayésien se fait en deux étapes. La première, dite qualitative, consiste à définir et à présenter sous forme d'un graphe les éventuelles relations d'influence entre les différentes paires de nœuds. Cette tâche est, selon D. M. Chickering[80], NP-difficile, principalement, à cause du fait que l'espace de recherche est exponentiel en fonction du nombre de variables décrivant le domaine. Plusieurs méthodes et approches ont été proposées pour la construction ou l'apprentissage automatique de la structure des réseaux bayésiens. Globalement ces approches peuvent être classées dans deux catégories[151]. La classe des méthodes **Score-Search** qui reposent sur deux éléments de base. A savoir une métrique de scoring et une procédure de recherche[80]. La métrique de scoring prend comme entrées un ensemble d'observations(données d'apprentissage) et une structure de graphe et retourne un score reflétant le degré d'adéquation des données à la structure. La procédure de recherche, quant à elle, consiste à trouver le réseau ayant le meilleur score. La deuxième classe englobe les méthodes à base de contraintes(Contraint-based-methods) dont l'idée globale consiste, étant donnée un ensemble de données d'apprentissage, à satisfaire, le plus possible, la plus grande indépendance. Des tests d'hypothèses statistiques sont utilisés pour déterminer la validité des indépendances conditionnelles. Il existe aussi des méthodes hybrides combinant les deux classes d'approches ainsi que des algorithmes de recherche utilisant un ordre dans les variables[151]. La deuxième étape de la construction du graphe, dite quantitative, quant

à elle, consiste à annoter le graphe, construit lors de la première étape, par une distribution jointe définie sur les variables. Autrement dit, elle consiste à quantifier les liens de dépendance qui existent entre les variables associées aux nœuds de la structure, préalablement apprises ou connue. Cela revient à construire les tables de probabilités conditionnelles θ_i locales relatives à chaque variable du graphe. Le calcule de ces probabilité peut se faire de deux manières différentes selon que les données d'apprentissage soient complètes ou incomplètes. Dans le cas des données complètes, les probabilités sont calculées par un processus d'apprentissage statistique basé sur le maximum de vraisemblance ou sur des estimations bayésiennes tel que le maximum a posteriori et espérance a posteriori. L'estimation du maximum de vraisemblance donne[297] :

$$\hat{P}(X_i = x_k | Pa(X_i) = x_j) = \hat{\theta}^{MV}_{i,j,k} = \frac{N_{i,j,k}}{\sum\limits_k N_{i,j,k}}$$

Où $N_{i,j,k}$ est le nombre d'apparition de l'événement $X_i = x_k$ sachant que les parents de X_i sont dans la configuration $x_j(Pa(X_i) = x_j)$.

Le principe de l'estimation bayésienne est quelque peu différent. Elle consiste à trouver les paramètres θ_i les plus probables sachant que les données ont été observées, en utilisant des aprioris sur les paramètres. La règle de Bayes nous dit que[297] :

$$P(\theta) \propto \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} (\theta_{i,j,k})^{\alpha_{i,j,k}-1} .$$

Où $\alpha_{i,j,k}$ sont les coefficients de la distribution de Dirichlet associée à la loi apriori $P(X_i = x_k | Pa(X_i) = x_j)$. L'approche de maximum à posteriori (MAP) nous donne alors :

$$\hat{P}(X_i = x_k | Pa(X_i) = x_j) = \hat{\theta}^{MAP}_{i,j,k} = \frac{N_{i,j,k} + \alpha_{i,j,k} - 1}{\sum\limits_k (N_{i,j,k} + \alpha_{i,j,k} - 1)}.$$

Où $\alpha_{i,j,k}$ sont les paramètres de la distribution de Dirichlet associée à la loi a priori $P(X_i = x_k | Pa(X_i) = x_j)$. Et L'approche d'espérance à posteriori(EAP) nous donne alors :

$$\hat{P}(X_i = x_k | Pa(X_i) = x_j) = \hat{\theta}^{EAP}_{i,j,k} = \frac{N_{i,j,k} + \alpha_{i,j,k}}{\sum\limits_k (N_{i,j,k} + \alpha_{i,j,k})}.$$

Dans le cas des données incomplètes, correspondant aux applications pratiques, où les données sont complètement manquantes ou ne sont observées que partiellement, on peut envisager d'utiliser l'algorithme EM[297].

Une variante simple des réseaux bayésiens est appelée réseau bayésiens naïfs qui représente la forme la plus simplifiée des réseaux bayésiens dont la partie graphique est constituée d'un graphe contenant un seul nœud racine, parent de tous les autres nœuds, et plusieurs nœuds feuilles(nœuds n'ayant pas de fils)(Fig 3.8).

Ce type de réseaux bayésiens, qui est plus adapté aux problèmes de classification, part de l'hypothèse que toutes les variables sont indépendantes. La classification est assurée en considérant le nœud parent comme une variable non observée précisant à quelle classe appartient chaque objet et les nœuds enfants comme étant des variables observées correspondant aux différents attributs spécifiant cet objet. Étant donnée un ensemble de classe $C = \{c_1, c_1, \ldots, c_k\}$, Un objet sera affectée à la classe c_r tel que :

$$r = \underset{c_i \in C}{\text{ArgMax}}\, P(x|c_i) \cdot P(c_i).$$

Afin d'alléger l'hypothèse d'indépendance conditionnelle des caractéristiques, il a été proposé d'augmenter la structure naïve en rajoutant des liens entre certaines caractéristiques[297]. Il

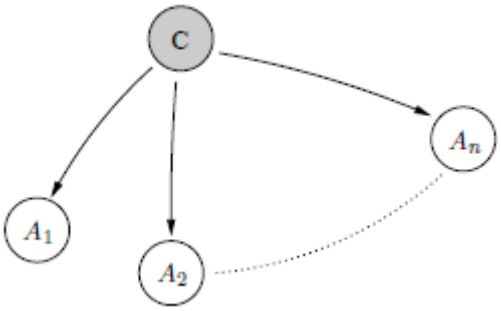


FIGURE 3.8 – Structures d'un réseau bayésien naïf

existe certaines spécialisations de réseaux bayésiens qui traitent des situations qui exigent, légèrement, plus de structure que le réseau bayésien général. Parmi ces structures on cite les modèles d'interaction causale, les réseaux Bayésiens dynamique et les diagrammes d'influences. Voir [198] pour plus de détailles.

Les Réseaux Bayésiens peuvent être utilisés pour l'apprentissage supervisé aussi bien que le non supervisé. Dans un cadre d'apprentissage supervisé où des variables cibles dépendent des variables explicatives et où les données sont complètes, Les réseaux bayésiens peuvent être utilisés comme une approche de classification et/ou de régression vue que la fonction de distribution $P(X_i|Pa(X_i))$ est essentiellement un modèle de classification et/ou de régression. En classification, les réseaux Bayésiens sont considérés comme des Classificateurs génératifs parce qu'ils codent la distribution de probabilité conjointe de la classe et des variables prédictives. Dans ce cas, l'estimation du maximum de vraisemblance des paramètres simple et, normalement, utilisée pour déterminer les paramètres du modèle. Néanmoins, plusieurs approches discriminantes pour les classificateurs Bayésiens ont été proposées. Par exemple, Huang et al [182] ont proposé une stratégie pour entrainer itérativement un classificateur Bayésien d'une manière discriminante par l'introduction dans la fonction d'optimisation un terme de pénalité décrivant la divergence entre les classes. Greiner et al [162] ont proposé une extension du modèle de régression logistique simple pour représenter n'importe qu'elle structure bayésienne. Ayant constaté que l'apprentissage des paramètres qui maximisent le log-vraisemblance est un problème NP-difficile, il proposèrent d'utiliser un algorithme du descente de gradient pour maximiser le log-vraisemblance conditionnel en utilisant le modèle de régression logistique étendu.

Un apprentissage Bayésien non supervisé peut parfois être inefficace, vu le nombre très élevé des paramètres nécessaires pour le calcul des distributions des probabilités conjointes d'une part et vu l'absence de l'information sur les classes au cours de l'apprentissage d'autre part. Cependant un modèle Bayésien non supervisé s'avère très précieux pour une exploration des données ou pour une classification préliminaire. AutoClass[77] est un simple exemple de programme de clustering utilisant un réseau Bayésien.

Historiquement, Les premières applications opérationnelles des réseaux Bayésiens ont été développées dans le domaine de la santé et particulièrement en diagnostic médical vue leur capacité à intégrer des ressources de connaissances hétérogènes, d'une part et à traiter des requêtes complexes d'autre part[297]. Aujourd'hui, les réseaux Bayésiens sont appliqués, avec succès pour créer des représentations probabilistes cohérentes de connaissances incertaines dans une large variété de domaines tel que le traitement du langage naturel, l'apprentissage automatique, la reconnaissance vocale, le datamining, les réseaux cellulaires, et bien d'autre domaines. Dans le domaine de la sécurité informatique, les modèles graphiques, les réseaux Bayésiens en particulier, sont devenus les principales approches. L'utilisation de l'approche bayésienne pour le développement des systèmes de détection d'intrusion a plusieurs avantages[403]. Les

méthodologies Bayésiennes permet le retour dans le passer pour déterminer les causes des évènements. Cette caractéristiques est très approprié pour connaître les causes des anomalies dans le trafic réseau[210]. Et étant basées sur l'hypothèse que les quantités d'intérêt(paramètres) sont régies ou dictées par des distributions de probabilité et que les décisions(classification) optimales peuvent être prises en raisonnant conjointement sur ces probabilités et sur les données observées[288, 427], les statistiques bayésiennes considèrent ces paramètres comme des variables aléatoires et non pas comme des constantes fixes mais inconnues qui peuvent être estimés à partir des échantillons aléatoires comme est le cas dans les techniques statistiques classiques. Il n'est pas possible d'avancer des assertions probabilistes sur les vrais paramètres s'ils sont considérés comme fixes et non pas aléatoires[403]. Avant de considérer les données courantes, les informations préalablement disponibles peuvent être utilisées pour construire des modèles de distribution apriori. Ainsi, les réseaux Bayésiens commencent par estimer des valeurs probables de ces paramètres inconnus, puis utilisent les données courantes pour ajuster les estimations trouvées. Et selon Kruegel et al[229], les réseaux bayésiens améliorent l'intégration des différentes sorties du modèle et permettent d'intégrer de façon transparente des informations supplémentaires offrant ainsi une manière plus sophistiquée à traiter les problèmes relatifs à la détection d'intrusion que les systèmes à base de règles ou à base de signatures. BenAmour et al [47] ont montré que les réseaux bayésiens naïfs, malgré leur simple structure et en dépit de leurs fortes hypothèses, sont réellement très compétitifs et ne présentent qu'une légère différence de performances par rapport aux arbres de décision dont la construction est en général, selon [188], un problème NP-Complet alors que celle des réseaux bayésiens naïfs est linéaire. Bien avant en 2001, Barbara et al [42] ont développé une méthode de détection d'intrusion basée sur une technique dite estimateurs pseudo-Bayés afin d'améliorer la capacité de leur système de détection baptisé ADAM(Audit Data Analysis and Mining)[41] qui utilise des techniques d'extraction de règles d'association pour séparer et détecter les événements malveillant dans les données du trafic réseau et un algorithme de classification pour les classés comme des instances de trafic normales ou instance anormales. L'avantage principale de cette méthode réside dans le fait qu'aucune connaissance sur les nouvelles attaques(inconnues) n'est nécessaire. Des probabilités apriori et a postériori sont dérivées à partir des informations relatives au trafic normal et aux attaques préalablement connues. L'intégration de la statistique bayésienne, via cette technique, à permet à ADM d'être en mesure de détecter de nouvelles attaques(non préalablement connues) et de réduire son taux de fausses alarmes. Valdes et al ont développé eBayes[396], un système de détection d'intrusion hybride, combinant les fonctionnalités de la détection à base de signatures et celles de l'approche comportementale. Et emploie une inférence bayésienne et un modèles de transition entre inférences pour déterminer si un fragment, particulier, de trafic réseau contient une attaque. L'efficacité de ce système a été démontrée par un ensemble de tests employant le jeux de données KDD'99 [91] et sur un site réel. Récemment, en 2015, L. Koc et al ont pu obtenir de plus meilleurs performances par l'intégration des réseaux Bayésien cachés(HNB-Hidden Naïve Bayes)[431] dans la construction de leurs modèle de détection d'intrusion[222]. Bien avant, en 2013, Elngar et al[122] ont proposé un système de détection d'intrusion combinant la méthode de sélection d'attributs PSO(Particle Swarm Optimization)[17], la technique de minimisation d'entropie d'information et les réseaux bayésiens cachées. Le processus de détection a été ainsi, accéléré tout en augmentant la précision. Il est à noter que les réseaux bayésiens ne sont plus robustes que si la taille de l'échantillon d'apprentissage est importante[156] et rencontrent beaucoup de difficultés avec les petits échantillons[427].

3.6.2 Arbre de décision

L'arbre de décision[59] est une structure de donnée hiérarchique implémentant la stratégie "diviser pour régner" qui consiste à identifier des sous problèmes, à leur trouver une solution,

puis à combiner ces solutions pour résoudre le problème général. C'est une méthode non-paramétrique très efficace qui est utilisée, en datamining, pour représenter aussi bien les modèles de classification que ceux de la régression. Elle est dite non-paramétrique dans le sens où aucune hypothèse n'est faite sur les densités des classes et la structure de l'arbre n'est pas fixée à priori mais nœuds et branches sont ajoutés durant le processus d'apprentissage selon la complexité du problème inhérent aux données[124]. Un arbre de décision est un arbre dirigé(un graphe connexe acyclique), qui peut être binaires ou n-aires, dans lequel :

- Chaque nœud intérieur, dit nœud de décision, correspond à un attribut décrivant les données,
- Chaque branche entre un nœud père et un nœud fils représente un test sur l'attribut de son nœud père. Usuellement, chaque test, associé à un nœud de décision, compare la valeur d'un unique attribut à une constante(Fig. 3.9), cependant, il existe des arbres dont les tests implémentent des fonction à un ou plusieurs attributs. Les réponses possibles aux tests correspondent aux étiquettes des branches issus de ce nœud[415].
- Chaque feuille représente la décision d'appartenance d'une instance de données à une classe vérifiant tous les tests du chemin menant de la racine à une feuille. Chaque chemin aboutissant à une feuille représente une règle conjonctive de classification.

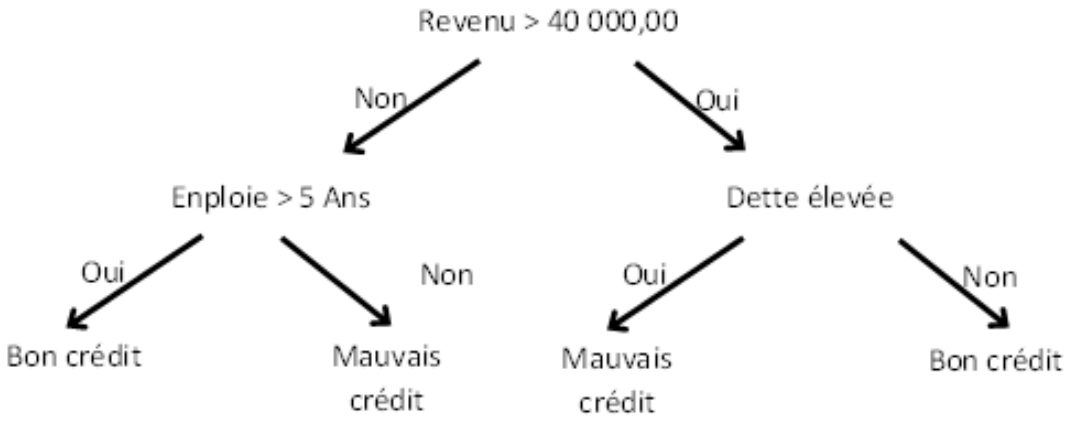


Figure 3.9 – Arbre de Décision Simple

Étant donné une base d'apprentissage, dans laquelle chaque instance de donnée est décrite avec un ensemble d'attributs, le processus de construction de l'arbre de décision consiste à[415]

- Créer le nœud racine de l'arbre
- Choisir un attribut à placer dans le nœud racine
- Faire étendre une branche pour chaque valeur possible de l'attribut choisi. Ce la revient à diviser l'ensemble des données en un nombre de sous ensemble, un pour chaque valeur,
- Répéter récursivement ce processus pour chaque branche ainsi crée en ne considérant que les instances de données ayant effectivement atteint cette branche. Si à un moment donné, tout les instances de donnée d'un nœud sont de la même classe on arrêt le développement de cette partie de l'arbre.

L'attribut sélectionné à chaque nœud, et la manière dont sa valeur doit être catégorisée sont choisis de façon à ce que les partitions résultantes soient "pures" dans la mesure du possible. L'impureté d'un nœud est généralement mesurée à l'aide de deux mesures : L'entropie et le coefficient de Gini.

1. L'entropie mesure la quantité du désordre ou de l'incertitude. Une partition ayant une faible entropie est relativement "pure" par contre une forte entropie exprime le fait que la

partition est un mélange de classes. En théorie d'information l'entropie d'une partition ou région $\mathcal{D}$ est définie comme suit :

$$H(\mathcal{D}) = -\sum_{i=1}^{K} P(c_i|\mathcal{D}) log_2 P(c_i|\mathcal{D})$$

où $P(c_i|\mathcal{D})$ est la probabilité de la classe c_i dans $\mathcal{D}$ et K est le nombre de classes. Si la région est pure alors l'entropie est nulle et atteindra sa valeur maximale $log_2 K$ si $\mathcal{D}$ est un mélange de classes ayant la même probabilité $P(c_i|\mathcal{D}) = \frac{1}{K}$. Si à un nœud la région $\mathcal{D}$ est scindée en deux sous-partition $\mathcal{D}_1$ et $\mathcal{D}_2$ alors l'entropie globale à ce nœud est donnée :

$$H(\mathcal{D}_1, \mathcal{D}_2) = \frac{n_1}{n} H(\mathcal{D}_1) + \frac{n_2}{n} H(\mathcal{D}_2)$$

où n, n_1 et n_2 représentent respectivement $|\mathcal{D}|$, $|\mathcal{D}_1|$ et $|\mathcal{D}_2|$.
Le Gain d'information apporté par chaque point de division(attribut) est défini, en terme de théorie de l'information, par :

$$Gain(|\mathcal{D}|, |\mathcal{D}_1|, |\mathcal{D}_2|) = H(\mathcal{D}) - H(\mathcal{D}_1, \mathcal{D}_2)$$

Ainsi, l'attribut pour lequel le gain d'information est le plus élevé sera choisi.

2. Le coefficient de Gini, basé sur l'entropie de Shannon, mesure la probabilité que deux instances de données, choisies aléatoirement(avec remise) dans un nœud, appartiennent à deux classes différentes. Pour un nœud t, le coefficient de Geni est défini comme suit :

$$G(\mathcal{D}) = 1 - \sum_{i=1}^{K} P(c_i|\mathcal{D})^2$$

Si la partition $\mathcal{D}$ est pure, la probabilité de la classe majoritaire est égale à 1 et celle des autres classes vaut 0 et le coefficient de Gini vaut 0. Dans le cas où toutes les classes constituant $\mathcal{D}$ ont la même probabilité $P(c_i|\mathcal{D}) = \frac{1}{K}$, alors le coefficient de Gini vaut $\frac{K-1}{K}$. Une partition ayant un faible coefficient de Gini est relativement "pure" par contre une forte valeur du coefficient de Gini exprime le fait que la partition est un mélange de classes(la quantité de désordre est importante). Le coefficient de Gini "weighted" peut être calculé par :

$$G(\mathcal{D}_1, \mathcal{D}_2) = \frac{n_1}{n} G(\mathcal{D}_1) + \frac{n_2}{n} G(\mathcal{D}_2).$$

où n, n_1 et n_2 représentent respectivement le nombres d'instances de données dans $\mathcal{D}$, $\mathcal{D}_1$ et $\mathcal{D}_2$. Le Gain d'information apporté par chaque point de division(attribut) est défini par :

$$Gain(|\mathcal{D}|, |\mathcal{D}_1|, |\mathcal{D}_2|) = G(\mathcal{D}) - G(\mathcal{D}_1, \mathcal{D}_2)$$

L'attribut pour lequel le gain d'information est le plus élevé sera choisi.

Il est à noté qu'en plus de l'entropie et du coefficient de Gini, il existe d'autre mesures qui sont aussi utilisées pour choisir le point de division d'une région nous citons en particulier le critère du χ^2 est celui de Lerman[242].

Selon Frank[148], les arbres de décision sont un exemple typique des méthodes de classification bien adaptées à la détection d'intrusion. Il s'avèrent très utiles pour trouver les structures dans des espaces à hautes dimensions et sont aussi utiles pour les mixtures continues et pour les données catégoriques, cependant ils ne sont pas appropriés aux problèmes non linéaires multivariés[349]. Dans le domaine de la détection d'intrusion, les arbres de décision on été

exclusivement utilisés pour la détection des abus d'utilisation car, selon [401], ils ne sont pas appropriés pour la détection des anomalies de plus, ils ne sont pas en mesure de détecter de nouvelles classes d'attaque, en général une nouvelle attaque est affectée à une classe par défaut comme est le cas l'algorithme C4.5 qui affect une nouvelle attaque à la classe "Normale". Cependant Bouzida et Cuppens[56] ont proposé une version modifiée du C4.5 dans laquelle une attaque nouvelle/inconnue est affectée à une nouvelle classe étiquetée "Inconnue". De leur côté Oha et al [331] ont tenté de réduire le taux des faux positif en revoyant la maniéré dont l'arbre est construit, la sélection des attributs produisant le moins de faux positifs et en prenant en considération les problèmes de sur échantillonnage des données d'apprentissage ainsi que les différents types d'erreurs pouvant être produits. Testée sur les données "KDD'99", cette nouvelle version s'est avérée plus performante que la version original mais a enregistré un taux de faux négatif plus élevé. Les test effectués par ben Amor et Al[47] sur les données KDD'99 ont montré que les arbres de décision fournissent des résultats légèrement meilleurs que ceux données par les classificateurs bayésiens naïfs, cependant le processus de construction de l'arbre de décision est plus lent. Utilisant le même jeux de données, G. Stein et Al[379] ont combiné les arbres de décision avec une technique de sélection d'attributs basée sur un algorithme génétique. Les attributs les plus appropriés ainsi obtenus sont utilisés pour construire l'arbre de décision qui a apporté un gain en détection pour certaines classes d'attaques(23 % pour l'attaque PROBE). Dans le même contexte, Sheen et Al[365] ont comparé les performances des arbres de décision combinés avec trois techniques de sélection d'attributs à savoir : χ^2, Gain d'information et ReleffF. Les résultat obtenus ont montré que la technique du chi^2 et celle du Gain d'information avaient des résultats équivalents et que la technique ReliefF[226] avait de faible performances.

3.6.3 Les algorithmes génétiques

initialement proposés par Holland[181] en 1975 puis développés par David Goldberg, sont une technique itérative de recherche basée sur la théorie d'évolution darwinienne appliquée aux modèles mathématiques. Le principe des algorithmes génétiques est très simple. Étant donné un ensemble de N individus choisi à priori au hasard et constituant une population initiale. Chaque individu x, dit chromosome, représente une solution potentielle à un problème donné et est constitué d'une chaîne de gènes. Le nombre de valeurs possibles pour une gène est dit cardinalité du gène. L'algorithme génétique(voir Fig. 3.10) fait évoluer, progressivement, la population initiale, au cours de plusieurs générations, tout en maintenant sa taille constante, par l'application de trois opérateurs de base : opérateur de sélection, de croisement et de mutation dans le but d'améliorer globalement la performance des individus. A chaque génération, la qualité de chaque individu est évaluée à l'aide d'une fonction positive f dite fonction d'adaptation(fitness). Les individus les plus aptes(ayant une valeur d'adaptation élevée) ont plus de chance d'être choisis pour participer à l'élaboration de la génération future. La probabilité $p_s(i)$ qu'un chromosome $i\,;\, i = 1, \cdots, TOP$(Taille de la population) soit sélectionné pour faire partie de la prochaine génération sachant sa fitness $f(i)$ est définie comme étant le rapport de sa fitness sur la somme des fitness associées aux individus de la population en cours ie :

$$p_s(i) = \frac{f(i)}{\sum_{k=1}^{TOP} f(k)}.$$

La façon la plus simple, pour faire la sélection, consiste à appliquer un opérateur de sélection stochastique pur. On construit, tout d'abord, un segment de longueur 1(voir Fig.3.11, puis on calcule la position de de l'individu i sur ce segment en calculant les probabilités de sélection

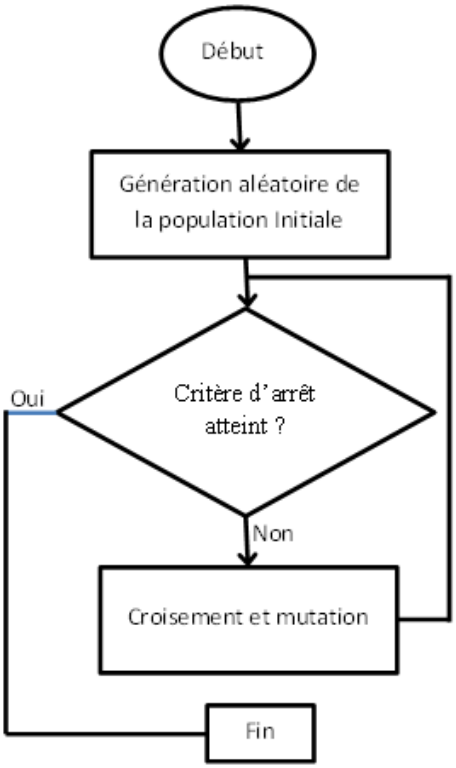


FIGURE 3.10 – Fonctionnement de l'algorithme génétique

cumulées[341] :

$$position(i) = \sum_{k=1}^{i} p_s(k).$$

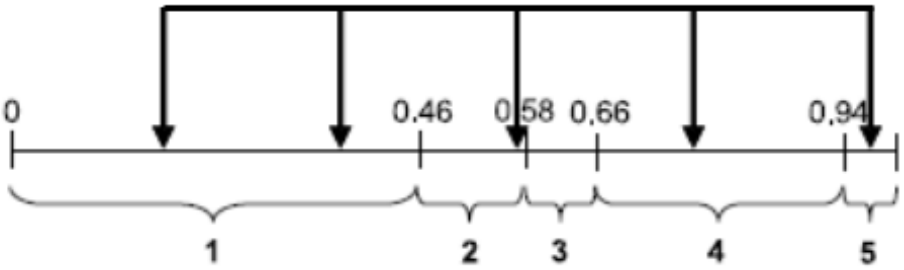


FIGURE 3.11 – Représentation de l'opérateur de sélection stochastique universelle pour des probabilités de sélection proportionnelles à la fitness(TOP = 5). Les chiffres indiquées sous les accolades représentes des zones de sélection de chacun des cinq individu[341].

Après quoi, en génère aléatoirement un nombre $u \in [0, 1]$ qui sera reporté sur le segment. Si $u \in [position(i-1), position(i]$ alors l'individu i sera choisi. Les individu, ainsi sélectionnés, sont alors croisés deux à deux selon une probabilité p_c dite probabilité de croisement qui est, en général, égale à 1. Le croisement consiste à choisir aléatoirement un((plusieurs) point(s) de césure sur le chromosome(père) puis, à permuter les portions de chromosomes de part et d'autre de ce(ces) point(s) de façon à créer de nouveaux individus(enfants). Le nombre d'individus parents à croisés et celui des enfants à engendrer doivent être choisi de façon à conserver la taille de la population constante. A titre d'exemple, concéderons le cas où deux parents P_1 et P_2 engendrent deux enfants E_1et E_2 qui remplaceront leurs parents. Si $P_1 = a_1a_2a_3a_4a_5a_6$ et

$P_1 = b_1b_2b_3b_4b_5b_6$ et si on choisi comme point de croisement le quatrième gène on obtient alors $E_1 = a_1a_2a_3a_4b_5b_6$ et $E_2 = b_1b_2b_3b_4a_5a_6$. Ce processus d'évolution imite les deux principaux mécanismes régissant l'évolution des êtres vivants :

— La sélection, qui favorise la reproduction et la survie des individus les plus performants.

— le croisement qui permet le brassage, la re-combinaison et les variations des caractères héréditaires des parents, pour former des descendants aux potentialités nouvelles.

Quelques individus issus de l'application des opérateurs de sélection et de croisement subissent un opérateur de mutation qui a pour but de garantir l'exploitation de l'espace de solutions et va permettre de maintenir une certaine homogénéité dans la génération en cours évitant, ainsi, une convergence trop rapide vers un optimum local. La proportion moyenne des individus devront subir une mutation est définie par un taux de mutation qui peut être constant pour tout l'algorithme comme il peut être une variable qui peut dépendre du temps[318], du locus[143] ou de la diversité de la population[339]. Pour permettre à l'algorithme génétique de converger on diminue le taux de mutation et à fin de lui permettre de sortir d'un éventuel optimum local on applique un taux de mutation élevé.

Les algorithmes génétique ont été largement utilisés dans la détection d'intrusion. L'une des premières tentatives d'utiliser les algorithmes génétiques pour la détection d'intrusion remonte à 1995 quand Crodbie et Spaffonrd[88] ont présenté une méthodologie à base de multi-agents utilisant des programmes génétiques pour détecter des comportements malveillants dans le trafic réseau. Chaque agent surveille un paramètre du trafic réseau. Cette approche était concluante quand plusieurs petits agents autonomes sont utilisés mais comportait un problème de communication entre agents de plus, si ces derniers ne sont pas correctement initialisés le processus d'apprentissage pourrait être trop long. Le schéma général de la détection d'intrusion à base d'algorithme génétique(Fig. 3.12) est globalement décrit comme suit[305] :

Dans une première étape, dite d'apprentissage, les données sur le trafic réseau sont collectées grâce aux différentes sondes du systèmes de détection. Puis un algorithme génétique est appliqué, en mode off-line, au données ainsi obtenues à fin d'extraire des règles de classification sous forme : if ⟨ condition ⟩ alors ⟨ action ⟩[367]. La condition décrit, habituellement, une correspondance entre la connexion réseau courante et un sous ensemble de règles stockées dans la base du système de détection. Le champ action fait référence à l'action prédéfinie dans la police de sécurité et qui peut être, par exemple, envoyer une alerte à l'intention de l'administrateur, inscrire un message dans le fichier journal, suspendre la connexion... Par exemple une règle peut avoir la forme suivante : "**if**(sce_IP_add = 124.12.5.18$_\wedge$ $dest\_IP\_add$ = 130.18.206.55$_\wedge$$dest_port_num = 21_\wedge con_time = 10.1\, seconds$) **then**(stop the connection)" et peut être interprétée comme suit : s'il existe une connexion ayant 124.12.5.18 comme adresse IP de l'hôte source(sce_IP_add), 130.18.206.55 comme adresse de l'hôte destination($dest_IP_add$), le port destination($dest_port_num$) ayant la valeur 21 et le temps de connexion(con_time) est 10.1 secondes alors la connexion doit être stoppée car l'adresse IP 124.12.5.18 est portée dans la liste noirs du système de détection. De ce fait toute requête provenant de cette adresse est rejetée. En réalité, seul les règles qui correspondes au activités malveillantes sont générées et testées sur l'historique des connexions. Les règles générées constituent la base de règles et son utilisées, en mode temps-réel, pour analyser les nouvelles connexion à fin de détecter tout trafic suspect. Une méthodologie de l'application des algorithmes génétique à la détection d'intrusion est succinctement présentée dans [246].

Dans le cadre d'intégration des algorithmes génétiques dans la détection d'intrusion, GASSATA[280] se compte parmi les premières tentatives. En fait, GASSATA est un prototype construit autour d'un algorithme génétique pour la détection des abus d'utilisations. Il définie un vecteur H de n hypothèses, où $H[i] = 1$ si une attaque i a lieu selon les hypothèse i, sinon $H[i] = 0$.

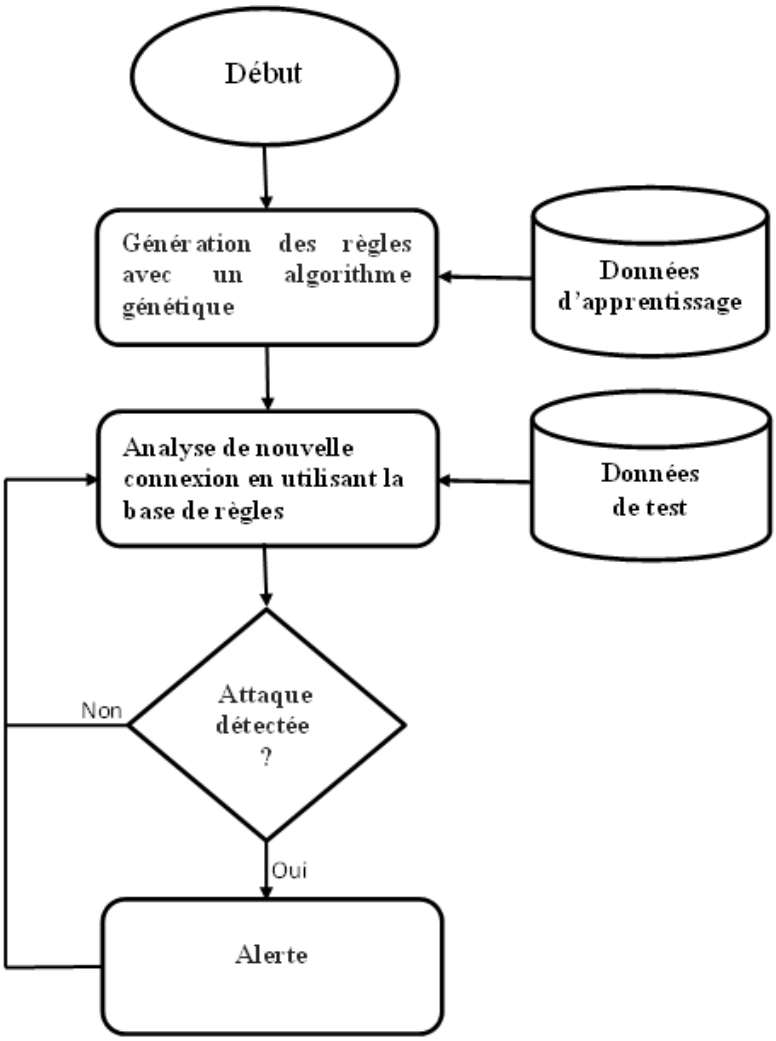


FIGURE 3.12 – Détection d'intrusion à base d'algorithme génétique

En conséquence la détection d'intrusion se ramène au problème de trouver le vecteur H qui maximise le produit WxH sujet aux contraintes $(AExH)_i \leq O_i$, où W est vecteur de n poids, AE une matrice d'événements d'attaques et O est un vecteur à n dimensions représentant une données d'audit. Chaque individu de la population correspond à un vecteur particulier H. la fonction de fitness est définie par :

$$Fitness = \sum_{i=1}^{n} W_i x I_i$$

où I_i représente un individu. Le système de détection se caractérise par un faible taux de fausse alarme est un taux de détection avoisinant 0.996. Cependant il n'est pas en mesure de localiser précisément une attaque, une intervention d'expert humain est requise pour analyser les données d'audit à fin d'ajuster la réponse. De son coté Chittur[81] à présenté un modèle de détection d'intrusion à base d'algorithme génétique ayant réaliser un taux de fausse alerte sensiblement bas. Pour déterminer la nature intrusive ou normale d'une instance de données d'audit, il utilisa une certaine formule C_i définie par :

$$C_i(x) = \sum_{i=1}^{n} \Re_{ij} \times x_j$$

où $\Re$ est l'**Ephemeral Random Constant-based coefficient** pour l'attribut x_j et n

est représente le nombre d'attributs. Une valeur seuil de C_i est prédéfinie, et chaque valeur dépassant ce seuil est classée comme attaque. La fonction de fitness utilisée est définie par :

$$F(\delta_i) = \frac{\alpha}{A} - \frac{\beta}{B}$$

.

où δ_i fait référence à un individu, α est le nombre d'attaque correctement détectées, A est le nombre total d'attaques, β représente le nombre de faux positifs et B est le nombre de connections normales. La valeur de la fitness varie de -1 à 1. Un fort taux de détection $\frac{\alpha}{A}$ et un faible taux de faux positifs $\frac{\beta}{B}$ induisent une forte valeur de la fonction de fitness pour un individu. Le principale problème de cet approche réside dans la sélection de valeur du seuil. Un seuil incorrecte peut facilement conduire à un taux très élevé de fausses alarmes dans la détection de nouvelles attaques. Lu et Traore[259] ont procédé à la génération des règle de classification à partir de l'historique du trafic réseau par le biais d'un programme génétique en utilisant la notion de support-confidence comme fonction de fitnes. Aussi ils ont utilisé des arbres d'analyse pour représenté leurs populations d'individus et non pas des chromosomes. Évalué avec l'ensemble de données de DARPA, ce modèle parvenait à détecter plusieurs intrusions et de nouvelles formes d'attaques également. Mais le taux de détection pour certaines attaques était très faible vue la nature aléatoires des paramètres de croisement et de mutation utilisés. Aussi d'un coté l'implémentation de ce modèle s'est avérée très difficile, vue l'utilisation de la programmation génétique et, d'un autre coté, son processus d'apprentissage nécessitait une grande masse de données et consommait beaucoup de temps.

En 2013, Moraveji et Al[290] proposèrent une approche de détection d'intrusion dans laquelle trois attributs seulement parmi quarante et un on été choisies, par une analyse aux composantes principale implémenté sous MATLAB, pour décrire une connexion. chaque attributs ainsi sélectionné représente une gène du chromosome. Comme chaque gène (attribut) est représentée par un bit, un chromosome représentant un individu est codé sur trois bits. Le but étant de choisir le plus petit ensemble d'attributs décrivant une connexion tout en garantissant un taux de détection élevé. La fitness de chaque règle est évaluée comme suit :

$$Fitness = \frac{a}{A} - \frac{b}{B}$$

où

- a : Nombre des attaques correctement détectées
- A : Nombre total des attaques dans la table d'apprentissage
- b : Nombre de faux positifs
- B : Nombre total des connexions normales dans la table d'apprentissage.

Ce modèle se caractérise par un fort taux de détection et un faible taux de faux positifs et un temps de réponse très réduit et peut être appliqué au réseaux à trafic intense.

En 2012, Uppaluri et Al[392] avaient déjà utilisé les même attributs avec une fonction de fitness définie par

$$Fitness = \frac{f(x)}{f(sum)}$$

avec $f(x)$ est la fitness d'une l'entité x et $f(sum)$ est le total des fitness de toutes les entités, afin de construire un système de détection, à base d'algorithme génétique, pour détecter uniquement huit attaques. Il ont enregistré un taux de détection avoisinant les 83.65%. De leurs coté, Azween et Al[35] ont proposé une approche à trois phases. Dans la première, ils ont effectué une transformation des attributs à l'aide d'une analyse linéaire discriminante. Puis, un algorithme génétique a été appliqué pour sélectionner un sous ensemble optimal d'attribut.

Dans la troisième phase la technique du "SVM kernels" à été utilisée pour la classification(Fig. 3.13).

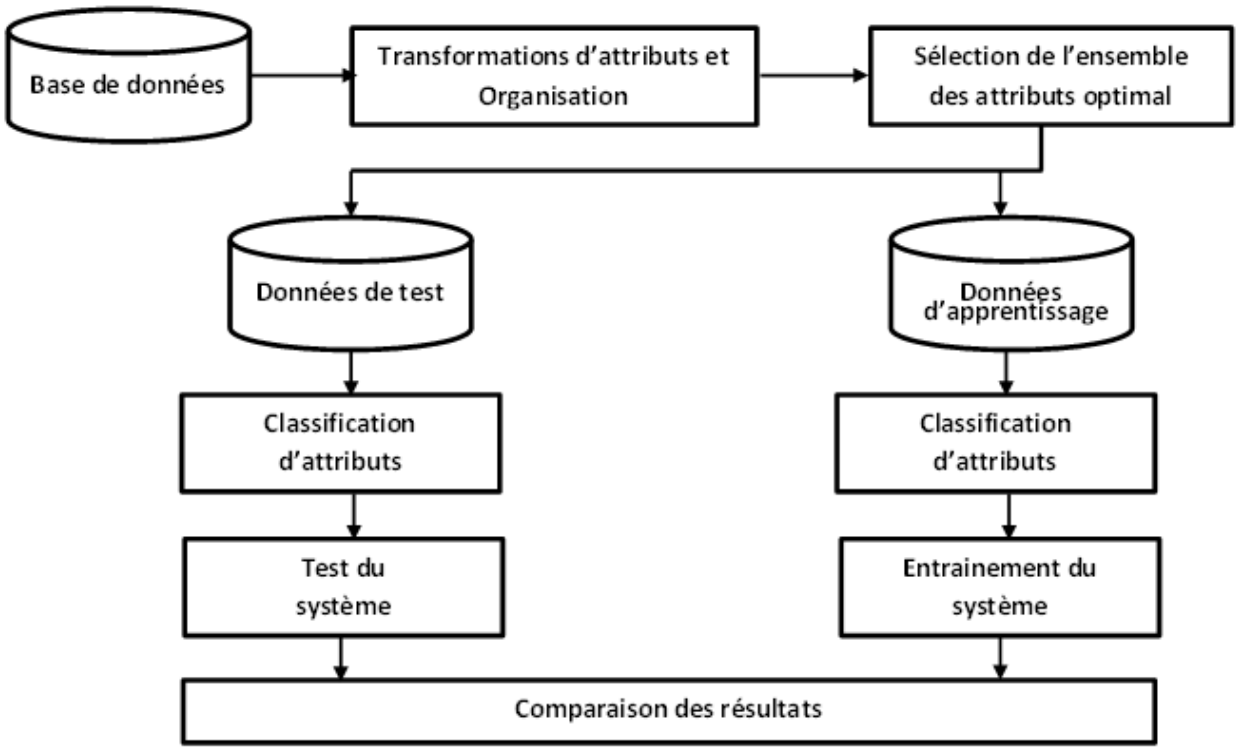


FIGURE 3.13 – Diagramme du système de détection proposé par Azween et Al.

Des revues et comparaisons des plus importantes approches de détection d'intrusion à base d'algorithmes génétiques peuvent être trouvées dans [123, 305, 205].

3.6.4 Les Réseau de neurones

Un réseau de neurones se définisse comme étant un modèle mathématique, adaptatif distribué pour le traitement parallèle des informations[153]. Son fonctionnement ressemble à celui du cerveau sur deux aspects :

- La connaissance est acquise par le réseau à travers un processus d'apprentissage
- Les connexions entre les neurones, connues sous le nom de poids synaptiques servent à stocker la connaissance

Il est constitué par la connexion d'un nombre très important d'unités de calcul simples, dites neurones formels(Fig. 3.14). Chaque neurone, opérant seulement sur l'information locale de façon asynchrone, calcule une sortie s unique sur la base de ses entrées e_i pondérées par des coefficients synaptiques (poids) w_i et combinées en une seule entrée $E = \sum w_i.e_i$.

La sortie s est généralement donnée par $s = f(\sum_{i=0}^{n} w_i x_i + \theta)$. f étant la fonction d'activation du neurone et θ est appelé biais. Les neurones se distinguent par la nature de leurs fonction d'activation f qui peut être une simple fonction d'identité pour les modèles linéaires, sinusoïdal $f(x) = \dfrac{1}{1+e^x}$, à seuil $f(x) = \mathbf{1}_{[0,+\infty]}(x)$, Gaussienne $f(x) = \sqrt{\frac{1}{2\pi}} e^{\frac{-x^2}{2}}$, ou stochastique $f(x) = 1$ avec la probabilité $\frac{1}{1+e^{\frac{-x}{H}}}$, 0 sinon, ext.

Les modèles linéaires et sinusoïdaux sont bien adaptés aux algorithmes apprentissage impliquant une rétro-propagation du gradient car leur fonction d'activation est différentiable. Le

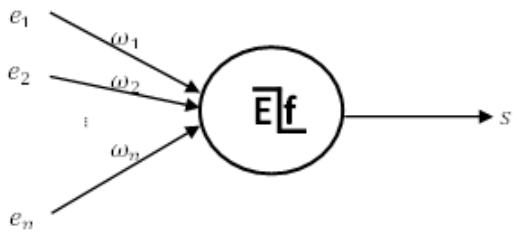
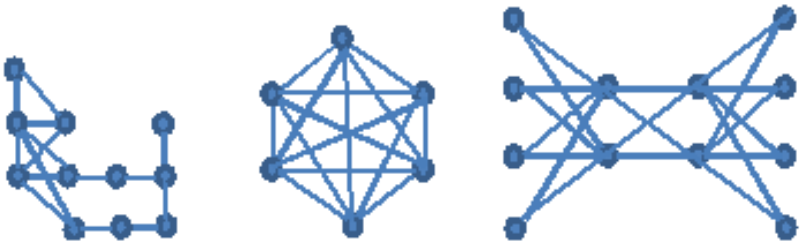


FIGURE 3.14 – Neurone formel

modèle à seuil est sans doute plus conforme à la "réalité" biologique mais pose des problèmes d'apprentissage. Le modèle stochastique, quant à lui, est utilisé pour des problèmes d'optimisation globale de fonctions perturbées ou encore pour les analogies avec les systèmes de particules.

Les possibilités d'arrangements entre les neurones sont multiples. Plusieurs configurations peuvent avoir lieu, mais quelques schémas typiques sont souvent utilisées et peuvent être catégorisés en quatre catégories(Fig. 3.15) :



(a) Connexion partielle. (b) connexion complète. (c) réseau à couches

FIGURE 3.15 – Différentes configuration de réseau de neurones

— Réseaux partiellement connectés(Fig. 3.15(a)) dont lesquels chaque neurone est connecté à quelques neurones appartenant à son voisinage.

— Réseaux à connexions complètes(Fig. 3.15(b)) où chaque neurone est connectés à tous les autres neurones du réseaux.

— Réseaux à couches où les neurones(Fig. 3.15(c)) sont répartis en couches. Les neurones d'une couche sont connectés aux neurones de la couche en aval. On compte une couche d'entré, un certain nombre de couches cachées et une couche de sortie. Les neurones de la couche d'entrée reçoivent les données sujets de l'analyse. Leurs nombre est directement déterminée par le nombre de variables d'entrées. Ceux de la couches cachée non aucun contact direct avec l'environnement extérieur et leurs fonctions d'activations sont en général non linéaires. Leurs nombre n'est pas implicite et doit être ajusté. Enfin, ceux de la couche de sortie donnent le résultat obtenu après compilation par le réseau des données entrées dans la première couche. Leurs nombre est est directement déterminé par le nombre de variables qu'on veut en sortie.

On compte aussi d'autre configuration ou architecture des réseaux de neurones tel que par exemple les réseaux de neurones à couches et à connexion locales et ceux à connexion récurrentes. La conception d'un réseau de neurones est un processus à quatre grandes étapes. La première consisté à préparer l'échantillon des données. Comme dans le cas de l'analyse des don-

nées, cette étape est cruciale va déterminer le choix de type de réseau, le nombre optimal des neurones ainsi que la façon dont l'apprentissage, les tests et la validation doivent être menés.

La deuxième étape consisté à élaborer la structure du réseau. Cette dernière dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : un perceptron standard, un réseau de Hopfield, un réseau à décalage temporel (TDNN), un réseau de Kohonen, un ART-MAP etc... Dans le cas du perceptron par exemple, il faudra aussi choisir le nombre de neurones dans la couche cachée. Une fois la structure du réseau établie, le réseau ainsi obtenu est sujet à un processus d'apprentissage qui consiste à calculer les meilleures valeurs des poids w_i et suit globalement l'algorithme 12.

Algorithme 12 : Algorithme d'apprentissage

Entrées : Ensemble de données d'apprentissage $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$;
Output : Les Poids w_i ; $i = 1, \cdots, |D|$;

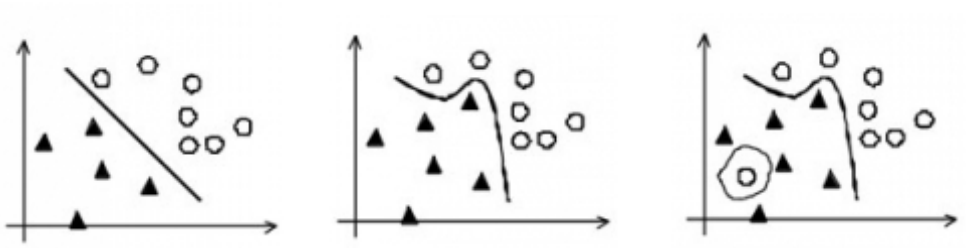
début
  $w_d \leftarrow 0$ Pour Tout $d = 1, \cdots |D|$;
  $\theta \leftarrow 0$ **pour** $iter \leftarrow 1$ **à** $MaxIter$ **faire**
    pour tous $(x, y) \in D$ **faire**
      $a \leftarrow \sum_{d=1}^{D} w_d x_d + \theta$;
      si $ya \leq 0$ **alors**
        $w_d \leftarrow w_d + yx_d$ Pour tout $d = 1, \cdots, |D|$;
        $b \leftarrow \theta + y$
      fin
    fin
  fin
  retourner $w_1, w_2, \cdots, w_n$
fin

En fin la quatrième étape, consiste à tester et à valider le modèle résultant. Dans le cas général, le test et la validation du réseau de neurones consiste à le faire fonctionner sur des portions de l'échantillon d'apprentissage préalablement préparés. Dans le cas de petits échantillons, on ne peut pas toujours avoir ces sous-échantillons, tout simplement parce qu'il n'est pas toujours possible d'avoir suffisamment de données. On a alors parfois recours à des procédures comme la cross-validation ou le le bootstrapping. Il est à noté que le problème de la configuration optimale d'un réseau de neurones a longtemps constitué une question ouverte, néanmoins il existe une variété de méthodes basées sur les statistiques[393, 394].

Les réseaux de neurones constituent une méthode d'approximation de systèmes complexes, particulièrement utile lorsque ces systèmes sont difficiles à modéliser à l'aide des méthodes statistiques classiques. Ils sont également applicables dans toutes les situations où il existe une relation non linéaire entre une variable prédictive et une variable prédite. Par leur nature et leur fonctionnement, ils peuvent détecter les interactions multiples non linéaires parmi une série de variables d'entrée, ils peuvent donc gérer des relations complexes entre les variables indépendantes et les variables dépendantes (Fig. 3.16).

Cependant, les réseaux de neurones ne fournissent pas d'explications sur le raisonnement l'ayant amenés à proposer un résultat ou une décision. De plus, le paramétrage d'un réseau de neurones est délicat et peut influer considérablement sur la pertinence des résultats fournis.

Les champs d'application des réseaux de neurones est très variés. Sans être exhaustif, on



(a) Discrimination linéaire. (b) non linéaire. (c) non continue.

FIGURE 3.16 – Discrimination entre les "triangles" et les "ronds"

cite entre autres la reconnaissance de la voix, de formes, de signaux et d'images, le diagnostic médical, analyse exploratoire de données, ext. En détection d'intrusion, l'application des réseaux de neurones peut être envisagée pour modéliser statistiquement, classer ou prédire les comportements des utilisateurs[279]. En détection d'anomalie, le système apprend à prédire la commande suivante à partir de la séquence, d'une certaine longueur prédéfinie, des précédentes de commandes[306]. Pour la détection des abus d'utilisation, un réseau neuronal peut être implémente de deux façons : La première, consiste à filtrer les données entrantes pour détecter tout évènement suspect et de le transmettre à un système expert. Dans la deuxième, le réseau neuronal est implémenté comme un système autonome de détection d'abus d'utilisation qui collecte et analyse les données[310]]. [214] prétend que le réseau de neurones est en mesure d'analyser les données issues du trafic réseau même si ces dernières sont incomplètes ou déformées. Ce qui rend leur application en détection d'intrusion très attractive. Ryan et al comptent parmi les premier chercheurs ayant tenter d'utiliser les réseaux de neurones en détection d'intrusion. En effet 1998, ils ont proposé un système de détection d'intrusion à base de réseau de neurones[345], à deux couches, entrainé à identifier les profiles des utilisateurs. A la fin de chaque session, il évalue les commandes de chaque utilisateur à la recherche d'éventuelles intrusion. Le système fonctionne sur des données issues des journaux système d'un environnement UNIX. Chaque vecteur d'attributs décrit les connexions d'un utilisateur durant tout la durée de la simulation effectuée sur le simulateur PlaNet Neural Network. il ont enregistré ainsi un taux de faux positifs avoisinant les 7% et un taux de faux négatifs de 4%. De son coté, Cannady[82] utilisa un réseau de neurones à trois couches pour classifier, en off-line, les connexion en normales et intrusives. Ces dernières sont décrite à l'aide de neufs attributs. Le jeu de données utilisé se compose de 10 000 enregistrement dont 1 000 corresponds à des attaques simulées. 30% des enregistrements ont été utilisés pour l'apprentissage. Le système ainsi construit parvenait à classer, correctement, les enregistrements (normal, attaques) dans 89 à 91% des cas. En 1999 Ghosh et Al [154] proposèrent un système de détection d'intrusion orienté hôte pour la détection des anomalies par l'analyse des profils de programmes. Ces derniers sont construits lors d'une phase d'apprentissage par un processus de capture des appels système effectués par les différents programmes. Le système, qui se présente comme un réseau de neurone à une seule couche caché, utilise l'algorithme "Lucky Bucket" pour mémoriser, temporairement, les événements malveillants récents. Appliquer à la détection d'anomalie, le système arrivait à identifier des attaques connues et inconnues avec un taux de détection de 77% avec 3% de fausses alarmes. Mais il a enregistré un très fort taux de fausses alarmes une fois utiliser pour la détection d'abus d'utilisation. Pour améliorer les performances du systèmes Ghosh et Al[155] ont fait recours au réseau d'Elamn qui est basé sur une topologie "feed-forward" avec en plus des nœuds contextuels retenant les informations des entrées précédentes. Le système ainsi construit parvenait à identifier 77% des attaques sans fausse alertes. Dans la même année, Cunningham et Libbmann[90] proposèrent une autre approche de détection d'intrusion basée sur un perceptron multi-couche

pour la détection des abus d'utilisation. Leur démarche consiste à chercher pour une attaque des mots clés spécifiques dans le trafic réseau à partir d'un hôte UNIX. De bonnes performance de détection ont été atteintes(80%) juste en utilisant 30 mots clés. En 2000 Rhodes[342] et Al proposèrent l'utilisation d'un réseau de neurones auto-organisé(self-organizing) pour la détection d'anomalie. Il(le réseau de neurones) utilise une collection de maps spécialisées pour traiter le trafic réseau pour chaque couche de protocole séparément. Un réseau neuronal est entrainé à reconnaître les activités normales d'un seul protocole.

P. K. Ganesh et D. Devaraj[152] proposèrent une méthodologie de détection d'intrusion combinant une technique de sélection d'attributs à base d'information mutuelle et d'un simple réseau de neurones en aval(feed-forward neural networks) entrainé par l'algorithme de rétropropagation(back propagation algorithm). Lors de la phase d'entrainement, le réseau de neurones est entraîné à capturer la relation sous-jacente entre les entrées choisies et les sorties. Les test effectués sur les données KDD'99 du DARPA ont montré que cette approche détecte les intrusions avec précision et est bien adapté pour des applications en temps réel.

Iftikhar Ahmad et al [15] Ont évalué cinq différentes méthodes de détection d'intrusion à base de réseaux de neurones. L'évaluation été basée sur deux critères. Le premier concerne l'adaptabilité, l'apprentissage minimum, les performances, la maturité et l'aptitude. Le second critère, considéré comme secondaire, concerne le taux minimum de faux négatifs, le coût, le temps, le traitement des intrusion coordonnées et variées. A l'issue de cette études, ils conclurent que les approches combinées utilisant les réseaux de neurones semblent être des tactiques plus appropriées pour la détection d'intrusion que les autre en matière de mise à jour, taux de détection, faux positifs, faux négatifs et sur le plan de flexibilité.

Al-Jarrah[25] a utilisé un réseau de neurones à convolution (TDNN : Time Delay Neural Network) pour identifier le comportement des attaques. Le système se compose de cinq modules : Un moteur de capture de paquets, module de pré-traitement, un module de reconnaissance de motif à base de réseau de neurones, un réseau de neurones classificateur et en fin un module d'alertes. Le renifleur capture, en temps réel, les paquets, de type ICMP, TCP, et UDP, passant par certaines interfaces physiques, puis extrait les attributs pertinents requis en phase de reconnaissance. Ces derniers serons transmis au module de pré-traitement via de canaux, un pour l'attaque d'écoute et l'autre pour l'attaque de balayage des ports. En réalité le module de pré-traitement et composé par deux sous-module. Un pour l'attaque d'écoute et l'autre pour l'attaque de balayage des ports dont le rôle consiste à extraire les attributs pertinentes pour décrire les attaques d'écoute et de balayage des ports et produire des descriptions possibles des comportements des attaques. Ces descriptions sont utilisées par un système de reconnaissance de motif à base de réseau de neurones à fin de reconnaitre les attaques. Le système ainsi entrainé est capable de produire une réponse immédiate dans un temps constant. Les tests d'évaluation des performances du système effectués sur l'ensemble de données de DARPA ont montré que ce système parvenait à identifier touts les types d'attaques de façon beaucoup plus rapide que les systèmes à base de règles tel que SNORT.

3.6.5 Les machines à support de vecteur

Les machines à support de vecteur (Support vector machines-SVM) sont une classe d'algorithmes d'apprentissage pouvant être utilisés pour la classification, la régression, l'estimation de fonction de densité et bien d'autres applications. Les SVMs sont essentiellement basées sur le principe de minimisation du risque structurel(SRM-Structural Risk Minimisation)[398] et la théorie d'apprentissage statistique de Vladimir Vapnik[397]. Par conséquence, elles fournissent de bonnes performances de généralisation en classification. Étant donné un ensemble de données d'apprentissage $S = \{(X_i, Y_i), i = 1, \ldots, n\}$ constitué d'une suite de couples de variables

aléatoires telles que :

— (X_i, Y_i) sont indépendantes et identiquement distribuées de loi inconnue.
— $X_i \in \mathcal{R}^d$ sont dites variables d'entrées
— $Y_i \in \{+1, -1\}$.

Les SVMs consistent, dans le cas des données linéairement séparables, à trouver un hyperplan(droite dans le cas de deux dimensions) qui sépare au mieux deux classes. L'équation de ce hyperplan séparateur est[268] :

$$\langle \omega, x \rangle + b = \omega^T x + b = 0$$

Où $\langle \cdot, \cdot \rangle$ dénote le produit scalaire, $\omega \in \mathcal{R}^d$ le vecteur poids et $b \in \mathcal{R}$ dit biais sont des paramètres du modèle. La fonction de décision, pour une observation x, peut être exprimer comme suit :

$$\begin{cases} \text{Si } \langle \omega, x_i \rangle + b > 0 \text{ alors } y_i = +1. \\ \text{Si } \langle \omega, x_i \rangle + b < 0 \text{ alors } y_i = -1. \end{cases}$$

Qui peut être simplifiée comme suit :

$$H(x) = sign(\omega^T x + b).$$

Et puisque les deux classes sont linéairement séparables, aucune instance de donnée ne se trouvera sur l'hyperplan séparateur, il conviendrait alors de considérer la fonction de décision décrite par les deux inégalités suivantes :

$$\begin{cases} \text{Si } \langle \omega, x_i \rangle + b > +1 \text{ alors } y_i = +1. \\ \text{Si } \langle \omega, x_i \rangle + b < -1 \text{ alors } y_i = -1. \end{cases}$$

qui peuvent être combinées en une même inégalité :

$$y_i(\omega^T x_i + b) \geq 1;\ i = 1, \ldots n$$

.

La région qui se trouve entre les deux hyperplans H_1 et H_2 donnés respectivement par : $\omega^T x + b = +1$ et $\omega^T x + b = -1$ est appelée région de généralisation de la machine d'apprentissage(Fig. 3.17).

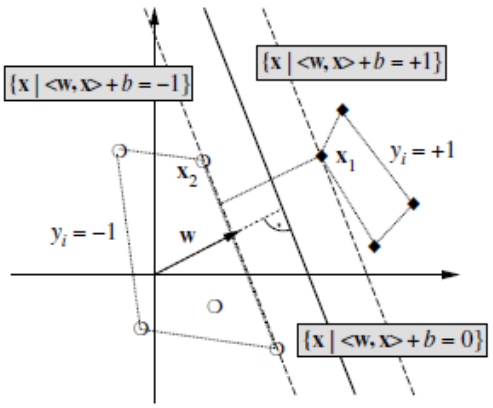


FIGURE 3.17 – Région de Généralisation[355]

Plus cette région est importante, plus est la capacité de généralisation des SVMs. La maximisation de cette région est l'objectif de la phase d'entrainement. Si on considéré une instance de donnée x_k appartenant à la classe y_k, on peut se trouver dans l'une des quatre situations suivantes[355]

1. $y_k \cdot (\omega^T x_k + b) > 1$: l'instance de donnée est bien classée et ne se situe pas dans la zone de la marge et ne représente pas un vecteur de support.
2. $y_k \cdot (\omega^T x_k + b) = 1$: l'instance de donnée est bien classée et se situe aux frontières de la zone de la marge et représente un vecteur de support.
3. $0 < y_k \cdot (\omega^T x_k + b) < 1$: l'instance de donnée est bien classée et se situe dans la zone de la marge et ne représente pas un vecteur de support.
4. $y_k \cdot (\omega^T x_k + b) < 0$: l'instance de donnée est mal classée(se trouve dans le mauvais coté de l'hyperplan séparateur) et ne représente pas un vecteur de support.

L'hyperplan optimal est celui qui assure une région de généralisation(la marge) maximale. Cette marge est définie comme étant la distance d'une observation à la surface de décision et dépend du vecteur poids ω. Les deux paramètres ω et b peuvent être re-dimensionnés tel que les points les plus proches de l'hyperplan satisfaits : $|(\omega \cdot x_i) + b| = 1$. On considérant deux observations x_1 et x_2 issues de classes différentes avec :
$|(\omega \cdot x_1) + b| = 1$ et $|(\omega \cdot x_2) + b| = 1$ alors la marge sera donnée par la distance perpendiculaire de ces deux observation à l'hyperplan c-a-d :

$$\frac{\omega}{\|\omega\|} \cdot (x_1 - x_2) = \frac{2}{\|\omega\|}$$

Parmi tous les hyperplans qui séparent les données, il existe un unique hyperplan qui maximise la marge de séparation entre classes :

$$\underset{(\omega,b)}{\text{Max}} \ \ min\{\|x - x_i\| : x \in \mathcal{R}^d; (\omega^T x_i + b) = 0; i = 1, \ldots, n\}$$

L'hyperplan optimal cherché est la solution du problème d'optimisation exprimé par :

$$\begin{cases} \underset{(\omega,b)}{\text{Min}} \ \frac{1}{2}\|\omega\|^2 \\ \text{Sous les contraintes : } y_i \cdot (\omega^T x_i + b) - 1 \geq 0; \ \forall i = 1, \ldots, n. \end{cases}$$

Ce dernier problème d'optimisation peut être convertie en un problème dual équivalent introduisant les multiplicateurs de Lagrange :

$$L(\omega, b, \alpha) = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{n} \alpha_i \{y_i \cdot (\omega^T x_i + b) - 1\}$$

Où les α_i sont les multiplicateurs de Lagrange non négatifs. L'optimisation du Lagrangien L est effectuée en le minimisant par rapport aux variable primaires ω et b et en le maximisant par rapport aux variables duales α_i. Le point selle doit satisfaire les conditions nécessaires de stationnarité correspondant aux conditions Karush-Kuhn-Tucker(KKT), nous trouvons pour les variables primaires, les équations suivantes :

$$\frac{\partial L}{\partial b} = 0 \text{ et} \frac{\partial L}{\partial \omega} = 0.$$

qui se traduisent par :

$$\sum_{i=1}^{n} \alpha_i y_i = 0; \ \ \omega = \sum_{i=1}^{n} \alpha_i y_i x_i$$

Ainsi, ω peut être calculer en fixant seulement n paramètres. On remplaçons ω dans le Lagrangien L par sa nouvelle formulation, le nombre de paramètre à fixer ne dégondera plus

de la dimension de l'espace d'entrée mais sera relatif à la taille de l'échantillon d'apprentissage. Nous obtiendrons le problème dual équivalent suivant :

$$\begin{cases} \underset{\alpha}{\text{Max}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T \cdot x_j) \\ \text{sujet à :} \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i \geq 0. \end{cases}$$

Ce dernier problème d'optimisation peut être résolu en appliquant les méthodes standards de programmation quadratiques. En notons par $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*)$ la solution optimale obtenue, le vecteur poids ω^* de l'hyperplan à marge maximale recherché s'écrit comme suit :

$$\omega^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

La valeur optimale du paramètre b, noteé b^*, peut être calculée à partir des contraintes primales :

$$b^* = \frac{\underset{y_i=-1}{\text{Max}} \langle \omega^*, x_i \rangle + \underset{y_i=+1}{\text{Min}} \langle \omega^*, x_i \rangle}{2}.$$

Une fois les paramètres α^* et b^* calculées, la fonction de décision peut être formulée, pour une nouvelle observation x, comme suit :

$$H(x) = sign\left(\sum_{i=1}^{n} \alpha_i^* y_i \langle x_i, x \rangle + b^* \right).$$

Il est à noter que seul les α_i^* correspondant aux observations se trouvant sur les hyperplans canoniques sont non nuls. Ses observations sont appelées vecteurs de supports et peuvent être considérés comme des représentants de leurs classes car si l'échantillon d'apprentissage n'était constitué que de ses observations on aurait obtenu le même hyperplan optimal. Les α_i^* correspondantes aux vecteur de supports sont appelés des valeurs de supports. En réalité, les données ne sont pas toujours linéairement séparable, ce qui signifie que le modèle à marge maximale n'est pas toujours valide. A fin de contourner ce problème les contraintes doivent être un peu relaxées afin de permettre une certaine tolérance aux erreurs de classification. On parle alors d'une marge de classification souple(Soft Margin). La relaxation des contraintes se fait par l'introduction de n variables $\xi_i \geq 0$, dites variables de relaxation ou d'écart $(\xi_i)_{1 \leq i \leq n}$[268] comme suit : $y_i(\omega^T x_i + b) \geq 1 - \xi_i$; $i = 1, \ldots n$.(Fig. 3.18).

Pour qu'une donnée d'apprentissage x_i soit mal classée, il faut que la variables ξ_i correspondante soit supérieur à 1. On doit chercher, donc, un hyperplan qui maximise, à la fois, la marge de classification et la somme des erreurs de classification $\sum_i \xi_i$. Le problème dual devient :

$$\begin{cases} \underset{\omega, b, \xi}{\text{Min}} \frac{1}{2} \|\omega\|^2 + C \sum_i \xi_i \\ \text{Sous les contraintes : } y_i \cdot (\omega^T x_i + b) \geq 1 - \xi_i; \ \forall i = 1, \ldots, n. \\ \xi_i \geq 0. \end{cases}$$

Où C, paramètre de pénalisation de relaxation, est une constante positive libre (mais fixe) qui représente une balance entre les deux termes de la fonction objective à savoir la marge de classification et les erreurs permises. Autrement dit entre la maximisation de la marge de classification et la minimisation de l'erreur de classification. Plus C est importante, moins

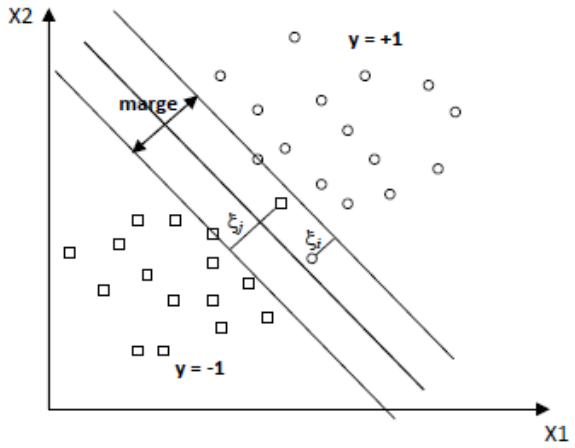


FIGURE 3.18 – SVM à marge souple

d'erreurs sont autorisées. Ce dernier problème d'optimisation peut être convertie en un problème dual équivalent introduisant les multiplicateurs de Lagrange α_i et β_i on obtient :

$$L(\omega, b, \alpha, \beta, \xi) = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i y_i(\omega^T x_i + b) - 1 + \xi_i - \sum_{i=1}^{n}\beta_i\xi_i.$$

dont la résolution nous conduit au problème dual suivant :

$$\begin{cases} \underset{\alpha}{\text{Max}} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j (x_i^T \cdot x_j) \\ \text{sujet à :} \\ \sum_{i=1}^{n}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C. \end{cases}$$

La seul différence avec le cas des données séparables réside dans le fait que les multiplicateurs de Lagrange α_i ne peuvent pas dépasser C. La fonction de décision conserve la même forme.

On peut avoir à la fois un modèle linéaire et un ensemble très riche de fonctions de décision non-linéaires en utilisant l'astuce de noyau, introduite par Cortes et Vapnik[87], avec des hyperplans à marge maximale. L'idée consiste à projeter les données d'apprentissage dans un espace F, dit espace de re-description, de dimension p plus élevée que celle de l'espace d'origine grâce à une fonction ϕ non-linéaire dite **Mapping function**(Fig 3.19) et d'appliquer, par la suit, la même méthode d'optimisation dans le nouveau espace. Le produit scalaire $\langle x_i, x_j\rangle$ sera remplacé par $\langle\phi(x_i), \phi(x_j)\rangle$ qui peut être facilement calculé avec une fonction symétrique K dite noyau et définie comme suit :

$$K(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$$

Cette dernière doit satisfaire les conditions de Mercer. On dispose de plusieurs famille de fonctions noyau qui sont très appropriées aux besoins des SVMs. Le tableau 3.1 présente quelques exemples de fonctions noyau couramment utilisées. Cependant, il a noté que la fonction sigmoïde ne satisfait pas les conditions de Mercer pour certaines valeurs de ces paramètres et certaines données[268]. Et selon Hsu et al[185], la fonction à base radiale (Radial Basis Function-RBF) est un premier choix très raisonnable.

L'utilisation de l'astuce du noyau nous permet d'avoir une forme plus générale de la fonction de décision :

$$H(x) = sign\left(\sum_{i=1}^{n}\alpha_i y_i K(x_i, x_j) + b\right)$$

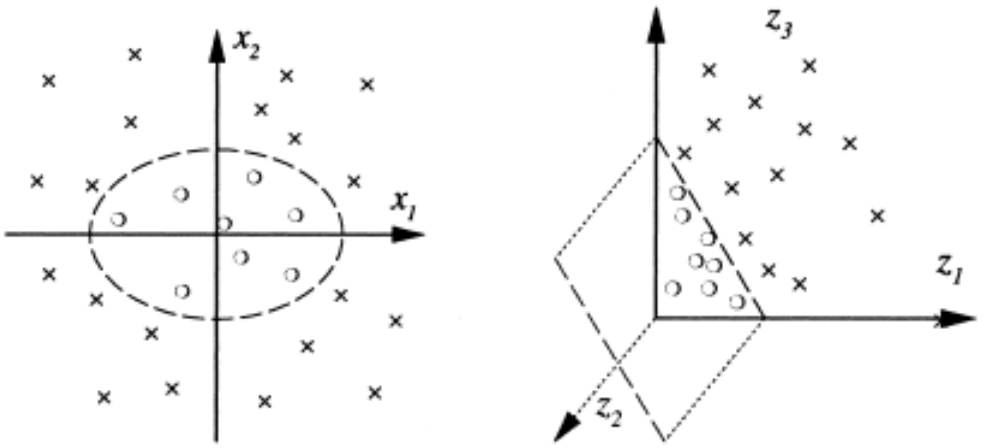


FIGURE 3.19 – SVM à base de noyau
Les données appartenant à $\mathcal{R}^2$ sont projetées dans $\mathcal{R}^3$ via une fonction $\phi : (x_1, x_2) \mapsto (z_1, z_2, z_3) \equiv (x_1^2, \sqrt{2}x_1 \cdot x_2, x_2^2)$[268]

TABLE 3.1 – Quelques fonctions noyaux

	Noyau	$K(x_i, x_j)$
1	Noyau linéaire	$K(x_i, x_j) = x_i^T \cdot x_j$
2	Noyau gaussien	$exp(-\frac{\|x_i - x_j\|^2}{2\sigma})$
3	Polynômial de degré d	$\left((x_i^T \cdot x_j) + \eta\right)^d$
4	Sigmoïde	$tanh(\gamma(x_i^T \cdot x_j) + \eta), \gamma > 0$
5	Fonction à base radiale(RBF)	$exp(-\gamma\|x_i - x_j\|^2), \gamma > 0$

et le problème d'optimisation quadratique suivants :

$$\begin{cases} \underset{\alpha}{\text{Max}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{sujet à :} \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \\ \alpha_i \geq 0. \end{cases}$$

Bien que les SVMs sont initialement conçues comme des classificateurs binaires, plusieurs approches ont été développées pour les problèmes multi-classes. Ces approches réduisent le problème de classification multi-classes à une composition de plusieurs problèmes de classification binaire. La méthodes dite Un-contre-Tous ou Un-contre-Reste[354], qui est la plus ancienne de ces méthodes, consiste à déterminer pour chaque classe C_k un hyperplan $H_k(w_k, b_k)$, défini par la fonction de décision $H_k(x) = sign\left(\omega_k^T x + b_k\right)$, la séparant des autres classes. A tour de rôle, chaque classe C_k est considérée comme étant la classe positive(+1) et les autres $K - 1$ comme classe négative(-1). Pour déterminer, la classe d'une nouvelle observation x on la présente à tous les K Classificateurs, et la décision s'obtient en appliquant le principe "winer takes all". La classe retenue est celle associée au classificateur ayant renvoyé la valeur la plus élevée. Autrement dit, on retient la classe C_r tel que :

$$r = \underset{1 \leq k \leq K}{\text{ArgMax}}(\omega_k^T x + b_k).$$

La méthode Un-contre-Un, due à Kner et Al[221], quant à elle tente de discriminé chaque classe d'une autre. Autrement dit elle construit un classificateur pour chaque paire de classes.

Ainsi, pour les K classes, $K(K-1)/2$ fonctions de décision sont apprises. Pour affecter une nouvelle observation, on la présente aux classificateurs ainsi construits, la décision s'obtient par vote majoritaire. La prédiction correspond alors à la classe qui gagne le plus de "duels".

Les SVMs se comptent parmi les méthodes d'apprentissage les plus populaires dans le domaine de détection d'intrusion en raison de leur bonne capacité de généralisation et de classification des données non-linéaire en utilisant différentes fonctions noyaux ainsi qu'à leur capacité de surmonter le problème de la malédiction de dimensionalité. Et, selon Lazarevic et al [238], elles sont les meilleurs en matière de détection de nouvelles attaques. Comparées aux réseaux de neurones, les SVMs présentent plusieurs avantages lorsqu'elles sont appliquées à la détection d'intrusion :

1. La vitesse qui une caractéristique très requise pour la détection d'intrusion en temps-réel.
2. L'évolutivité qui une qualité importante pour les grande cyber-infrastructures à flux d'information intensif.
3. La capacité à mettre à jour les échantillons d'apprentissage dynamiquement. Ce qui rend les SVM plus performants notamment contre les modèles d'attaques polymorphes.
4. Peuvent atteindre l'optimum globale ce qui leur permet de contrôler facilement le problème de sur-apprentissage ou sur-ajustement(overfitting).

La supériorité des SVMs sur les réseaux de neurones en matière de détection d'intrusion à été prouvée par plusieurs auteurs, notamment par Mukkamala et al[293], qui ont utilisé cinq classificateur SVM dont un pour identifier le trafic normal, et les quatre autres pour identifier chacun des quatre types d'intrusions injectées dans la base de donnée KDD'99[91]. Les performance de chaque algorithme, appliqué sur sept ensembles différents d'attributs, ont atteint le seuil de 99%, alors que les réseaux de neurones, avec une longue période d'entraînement, n'ont pas dépassé le seuil de 87,07%.

Globalement, Les SVMs ont été utilisées en détection d'intrusion soit comme outil de construction de modèles de comportement normal, soit comme outil de sélection d'attributs pertinents pour la détection d'intrusion ou encore, elles ont été combinées à d'autre techniques d'apprentissage automatique pour construire des systèmes de détection d'intrusion. Le temps d'apprentissage des SVMs est le plus grand obstacle à leurs utilisation notamment en détection d'intrusion où de grande masses de données à très hautes dimensionalité sont à analyser. Selon Yu et al [425] il faudrait une année pour entrainer une machine à support de vecteurs opérant sur un jeux de données constitué d'un million d'enregistrements. Pour remédier à ce problème, plusieurs approches et techniques ont été proposées à fin d'augmenter les performances d'entrainement de ces classificateurs par une sélection aléatoire ou par une approximation de la marge de classification[7, 70, 136, 215]. D'autre auteurs, tel que J. Haweliya et al [174], ont suggérer l'utilisation des SVMs semi-supervisées. En effet ces dernières, ayant hérité des SVMs leur solide théorie, permettent d'améliorer le pouvoir de généralisation d'une part et permettent l'exploration et l'analyse des données non étiquetées[112]. Et à fin de réduire la complexité algorithmique des SVMs qui est de l'ordre de $\mathcal{O}(m^2 \cdot p)$ où p représenté la dimension des données et n la taille de l'échantillon d'apprentissage dans les SVms à noyau, des chercheurs, tel que P. Yadav et D. Singh [418] ont fait recours aux implémentations parallèles des SVMs. Dans [33, 301, 340, 344] les auteurs présentent des revues des différentes approches de détection d'intrusion à base des SVMs, seules ou combinées avec d'autre méthodes de datamining, ayant été proposées dans la littératures.

3.6.6 La logique floue

La logique floue est une extension de la logique booléenne classique dont l'objet consiste à étudier la représentation des connaissances imprécises et le raisonnement approché. Gacôgne[149] la situe à côté des heuristiques de résolutions de problèmes, des systèmes experts, de l'apprentissage, de l'intelligence artificielle distribuée et même du traitement de la langue naturelle, domaines qui composent les techniques d'intelligence artificielle au sein des sciences cognitives. Initialement introduite par Lotfi Zadeh[428] en 1965 en se basant sur sa théorie mathématique des ensembles flous qui repose sur la notion d'appartenance partielle : chaque élément appartient partiellement ou graduellement aux ensembles flous qui ont été définis. Les contours de chaque ensemble flou (Fig. 3.20) ne sont pas "nets" , mais "flous" ou "graduels". Dans cette théorie, un ensemble flou A est défini par sa fonction d'appartenance μ_A, qui à tout élément x d'un univers de référence $\mathcal{X}$ associe un degré d'appartenance avec lequel x appartient à un sous ensemble flou A. Formellement, cette fonction d'appartenance est définie comme suite :

$$\mu_A : \mathcal{X} \to [0,1]$$

$$x \to \mu_A(x)$$

qui est une sorte de généralisation de la traditionnelle fonction caractéristique d'un ensemble ordinaire $B \subset \mathcal{X}$

$$\mu_B : \mathcal{X} \to \{0,1\}$$

$$x \to \mu_B(x)$$

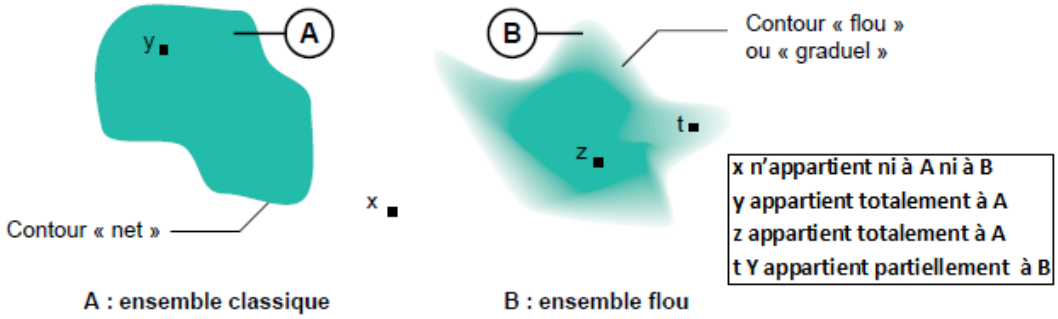


FIGURE 3.20 – comparaison d'un ensemble classique et d'un ensemble flou.

En réalité, la fonction d'appartenance diffère d'une fonction caractéristique par le fait qu'elle peut prendre n'importe quelle valeur dans l'intervalle [0,1](Fig. 3.21).

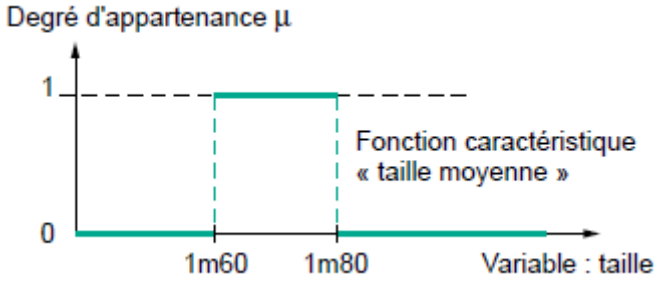
Une fonction d'appartenance peut avoir plusieurs formes, dont les plus utilisées sont présentées dans la figure suivante(Fig. 3.22).

Tout sous ensemble flou A d'un univers $\mathcal{X}$ peut être représenté par :

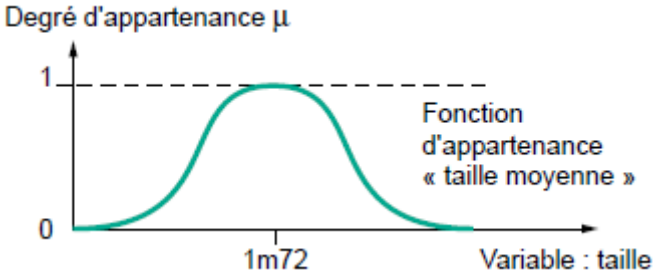
$$A = \{(x, \mu_A(x)), x \in \mathcal{X}\}.$$

On observe trois cas possibles :

1. $\mu_A(x) = 0$ si x n'appartient pas à A.
2. $0 < \mu_A(x) < 1$ si x appartient partiellement à A.
3. $\mu_A(x) = 1$ si x appartient entièrement à A.



(a) Fonction caractéristique.



(b) Fonction d'appartenance.

FIGURE 3.21 – Fonction caractéristique Vs Fonction d'appartenance.

La fonction d'appartenance $\mu_A(x)$ inclut ou exclut donc ses extrémité, tout élément x au sous ensemble A, mais entre les valeurs extrêmes le degré d'appartenance varie à proportion de la proximité à l'ensemble(Fig. 3.20).

Un ensemble flou A est caractérisé par un support, un noyau, une hauteur, un cardinal, et une α-coupe[410].

— Le support d'un sous-ensemble flou $A \in \mathcal{X}$, noté $S(A)$, est l'ensemble de tous les éléments qui lui appartiennent au moins un petit peu. Formellement il est défini par : $S(A) = \{x \in \mathcal{X} | \mu_A(x) > 0\}$. L'ensemble flou dont le support est un singleton est appelé "Singleton flou".

— Le noyau $N(A)$ est l'ensemble ordinaire qui contient tous les éléments x de $\mathcal{X}$ réellement dans A et est défini formellement par : $N(A) = \{x \in \mathcal{X} | \mu_A(x) = 1\}$.

— la hauteur $H(A)$ est définie comme étant la plus grande valeur du degré d'appartenance dans A. Et est formellement défini par $H(A) = \sup_{x \in \mathcal{X}} \mu_A(x)$. Le sous-ensemble flou A est dit "normal" si $H(A) = 1$ et est dit sous normal si $H(A) < 1$. S'il y'a un seul point ayant un degré d'appartenance égale à 1, alors ce point est appelé la valeur modale de A.

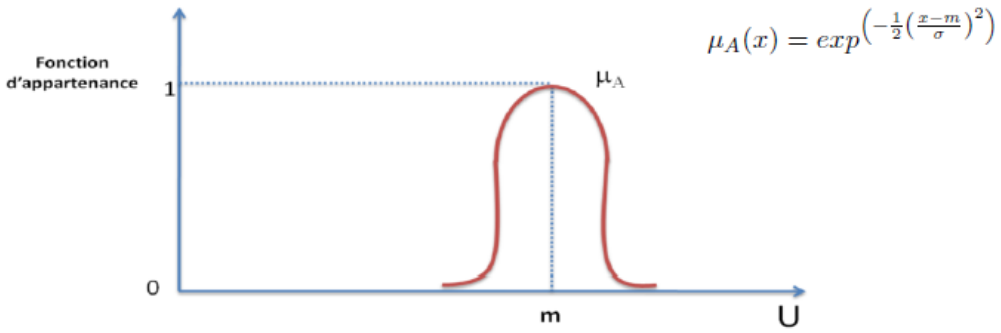
— Le cardinal d'un ensemble flou A de support fini est égale à la somme de degrés d'appartenance des éléments de ce support :

$$|A| = \sum_{x \in S(A)} \mu_A(x).$$

Dans le cas d'un support infini, le cardinal est donné comme suit :

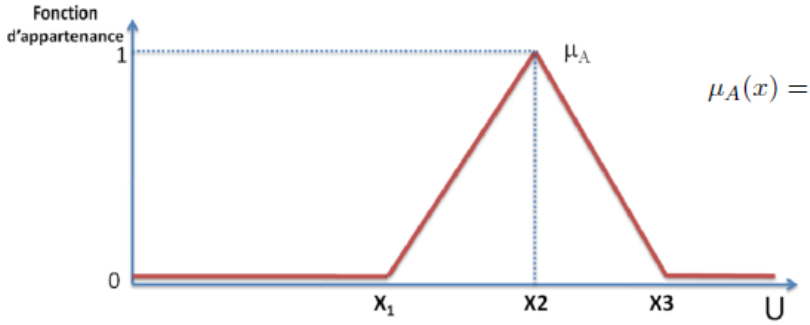
$$|A| = \int_x \mu_A(x) dm(x).$$

avec $\int_x dm(x) = 1$. Si A est sous-ensemble ordinaire de $\mathcal{X}$, son cardinal est le nombre d'éléments qui le composent.



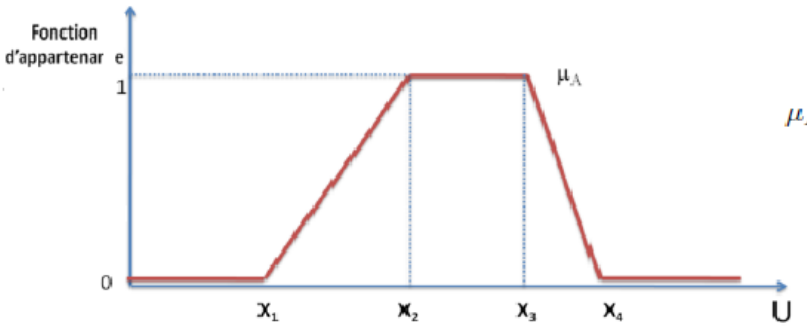
$$\mu_A(x) = exp^{\left(-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right)}$$

(a) Fonction d'appartenance gaussienne.



$$\mu_A(x) = \begin{cases} \frac{x-x_1}{x_2-x_1} & si\ x_1 \leq x < x_2 \\ \frac{x-x_3}{x_2-x_3} & si\ x_2 \leq x < x_3 \\ 0 & sinon \end{cases}$$

(b) Fonction d'appartenance triangulaire.



$$\mu_{\cdot} \begin{cases} \frac{x-x_1}{x_2-x_1} & si\ x_1 \leq x < x_2 \\ 1 & si\ x_2 \leq x < x_3 \\ \frac{x-x_4}{x_3-x_4} & si\ x_3 \leq x < x_4 \\ 0 & sinon \end{cases}$$

(c) Fonction d'appartenance trapézoïdale.

FIGURE 3.22 – Différentes formes d'une fonction caractéristique.

— La α-coupe A_α de $\mathcal{X}$ associé à A pour le seuil α est l'ensemble des éléments qui appartiennent à A avec un degré au moins égales à α. Formellement elle est donnée par : $A_\alpha = \{x \in \mathcal{X} | \mu_A(x) \geq \alpha\}$. A_α est un sous-ensemble ordinaire de fonction caractéris-

tique :

$$_{\mathcal{X}}A_{\alpha}(x) = \begin{cases} 1 & \text{si } \mu_A(x) \geq \alpha \\ 0 & \text{sinon} \end{cases}$$

Si A est un sous-ensemble flou d'un univers $\mathcal{X}$, de fonction d'appartenance μ_A, on a, selon le théorème de décomposition : $\forall x \in \mathcal{X}, \mu_A(x) = \sup_{\alpha \in]0,1]} \alpha._{\mathcal{X}}A_{\alpha}(x)$.

Il est à noter que si A est un ensemble classique, on aura alors $S(A) = N(A)$ et $H(A) = 1$ ou $H(A) = 0$ si $A = \phi$, on retrouvera ainsi les propriétés usuelles des ensembles classiques. Afin de pouvoir manipuler aisément les ensembles flous, Les opérateurs unaires et d'agrégation, tel que l'égalité, l'union, l'intersection, la déférence..., de la théorie des ensembles classiques ont été redéfinis et adaptés à la logique floue permettant des valeurs strictement entre 0 et 1(voir [410]). Les opérateurs logiques de Négation, conjonction et disjonction ont été, respectivement, redéfinis comme suite :

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x)$$
$$\mu_{A \cap B(x)} = min(\mu_A(x), \mu_B(x))$$
$$\mu_{A \cup B(x)} = max(\mu_A(x), \mu_B(x))$$

Le processus de conception et de construction d'une fonction d'appartenance est dit "fuzzification"[410]. Et consiste à déterminer le degré d'appartenance d'un élément donné à un ensemble flou. En réalité, il y a trois éléments cruciales à déterminer dans le processus de conception de la fonction d'appartenance à savoir : la forme, le nombre et les paramètres des fonctions d'appartenance.

La définition des fonctions d'appartenance est un point très délicat car la seule restriction qu'une fonction d'appartenance doit satisfaire est que ses valeurs doivent être dans l'intervalle [0,1]. De ce fait un ensemble flous peut être représenter par un nombre infini de fonctions d'appartenance[50]. Ces dernières peuvent âtre fonctionnelle ou numérique. Une fonction d'appartenance numérique est utile lorsque l'univers du discours est discret et peut être facilement implémentées par une table vérité. Dans le cas où la fonction d'appartenance est fonctionnellement définie, elle peut avoirs plusieurs formes dont les plus utilisées sont présentées par la figure 3.22. Le fait qu'un ensemble flou peut être décrit par un nombre infini de fonctions d'appartenance est en même temps une faiblesse et un avantage : l'unicité est sacrifiée au profit de la flexibilité, ce qui permet l'ajustement d'un modèle flou[50]. Les paramètres de la fonction d'appartenance à déterminer sont essentiellement son support et son noyau. Selon[20], Les méthodes de construction d'une fonction d'appartenance sont classées en quatre catégories : Méthodes automatiques, statistiques, psychométriques et géométriques.

— Les méthodes automatiques consistent, en premier lieu, a définir une première fonction d'appartenance mal ajustée voir aléatoire et de l'ajuster par la suite. Trois catégories de méthodes automatiques sont à distinguées, à savoir : Celles à base de réseaux de neurones, à base de classification et celles à base d'algorithmes génétique.

— Les méthodes statistiques décrivent les attributs linguistiques par des fonctions d'appartenance progressives. Les données de l'univers de référence $\mathcal{X}$ sont ordonnées par une structure d'ordre : $x_1 \geq x_2$ (signifie que $x_1 \in A$ est au moins aussi vrai que $x_2 \in A$). Et sont souvent exprimées par des histogrammes de fréquences ou par d'autres courbes de probabilités. Parmi les méthodes statistiques les plus utilisés, on cite à titre d'exemple, la méthode "oui-non", l'estimation d'ensemble et la méthode de Hisdal. D'un point de vue de l'effort à fournir pour l'obtention de ces fonctions d'appartenance, nous remarquons que dans les méthodes automatiques cet effort est minimal, puisque l'expert n'a pas à intervenir dans le processus d'acquisition.

— Les méthodes psychométriques, considérées comme des stratégies naturelles pour l'extraction des fonctions d'appartenance, consistent à interroger un ou plusieurs experts pour spécifier la fonction d'appartenance la plus appropriée au problème donnée. Comme il y a une infinité de fonction d'appartenance pour un problème, le choix est souvent limité à certaines fonctions prédéfinies comme par exemple, des simples fonction triangulaires avec un support donné par une intervalle fermée est un noyau constitué par singleton. Cela simplifie le choix ; l'expert n'aura qu'a choisir la valeur centrale et la pente de la courbe de chaque côté. Dans le cas où plusieurs experts sont consultés, une fonction consensuelle est construite au moyen d'agrégation.On compte plusieurs méthodes psychométriques pour l'obtention de fonctions d'appartenance ; on cite entre autres : Méthodes "Noyau-Support", méthode de Quantification structurelle et la méthode de "grille répertoire".

— Les méthodes géométriques nécessitent de connaître un nombre raisonnable de points appartenant à la fonction d'appartenance à construire sur un univers de référence $\mathcal{X}$ continu. À partir de ces points, il est possible de construire une fonction par interpolation tout en vérifiant certaines contraintes pour que la courbe obtenue corresponde à une fonction d'appartenance correcte.

La théorie des ensemble flous a attiré l'attention des chercheurs dans diverses disciplines. Depuis 1965, beaucoup d'efforts ont été consentis pour le développement de cette théorie et ses applications. En Datamining, l'approche floue constitue un moyen, efficace, de relier les mondes symbolique et numérique, notamment au travers des variables linguistiques, dont chaque concept est représenté par un ensemble flou. Elle vient combler la faiblesse en matière de description des connaissance humaine des techniques de discrétisation habituellement appliquées dans de nombreuses approches d'apprentissage telles que les arbres de décision. Dans certaines situations, la connaissance humaine correspond exactement à de tels discrétisation des attributs continus. Par exemple, la durée de connexion à partir d'une même adresse IP est divisée en deux intervalles par le seuil de 10.000 dans le schéma suivant : "Si un hôte reçoit plus de 10.000 demandes de connexion provenant de la même adresse IP en une seconde, cette adresse IP est considérée comme suspecte. Cependant, dans d'autres situations, la discrétisation en intervalles ne convient pas pour décrire la connaissance humaine. Par exemple, la connaissance : "Lorsque la durée d'une connexion est très courte, cette connexion est considérée comme une activité normale" ne peut être représenter adéquatement en utilisant la discrétisation du domaine de la durée de connexion en intervalles. En effet, le terme "très court" ne peut pas être représenté de manière appropriée par un intervalle. Si on choisit une mesure quantitative, une plage de valeurs ou un intervalle pour représenter une valeur normal, toute les valeurs qui n'appartiennent pas à l'intervalle seront considérées comme aberrantes au même degré sans prise en compte de leurs variations en distances par rapport à l'intervalle. Au même titre, toutes les valeurs appartenant à l'intervalle seront considérées comme normal au même degré de normalité. Ce problème est connue sous le nom de "frontière nette". Plusieurs algorithmes de détection de valeurs aberrantes supposent l'existence de cette "frontière nette" entre les valeurs normales et aberrantes. Cette hypothèse provoque une séparation brutale entre normalité et anomalie. Cependant la normalité est un concept vague. Une façon naturelle de caractériser un comportement normal est de lui définir un degré de normalité. Par conséquent une meilleure caractérisation de la frontière entre normal et anormal est nécessaire pour augmenter la précision de la détection des valeurs aberrantes. Il est, donc, nécessaire d'intégrer la logique floue dans le processus d'exploration de données à fin de traiter les questions d'incertitude. Des recherches récentes ont montré que la théorie des ensemble flous a été impliquée dans tous les aspects du datamining, y compris le nettoyage des données, pré-traitement, découverte de motifs, interprétation et évaluation des motifs. Une tendance nouvelle, consiste à intégrer la

théorie des ensembles flous dans le processus de sélection et de réduction des attributs afin d'éliminer les attributs redondants. Cette approche réduit, considérablement, la taille de l'ensemble de données d'origine tout en maintenant la même quantité d'information[244]. L'analyse floue des données peut être aborder de deux principales façons. La première consiste à étendre les méthodes d'analyse classique d'une manière assez générique au moyen de Fuzzification. La secondes, plus sophistiquée, est basée sur la projection des données dans un espace mathématique plus complexe, doté d'une métrique floue, et d'effectuer l'analyse dans cet espace[184].

Le raisonnement en logique floue, également appelé raisonnement approximatif, est le processus par lequel une conclusion éventuellement imprécise est déduite d'un ensemble de prémisses imprécises. Il est, dans sa grande partie, un raisonnement qualitatif plutôt que quantitatif[117]. Son intérêt réside dans le fait qu'il se base sur des règles floues exprimées en langage naturel en utilisant des variables linguistiques. Ces règles floues permettent de modéliser une connaissance experte, en reliant de façon non-linéaire des entrées floues avec des sortie "nettes" ou floues, et sont données sous la forme :

$$\textbf{Si } x \in A \textbf{ et } y \in B \textbf{ alors } z \in C$$

avec A, B et C sont des ensembles floues. Le caractère flou de la règle provient du fait que la prémisse(Condition), et éventuellement la conséquence(conclusion), sont définies par des concepts linguistiques, implémentés par des ensemble flous. La prémisse(condition) d'une règle est remplacée par une valeur de vérité issue de la fuzzification qui sera par la suite appliquée à un ensemble floue. Si la prémisse est constituée d'une conjonction de deux (ou plusieurs) conditions, le système flou prend la valeur de vérité minimale de ces conditions. Dans le cas d'une disjonction la plus grande valeur de vérité est considérée. Un système d'inférence floue est constitué de quatre modules(Fig. 3.23) :

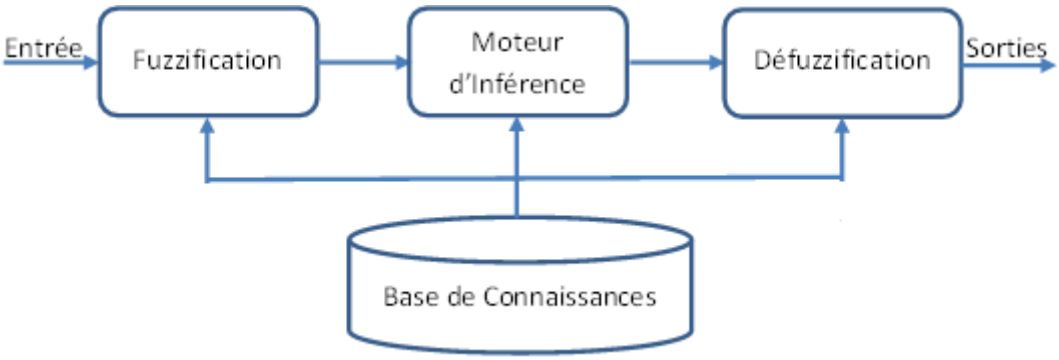


FIGURE 3.23 – Système à inférence floue.

— Le module de fuzzification définie les fonctions d'appartenance. Il transforme les valeurs numérique en degrés d'appartenance au différents sous-ensemble flous de la partition.

— L'unité de décision ou moteur d'inférence exploite un raisonnement approximatif pour déduire une décision à partir d'un fait observé de la base des règles. Dans son processus d'inférences, Le moteur d'inférence réalise les opérateurs flous en se basant sur plusieurs méthodes d'inférences directes et indirectes(Fig. 3.24). Les méthodes directes, regroupent la méthode de Mamdani[269], et celle de Sugeno[383] et sont les plus utilisées vu leur simplicité. Les méthodes indirectes ont un mécanisme de raisonnement complexe.

— Une base de connaissances constituée d'une base de données et une base de gèles floues. La base de données contient la définition des ensembles flous, les facteurs d'échelle pour

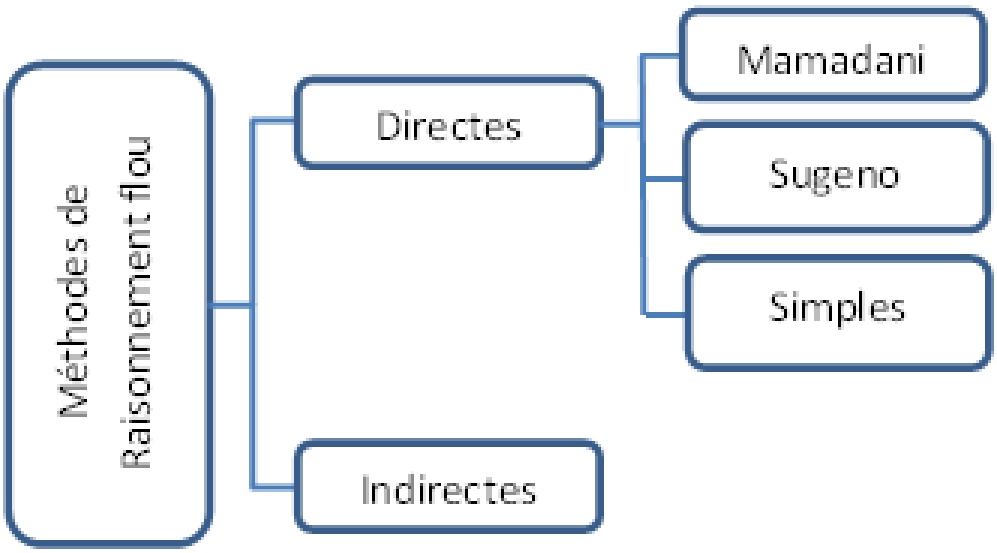


FIGURE 3.24 – Méthodes d'inférence floue.

la normalisation des ensembles de référence et la partition de l'espace flou d'entrée et de sortie.

— Le module de defuzzification permet d'inférer une valeur nette ou de prendre des décisions, à partir du résultat de l'agrégation des règles.

Un système d'inférence flou peut produire plusieurs sorties. Chaque sortie étant un ensemble de valeur possibles dont l'interprétation varie avec le type de règles.

La logique floue est appropriée pour le problème de détection d'intrusion pour deux raisons principales[62]. D'abord, beaucoup d'attributs quantitatifs, ordinales et catégoriques, sont impliqués dans la détection d'intrusion et peuvent, potentiellement, être considérées comme variables floues. Comme exemples des mesures ordinales ont cite : le temps d'utilisation du processeur et la durée d'une connexion. Le nombre des différents services TCP/UDP lancés par un même hôte source est un exemple d'une mesure catégorique linéaire. La seconde raison est le fait que la sécurité elle même comprend du flou, généralement, les approches de détection connues reposent sur un processus de classification des activités initiées par un utilisateur ou une application en deux catégories : normales et malveillantes en utilisant usuellement des mesures quantitatives. L'introduction de la logique floue à ces attributs rend la séparation brutal entre les classes d'activité réseaux plus lisses. Dans ce contexte des travaux forts intéressants ont combiné les mesures statistiques à des approches de classification en utilisant la logique floue. Par exemple Dickerson et al [108] classent des portions de données on se servant d'une variété de métriques puis il créent et appliquent des règles de logique flou sur ces portion de données afin de les classer en données normales ou non. Cette approche s'est avérée particulièrement efficace contre les attaques de types "port scan". Dans un travail ultérieur [109], ils ont utilisé des règles floues pour corréler des informations issues de plusieurs sondes. De son côté J. Luo[263] a ajouté la logique flou aux règles d'association et aux épisodes fréquents construites inductivement à partir des données. Il note que la logique floue offre des modèles plus abstraits et plus flexibles pour la détection. Et rajoute que la détection d'intrusion est une application naturelle pour la logique floue car, fréquemment, nous ne pouvons pas déterminer correctement si une connexion est normale ou malveillante, nous utilisons la logique floue pour quantifier le degré de malveillance d'une connexion au lieu de d'utiliser un seuil pour déterminer si une connexion est saine ou pas.

3.6.7 Technique d'immunologie

Le principal défi en détection d'intrusion est de pouvoir établir une nette discrimination entre comportement normal et comportement intrusif. A fin de lever cet défi, plusieurs approches et techniques ont été proposées. Initialement, acteurs en matière de securité informatique(chercheurs, et développeurs) en fait recours à des techniques, dite classiques, issues des statistique, de l'intelligence artificielles et de l'apprentissage automatique. Cependant, la croissance et la complexité des systèmes et réseaux informatiques nécessitent le développement d'outils de défense automatisés et adaptatifs. Des solutions prometteuses voient le jour avec l'informatique inspirée de la biologie, et, en particulier, l'approche immunologique.

Les systèmes immunitaires artificiels représentent un nouveau paradigme de calcul inspiré des principes de l'immunologie théorique. Il exploitent les propriétés des systèmes immunitaires biologiques. tel que : l'apprentissage, la mémorisation, l'auto-organisation, l'adaptation, la reconnaissance, la robustesse, l'involutivité. L'extraction de caractéristiques, la diversité, détection distribuée, La reconnaissance du soit, Protection dynamique... (voir [92] pour une description détaillée des propriétés clé des systèmes immunitaires). Et peuvent être définis comme étant des compositions de méthodologies intelligentes inspirées par les systèmes immunitaires naturels afin de résoudre des problèmes issus du monde réel[92]. Et selon De Castro et Timmis[69], un système ne peut être qualifié d'immunitaire artificiel que s'il comporte, au minimum, un modèle élémentaire d'un composant immunitaire(Cellule, molécule, organe), qu'il a été conçu en incorporant des idées inspirées de l'immunologie théorique ou expérimentale et qu'il est destiné à résoudre des problèmes. Dans ce contexte, Ils proposent un framework simplifiant le processus de correspondance entre les systèmes immunitaires naturels et les systèmes immunitaires artificiels[69](Fig. 3.25). Ce dernier est composé des trois éléments suivants :

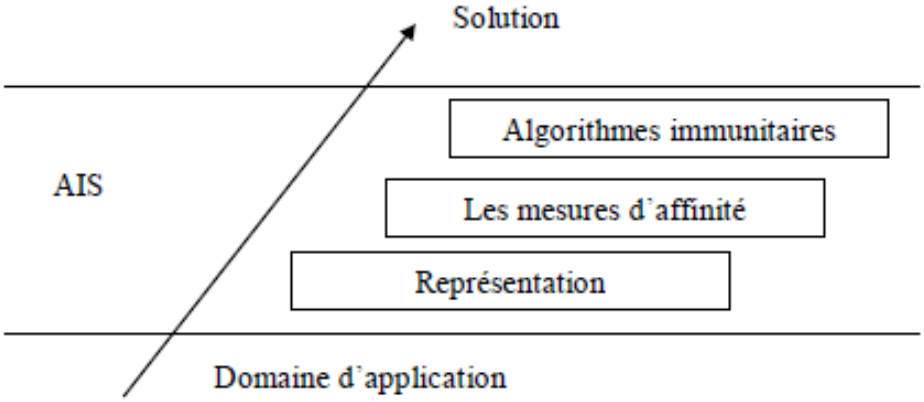


FIGURE 3.25 – Structure conceptuelle d'un système immunitaire artificiel.

1. Représentation des composants du système.
2. Un ensemble de mécanismes d'évaluation de l'interaction inter-composants et l'interaction de ces derniers avec leurs environnement(mesures d'affinité).
3. Un ensemble d'algorithmes immunitaires contrôlant l'évolution et la dynamique du système.

L'affinité peut être estimée via n'importe quelle mesure de distance entre chaînes ou vecteurs(Voir Tableau 3.2 pour quelques exemples de distance). Les mesures d'affinité quantifient les interactions entre les composants du système immunitaire et sont partiellement dépendantes des représentations adoptées. Plusieurs mécanismes ou algorithmes immunitaires artificiels ont

été proposés pour faire face à de nombreux problèmes issus de plusieurs domaines tel que sécurité des réseaux, la reconnaissance de caractères, l'alignement d'image, et l'alignement multiple de séquences. Chacun de ces algorithmes s'inspire d'un comportement particulier du système immunitaire naturel.

Distance	$d(Ab_i, Ag_j)$
Distance Euclidienne	$\sqrt{\sum_{i=1}^{L}(Ab_i - Ag_i)}$
Distance de Manhattan	$\sum_{i=1}^{L} \lvert Ab_i - Ag_i \rvert$
Distance de Hamming	$\sum_{i=1}^{L} \delta_i$ avec $\delta_i = 1$ *si* $Ab_i \neq Ag_i, 0$ sinon

TABLE 3.2 – Quelques mesures de distance.
Ab est un anticorps et Ag est un antigène avec
$Ab = (Ab_1, Ab_2, \ldots, Ab_L)$, et $Ag = (Ag_1, Ag_2, \ldots, Ag_L)$

Ainsi, on distingue trois types d'algorithmes :

1. L'algorithme de la sélection négative
2. L'algorithme de la sélection clonale
3. L'algorithme du réseau immunitaire

L'algorithme de sélection négative(Algorithme 13) traduit la capacité du système immunitaire biologique à différencier le Soi du non Soi. Il est appelé à générer un ensemble de cellules immunitaires, dites détecteurs, capables de reconnaître toute sorte de cellule faisant partie de l'espace du non Soi.

Globalement, un algorithme de sélection négative consiste, étant donné un ensemble de modèles de Soi à protéger S, à gérer un ensemble de détecteurs D qui ne doivent identifier aucun élément Soi. Puis à vérifier l'occurrence des motifs non Soi. Plusieurs alternatives de cet algorithme ont été proposées par la suite. Cependant les caractéristiques principales de la version originale sont maintenues en particulier son objectif qui consiste à couvrir l'espace de non Soi avec un ensemble approprié de détecteurs. Une de ces alternative, dite sélection positive proposée par Forrest et al [147], consiste à générer des détecteurs pour les éléments du Soi au lieu de ceux qui détectent des éléments du non Soi. Tout élément non Soi suspect est comparé à l'ensemble des détecteurs de Soi. Si aucune correspondance n'est enregistrée, l'élément est considéré comme non Soi.

L'algorithme de sélection clonale(Algorithme 14) est basée sur la théorie de la sélection clonale proposée il y a plus de cinquante année. Il présente une abstraction des mécanismes de mémorisation des systèmes immunitaires biologiques et traduit leur capacité à générer une réponse immunitaire capable d'éliminer les pathogènes. Bien qu'il ait une source d'inspiration biologique différente, il est très semblable aux algorithmes evolutionnaires, les algorithmes génétique en particulier, du fait que lui aussi fait évoluer une population initiale aléatoirement générée ver un optimum global. Les cellules immunitaires capables de reconnaître des antigènes se reproduisent, avec un mécanisme de clonage, proportionnellement à leur degré d'affinité avec les antigènes. Et subissent, éventuellement, des mutations leurs permettant d'améliorer leur degré d'affinité. La sélection clonale peut être utilisées pour les problème de recherche et d'optimisations.

Algorithme 13 : Algorithme de séléection négative

Entrées : S : Ensemble d'éléments du soi ;
M : Ensemble de Contrôle ;
$SeuilAff$: Seuil d'affinité ;
$NbrDetecteurs$: Nombre de détecteurs ;

Output : Étiquette Soi ou non Soi pour chaque $m \in M$;

```
1  début
2      Phase de génération de détecteurs;
3      D ← Φ;
4      tant que (|D| < NbrDetecteurs) faire
5          Générer, aléatoirement, un détecteur d;
6          si ( ∄i tel que affinite(S_i, d) > SeuilAff) alors
7              D ← D ∪ {d_i};
8          fin
9      fin
10     Phase de génération de vérification ;
11     pour chaque m ∈ M faire
12         si ( m concorde avec un détecteur d ∈ D) alors
13             Classer m comme non Soi;
14         sinon
15             Classer m comme Soi;
16     fin
17 fin
```

Algorithme 14 : Algorithme de séléection Clonale

```
1  début
2      Générer aléatoirement une population initiale de cellules immunitaires;
3      répéter
4          pour chaque (Antigène) faire
5              Calculer Son affinité avec chaque cellule immunitaire;
6              Sélectionner les cellules de plus grande affinité;
7              Cloner les cellules immunitaires sélectionnées proportionnellement à leur
                affinité avec l'antigène;
8              Muter chaque clone avec un taux de mutation inversement proportionnel à
                leur degré d'affinité
9          fin
10     jusqu'à Critère d'arrêt satisfait;
11 fin
```

L'algorithme du réseau immunitaire(algorithme 15) est inspiré de la théorie du réseau immunitaire introduite par Jerme[199] suggérant que le système immunitaire est un réseau autorégulé de molécules et de cellules qui se reconnaissent entre elles même en absence d'anti-

gènes. En absence d'antigènes, le réseau est dans un état stable, en revanche, lorsqu'un antigène se présente, l'ensemble des cellules immunitaires réagissent en parallèle. Cette interaction est assurée par des récepteurs spécialisés se trouvant sur la surface des anticorps et fut modélisées par plusieurs modèles.

Dans le modèle aiNet(artificial immune NETwork) proposé par Von Zuben et al [436] par exemple, le réseau immunitaire est initialisé, aléatoirement, par un petit nombre d'anticorps modélisés par une chaine avec le modèle Forme-Espace[313] qui permet la description quantitative des interactions des récepteurs et des antigènes. Dans une deuxième phase, l'affinité de chaque antigène avec chaque élément du réseaux est évaluée selon la distance Euclidienne(Tableau 3.2). Des anticorps à forte affinité sont clonés proportionnellement à leurs degré d'affinité. Les clones produits subissent une hyper-mutation somatique inversement proportionnelle à leurs affinité antiginéque. Un certain nombre des clones à forte affinité sont sélectionnés pour peupler la mémoire clonale. Les anticorps à faible affinité sont supprimés du réseau(suppression clonale) avec tous les anticorps dont l'affinité avec l'antigène est inférieure à un certain seuil. Puis des anticorps, aléatoirement produits, sont incorporés dans le réseau. Par la suit l'affinité entre chaque paire d'anticorps est calculée. Et chaque anticorps correspondant à une affinité inférieure à un certain seuil est supprimé.

Algorithme 15 : Algorithme de réseau immunitaire

Entrées : S : Ensemble de motifs à reconnaitre ;

Output : ensemble de détecteurs capable de classer des nouveau motifs;

```
1  début
2      C ← Φ;
3      Générer aléatoirement une population initiale d'anticorps N;
4      répéter
5          pour chaque (motif s_i ∈ S) faire
6              Calculer son affinité avec chaque anticorps de N;
7              Cloner, proportionnellement à leurs degré d'affinité, les anticorps ayant les
                plus hautes affinités ;
8              Effectuer une mutation de ces clones ;
9              Placer les clones à hautes affinité dans la mémoire clonale C;
10             Supprimer c_i ∈ C; i = 1, ..., |C| tel que affinite(c_i, s_i) < Seuil prédifini;
11             Calculer l'affinité entre chaque paire d'anticorps de C ;
12             Supprimer de C tout les c_i de faible affinité N ← N ∪ C;
13         fin
14         Calculer l'affinité entre chaque paire d'anticorps de N;
15         Supprimer de N tout n_i ayant faible affinité;
16         N ← N ∪ {un nombre aléatoire d'anticorps générés aléatoirement};
17     jusqu'à Critère d'arrêt satisfait;
18     retourner C ;
19 fin
```

Les systèmes immunitaires artificiels se distinguent des autres modèles bio-inspirés tel que les réseaux de neurones et les algorithmes génétiques par les faits suivants :

1. Contrairement aux réseaux de neurones qui ne peuvent avoir qu'une structure réseaux et aux algorithmes génétique qui manipulent des simples individus non connectés, les systèmes immunitaires artificiels peuvent avoir une structure d'un réseau reliant un ensemble d'éléments, comme ils peuvent être considérés comme un ensemble d'éléments fonctionnant ensemble sans notion de communication.
2. Contrairement aux réseaux de neurones les systèmes immunitaires artificiels sont capable à réagirent, grâce à la mutation, à des nouvelles situations jamais rencontrées.
3. Les systèmes immunitaires artificiels sont en mesure de reconnaître des modèles qui leurs sont fournis tous comme les réseaux de neurones et à évaluer des objectifs de la même manière que les algorithmes génétiques
4. A l'instar des Les systèmes immunitaires biologiques, les systèmes immunitaires artificiels évoluent en fonction du temps et sont aptes à se "souvenir", alors que les réseaux de neurones qui ne disposent que de la faculté d'apprentissage et les algorithmes génétique ne peuvent qu'évoluer.

En conséquence, les systèmes immunitaires artificiels sont plus flexibles et sont applicables dans une plus grande variété de domaines tel que l'apprentissage(Classification, clustering, robotique et contrôle), la sécurité informatique, et l'optimisation. Mais la sécurité informatique semble être le domaine le plus approprié et le plus naturel vue l'analogie évidente entre les systèmes immunitaires biologiques et les systèmes de sécurité informatique[412]. Cette analogie a été reconnue pour la première fois en 1987 lorsque Adelman a introduit le terme "virus informatique". Le lien entre immunologie et sécurité informatique à été mis en évidence en 1994 avec les travaux de Forrest et al [146] et Kephart[213]. Plus particulièrement, plusieurs recherches ont fait ressortir des ressemblances entre les systèmes de détection d'intrusion et les systèmes immunitaires biologiques. Ces ressemblances couvrent leurs fonctionnalités, leurs modes de détection et leurs environnements. Ainsi, après une étude comparative entre les systèmes immunitaires biologique et les systèmes de détection d'intrusion, Somayaji et al [374] sont parvenus à énumérer l'ensemble des propriétés des systèmes immunitaires biologiques pouvant servir comme principes de base dans la conception des systèmes de détection d'intrusion. Ces propriétés permettent aux systèmes immunitaires non seulement d'être efficaces en détection et élimination des intrus, mais augmentent également leurs niveau de tolérances aux pannes :

1. Ditribualité : Les lymphocytes sont capables de détecter localement toute infection. Aucune coordination centrale n'est nécessaire.
2. Multi-couches : Aucun mécanisme ne confère une sécurité complète, mais plusieurs couches de différents mécanismes sont combinés pour assurer un haut niveau de sécurité globale.
3. Diversité : Chaque individu d'une même population, dispose d'un système immunitaire unique. Ainsi les individus ne sont pas vulnérables aux mêmes infections aux même degré. Cela garantie une haute survabilité de la population.
4. Disposabilité : Aucune composante du système immunitaire n'est essentiel pour le fonctionnement du système, par conséquent toute cellule peut être remplacée par un mécanisme de reproduction. Sur le plan informatique, cette reproduction n'est possible qu'au niveau processus et il serait bénéfique de pouvoir la contrôlée.
5. Autonomie : Les systèmes immunitaires n'ont besoin d'aucune assistance ou maintenance extérieure. Chaque pathogènes est automatiquement détectée et éliminée et le cellules endommagées sont reproduites.
6. Adaptabilité :Le système immunitaire est capable d'apprendre à détecter de nouvelles pathogènes et est doté d'une mémoire immunitaire lui permettant de reconnaître toute pathogène précédemment rencontrée.

7. Pas de couches sûres : Toutes les cellule du corps, y compris celles du système immunitaire peuvent être attaquées par une pathogènes.
8. Couverture continuellement changeante : Le système immunitaire ne peut maintenir qu'un petit échantillon de ses détecteurs(lymphocytes). Cet échantillon est en constance évolution grâce au processus de reproduction des cellules mortes.
9. identité via comportement : Des fragment de protéine servent d'indicateurs de comportement à travers les quelles l'identité est vérifiée.
10. Détection d'anomalies : le système immunitaire est capable de détecter de nouvelles pathogènes jamais vue auparavant.
11. Détection imparfaite : Aucune pathogène n'est reconnue parfaitement ou exactement par un détecteurs pré-existant. Cela augmente la flexibilité avec laquelle le système immunitaire peut allouer ses ressources.

Partant de cette analogie, Plusieurs approches de détection d'intrusion intégrant et exploitant les concepts de l'immunologie théorique et expérimentale ont été proposées. Forrest et al [145, 146] sont considérés comme les premiers chercheurs a avoir proposer un modèle simulant le principe de discrimination entre Soi et non Soi d'un système immunitaire biologique pour la détection des altérations dans les fichiers et les séquences d'appels système. Ce modèle reposait essentiellement sur l'algorithme de sélection négative. En s'inspirant du travail de Forrest et al [146], Somayaji et al [373], ont proposé un système de détection basé hôte qui contrôle les processus privilégiés et consiste, dans une phase d'apprentissage, à collecter, sous forme de séquences de commandes sendmail, les informations nécessaires à définir le Soi. Puis, lors de la phase de test, il vérifie l'occurrence des nouvelles séquences ne faisant pas partie du Soi du programme en cours. Chaque séquence non reconnue est considérées comme une erreur. Une anomalie est déclenchée si le nombre des ces erreurs atteint un seuil prédéfini.

Le même principe a été repris par Hofmeyr[179] en introduisant quelque améliorations. Les séquences d'appels système sont représentées dans des fenêtres d'appels système qui sont confrontées à des modèles de comportements normaux. La similarité entre deux séquences est calculée en utilisant la distance de Hamming. Toute déviation du comportement normal est considérée comme erreur et si le nombre des erreurs dépasse un seuil prédéfini une alerte est générée.

En 2000, Hofmeyr et al [180] on proposé LISYS(Lightweight Immune SYStem), un des plus anciens systèmes de détection d'intrusion. En réalité LISYS est une implémentation de ARTIS(ARTificial Immune System) qui est un frameworke modélisant la plus part des processus et propriétés d'un système immunitaire biologique dont le rôle consiste à spécifier les éléments d'un système distribué adaptatif sans faire référence à aucune application spécifique. Ses éléments générique doivent être particularisés selon les caractéristiques de l'application[141]. Ainsi LISYS est un système de détection reposant sur des concepts immunologiques destinés à fonctionner dans un environnement distribué et se compose d'un ensemble de détecteurs qui sont analogues à des lymphocytes. Ces détecteurs sont continuellement comparés aux paquets. Toute correspondance entre un détecteur et un événement est interprétée comme une anomalie. ARTIS a été, aussi, utilisé par Harmer[171] pour la détection de virus et par Williams et al[411] pour la détection des intrusions.

De son côté, Dasgupta[93] a proposé une approche de détection à temps réel, à base de multi-agents immunitaires. Ces agents, pouvant apprendre et s'adapter à leurs environnement, errent autour des machines(nœuds et routeurs) à la recherches d'éventuels anomalies, défauts , ou abus d'utilisations. En cas de détection d'un disfoncionnement, ces agents sont en mesure d'effectuer les actions appropriés selon la police de sécurité installée. Comparé à d'autres systèmes de détection à base d'agents, ce système se distingue par le fait qu'il comporte simultanément une

surveillance multi-niveaux, un mécanisme de détection et de réponse hiérarchique, comme il est en mesure de détecter les attaques connues et inconnues.

Kim et Bentley[218] ont proposé une approche de détection d'intrusion à base d'un modèle immunitaire évolutionnaire à trois étapes : La première étape concerne l'évolution de la librairies des gènes, la deuxième présente la sélection négative et la troisième la sélection clonale. Les deux premières étapes sont exécutées par un système de détection primaire et la troisième par un système secondaire. Au lieu d'être généré aléatoirement, les détecteurs immunitaires sont crées en sélectionnant et réarrangent les gènes utiles. Les gènes des détecteurs performants sont maintenus et ajouté à la librairies alors que ceux des détecteurs défaillants sont éliminés. Lors de la sélection clonale, diverse intrusion sont détectées avec un nombre limité de détecteurs, et la mémoire clonale est construite.

Dans [18], les auteurs remettent en cause le point de vue classique reposant sur le concept de Soi et non Soi et proposent de le remplacer par des idées issues de la théorie de danger[275] qui stipulent que les systèmes immunitaires ne reposent pas uniquement sur la discrimination entre Soi et Non-soi mais réagissent à des menaces sur la base de la corrélation entre divers signaux(dangers). Et selon leurs avis, cette théorie, objet de débats entre immunologistes, n'a jamais été appliquée à la détection d'intrusion au paravent, est la clé qui ouvrira le véritable potentiel des système immunitaires artificiels permettant de construire des systèmes de détection d'intrusion, viables, capables d'évoluer. L'idée d'intégrer les concepts de la théorie de danger dans la détection d'intrusion a été, par la suite, adoptée par plusieurs chercheurs tel que Mark Vella et al [402], Azuan Ahmad et al [16], Al-Dhubhani[23], S. Vasanthi et al [400].

Récemment Z. Yabin[419] proposa un modèle de détection d'intrusion, à base de l'immunité artificielle, constitué de trois modules : Module de génération du Soi, Module de génération de détecteurs et un module de génération de détecteurs mémoire(Fig 3.26).

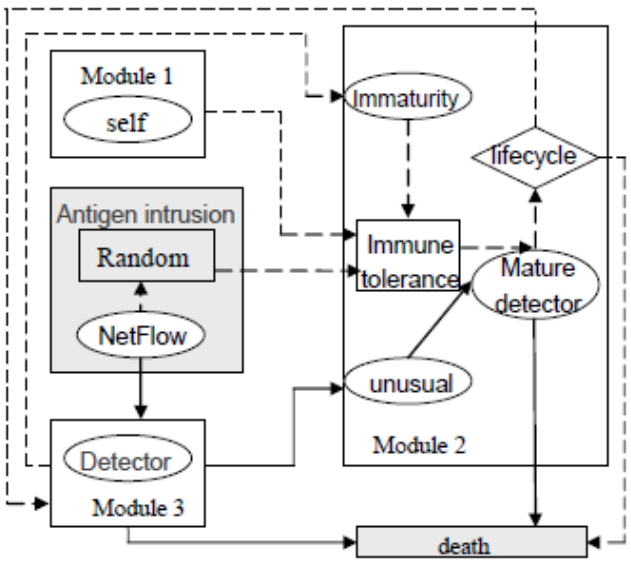


FIGURE 3.26 – Un modèle de détection d'intrusion à base d'immunité artificielle[419].

Se système comprend essentiellement deux processus. Un pour la détection des antigènes, représenté par des lignes continues dans la figure 3.26, et l'autre, représenté par des lignes pointillées, pour la détection des anticorps représentés.

De leur côté I. Dutt at al [119] ont proposé une méthodologie de détection d'intrusion basée sur un système immunitaire artificiel qui imite le système immunitaire biologique. Cette

méthodologie consiste à placer au niveau de chaque hôte un système de détection d'intrusion à deux couches. La première couche, considérée comme la première ligne défense, correspond au sous-système inné du système immunitaire naturel surveille les fichiers sur l'hôte. Les fichier exécutables(.exe,.bat) ou de type anonyme sont dirigés vers la deuxième ligne de défense (deuxième couche) en changeant leurs extensions afin de minimiser leur vulnérabilité à se multiplier ou résider dans le disque dur. Une fois la deuxième couche, correspondant au sous-système adaptatif du système immunitaire naturel, alertée elle invoque le module d'activation des cellules **B** afin de détecter la vulnérabilité des fichiers qu'elle a reçu. Si après analyse, le fichier s'avère malicieux il sera supprimer par le module d'activation des cellules **T** imitant les cellules **T** biologiques. Le système maintient à jour une table, dite Table B-cellule, dans la quelle il stock des informations (nom original, extension, temps d'arrivée...) sur tout fichier.

Les mécanismes du système immunitaire ont été tout récemment impliqués dans la détection des menaces ciblant l'environnement IoT(Internet of Things). Voir à ce sujet [78, 256, 257]. Une autre méthode biologique, basée sur la séquence de l'A.D.N a été proposée par Yu et al[424].

3.6.8 Les essaims intelligents

Récemment, Une famille de méthodes, dites bio-inspirées, ont fait leur apparition dans une variété de domaines allant de l'ingénierie, de l'informatique, l'économie, la médecine et les sciences sociales et également dans la détection d'intrusion. Parmi ces approches ont compte la technique des essaims intelligents, initialement introduite par Beni et Wang[48]. Cette technique cherche de l'inspiration dans le comportement des essaims d'insectes, de poissons, d'oiseaux et d'autres animaux[163]. Dans ces essaims, chaque individu semble avoir une intelligence limitée, mais une fois entré en interaction sociale avec d'autres individus de son groupe et avec son environnement, il arrive à accomplir des tâches complexes tel que la détermination du plus court chemin vers une source de nourriture, l'organisation de leurs nids, la synchronisation de leurs mouvements et le voyage, à haute vitesse, en une seule entité cohérente... Il est à noter que l'accomplissement de telles tâches se fait sans aucune présence d'une autorité centralisée via des interaction(communications) directes ou indirectes quand les individus rodent dans leurs espace de recherche. Ainsi, ces agents peuvent être utilisés à fin de trouver des règles de classification pour la détection d'abus d'utilisation, pour découvrir les clusters pour la détection d'anomalie et pour tracer les intrus. En effet les caractéristiques d'auto-organisation et de la distributivité sont très appréciables en détection d'intrusion. Il serait très intéressant de pouvoir diviser le problème difficile de détection en un ensemble de problèmes plus simples et d'assigner chaque sous-problème à un agent. Cette potentialité rend les systèmes de détection d'intrusion autonomes, hautement adaptatifs, parallèles, rentables comme elle les dote de la capacité d'auto s'organiser. Les deux principale catégories de méthodes de cette tendance ayant été les plus utilisées en détection d'intrusion sont : Les techniques d'optimisation de colonies de fourmis (ACO-Ant Colony Optimisation) introduites par Colorni et al [85] en 1991 et les techniques d'optimisation par essaims particulières proposées en 1995, par Eberhart et Kennedy[211, 120].

Dans le contexte de l'application des techniques d'optimisation de colonie de fourmis dans la détection d'intrusion, Fenet et al[135] ont proposé une architecture utilisant la métaphore des colonies de fourmis pour localiser la source d'une attaque. Cette architecture est constituée de trois sortes d'agents mobiles statiques :

1. Le serveur de phéromone installé sur chaque hôte qui est chargé, entre autre, de diffuser un message d'alerte(sous forme de phéromone) en cas d'intrusion.
2. L'agent observateur(guetteur), Composante principale du sous-système de détection, surveille les processus et les connexions réseaux du hôte où il est installé.

3. Les agents lymphocytes parcourent aléatoirement le réseau à la recherche des traces du phéromone. A la découverte de traces de phéromone, ils convergent vers la machine menacée et prennent les mesures défensives appropriées. Les lymphocytes constituent le module d'alerte.

Ainsi, l'ensemble de l'architecture constitue un système de détection d'intrusion et de réponse entièrement distribué. De sont côté Foukia[144], en adoptant une méthodologie similaire, a proposé un système de détection d'intrusion constitué de deux composantes. Un mécanisme de détection constitué d'un système immunitaire artificiel et un autre pour la réponse aux attaques basé sur le paradigme d'ACO.

Dans cette architecture, chaque nœud exécute une plateforme d'agents mobiles accueillant les agents de détection et ceux de réponse. Agents de détection et de réponse sillonnent, au hasard, le réseau. Les agents de détection calculent l'index de suspicion en fonction du statut local des hôtes. Si ce dernier dépasse un seuil prédéfini, l'agent de détection génère et diffuse une quantité appropriée du phéromopne. A la découverte des traces du phéromone, les agents de réponse remontent le chemin jusqu'à la source où ils déclenchent une réponse à l'attaque. Ramos et al[335] ont appliqué l'algorithme de clustering à base de colonies de fourmis, initialement proposé par Deneubourg et al [104] et amélioré par Lumer et Faieta [261] en le dotant d'une capacité de mémoriser les derniers items transportés et leurs positions. Les performances obtenues étaient compatibles à celles obtenues avec les SVMs, les arbres de décision et la programmation génétique linéaire. Cet algorithme de clustering s'est avéré plus adapté au problème de détection d'intrusion vu sa capacité de traiter de nouvelles classes, de s'auto-organisé et de fonctionner en mode en ligne. Et il a été ré-adopté dans les travaux de Feng et al[137, 138, 139] et de Tsang2005 et al [390, 391] qui ont tenté de corriger ces lacunes.

Abadi et Jalali[1] ont proposé une approche algorithmique, basé sur la technique d'optimisation par colonie de fourmis, dans laquelle toutes les attaques possibles ont été représentées par un graphe dans lequel chaque chemin complet reliant un nœud à un nœud cible correspond à un scénario d'attaque. La minimisation de ce graphe désigne l'ensemble minimal d'exploits qui devraient être éliminer pour assurer qu'aucun scénario d'attaque se soit réalisable. Chaque exploit est associé à une quantité de phéromone, sur un chemin, indiquant l'opportunité de l'inclure dans une solution d'une fourmis. Dans une première étape, l'algorithme défini les paramètres et initialise les traces de phéromone. Puis, à la base du graphe généré, un certain nombre de fourmis construisent itérativement, un ensemble d'exploits critiques, initialement vide, en ajoutant, à chaque itération, un exploit jusqu'à ce tous les scénarios d'une attaque soient couverts. A chaque itération, chaque fourmi choisit, selon une probabilité, un exploit à la base de la quantité de phéromone qui lui est associée. Les exploits critiques redondants doivent être éliminés. En suite, la meilleure solution de l'itération courante est améliorée avec une heuristique de recherche locale puis, les quantités ou concentrations du phéromone des chemins sont mis à jour en utilisant une règle de mise à jour globale. Les itérations de l'algorithme sont exécutées jusqu'à ce qu'une condition d'arrêt soit satisfaite(par exemple un nombre maximal d'itérations atteint). L'efficacité de ce système semble dépendre fortement de la précision des résultats de l'analyse des vulnérabilités. Néanmoins, dans des scénarios de la vie réelle, les vulnérabilités ne sont pas toutes connues à l'avance. De plus, il est prévu de manière réaliste, que les graphes générés peuvent être larges et complexes.

Plus récemment, Aghdam et al [10] on proposé une approche de sélection d'attributs les plus pertinents pour la détection d'intrusion en utilisant l'optimisation par colonie de fourmis. La méthode proposée, est qualifiée par ses auteurs, d'être simple à implémenter et d'une faible complexité de calcul. Les tests effectués ont montré que cette approche offre , avec un nombre réduit d'attributs, une plus grande précision de détection et de faible taux de fausses alertes. Dans le même contexte, Mehmod et al ont utilisé la même technique, sur l'ensemble de don-

nées KDD'99[91] afin d'extraire un sous-ensemble de données réduit et optimal qui a été validé avec la technique des SVM. Les résultats ont montré une amélioration significative. Il est à noter qu'il existe, dans la littérature, d'autres approches hybrides ayant proposé de combiner le modèle de clustering à base de colonies de fourmis avec d'autres techniques d'apprentissage machine et du soft computing incluant les automates cellulaires[21], K-means[289], Les carte auto-organisatrices[284], C-means[207], les systèmes de règles floues[353] et les SVMs[322], l'entropie flue relative[336]...

Les techniques d'optimisation par essaims(PSO) particulières ont montré de bonnes performances dans la résolution des problèmes numériques et ont été utilisées dans la détection d'intrusion pour l'apprentissage des règles de classification, essentiellement, dans le cadre de la détection d'anomalie combinées à d'autres techniques d'apprentissage automatique, notamment, avec différentes variantes de réseaux de neurones, les SVMs et l'algorithme K-Means. Kolias et al [225] (Michailidis et al [283] furent les premiers à avoir fusionner PSO et réseaux de neurones pour construire un système de détection d'intrusion plus performant. Ce dernier a été implémenté sous java est consiste, dans une phase d'apprentissage, à apprendre , récursivement, les poids synaptiques du réseau de neurones à l'aide d'un algorithme à base de PSO. En effet chaque particule du PSO correspond à un poids synaptique. Dans la phase de test, un réseaux de neurones, effectuant la tâche de classification, est alimenté avec les poids synaptiques optimaux, ainsi appris. Dans [254, 255] les auteurs proposent d'entrainer des réseaux de neurones d'ondelettes(WNN-Wavelet Neural Network)[430] à base de Quantum Particle Swarm Optimization(QPSO)[420] et la Modified Quantum Particle Swarm Optimization (MQPSO), deux variantes du PSO. A fin d'empêcher la PSO de converger vers un minimum local, d'accroitre la diversité des populations et d'élargir la porté de recherche, Tian et al. [386] ont augmenté leur système issu de l'hybridation PSO-Réseau de neurones par une étape supplémentaire exécutant un algorithme de mutation évolutionnaire. Cette même logique d'hybridation à été adoptée par Qiu et al [328] afin de remédier aux imperfection des systèmes de détection à base de réseaux de neurones à rétro-propagation classiques, notamment en ce qui concerne le taux de détection et la vitesse de convergence. P. Pawar et al. [307] ont continué dans la direction d'optimisation des performances de leur système par la sélection des paramètres d'entrées appropriés à l'aide du PSO. D'autre chercheurs, voulant profiter de la bonne capacité d'apprentissage des SVMs ainsi que leur pouvoir de généralisation dans le cas des données à haute dimension et/ou bruitées en suggérer de construire des systèmes de détection hybrides à base de combinaisons PSO-SVM. Parmi les travaux qui ont été conduits dans cette logique on cite : [212, 265, 351, 404]. Par ailleurs, d'autres chercheurs tel que Saxena et al[352], Xiao et al. [417], Li et al. [248], Ke-Wei Wang et al [405], se sont intéressés à la simplicité et la bonne recherche local de l'algorithme K-Means[267].

La technique d'optimisation par les colonies d'abeilles, proposée en 2005 par Karaboga [209], semble, elle aussi, avoir attiré l'intention de plusieurs chercheurs issue d'une large variété de domaines et en détection d'anomalie en particulier. En effet plusieurs chercheurs ont proposé des approches de détection d'anomalie à base de cette technique hybridée avec d'autre méthodes d'apprentissage, à l'instar de la technique d'optimisation par les colonie de fourmis. Comme exemple de cette orientation on cite les travaux de : Aldwairi et [24], Bae et al [37], Goodarzi et [158], Gupta et al [165], et Qian et al [327].

3.6.9 Autres Méthodes

Plusieurs d'autres méthodes ont été suggérées pour l'extraction de connaissances liées à la sécurité informatique à partir des données relatives au trafic réseau. Dans cette section, nous présentons quelques techniques ayant été appliquées à la détection d'intrusion :

1. Les réseaux de Pétri colorés[197], qui se définissent comme un langage graphique pour la conception, la spécification, la simulation et la vérification des systèmes, particulièrement, bien adapté aux systèmes dans lesquels communication, synchronisation et partage de ressources sont importantes, ont été appliqués avec succès à la détection d'abus d'utilisations. Ils offrent une superbe intelligence de détection d'activités malveillantes[113] et permettent de généraliser les signatures d'attaques, connues, de la base de connaissances établie par des experts et de présenter graphiquement les attaques[361]. Comme ils facilitent, aux administrateurs système, l'ajout de nouvelles signatures. Comme exemples des systèmes de détection d'intrusion à base des réseaux de Pétri ont cite :
   (a) IDIOT(Intrusion Detection In Our Time)[89, 232] développé au laboratoire COAST à Purdue University,
   (b) POSTAT (Partial Order State Transition Analysis Technique) développé par Yuan Ho à l'université de Idho à Moscou [178].
   (c) La frameWork de détection d'intrusion proposée par Z. Gou et al [160].
2. Les techniques issues du traitement de signal ont été appliquées, avec succès, à la détection d'anomalie en raison de leur capacité à détecter de nouvelles intrusions non encore spécifiées ainsi que celles que les systèmes à base de signatures n'arrivent pas à détecter et à leur capacité de transformer les données. Les techniques d'ondelettes[408, 416], d'analyse spectrale[380], d'analyse en composantes principales[203] et l'estimation du maximum d'entropie[195] en sont des exemples de ces techniques ayant été les plus utilisées dans le domaine de la détection d'intrusion. Parmi les chercheurs qui ont utilisé cette catégorie de techniques on cite. Andrysiak et al [29], Barford et al[43], Berezinski et al [49], Bouzida et al [55], Chabathula[72] Deepthi et al[101], Gu et al [164], Huang et al[183], Ji et al [200], Lu et al [260], Luo et al [264], Mozzaquatro et al [291], Rawat et al [337], Santiago et al [350], Vasan et al [399],

3. Plusieurs travaux ont proposé d'impliquer les modèles markoviens [382] dans le processus de détection d'intrusion en prétendant que la les systèmes de détection d'intrusion à base de chaînes de Markov sont en mesure de détecter les attaques complexes reparties sur plusieurs étapes exécutées sur une période prolongées et où des actions spécifiques peuvent être interchangeables. Le modèle markovien est apprit à partir de l'historique du comportement normal du système. Le comportement est analysé pour déterminer la probabilité du processus. Une faible probabilité indiquerait une anomalie. Malheureusement, il s'est avéré que la construction des chaînes markoviennes est un processus complexe et consomme beaucoup de temps Toutefois leur utilisation peut être plus faisable dans un environnement off-line. Parmi les chercheurs ayant appliqué les chaines et les modèles morkoviens dans la détection d'intrusion ont cite : Warrender et al [406], Debar et al [97], Nong Ye [422], N. Gornitz[159], Sharma2015 et al [364], Floreano et al [142].

4. Les systèmes multi-agents(SMA)[30, 58] se sont avérés très adaptés pour implémenter, efficacement, la sécurité des réseaux informatiques de la nouvelle génération qui sont caractérisés par leurs grandes vitesses de transmission, de leurs énormes trafic et par leurs nombreux et complexes services. Les systèmes de détection d'intrusion représentent le mécanisme typique de la sécurité informatique ayant, le plus, profité de l'architecture distribuées des systèmes multi-agents et de leurs propriétés. L'autonomie, l'adaptabilité, la coordination, la réactivité, la distribution et la communication sont les principales propriétés ayant fait des SMA une approche très appropriée au problème de la détection d'intrusion. Plusieurs efforts ont été consentis pour intégrer le paradigme d'agents mobiles dans la détection d'intrusion. Parmi les premiers systèmes proposés dans ce contexte on cite

JAM(Java Agent for Meta Learning)[381], IDA(Intrusion Detetion Agent System)[32], Micael[329], AAFID(Autonomous Agents For Intrusion Detection)[356], General Multi-agents system framwork for intrusion detection[176], [286], SPIDeR-MAN (Synergistic and Perceptual Intrusion Detection with Reinforcement in a Multi-Agent Neural Network), CIDS (Cougaar-based IDS)[95] et Multi-Agent System comprising intelligent agents[295]. [193] présente une revue et une analyse comparative de certains systèmes de détection d'intrusion à base d'agents qui ont été proposés entre 2011 et 2015.

5. Récemment, un grand intérêt a été donné au méthodes d'**ensemble learning** ayant largement influencer le développement du data mining et de l'apprentissage automatique[359]. Globalement, ces méthodes consistent à entrainer plusieurs modèles et de combiner ou agréger leurs résultats. Hansen et Salamon[170] avaient montré que la combinaison de plusieurs réseaux de neurones peut améliorer considérablement la précision des décision. De son côté Shapire[363] affirme qu'il est théoriquement possible d'obtenir une haute précision, par la combinaison d'un ensemble de classificateurs faibles. Dietterich [110], voie que les méthodes de l'ensemble learning sont plus performants qu'un simple classificateurs pour trois raisons. Premièrement, le plus souvent les données d'apprentissage ne fournissent pas assez d'informations pour sélectionner une unique hypothèse précise et l'apprentissage d'un faible classificateur pourrait être imparfait et troisièmement l'espace des hypothèses peut ne pas contenir la vraie fonction cible. Selon Polikar[319] la construction d'un système d'ensemble learning se fait en trois étapes. La première consiste à sélectionner les données assurant une diversité. La deuxième étape consiste à entrainer un ensemble de classificateurs à base de plusieurs algorithmes concurrents tel que le boosting[362], bagging[60] et les forêts aléatoires[61]. Les résultats obtenus par les différents classificateurs sont combinés via des stratégies appropriées tel que le vote majoritaire, la moyennisation pondérée ou simple ou via une combinaison linéaire. En prenant en compte le fait qu'un système de détection d'intrusion ne peut couvrir qu'un nombre limité de différents types de données et ne peut identifier qu'un nombre limité d'attaques[230], l'ensemble learning parait un bon moyen d'améliorer la précision et les performance des systèmes de détection. En effet, Mukkamala et al, dans [294], ont montré qu'un ensemble composé de différents types de réseaux de neurones, de machine à support de vecteurs (SVM) et de la régression multivariée par spline adaptative(MARS-Multivariate Adaptive Regression Splines) combiné avec les techniques de Bagging surpasse les algorithmes traditionnels.

   De leur côté, Chebrolu et al[76] ont combiné les réseaux Bayésien(RB) avec les arbres de régression (CART) dans un ensemble utilisant la technique du Bagging. Et Dans [3, 311] les auteurs ont combiné les SVM, les arbres de décisions(DT). Les résultats ainsi obtenues, dans les deux cas dépassait celles obtenues lorsque chaque technique étaient exécutées séparément. La figure 3.27 présente les méthodes de datamining ayant été combinées ensemble dans d'autres recherches. Les lignes plus épais relient les méthodes les plus souvent combinées ensemble.

   Globalement L'ensemble learning a été utilisé en détection d'intrusion comme suit : En premier lieu chaque sous-espace d'attributs est utilisé, indépendamment, pour détecter des attaques. Puis les évidences sont combinées à fin de produire une décision finale(Fig. 3.28).

   Selon Balon-Perin ce paradigme a été introduit en détection d'intrusion, pour la première fois en 2003. La majorité des travaux portant sur l'utilisation de l'ensemble learning dans la détection d'intrusion ont été effectué entre 2004 et 2005. puis il y a eu un regain d'intérêt pour cette approche[39]. [4, 19, 26, 75, 107, 114, 133, 150, 173, 274, 324, 326,

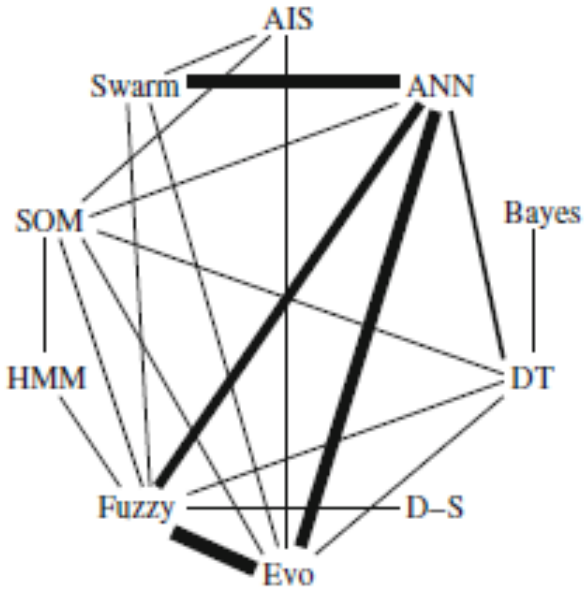


FIGURE 3.27 – méthodes ayant été utilisées ensemble[235]

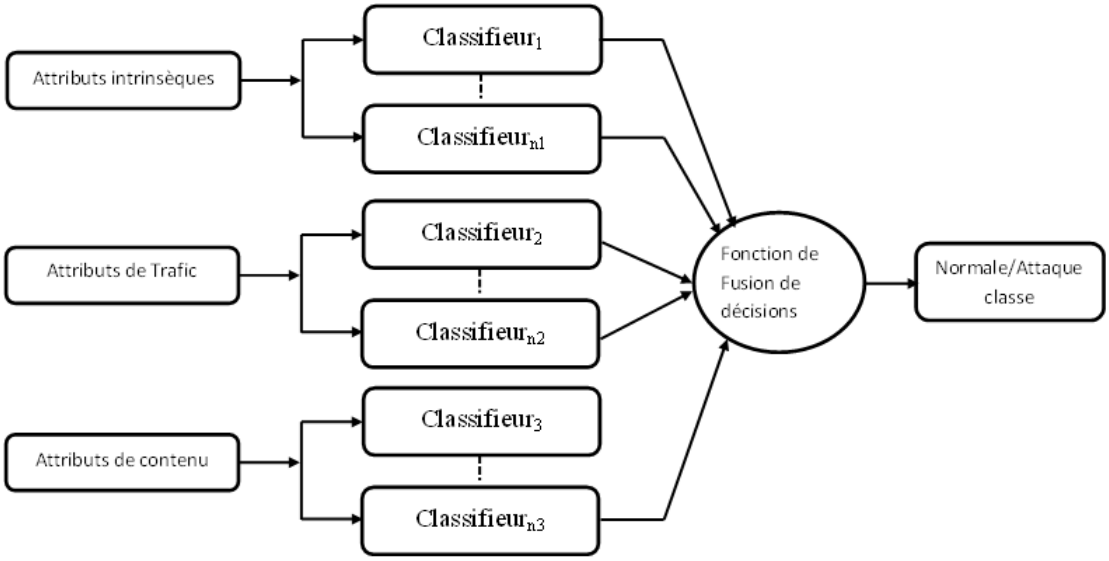


FIGURE 3.28 – Ensemble learning pour la détection d'intrusion

338, 343, 384, 375, 426, 433, 434] se comptent parmi les travaux ayant porté sur cette problématique.

3.7 Conclusion

Dans cette section nous avons fourni une vue d'ensemble des diverses méthodes de datamining qui ont été utilisées ou proposées pour la détection d'intrusion dans les réseaux. La plupart des techniques présentées ici semblent être bonne pour détecter au moins un type d'activité malveillante, toutefois aucune de ces techniques ne prétend être en mesure de détecter tout type d'intrusion.

A l'issue de cette étude, nous pouvons conclure, d'une part, que les caractéristiques souhaitées d'un système de détection d'intrusion dépendent, à la fois, de la méthodologie et de l'approche de modélisation utilisées dans sa conception et sa réalisation. Et, d'autre part, l'utilisation d'une seule méthode de datamining pour construire des systèmes de détection d'intrusion n'assure pas de bonnes performances. Différents types de détecteurs sont nécessaires pour détecter les différents types d'attaques. La combinaison et la corrélation des résultats de plusieurs techniques ou détecteurs semble être l'approche la plus appropriée pour améliorer la performance et l'efficacité des systèmes de détection d'intrusion. En effet cette approche s'est avérée plus performante que les SVM et les réseaux de neurones.

Deuxième partie

Contributions

CHAPITRE 4

DESCRIPTION ET PRÉ-TRAITEMENT DES DONNÉES

4.1 Introduction

Le processus de construction d'un système de détection englobe trois aspects à savoir : la collection de données, le pré-traitement et la détection d'intrusion. Dans le cadre de collection de données, nous avons, dans un premier temps, réaliser des simulations d'attaques sur des réseaux locaux, et des snifer à fin de collecter des jeux de données dans le but d'entrainer et tester nos différents systèmes de détection d'intrusion. Mais nous nous somme vite convaincus de la nécessiter d'un benchmark standard nous permettant de comparer nos classifieurs avec d'autres issus de la littérature spécialisée. Nos recherches dans ce contexte, nous ont conduit aux conclusion suivantes :

— "The information Exploration Shootout (IES)" fut le premier ensemble de données ayant été largement utilisé dans le contexte de détection d'anomalie. Crée en 1996, l'IES été composé de cinq fichiers de 50 Mo chacun. Parmi lesquels un ne contenait que du trafic normal et les quatre autres contenait quatre types différents d'attaques. Les données ont été collectés à partir d'une passerelle sur le réseau de MITRE Corporation [1].

— Cette base de données n'est plus disponible et a cédé sa place à l'ensemble de données de DARPA [2] qui est devenu, non seulement, plus populaire grâce à la compétition international de découverte de connaissances(KDD'99) tenue en 1999 à l'occasion de la cinquième conférence internationale d'ACM [3] sur la découverte de la connaissance et du datamining, mais aussi devenu de facto un benchmark pour l'évaluation des performances des systèmes de détection d'intrusion. Selon A. Özgür et al. [325], cet brenchmark a été utilisé dans pas moins de 149 articles issus de 65 journaux durant la période 2010-2015(Fig. 4.1).

1. La **MITRE Corporation** est une société à but non lucratif qui exploite des centres de recherche et de développement financés par le gouvernement fédéral des états unis.

2. Defense Advanced Research Projects Agency est une agence du département de la Défense des États-Unis chargée de la recherche et de développement des nouvelles technologies destinées à un usage militaire.

3. Association for Computing Machinery

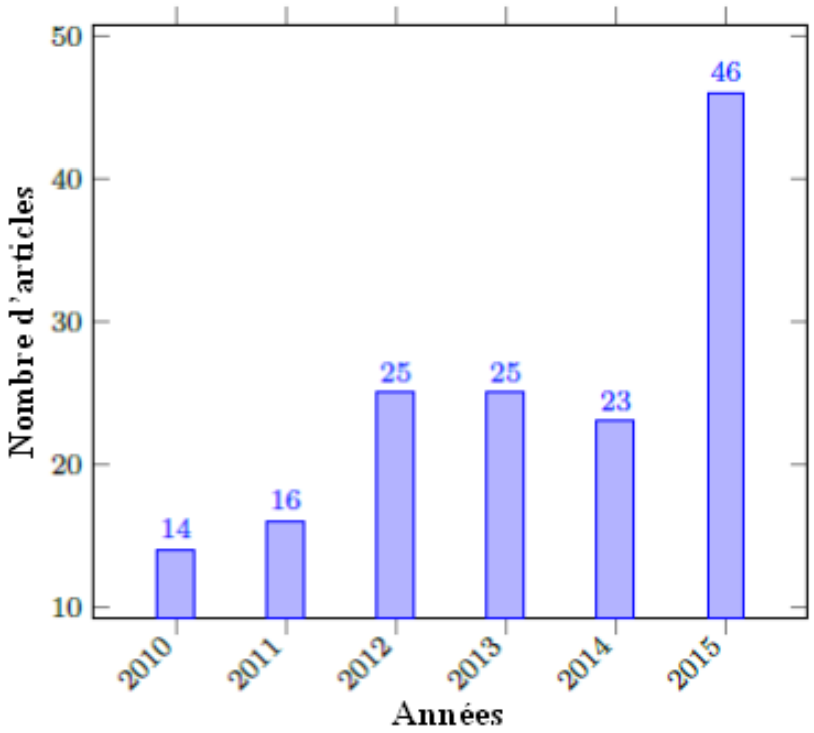


FIGURE 4.1 – Utilisation du KDD'99 entre 2010 et 2015([325]).

Pour ces raisons, nous avons adopté cet ensemble de données comme brenchmark pour évaluer nos différents classifieurs développés dans le cadre de cette thèse.

4.2 L'ensemble de données de DARPA

En fait, cet ensemble de données est le fruit d'une simulation d'attaques sur un réseau de l'armée de l'aire américaine durant neuf semaines. Chaque connexion est étiquetée en tant que connexion normale ou attaque. Chaque attaque est identifiée par une étiquettes spécifique. Et peut appartenir à l'une des quatre catégories suivantes (table 4.1) :

TABLE 4.1 – Catégories d'attaques dans KDD'99.

Categorie	Attaque
Dos	back, land, neptune, pod, smurf, teardrop
Probe	ipsweep, nmap, portsweep, satan
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warez-client, warezmaster
U2R	buffer_overflow, loadmodule,perl, ps, rootkit

— **DOS(Denial of Service Attack)** : Regroupe les attaques visant à porter atteinte à la disponibilité des services en saturant les ressources de la machine cible. Certaines attaques de cette catégorie exploitent les bugs des applications et d'autres les vulnérabilités dûs aux mauvaises implémentations ou aux faiblesses des protocoles.

— **U2R (User To Root Attack)** : Dans cette catégorie d'attaques, L'attaquant, qui est un utilisateur interne légitime, tente d'acquérir les droits d'un utilisateur root(Administrateur) à partir d'un simple compte utilisateur par l'exploitation des vulnérabilités. Cette catégorie d'attaques exploitent généralement la saturation des Buffers causée par les erreurs de programmation.

— **R2L(Remote to Local Attack)** : Regroupe les attaques visant à contourner ou usurper, à partir d'une machine distante, les paramètres d'authentification d'une machine cible en exploitant ses vulnérabilités afin d'acquérir un accès illégal. La plupart de ces attaques sont issues de la sociale ingénierie.

— **Surveillance et écoute(Probing Attack)** : Ensemble d'attaques dont l'objectif consiste à collecter les informations sur une ou plusieurs machines. Les attaques de cette catégorie utilisent des techniques de balayage des ports afin de connaître les services offerts par le système de la machine cible, la topologie du réseau, les protections déployées, etc. Il existe plusieurs types attaques probes : certaines abusent les utilisateurs légitimes et d'autres utilisent les techniques d'ingénierie pour collecter les informations. Ces attaques sont les plus perpétuées car elle ne nécessité qu'une expertise technique minime.

Cet ensemble de données est constitué de 4 898 431 enregistrements dont 972 781 sont issus d'un trafic normal et 3 925 650 correspondent à 22 attaques. Un sous-ensemble ne contenant que 10% des données pris au hasard de cet ensemble de données a été créé et est, généralement, utilisé dans le cadre d'un processus d'apprentissage. Les tableaux 4.2 et 4.3 présentent les distributions des différentes classes dans, respectivement, l'ensemble des données d'apprentissage et celui des données de test.

Classe	Taille	%
Dos	391 458	79.24
Normal	97 278	19.69
Prob	4 107	0.83
R2L	1 126	0.23
U2R	52	0.01
Total	494 021	100

TABLE 4.2 – Répartition des classe dans l'ensemble d'apprentissage

Classe	Taille	%
Dos	3 883 370	79.278
Normal	972 781	19.859
Prob	41 102	0.839
R2L	1 126	0.023
U2R	52	0.001
Total	4 898 431	100

TABLE 4.3 – Répartition des classe dans l'ensemble de test

Chaque enregistrement décrit une connexion à l'aide de 41 attributs dont 7 qualitatifs et les 34 autres sont quantitatifs. Ces attributs sont regroupés en trois classes décrites comme suit :

— Les Attributs de base(table 4.4) : Cette catégorie regroupe neuf attributs intrinsèques décrivant les données au niveau paquet. Ces attributs sont directement obtenus à partir des paquets capturés et sont utilisés pour calculer d'autres attributs et peuvent être utilisés pour détecter plusieurs attaques. L'attribut **Src_bytes**, par exemple, représentant la quantité de données envoyées de la source vers la destination, sert à détecter les attaques par $buffer\_overflow$.

— Les Attributs de contenu(table 4.5) : Ces attributs sont relatifs aux contenus des paquets d'une connexion et permettent de révéler certaines actions malveillantes tels que les accès aux fichiers systèmes, tentatives d'accès non autorisés, etc. Ils sont particulièrement utiles pour la détection des attaques U2R (User to Root) et R2L (Remote to Local access). La définition de ce type d'attributs nécessite des connaissances à priori sur les différentes stratégies d'attaques.

— Attributs temporels (time-based features)(Table 4.6) : Cette catégorie d'attribut est constituée de deux sortes d'attributs : Des attributs relatifs aux connexion ayant le même hôte de destination que la connexion courante durant les deux dernières minutes et des attributs décrivant des connexions ayant le même service que la connexion courante durant les deux dernières minutes.

N	**Attribut**	**Description**	**Type**
1	duration	durée de la connexion	continu
2	protocol_type	type du protocole	discret
3	service	service réseau (destination)	discret
4	flag	statut de la connexion	continu
5	src_bytes	nb de données (en octets) de la source vers la destination	continu
6	dst_bytes	nb de données (en octets) de la destination vers la source	continu
7	land	1 si la connexion est de/vers le même hôte/port ; 0 sinon	continu
8	wrong_fragment	nb de fragments erronés	continu
9	urgent	nb de paquets urgents	continu

TABLE 4.4 – Attributs de base d'une connexion TCP individuelles

N	**Attribut**	**Description**	**Type**
10	hot	nb d'indicateurs hot	continu
11	num_failed_logins	nb d'essais login ratés	continu
12	logged_in	1 si succès du login ; 0 sinon	discret
13	num_compromised	nb de conditions de compromis	continu
14	root_shell	1 si la racine shell est obtenue ; 0 sinon	discret
15	su_attempted	1 s'il y a tentative de la commande racine su ; 0 sinon	discret
16	num_root	nb d'accès à la racine	continu
17	num_file_creations	nb de créations d'opérations de fichiers	continu
18	num_shells	nb de shell prompts	continu
19	num_access_files	nb opérations sur les fichiers de contrôle d'accès	continu
20	num_outbound_cmds	nb de commandes outbound dans une session ftp	continu
21	1is_host_login	1 si le login appartient à la liste hot ; 0 sion	discret
22	is_guest_login	1 si le login est login invité ; 0 sinon	discret

TABLE 4.5 – Attributs de contenu

N	**Attribut**	**Description**	**Type**
23	count	nb de connexion pour le même hôte	continu
24	srv_count	nb de connexion pour le même service	continu
25	serror_rate	% de connexion pour le même hôte ayant l'erreur SYN	continu
26	srv_serror_rate	% de connexion pour le même service ayant l'erreur SYN	continu
27	rerror_rate	% de connexion pour le même hôte ayant l'erreur REJ	continu
28	srv_rerror_rate	% de connexion pour le même service ayant l'erreur REJ	continu
29	same_srv_rate	% de connexion pour le même hôte utilisant le même service	continu
30	diff_srv_rate	% de connexion pour le même hôte utilisant différents services	continu
31	srv_diff_host_rate	% de connexion pour le même service utilisant différents hôtes	continu
32	dst_host_count	nb de connexion pour le même hôte	continu
33	dst_host_srv_count	nb de connexion pour le même hôte utilisant le même service	continu
34	dst_host_same_srv_rate	% de connexion pour le même hôte utilisant le même service	continu
35	dst_host_diff_srv_rate	% de connexion pour le même hôte utilisant différents services	continu
36	dst_host_same_src_port_rate	% de connexion pour le même hôte ayant le port src	continu
37	dst_host_srv_diff_host_rate	% de connexion pour le même hôte et le même service utilisant différents hôtes	continu
38	dst_host_serror_rate	% de connexion pour le même hôte ayant l'erreur SYN	continu
39	dst_host_srv_serror_rate	% de connexion pour le même hôte et le même service ayant l'erreur SYN	continu
40	dst_host_rerror_rate	% de connexion pour le même hôte ayant l'erreur REJ	continu
41	dst_host_srv_rerror_rate	% de connexion pour le même hôte et le même service ayant l'erreur REJ	continu

Table 4.6 – Liste des Attributs calculés durant deux secondes

Par ailleurs nous avons constater que cet ensemble de données souffre, essentiellement, des lacunes suivantes :

— Contient un nombre important d'enregistrement redondants. En présence de telle redondance, tout algorithme de classification se trouve biaisé vers les attaques fréquentes et donnera moins d'importance aux attaques rares tel que R2L et U2R qui sont, généralement, les plus nuisibles.

— Contient des attributs non pertinents qui ajoutent, simplement, du bruit aux données et affectent, négativement, la précision de tout modèle de classification.

A fin de lever ces lacunes, nous avons procéder à une étape de pré-traitement.

4.3 Pré-traitement des données

Le pré-traitement des données constitue l'une des plus cruciales étape dans le processus d'extraction de connaissances à partir de données (ECD). Il traite de la préparation et de la transformation du jeu de données initial. et englobe les étapes de consolidation, de nettoyage, de transformation, et réduction de données(Fig. 4.2).

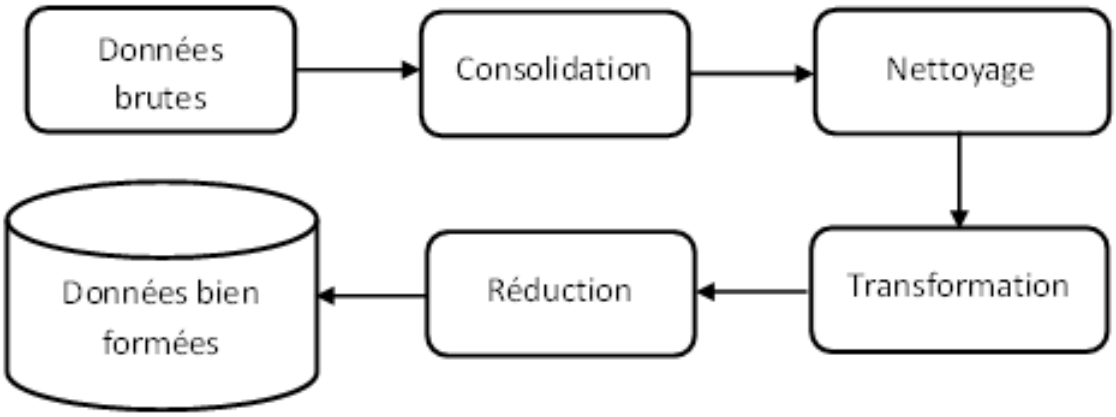


FIGURE 4.2 – Préparation de données

— La consolidation des données consiste à recueillir, sélection et intégrer les données provenant de plusieurs sources en un seul ensemble de données afin d'obtenir un rapport structuré, plus facile à consulter que l'information brute, mais avec le moins de perte d'information possible.

— Le nettoyage consiste à assigner les valeurs manquantes, à supprimer les données bruitées ou non pertinentes.

— La transformation des données consiste à mettre les données dans un format approprié pour la fouille et implique, globalement les tâches de normalisation, standardisation et de lissage. La normalisation consiste à mettre en échelle des attributs numérique afin qu'ils tombent dans une intervalle précise $[\alpha, \beta]$ via l'équation :

$$v_i{}^* = \frac{v_i - min_A}{max_A - min_A} * (\beta - \alpha) + \alpha \tag{4.1}$$

Où min_A et max_A dénotent respectivement le minimum et le maximum des valeurs prisent par l'attribut A, v_i une de ces valeurs et v_i^* la nouvelle valeurs obtenue après transformation de v_i. Les bornes α et β sont généralement identifiées à 0 et 1.

La standardisation, signifiant étymologiquement "centrer réduire", consiste à transformer les données de sorte qu'elles aient une moyenne nulle et un écart type égale à 1 via l'équation :

$$v_i{}^* = \frac{v_i - \overline{A}}{\sigma_A} \tag{4.2}$$

Où $\overline{A}$ et σ_A signifient respectivement la moyenne et l'écart type de l'attribut A. v_i dénote une valeur prise par l'attribut A et v_i^* sa transformée.

— Le concept de réduction de données englobe, à la fois, la réduction du volume et la réduction des dimension(nombre d'attributs).

Dans la majorité des cas, le pré-traitement doit préparer des informations globales sur les données pour les étapes qui suivent telles la tendance centrale des données (moyenne, médiane, mode), le maximum et le minimum, le rang, les quartiles, la variance, etc. Plusieurs techniques de visualisation des données telles que les courbes, les diagrammes, les graphes, etc, peuvent aider à la sélection et le nettoyage des données.

Dans le cadre de cette thèse, l'étape de pré-traitement est justifiée par le fait que l'ensemble de données utilisé pour entrainer et tester nos différents classifieurs comporte des données redondantes et des attributs non pertinents. Ces lacunes peuvent être la principale source :

— De confusion pour nos algorithmes de classification,

— De connaissances inexactes ou inefficaces,

— De temps de calcule non raisonnable.

Ainsi, dans le contexte de pré-traitement nous avons effectuer la tâches suivantes :

— Dans un premier temps nous avons structuré notre ensemble de données sous forme matricielle sous laquelle nous avons effectué les opération suivantes :

— Élimination des enregistrement doubles dont la majorité sont issue du trafic normal, ce qui nous a permit de réduire, considérablement, la taille du jeux de données comme le montre les tableaux 4.7, 4.8.

Classe	Taille	%
Dos	54 572	37.484
Normal	87 832	60.330
Prob	2 131	1.464
R2L	999	0.686
U2R	52	0.036
Total	145 586	100

TABLE 4.7 – Elimination de la redondance des données d'apprentissage

Classe	Taille	%
Dos	247 267	23.002
Normal	812 814	75.611
Prob	13 860	1.289
R2L	999	0.093
U2R	52	0.005
Total	1 074 992	100

TABLE 4.8 – Elimination de la redondance des données de test

— Élimination des attributs constants. Dans ce contexte, nous avons éliminé les attributs 20 et 21(voir tableaux 4.5) que nous avons jugés non pertinents.

— Après l'étape de nettoyage des données nous avons procédé à la conversion des attributs symbolique en numérique. en remplaçant chaque valeur d'un attribut par son rang dans la liste des valeurs possible pour cet attribut. Les attributs concernés par cette conversion sont : *protocol_type*, *service* et *flag*(Voir tableau 4.4). Les différentes conversions sont représentées dans les tableaux 4.9, 4.10, et 4.11.

Protocol	Valeur
icmp	1
tcp	2
udp	3

TABLE 4.9 – Conversion de l'attribut "*protocol_type*"

Flag	Valeur	Flag	Valeur
OTH	1	S1	7
REJ	2	S2	8
RSTO	3	S3	9
RSTOS0	4	SF	10
RSTR	5	SH	11
S0	6		

TABLE 4.10 – Conversion de l'attribut "*flag*"

Service	Valeur Numérique	Service	Val. Numérique	Service	Val. Numérique
aol	1	imap4	26	rje	51
auth	2	IRC	27	shell	52
bgp	3	iso_tsap	28	smtp	53
courier	4	klogin	29	sql_net	54
csnet_ns	5	kshell	30	ssh	55
ctf	6	ldap	31	sunrpc	56
daytime	7	link	32	supdup	57
discard	8	login	33	systat	58
domain	9	mtp	34	telnet	59
domain_u	10	name	35	tftp_u	60
echo	11	netbios_dgm	36	tim_i	61
eco_i	12	netbios_ns	37	time	62
ecr_i	13	netbios_ssn	38	urh_i	63
efs	14	netstat	39	urp_i	64
exec	15	nnsp	40	uucp	65
finger	16	nntp	41	uucp_path	66
ftp	17	ntp_u	42	vmnet	67
ftp_data	18	other	43	whois	68
gopher	19	pm_dump	44	X11	69
harvest	20	pop_2	45	Z39_50	70
hostnames	21	pop_3	46		
http	22	printer	47		
http_2784	23	private	48		
http_443	24	red_i	49		
http_8001	25	remote_job	50		

TABLE 4.11 – Conversion de l'attribut "*service*"

— La dernière étape de la préparation des données consiste à normaliser l'ensemble de données en appliquant à chaque valeur x_i, prise par un attribut, l'équation 4.1 en prenant $\alpha = 0$ et $\beta = 1$. Cette étape est nécessaire pour éviter que certains attributs ayant des plages numériques larges deviennent plus influents que ceux ayant de petites plages numériques.

Par ailleurs nous avons utilisé quatre mesures de performance pour évaluer et comparer nos classifieurs :

— T.N. (True negatif) Mesures le pourcentages des connections normales dans l'ensemble des données de test classées comme normales.

— F.N. (False negatif)Mesures le pourcentages des connections intrusives dans l'ensemble des données de test classées comme normales.

— F.P. (False Positif) Mesures le pourcentages des connections normales dans l'ensemble des données de test classées comme intrusions.

— T.P. (True Positif) Mesures le pourcentages des connections intrusives dans l'ensemble des données de test classées comme Intrusion.

Dans les deux chapitre suivants, nous présenterons les deux travaux, effectués dans le cadre de cette thèse, ayant aboutis à des publications internationales.

CHAPITRE 5

UNE MÉTHODE DE CLASSIFICATION À BASE DE COPULES POUR LA DÉTECTION D'INTRUSION

5.1 Introduction

Soit un ensemble de d attributs $(a_1, a_2, \ldots, a_d)$ caractérisant un espace vectoriel E. Soit aussi $(x_1, x_2, \ldots, x_n)$ un ensemble E utilisé comme ensemble d'apprentissage sur m classes $(\omega_1, \omega_2, \ldots, \omega_m)$ qui en fait sont des sous-ensembles disjoints de E. Pour éviter le recours à certaines lois de probabilité prédéterminées systématiquement sur attributs, nous construisons un modèle de classification basé sur les copules estimant la vraie lois des attributs et leurs dépendance. Par la suite, nous assignons chaque éléments de E à sa plus probable classe ω_i; $i \in \{1, \ldots, m\}$. Le dit élément sera bien classé s'il vérifie un certain critère probabiliste optimale.

En classification déterministe, ce modèle construit sur l'ensemble E, une relation d'équivalence $\mathcal{R} \subset E \times E$ où $E/\mathcal{R}$ est une partition E. Alors que dans une classification non déterministe, les classes sont construites en utilisant une distribution de probabilité avec un risque adapté. L'affectation de k éléments à m classes nécessite seulement k étapes dans le cas d'une classification déterministe où chaque étape exécute m test simples. Cependant dans une classification non déterministe on compte m^k possibilité pour distribuer les k éléments sur les m classes. Chaque possibilité nécessite k étapes d'affectation. Chaque élément est affecté à la classe ω_j via une probabilité conditionnelle $f(x \mid j)$ qui peut être estimée en utilisant des données d'entrainement. En fait, nous cherchons la classe la plus probable(estimation de maximum de vraisemblance) solution de :

$$k = \arg\max_j \left(f(x \mid j)\right)$$

où $f(x \mid j)$ dénote la fonction de densité de probabilité conditionnelle pour qu'un élément x soit membre de la classe ω_j.

A fin de réduire la complexité du probléme, nous affectons les éléments à leurs classes respectives de façon déterministe. De cette façon nous aurons besoin que de $k * m$ testes. Dans ce qui suit, nous dénotons la fonction de densité de probabilité conditionnelle par $f^j(x)$ au lieu de $f(x \mid j)$. Plusieurs algorithmes et modèles ont été proposés pour l'estimation de cette fonction de densité de probabilité conditionnelle : kernel-density estimator [385], k-nearest-neighbours (KNN) method [282], Learning Vector Quantisation (LVQ) [223], Support Vector Machines (SVM)[292].

Dans le cadre de ce travail, nous utilisons la copule empirique comme alternative pour décrire la structure de dépendance dans un classificateur probabiliste supervisé.

L'ensemble E est identifié à un espace vectoriel $\mathbf{R}^d$ sur le champs $\mathbf{R}$ et nous utilisons la loi du phénomène considéré sur E qui peut être bien estimée si l'échantillon d'apprentissage est de taille importante. Ainsi la fonction de densité conditionnelle $f^j(x)$ est estimée selon l'algorithme 16.

Algorithme 16 : Estimation de densité de probabilité conditionnelle

Entrées :

$\{X_i\}_{i=1}^n$ un échantillon à partir de la $d-$ distribution F de densité f.

$\Omega = \{\omega_1, \cdots, \omega_m\}$ m classes d'apprentissage.

Output : Densités conjointes $f^j(x)$; $i = 1, \cdots, m$;

début
  pour chaque $j \in \{1, \cdots, m\}$ **faire**
    Transformer les observations X_i^j en $U_i^j = F_{ni}^j(X_i)$ où F_{ni}^j ;
    Estimée la $i^{\text{iéme}}$ distribution limité à la classe ω_j
    et X_i^j dénote une observation de la classe ω_j ;
    Estimer la densité marginale f_i^j pour classe ω_j;
    Estimer la densité conjointe des données transformées de la classe ω_j. La densité est notée c^j et est équivalente à la densité de la copule.;
    Estimer la densité conjointe des données originales de la classe ω_j avec :

$$f^j(x) = c^j\left(F_1(x_1), \ldots, F_d(x_d)\right) \prod_{i=1}^{d} f_i^j(x_i)$$

  fin
fin

Cette approche nous permet, d'une part, d'atténuer la malédiction de la dimensionnalité et nous permet, d' autre part, de traiter les données même dans le cas où la variance n'existe. Comme elle tiens compte des relations non-linéaires qui peuvent exister entre les attributs. Une nouvelle observation x sera affectée à la classe ω_r solution de :

$$r = \arg\max_j f^j(\mathbf{x}) \tag{5.1}$$

Avant d'entamer, en détaille le modèle de classification développé, il nous semble important de présenter une courte introduction aux copules.

5.2 Les Copules

Les copules jouent un rôle très important dans plusieurs domaines des statistiques et de l'apprentissage automatique comme outil d'études des mesures sans échelle de dépendance et comme point de départ dans la construction des familles de distributions bidirectionnelles en particulier dans les applications où les dépendances sont non-linéaires.

La meilleure définitions des copules est celle donnée par le fameux théorème de Sklar-[370, 276] qui précise le lien entre la fonction de copule et la fonction de distribution conjointe.

Theorem 1. *(Théorème de Sklar)*

Soit F une fonction de distribution à d-dimensions sur des variables aléatoires réelles avec des fonctions marginales $f_1, f_2, ..., f_d$, il existe alors une fonction de copule C tel que $\forall x \in \bar{\boldsymbol{R}}^d$ on a :

$$F(x_1, \ldots, x_d) = C(f_1(x_1), \ldots, f_d(x_d)) \tag{5.2}$$

où $\bar{\boldsymbol{R}}$ dénote l'axe réel étendu $[-\infty, \infty]$ et $C : [0,1]^p \to [0,1]$.

La distribution de copule peut, également, être définie comme une distribution conjointe des variables aléatoires uniformément distribuées :

$$C(u_1, \ldots, u_p) = P(U_1 \leq u_1, \ldots, U_p \leq u_p) \tag{5.3}$$

où $U_i \sim U(0,1)$ for $i = 1, \ldots, p$.

Il est à noté que si $f_1(x_1), \ldots, f_d(x_d)$ dans (5.2) sont toutes continues alors, C est unique. Sinon , C est uniquement déterminée sur $Ran(f_1) \times Ran(f_2) \times \cdots \times Ran(f_d)$, où Ran signifie le Rang.

Inversement, si C est une d-copule et $f_1, \ldots, f_d$ sont des fonctions de distribution alors la fonction F définie ci-haut est une fonction de distribution à d dimensions ayant comme marginales $f_1, \ldots, f_d$. (La démonstration est donnée dans [370]).

A partir du théorème de sklar, on voie que pour des fonctions de distribution multivariées continues , les marginales uni-variées et la structure de dépendance multivariée peuvent être séparées et que la structure de dépendance peut être représentée par une copule. Une importante conséquence du théorème 1 est que la $d-$densité conjointe F et les densités marginales $f_1, f_2, \ldots, f_d$ sont aussi reliées

$$f(x_1, \ldots, x_d) = c(F_1(x_1), \ldots, F_d(x_d)) \prod_{i=1}^{d} f_i(x_i) \tag{5.4}$$

où c dénote la densité de la copule C. L'équation (5.4) montre que le produit des densités marginales et la copule "construisent" une $d-$densité conjointe. L'unique fonction de copule, liée à la distribution multivariée F ayant $f_i; 1 \leq i \leq d$ comme marginales continues, est déterminée par

$$C(u_1, \ldots, u_d) = F(F_i^{-1}(u_1), \ldots, F_i^{-1}(u_d)) \tag{5.5}$$

où $F_i^{-1}(s) = \{t \mid F_i(t) \geq s\}$ dénote le pseudo-inverse de la marginales uni-variée F_i ; $i = 1, \cdots, d$. La copule est essentiellement un moyen pour transformer la variable aléatoire $(X_1, \cdots, X_d)$ en une autre variable aléatoire $(U_1, \cdots, U_d) = (F_1(X_1), \cdots, F_d(X_d))$ ayant des marginales uniformes sur $[0,1]$ est préserve les dépendances entre ses composantes. Sans l'hypothése de continuité, l'équation 5.5 doit être utilisée avec précaution (voir [298] ou [271]).

5.3 Estimation de la fonction de Copule

Estimer la fonction copule revient en premier lieu, à estimer séparément les marginales et la loi conjointe. D'ailleurs, certaines de ces fonctions peuvent être entièrement connues. Selon les prétentions faites, quelques quantités doivent être estimées de façon paramétriquement, ou semi ou même non-paramétrique. Dans le dernier cas, nous devons choisir entre la méthodologie habituelle d'employer "les contre-parties empiriques" et les méthodes de lissage bien connues dans les statistiques tel que : Noyaux, ondelettes, polynômes orthogonaux, les voisins les plus proches, etc.

Ici, nous ferrons recours à une méthode non paramétrique pour estimer la copule car d'une part, l'estimation non paramétrique des copules ne nécessite la définition d'aucun paramètre ou seuil. De plus, elle offre un meilleur pouvoir de généralisation et peut fournir les informations initiales requises pour un modèle paramétrique. Et d'autre part, les distributions marginales et la distribution conjointe sont directement observables alors que la copule est une structure de dépendance cachée, ceci rend la tâche de proposer un modèle paramétrique de copule approprié non triviale.

En fait, l'estimation non paramétrique des copules remonte à Deheuvels [102], qui a proposé la copule dite empirique défini par :

$$C_n(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left(F_{n,1}(X_{i1}) \leq u_1, \ldots, F_{n,d}(X_{i,d}) \leq u_d\right) \tag{5.6}$$

où $F_{n,i}$ sont des fonctions de distribution empiriques données par :

$$F_{n,j}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(X_{i,j} \leq x) \tag{5.7}$$

avec $j = 1, \ldots, d$ et $\mathbf{u} \in [0,1]^d$.

soit R_i le rank de X_i sur l'échantillon $X_1, \ldots, X_n$. Notez que C_n est une fonction de $R_1, R_2, \ldots, R_n$, car $F_{n,j}(X_i) = \dfrac{R_{i,j}}{n} \quad i = 1, \ldots, n$, à savoir :

$$C_n(\mathbf{u}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left(\frac{R_{i,1}}{n} \leq u_1, \ldots, \frac{R_{i,d}}{n} \leq u_d\right). \tag{5.8}$$

A partir de cette représentation, on peut considérer $C_n(\mathbf{u})$ comme une distribution multivariée avec des marginales uniformes prenant des valeurs dans l'ensemble $\left[\frac{1}{n}, \frac{2}{n}, \ldots, 1\right]$. ainsi sa densité donnée par :

$$c_n(\mathbf{u}) = \frac{\partial C(u_1, ..., u_d)}{\partial u_1, \cdots, \partial u_d}. \tag{5.9}$$

peut être estimée par une fonction noyau standard :

$$\hat{c}_n(\mathbf{u}) = \frac{1}{n}\sum_{j=1}^{n}\prod_{i=1}^{d} h_i^{-1} K\left(\frac{u_i - U_{ji}}{h_i^{-1}}\right) \tag{5.10}$$

où U_i est la transformée de la donnée originale :$U_i = F_{n,i}^{j}(X_i)$.

La fonction noyau uni-variée $K(u)$ peut être n'importe quel fonction qui satisfait les conditions suivantes :

(a) $K(x) \geq 0$ et $\int_{\mathbf{R}} K(x)dx = 1$.

(b) $\int_{\mathbf{R}} xK(x)dx = 0$ (Symétrique sur l'origine).

(c) admet un second moment finie e.g. $\int_{\mathbf{R}} x^2 K(x)dx < \infty$.

Nous avons à choisir la fonction noyau K ainsi que son paramètre de lissage ou sa bande passante h. En fait, le choix de K est un problème de moindre importance, différentes fonctions produisant de bons résultats peuvent être utilisées(Voir la table 5.1 pour quelque exemples).

Nous utiliserons, comme noyau, une fonction gaussienne donnée par :

$$K(v) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{v}{2}). \tag{5.11}$$

TABLE 5.1 – Quelques fonctions noyaux.

	Kernel	**K(x)**
1	uniform	$\frac{1}{2}\mathbf{1}_{(\lvert x\rvert\leq 1)}$
2	Epanechnikov	$\frac{3}{4}(1-x^2)\mathbf{1}_{(\lvert x\rvert\leq 1)}$
3	Gaussian	$\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x}{2}\right)$
4	triangular	$(1-\lvert x\rvert)\mathbf{1}_{(\lvert x\rvert\leq 1)}$
5	Triweight	$\frac{35}{32}(1-x^2)^3\mathbf{1}_{(\lvert x\rvert\leq 1)}$
6	Tricube	$\frac{70}{81}(1-x^3)^3\mathbf{1}_{(\lvert x\rvert\leq 1)}$
7	Biweight(Quartic)	$\frac{15}{16}(1-x^2)^2\mathbf{1}_{(\lvert x\rvert\leq 1)}$
8	Cosine	$\frac{\pi}{4}cos(\frac{\pi}{2}x)\mathbf{1}_{(\lvert x\rvert\leq 1)}$

Dans la pratique le choix d'une méthode efficace pour calculer h pour un ensemble d'observations est plus complexe vue l'influence de la bande passante sur la forme de l'estimateur correspondant. Si la bande passent est trop petite, on obtiendra un estimateur avec une haute variabilité et un sous-lissage. Et si la valeur de h est importante, l'estimateur résultant sera très lisse et plus éloigné de la fonction que nous essayons d'estimer[330](Fig. 5.1).

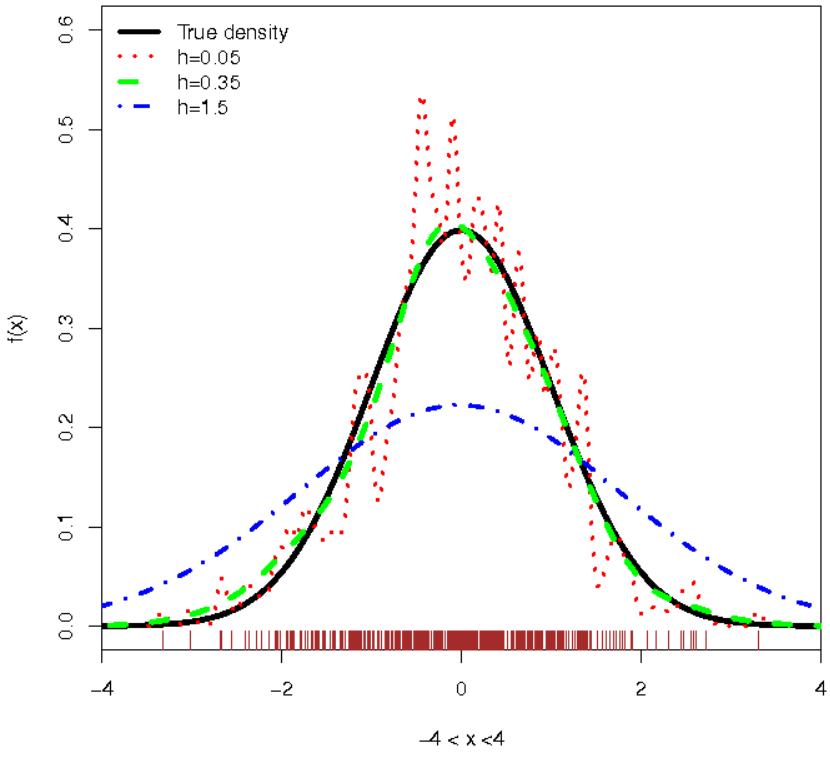


FIGURE 5.1 – Densité de distribution normale standard et l'estimateur noyau(KDE) de sa densité obtenues à partir d'un échantillon aléatoire de taille 500 ; avec des bandes passantes différentes : Ligne continue : Vrai densité (normal standard). ligne pointillée : KDE avec h = 0,05. La ligne en tiret : KDE avec h = 0,35. ligne en point-tiret : KDE avec h = 1,5.

Pour évaluer le compromis entre le biais et la variance, Silverman [366] a suggéré l'utilisation de la bande passante donnée par :

$$h_n = 0.9(min(\hat{\sigma}, \frac{IQR}{1.34})n^{\frac{1}{5}} \tag{5.12}$$

où IRQ est l'intervalle interquantille et et $\hat{\sigma}$ est l'écart type de l'échantillon. Comme dans toutes les procédures de sélection de bande passante souhaitables, cette bande passante devient plus petit que le nombre d'observations n augmente, mais ne va pas à zéro "trop vite" [111].

5.4 Le classificateur probabiliste

Comme mentionner ci-haut, le but de ce chapitre est de développer une méthode de classification non paramétrique en utilisant les copules pour estimer la densité conditionnelle $f^j(x)$ pour qu'un élément x soit membre d'une classe ω. En réalité, nous utilisons la copule empirique comme outil d'estimation de $f^j(x)$ donnée par l'équation 5.4.

Considérons un ensemble m classe ω_i ; $i = 1, \cdots, m$. Chaque classe ω_i est caractérisée par un vecteur aléatoire à d-dimension $\mathbf{X}^i = (X_1^i, \ldots, X_d^i)$.

Soit $(X_{11}^i, \ldots, X_{1d}^j), \ldots, (X_{n1}^j, \ldots, X_{nd}^j)$ un échantillon aléatoire issue de la classe ω_j. La distribution de la composante $\mathbf{X}_i^j$ du vecteur aléatoire $\mathbf{X}^j$ peut être estimée par

$$F_{n,i}^j(x_i) = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}\left(X_{ki}^j \leq x_i\right).$$

La fonction de densité de cette composante est, aussi, estimée par

$$\hat{f}_i^j(x_i) = \frac{1}{n}\sum_{j=1}^{n} K(x_i - X_{ji})$$

où

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$$

La densité du vecteur $\mathbf{X}_j$ peut être estimée par

$$\hat{f}^j(\mathbf{x}) = \hat{c}^j\left(F_{n,1}^j(x_1), \ldots, F_{n,d}^j(x_d)\right) \prod_{i=1}^{d} \hat{f}_i^j(x_i) \tag{5.13}$$

où $\hat{c}^j$ dénote l'estimateur de la densité de la copule associée au vecteur aléatoire $\mathbf{X}^j$ estimée par une fonction noyau standard dans l'équation 5.10.

Ainsi, tous les éléments de notre classificateur sont construits, à savoir :

— $\hat{c}$ l'estimateur de la densité de la copule,

— $\hat{f}_i^j$ l'estimateur de la densité marginale,

— et $\hat{f}^j$ l'éstimateur de la densité conjoint.

Le but du classificateur et de déterminer, étant donnée une nouvelle observation x, sa classe la plus probable ω_r choisie comme suit :

$$r = \arg\max_j \hat{f}^j(\mathbf{x})$$

Finalement, l'algorithme 17 décrit les principales étapes de notre classificateur.

Algorithme 17 : Algorithme du Classificateur probabiliste

début
  Soit $\mathbf{x} = (x_1, \ldots, x_d)$ une nouvelle observation;
  pour chaque $j \in \{1, \cdots, m\}$ **faire**
    pour chaque $i \in \{1, \cdots, d\}$ **faire**
      $u_i^j \leftarrow F_{n,i}^j(x_i)$;
      Calculer $\hat{f}_i^j(x_i)$;
    fin
    Calculer $\hat{c}^j \left(F_{n,1}^j(x_1), \ldots, F_{n,d}^j(x_d)\right)$ comme décrit ci-haut;
    Calculer $\hat{f}^j(\mathbf{x})$ à partir de l'équation(5.13);
  fin
  affecter l'observation x à la classe ω_r tel que

$$r = \arg\max_j \hat{f}^j(\mathbf{x})$$

fin

5.5 Test et comparaisons

Pour Vérifier la faisabilité et l'efficacité du classificateur, nous utilisant le jeu de données KDD'99([91]) décrite dans la section **??**. Les calculs sont effectués sous l'environnement statistique R [333, 334] en utilisant Les packages parallèles snow[371] et snowfall[372] sous Linux RedHat enterprise 6 workstation sur un Intel Core I7 avec 16 Go de RAM Ram et 4 cœurs physiques.

La table table 5.2 représente la matrice de confusion entre les cinq catégories de comportements. Cette version condensée de la matrice de confusion nous permet de comparer nos résultats avec ceux obtenus par d'autres auteurs ayant utilisé le même jeu de données.

TABLE 5.2 – Résultats par catégories d'attaques.

	Normal	Dos	Probe	R2L	U2R
Normal	97.375	0.406	2.038	0.175	0.006
Dos	0.068	97.357	2.563	0.010	0.002
Probe	4.928	4.199	90.548	0.094	0.231
R2L	0.000	0.000	0.000	100.000	0.000
U2R	0.000	0.000	0.000	0.000	100.000

Les distributions conditionnelles sont données sur les lignes de la table. Par exemple, la première ligne montre que le comportement normal est identifié comme normal avec une probabilité estimée à 97.375%(Vrai négatif), comme attaque DOS avec une probabilité de 0.406%, comme PROB à 2.038%, comme R2L avec une probabilité égale à 0.175% et enfin comme U2R avec 0.006%. Les quatre dernières identifications sont dits "Faux positifs". A partir de la seconde ligne, quand une attaque est identifiée comme activité normale on dit qu'on est en présence d'un "Faux négatif" autrement on est devant un "Vrai négatif".

Afin d'évaluer les performances de notre méthode, nous avons comparé, dans le tableau 5.3, nos résultats avec ceux obtenus par d'autres auteurs ayant utilisé le même ensemble de données.

TABLE 5.3 – Comparaison de performances du Classificateur proposé.

Méthode	normal	DOS	PROBE	U2R	R2L
PNRule[6]	99.50	96.9	73.20	06.60	10.70
PSM & SVM[22]	99.80	97.90	98.60	68.90	19.50
CSFDTM[44]	99.20	100	71.40	84.40	99.50
NB-DT[47]	96.64	96.38	78.18	11.84	7.11
ADWICE[57]	97.00	99.00	99.00	92.00	31.00
PCA-SVM [121]	99.80	92.50	98.30	05.10	70.20
GP Multi-Transformation[129]	99.93	98.81	97.29	45.20	80.22
WANBT[132]	99.93	99.91	99.84	99.47	99.63
SVM + DGSOT[215]	95.00	97.00	91.00	23.00	43.00
MCAD[233]	95.20	99.20	97.0	72.80	69.20
M.C.S[299]		97.40	83.80	32.80	10.70
KDD cup 99 Winer [315]	99.50	97.10	83.30	13.20	08.40
Multi-C [346]		97.30	88.70	29.80	09.60
GDA+ANN [369]	98.95	98.63	96.25	24.12	12.08
GDA+C4.5	99.68	98.60	99.61	57.01	66.25
I.C.A.[421]	69.60	98.00	100.00	71.40	99.20
Parzen-window [423]	97.38	96.71	99.17	93.57	31.17
C.N.B.D.[131]	99.72	99.75	99.25	99.20	99.26
ESC-IDS-1[388]	98.20	99.5	84.10	14.10	31.50
Model 1(a)[186]		97.40	83.80	32.80	10.70
C.L.C. [243]	73.95	99.88	87.83	61.36	98.50
Notre méthode[216]	**99.80**	**99.997**	**99.48**	**100.00**	**98.57**

5.6 Conclusion

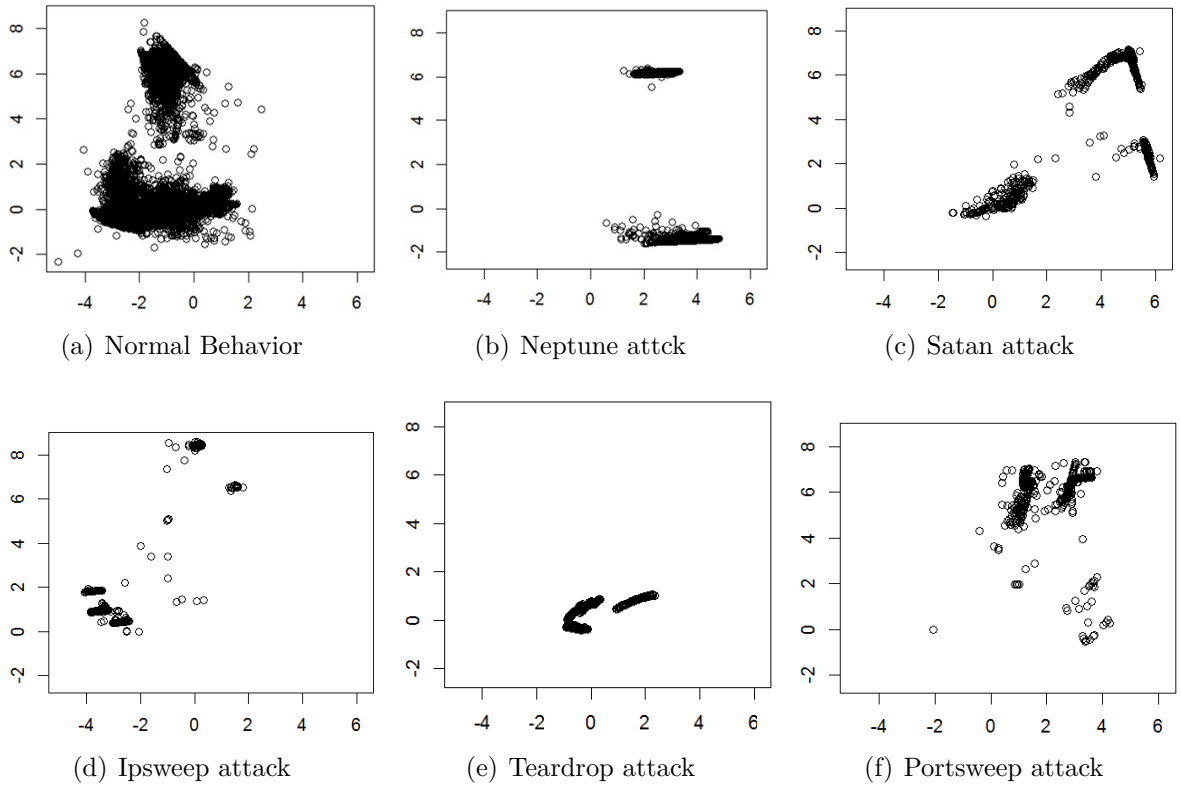
Les résultats des tests et des comparaisons prouvent, d'une part, l'efficacité du classificateur ainsi construit et confirme, d'autre part, le fait que les copules sont flexibles et puissantes en matière d'études de la dépendances et de construction des familles de distributions multivariées, notamment, dans le cas où des dépendances non-linéaires sont impliquées et doivent être représentées, essentiellement, si les lois de probabilité associées au attributs sont non gaussiennes.

CHAPITRE 6

DÉTECTION D'INTRUSION AVEC UNE REPRÉSENTATION MULTI-CONNEXE

6.1 Introduction

Récemment, les méthodes d'extraction de connaissances et d'apprentissage machine sont devenus la base principale des systèmes de détections d'intrusion. Les méthodes de détection qui ont été proposées sont très souvent à base de techniques statistiques ou de l'intelligence computationnelle et sont globalement répertoriées en deux grandes catégories d'approches à savoir la détection d'abus d'utilisation et la détection d'intrusion. Ces méthodes considèrent généralement que les comportements aussi bien normaux qu'intrusif représentés dans un espace topologique sont implicitement connexes. Cette hypothèse n'est pas évidente, une simple projection 2D, issue d'une analyse aux composantes principales sur l'ensemble de données de DARPA[91], montre que certaines représentations peuvent être non connexes(Fig. 6.1).



(a) Normal Behavior (b) Neptune attck (c) Satan attack

(d) Ipsweep attack (e) Teardrop attack (f) Portsweep attack

FIGURE 6.1 – Projection 2D de quelques comportements.

Selon le théorème de l'hyperplan séparateur (Théorème de Hahn-Banach et ses corollaires), cette non-connexité persiste dans les espaces de grande dimension, même si la dimension est infinie. Par conséquent, si pour un certain nombre d'attributs, la représentation géométrique d'un comportement(normal ou anormal) est non connexe, l'ajout d'autres attributs préservera cette non connexité. De ce fait les systèmes de détection d'abus d'utilisation aussi bien que ceux de la détection d'anomalie se trouvent considérablement affectés s'ils adoptent cette hypothèse de connexité. Nous supposons que la connexité des représentations de comportements, considérée implicitement comme une hypothèse, est inappropriée (Fig 6.1). De ce fait nous adoptons une nouvelle démarche qui consiste à décomposer chaque classe décrivant un comportement normal ou intrusif en un ensemble de sous-groupes connexes, que nous appellerons classes naturelles. Nous utiliserons, en premier lieu, ces classes naturelles comme outil de description des classes d'apprentissage préalablement construites par des experts. Dans l'étape de détection, chaque comportement capturé peut être affecté à la classe naturelle la plus appropriée sous certaines conditions. De toute évidence, si les classes naturelles constituent une partition de l'espace de représentation des données et si chaque classe d'apprentissage est une véritable union des classes naturelles, il existe, alors une estimation presque parfaite de sa vraie classe d'apprentissage. Or ce dernier cas est rarement réalisable car les partitions générées par les classes naturelles n'expliquent pas exactement celles générées par les classes d'apprentissage. Nous proposerons, une approche de construction des classes naturelle à base de certaines caractéristiques topologiques et stochastiques. Cette méthode à l'avantage d'être facile à implémenter sur n'importe quel type de réseau disposant d'un outil de surveillance et d'audit de trafic ; comme elle peut être utilisée en temps réel. Elle peut être mise à jour automatiquement autant sur le plan d'une meilleure discrimination des classes que l'introduction d'une nouvelle classe d'intrusion.

6.2 Principe de l'approche

Nous considérons, que les données sont représentées dans un espace métrique à p dimensions. Nous considérons aussi une mesure obtenues à partir d'une mixture $g(x)$ de m densités gaussiennes notées f_j ; $j = 1, \cdots, m$. Chaque densité f_j est caractérisée par une moyenne μ_j et une matrice de variances σ_j ;$j = 1, \cdot, m$. La mixture, qui est une combinaison convexe des f_j, qui est à mode unique, peut être écrite comme suit :

$$g(x) = \sum_{j=1}^{m} \alpha_j f_j(x)$$

avec

$$\sum_{j=1}^{m} \alpha_j = 1,\ \alpha_j > 0,\ j = 1, ..., m.$$

Les points intéressants du modèle g sont essentiellement ses modes qui sont, en réalité, ses maximum locaux. Cependant, le nombre de maximum de g, noté k, est généralement plus grand que le nombre de classes d'apprentissage, noté m, à cause de l'inter-action entre classes. Si des classes sont multi-connexes le nombre k doit être le plus grand. En premier lieu, nous devons différencier entre les modes effectifs, associés aux densités à mode unique f_j, $j = 1, \cdots, m$, composant le mélange, et les modes synthétiques obtenus par des interactions entre classes.

Les modes sont représentés par des points où la densité atteint son maximum. Dans le cas expérimental, le maximum est estimé par un point dont la concentration de voisinage est la plus grande. Dans cette étude, ce point, dit point d'accumulation, est estimé par des techniques de clustering appliquées aux données d'apprentissage. Chaque cluster obtenu est une approximation d'une classe naturelle. Ces classes naturelles, notées, C_i; $i = 1, ..., k$, peuvent être

représentées par une partition connexe de $\mathbb{R}^p$, notée $\{C_i;\ i = 1, ..., k\}$ où $\forall x \in \mathbb{R}^p$, $\exists \varepsilon(x) \in \mathbb{R}^*_+$, tel que $\{x;\ x \in \mathbb{R}^p;\ g(x) > \varepsilon(x)\} \subset C_i$.

Si les classes naturelles constituent une sous-partition de la partition générée par les classes d'apprentissage, notre étude devient très facile. Cependant, ce cas se produit rarement dans la pratique. Les modes synthétiques produisant des classes naturelles synthétiques doivent être évités. La discrimination entre classes naturelles effectives(classe naturelle est associée à un mode effectif) et synthétiques est faite par la méthode de maximum de vraisemblance quand la dimension de l'espace de représentation est faible[278]. Nous avons juste à résoudre l'équation :

$$\nabla_\theta Log(L(x_1, x_2, ..., x_n; \theta)) = 0 \tag{6.1}$$

où L correspond au produit de n densité où chaque densité est elle même une mixture de m densités pondérées.

Chaque classe naturelle est caractérisée par un centre μ_j et une matrice de variance Σ_j et un poids α_j. Le paramètre général θ est identifié à (α_1,..., α_m,μ_1,..., μ_m, Σ_1,..., Σ_m) qui est construit par $m(1+p+p(p+1)/2)$ scalaires inconnus. Quand l'équation 6.1 a une seule solution, la méthode peut être considérée comme un outil performant pour estimer les sommets réels des classes naturelles. Comme le système 6.1 est non linéaire, avec un grand nombre de paramètres, l'implémentation numérique d'une bonne solution approximative est très complexe. Cependant, si les réelles classes naturelles sont localisées, le problème se réduit à m sous-systèmes. Chaque sous-système nécessite l'estimation de juste un centre et une matrice de variance. Dans le cadre d'une implémentation parallèle de la solution, on est très souvent tenté de construire une métrique locale pour chaque sous-système. Cependant, nous avons observé que cette approche produit un important taux de mauvaise classification. Pour éviter ce problème, nous utilisons une métrique topologique globale adaptée. On rappel que dans le cas d'un espace à haute dimension, des auteurs utilisent des techniques de réduction de dimension comme "Mixture of Factor Analyzers(MFA)"[38].

6.3 Modèle développé

Pour exploiter la méthode dans le cadre de l'apprentissage, on considère un ensemble de n observations $x_1, x_2, ..., x_n$ faites sur p variables décrivant m comportements dont un correspond à un trafic normal et les autres, $m-1$, correspondent à des attaques. Les données sont présentées sous forme de tableau comme suit :

$$X = \begin{pmatrix} x_{11} & x_{12} & ... & x_{1p} \\ x_{21} & x_{22} & ... & x_{2p} \\ ... & ... & ... & ... \\ x_{n1} & x_{n2} & ... & x_{np} \end{pmatrix}$$

Chaque classe naturelle C_i (aussi bien réelle que synthétique) est caractérisée par son centre c_i donné par :

$$c_i = \frac{1}{n_i} \sum_{\{j;\, x_j \in C_i\}} x_j$$

où n_i est le cardinal de C_i et x_i est un vecteur de $\mathbb{R}$

L'information qui va être exploitée correspond en particulier à la dispersion des observations autour de la moyenne exprimée en termes de variances et de covariances.

Étant donné un nouveau vecteur t représentant une nouvelle connexion et une base $(\overrightarrow{v_1}, \overrightarrow{v_2}, ..., \overrightarrow{v_p})$ dans l'espace de représentation qui un espace affine, on a

$$\overrightarrow{ot} = \overrightarrow{oc_i} + \Sigma_{i=1}^{p} \beta_i \overrightarrow{v_i}$$

où o est l'origine.

On utilisant les propriétés de l'espace on peut écrire :

$$\overrightarrow{c_i t} = \Sigma_{i=1}^{p} \beta_i \overrightarrow{v_i}$$

En notons $\sigma^2_{\overrightarrow{c_i t}}$ la variance conditionnelle à la direction $\overrightarrow{c_i t}$ à un risque fixé α, la règle de décision peut être construite dans l'intervalle de confiance :

$$I_{\overrightarrow{c_i t}} = \left[c_i - r_{\alpha/2} \sigma_{\overrightarrow{c_i t}} \frac{\overrightarrow{c_i t}}{||\overrightarrow{c_i t}||},\ c_i + r_{\alpha/2} \sigma_{\overrightarrow{c_i t}} \frac{\overrightarrow{c_i t}}{||\overrightarrow{c_i t}||} \right] \tag{6.2}$$

où $r_{\alpha/2}$ est le quantile d'ordre $1 - \alpha/2$ et $\frac{\overrightarrow{c_i t}}{||\overrightarrow{c_i t}||}$ est le vecteur unitaire dans la direction de $\overrightarrow{c_i t}$. l'origine est translaté à la position $\overrightarrow{c_i}$. La quantité $\sigma_{\overrightarrow{c_i t}}$ est obtenue à partir de la variance intra-class (within variance) projeté dans la direction de $\overrightarrow{c_i t}$. Comme la projection est linéaire, la loi des points projetés sur la ligne générée par $\overrightarrow{c_i t}$ reste gaussienne ce qui justifie 6.2. L'affectation de la nouvelle observation t à une classe naturelle est, en générale, facile. Cependant son affectation à une classe d'apprentissage reste difficile, notamment si la classe d'apprentissage est multi-connexe. En effet, si les classes naturelles sont réellement des sous-partitions des classes d'apprentissage, l'identification des comportements devient sans risque. Si non le taux de mauvaise classification devient significatif. Les classes naturelles sont construites selon l'algorithme 18.

Une fois les classes naturelles construites, la nouvelle observation t sera affectée à la classe la plus appropriée comme suit : En premier lieu, l'observation t est affectée à la classe naturelle C_i solution de :

$$i = arg \min_{j \in \{1,\ldots,m\}} d(t, \bar{c_j}) \tag{6.3}$$

où $d(t, \bar{c_j})$ est distance de Mahalanobis entre t et $\bar{c_i}$ le centre de la classe naturelles C_j et m est le nombre des classes naturelles . Il est à rappeler que la distance de Mahalanobis est associé à l'inverse de la de la matrice des variance-covariance. Si :

- C_i est complètement incluse dans une classe d'apprentissage L_j, alors t est affecté à L_j.
- C_i est partitionnée sur plusieurs classes naturelles $\{L_{j1}, ..., L_{js}\}$, alors t est affecté à la classe d'apprentissage L_{jk} ; $jk \in \{j1, ... js\}$ la plus proche selon(6.3).

De cette façon on aura pas à se soucier de la séparation des classes naturelles réelles des classes naturelles synthétiques générées par interaction. Il est à noter que si le cardinal de la classe C_i est plus petit que le nombre de variables p, le modèle échoue car la matrice de variance-covariance dégénère et on ne pourra pas utiliser la distance de Mahalanobis. Une solution alternative consiste à combiner les variances des classes voisines proportionnellement à leurs cardinaux pour construire une metric de Mahalanobis généralisée. Éventuellement, si le problème de régularité persiste on aura à élargir le voisinage des classes naturelles considérées pour avoir une nouvelle métrique de Mahalanobis locale. Comme il y a beaucoup de classes naturelles de cardinal plus petit que p, La métrique de Mahanalobis généralisée est remplacée par la métrique globale.

Quand une classe L_j est choisie, la distance entre son centre et l'observation t est calculée. L'affectation de t se confirmera seulement si cette distance tombe dans l'intervalle de confiance donnée par 6.2.

Algorithme 18 : Algorithme de construction des classes naturelles

Entrées : $D \leftarrow \{t_1, \ldots, t_n\}$ L'ensemble des classes d'apprentissage;
$C \leftarrow \{c_1, \ldots, c_k\}$. Classes initiales;
$s_0 \leftarrow 1$ maximum de déviation standard initial;
Output : Les classes naturelles

début
  tant que Vrai **faire**
    pour chaque $t \in D$ **faire**
      affecter t à la classe c_j la plus proche selon la distance de Mahalanobis
    fin
    $C_{new} \leftarrow \Phi$ Ensemble vide;
    pour $c_j = \{A_{j1}, \ldots, A_{jq}\} \in C$ **faire**
      Ajouter $\{A_{j1}, \ldots, A_{jq}\}$ à C_{new};
    fin
    $C \leftarrow C_{new}$;
    pour chaque c_j in C **faire**
      Recalculer le centre c_j;
    fin
    $S \leftarrow \Phi$;
    pour c_j in C **faire**
      $V_{c_j} \leftarrow$ Covar (c_j);
      $s_j \leftarrow$ sqrt(sum(diag(V_{c_j})));
      ajouter s_j à S;
    fin
    $s \leftarrow max(S)$;
    si $0.9 \leq \frac{s}{s_0} \leq 1$ **alors**
      retourner C
    fin
    sinon
      $s_0 \leftarrow s$
    fin
  fin
fin

6.4 Test et Résultats

Nous avons tester notre modèle sur l'ensemble de données KDD'99 sous l'environnement R[333]. En premier lieu, nous avons appliqué l'algorithme 18 sur la table d'apprentissage contenant 494 021 enregistrements dont 97 278 décrivent des comportements normales et les autres décrivent 22 attaques. Nous avons, ainsi, obtenu 672 classes naturelles dont les centres et les matrices de variances sont calculés avec les méthodes d'estimation usuelles. Le processus de détection d'intrusion est lancer on utilisant les résultats de la phase d'apprentissage à savoir les différentes classes naturelles et leurs centres et matrices de variances sur un l'ensemble de données de test contenant 4 898 431 dont 972 781 enregistrements correspondent à un trafic normal. Nous avons obtenus les résultats mentionnés dans le tableau 6.1. Où Fq, T.D., T.D.R., F.N., F.N.R., C et C.R correspondent respectivement à : la fréquence, vrai détection, Taux de vrai détection, Faux négatifs, Taux de Faux négatifs, Confusion entre attaques, et le Taux de confusion entre attaques. La matrice de confusion entre les cinq catégories de comportements (NORMAL, DOS, U2R, R2L and PROBE) est présentée dans le tableau 6.2.

Cette présentation condensée nous permet de comparer les résultats obtenus avec d'autres résultats issues d'autres travaux de recherches ayant utiliser les même ensembles de données pour l'apprentissage et les tests. Le tableau 6.3 résume cette comparaison.

Les comparaisons montrent que notre méthodes semble plus performante en matière de détection des comportements normal, DOS et U2R est reste aussi efficace que les autres méthodes en ce qui concerne PROBE et R2L.

TABLE 6.1 – Résultats de Classification

	Class	Fq	T.D.	T.D.R.	F.N.	F.N.R.	C	C.R.
1	back	2203	2203	100.000	0	0.000	0	0.000
2	buffer_overflow	30	30	100.000	0	0.000	0	0.000
3	ftp_write	8	8	100.000	0	0.000	0	0.000
4	guess_passwd	53	53	100.000	0	0.000	0	0.000
5	imap	12	12	100.000	0	0.000	0	0.000
6	ipsweep	12481	12374	99.143	61	0.489	46	0.369
7	land	21	21	100.000	0	0.000	0	0.000
8	loadmodule	9	9	100.000	0	0.000	0	0.000
9	multihop	7	7	100.000	0	0.000	0	0.000
10	neptune	1072017	1071975	99.996	15	0.001	27	0.003
11	nmap	2316	2239	96.675	12	0.518	65	2.807
12	normal	972781	970929	99.810	0	0.000	0	0.000
13	perl	3	3	100.000	0	0.000	0	0.000
14	phf	4	4	100.000	0	0.000	0	0.000
15	pod	264	259	98.106	5	1.894	0	0.000
16	portsweep	10413	10351	99.405	27	0.259	35	0.336
17	rootkit	10	10	100.000	0	0.000	0	0.000
18	satan	15892	15789	99.352	70	0.440	33	0.208
19	smurf	2807886	2807825	99.998	61	0.002	0	0.000
20	spy	2	2	100.000	0	0.000	0	0.000
21	teardrop	979	979	100.000	0	0.000	0	0.000
22	warezclient	1020	1004	98.431	16	1.569	0	0.000
23	warezmaster	20	20	100.000	0	0.000	0	0.000
		4898431	4896086	99.952	267	0.007	206	0.005

TABLE 6.2 – Résultats par catégories d'attaques

	NORMAL	DOS	U2R	R2L	PROBE
NORMAL	99.810	0.102	9e-03	0.049	0.030
DOS	0.002	99.997	0e+00	0.000	0.001
U2R	0.000	0.000	100	0.000	0.000
R2L	1.421	0.000	0e+00	98.579	0.000
PROBE	0.414	0.092	2e-03	0.007	99.484

TABLE 6.3 – Table de comparaison des performances de l' algorithme proposé.

Méthode	normal	DOS	PROBE	U2R	R2L
PNRule[6]	99.50	96.9	73.20	06.60	10.70
PSM & SVM[22]	99.80	97.90	98.60	68.90	19.50
CSFDTM[44]	99.20	100	71.40	84.40	99.50
NB-DT[47]	96.64	96.38	78.18	11.84	7.11
ADWICE[57]	97.00	99.00	99.00	92.00	31.00
PCA-SVM [121]	99.80	92.50	98.30	05.10	70.20
GP Multi-Transformation[129]	99.93	98.81	97.29	45.20	80.22
WANBT[132]	99.93	99.91	99.84	99.47	99.63
SVM + DGSOT[215]	95.00	97.00	91.00	23.00	43.00
MCAD[233]	95.20	99.20	97.0	72.80	69.20
M.C.S[299]		97.40	83.80	32.80	10.70
KDD cup 99 Winer [315]	99.50	97.10	83.30	13.20	08.40
Multi-C [346]		97.30	88.70	29.80	09.60
GDA+ANN [369]	98.95	98.63	96.25	24.12	12.08
GDA+C4.5	99.68	98.60	99.61	57.01	66.25
I.C.A.[421]	69.60	98.00	100.00	71.40	99.20
Parzen-window [423]	97.38	96.71	99.17	93.57	31.17
Notre méthode[217]	**99.80**	**99.997**	**99.48**	**100.00**	**98.57**

6.5 Conclusion

Le fait de considérer que les différents comportements peuvent avoir une représentation multi-connexe nous a permis de développer une méthode de détection d'intrusion ayant un taux de détection élevé et de faibles taux de faux négatifs et de faux positifs. Cette approche est flexible et permet de combiner les avantages de la détection d'anomalie et la détection d'abus d'utilisation. Elle a l'avantage d'être rapidement et facilement mis en œuvre et mis à jour. Les résultats de comparaisons avec d'autre travaux ont montré que notre modèle est très compétitif avec des modèles hybride ce qui nous pousse à croire que notre méthode peut être considérablement améliorée par une hybridation avec une autre technique. Aussi le fait d'utiliser d'autres métriques à la place de celle de Mahanalobis lorsque cette dernière devienne inappropriée peut contribuer à l'améliorer. A fin de réduire le temps de réponse de cette méthodes nous comptons développer une version parallèle de nos algorithmes.

CONCLUSION GÉNÉRALE

L'utilisation omniprésente des ordinateurs et des réseaux informatiques dans la société d'aujourd'hui, d'une part et la complexité des technologies utilisées, la croissance exponentielle des terminaux à protéger ainsi que la prolifération de nouvelles menaces de plus en plus sophistiquées d'autre part, ont attribué à la sécurité informatique une priorité primordiale. La directrice du Swiss Cybersecurity and Advisory Research Group Solange Ghernaouti, lors de son intervention au cours de la 4^e édition du symposium international sur la cybercriminalité en Algérie organisé par le World Trade Center Algeria du 04 au 05 octobre 2015, avait insisté sur l'importance pour les petites et moyennes entreprises de se prémunir contre les attaques informatiques, précisant que les risques ne sont pas virtuels et que le crime est bien réel. "Il existe chaque jour de nouveaux moyens pour nuire, destabiliser, influencer, conquérir et faire la guerre". Les attaques électronique deviennent de plus en plus sophistiquées, diversifiées, complexes et dont le nombre ne cesse d'augmenter comme le montre les statistiques issues de la Gendarmerie nationale. Cette dernière avait recensé, en 2015, près de 164 plaintes concernant des cybercrimes contre, seulement, 18 plaintes ayant été déposées en 2009. De sont côté M. Sid Ahmed Tibaoui, le directeur du World Trade Center Algeria, a souligné qu'il faudra apporter des approches pluridisciplinaires pour cerner les enjeux de la cyber-sécurité en liaison avec la souveraineté des données qui sont à la fois politiques, juridiques, technologiques, diplomatiques, économiques, sécuritaires et sociétales. Dans ce contexte de considérables efforts de recherche et de développement d'outils pour lutter contre la cyber-sécurité ont été consentis. Parmi les aboutissements de ces efforts, on compte les systèmes de détection d'intrusion devenus une composante incontournable de toute architecture de sécurité informatique. Depuis leur introduction par J. P. Andersson en 1980, ces systèmes n'ont pas cessé de se développer et de gagner en précision et en performance grâce à l'intégration des techniques et méthodes issues de différents domaines tel que la statistique, l'intelligence artificielle et la fouille des données.

Dans le cadre de cette thèse, nous nous sommes intéressés particulièrement à la contribution des méthodes de la fouille des données dans l'amélioration des performances des systèmes de détection d'intrusion. Notre point de départ a été de présenter un aperçu global sur cette sous-branche de la sécurité informatique. Nous avons présenté en quoi consiste-t-elle, quels sont ses outils, ses techniques, ses approches, ses défis, ses avantages et ses limites. En deuxième lieu, nous avons introduit le concept de la fouille des données où nous avons présenté ses méthodes et algorithmes ainsi que ses promesses et ses limites afin de voir comment cette discipline pourrait contribuer à la "construction" d'une solution optimale au problème de la détection d'intrusion. Par ailleurs, nous avons présenté un état de l'art sur les approches de détection d'intrusion à base de méthodes et algorithmes de la fouille des données. A l'issue de cette étude, nous concluons que :

1. Les efforts de recherches et développement consentis en matière de lutte contre les menaces et les attaques ciblant les systèmes informatiques ont abouti à un certain nombre considérable de technologies tels les firewalls, les techniques de cryptage de données, les

mécanismes d'authentification et de vérification de vulnérabilités. Toutefois ces outils, souffrant d'un nombre considérable d'inévitables vulnérabilités, ne sont pas en mesure de faire face, efficacement, aux différentes attaques qui sont continuellement sophistiquées, diversifiées et adaptées à exploiter les faiblesses des systèmes informatiques. Ainsi, même équipés avec cette panoplie de mesures de protection, les systèmes restent toujours exposés aux intrusions profitant ainsi de ces failles et des astuces de l'ingénierie sociale. Une machine non connectée à aucun réseau reste vulnérable aux employés mécontents ou à des externes profitant de leurs privilèges. Compte tenu de cette situation, qui perdure, il est très judicieux d'établir une deuxième ligne de défense sous forme d'un système de détection d'intrusion. Or pour être efficaces, ces systèmes doivent répondre à un certain nombre d'exigences de fiabilité, de réactivité, d'adaptabilité, d'exactitude et de tolérance aux pannes.

2. D'énormes efforts ont été consentis afin de rendre les systèmes de détection d'intrusion plus conformes aux exigences souhaitées. Ainsi, dès leur premières apparition en 1980, plusieurs approches et modèles de détection ont été proposées dans la littérature. Initialement, ces approches et modèles étaient à base de techniques et méthodes issues du domaine de la statistique puis de l'intelligence artificielle, notamment les systèmes experts. Or les systèmes de détections d'intrusion implémentant ces modèles et approches n'ont pas été à la hauteur des exigences. Il a fallu donc puiser dans d'autres domaines scientifiques. Et comme la détection d'intrusion est, par nature, un processus de classification traitant de grandes masses de données, la fouille des données semble être un domaine approprié. Cela est dû essentiellement au fait que, d'une part, la fouille des données offre un ensemble de méthodes et d'algorithmes d'extraction de relations et de similarités non triviales cachées dans de grandes masse de données ainsi que des techniques de réduction de données et de sélection d'attributs les plus pertinents permettant de réduire, considérablement, délais et besoins en ressources calculatoire. Les résultats fournis par ces techniques peuvent être utilisés dans des systèmes de prise de décision automatique d'autre part.
3. Ainsi, dans le contexte d'intégration des méthodes et techniques du Datamining dans la détection d'intrusion, plusieurs plateformes et architectures de détection d'intrusion, de plus en plus performantes et de plus en plus précises, ont été proposées. Ces dernières ont été essentiellement basées sur la détection d'anomalie. Mais, lors de cette dernière décennie, la détection d'abus d'utilisation à base de fouille des données semble gagner plus d'intérêt. Par ailleurs, l'hybridation des techniques est devenue très courante dans les systèmes de détection d'intrusion récents. Cette tendance exige l'intégration de module prenant en considération la multiplicité décisionnelle.
4. Généralement, les techniques proposées dans la littérature semblent être bonnes pour détecter au moins un type d'activité malveillante, toutefois aucune de ces techniques ne prétend être en mesure de détecter tout type d'intrusion. Ainsi un système de détection d'intrusion capable de détecter toute intrusion(connue ou non) au moment opportun semble être toujours une utopie. Puisque malgré le déploiement de tel systèmes, le niveau d'alerte reste à un niveau élevé et de nouveaux incidents sont répertoriés de jours en jours. L'être humain reste le maillon le plus faible de la chaîne de sécurité ce qui permet des attaques sophistiquées où l'utilisateur légal est manipulé afin d'exécuter, sans se rendre compte, des attaques déguisées en se servant de ses autorisations. Pour faire face à ces lacunes, de nouveaux concepts de soutien et de compréhension des utilisateurs dans le processus de sécurité sont nécessaires. Des analystes humains, des équipes de réponse aux incidents et des juristes doivent être impliquées dans toute entreprise pour compléter les systèmes de détection d'intrusion.
5. Doter les systèmes de détection d'intrusion de mécanismes de traitement parallèle et

d'une architecture hiérarchique leur procurera une meilleure involutivité et réflexivité. Par ailleurs, il est préférable de concevoir des méthodes spécialement adaptées au problème de détection d'intrusion au lieu de tenter d'adapter les méthodes à but général du datamining.

à travers cette étude, nous avons montré comment les méthodes issues du datamining pouvaient contribuer à l'amélioration des performances des systèmes de détection d'intrusion. Après avoir testé un ensemble de méthodes tels, entre autres, les réseaux bayésiens, les réseaux de neurones, les règles d'association, les K-means, les KNN, les algorithmes génétiques, les SVM et, notamment, l'analyse en composantes principale (ACP), nous avons remarqué que les comportements, à la fois normaux et intrusifs, ne sont par nécessairement représentés par des classes connexes comme le suggèrent la majorité des approches proposées dans la littérature. De ce fait, nous avons proposé un algorithme pour re-segmenter, suffisamment, chaque représentation des comportements (normal ou malveillant) par des sous-ensembles connexes, dites classes naturelles, qui ont été utilisées, avec des métriques appropriées, comme données d'apprentissage pour entraîner des Classificateurs [216, 217]. Les résultats de comparaisons avec d'autres travaux ont montré que ces deux modèles sont très compétitifs.

Comme perspectives immédiates, nous comptons implémenter ces deux modèles [216, 217] sur des environnements parallèles tels les GPU ; travail déjà entamé. Nous allons faire migrer notre solution sur le cluster récemment installé au niveau de l'université Djillali liabes dès sa mise en exploitation. Nous comptons exploiter également d'autres méthodes telles les approches issues de la vie artificielle, notamment les automates cellulaires, qui semblent très prometteuses, d'une part, et la théorie des jeux qui nécessiterait une bonne modélisation des stratégies de l'administrateur et celles des attaquants potentiels, d'autre part.

BIBLIOGRAPHIE

[1] M. Abadi, S. Jalali, An ant colony optimization algorithm for network vulnerability analysis, Iranian Journal for Electrical and Electronic Engineering, Vol. 2, Nos. 3 & 4, pp. 106-120, 2006.

[2] T. Abraham, IDDM : Intrusion Detection Using Data Mining Techniques, Technical report DSTO-GD-0286, Defence Science and Technology Organisation(DSTO) Electronics and Surveillance Research Laboratory, `http://dspace.dsto.defence.gov.au/dspace/bitstream/1947/3750/1/DSTO-GD-0286%20PR.pdf`, 2001.

[3] A. Abraham & J. Thomas, Distributed Intrusion Detection Systems : A Computational Intelligence Approach, In Applications of Information Systems to Homeland Security and Defense,Edited by Hussein A. Abbass and Daryl Essam, Vol. 5, pp. 107-137, IDEA Group Publishing, 2006.

[4] A. O. Adetunmbi, A bagging approach to network intrusion detection, Journal of the Nigerian Association of Mathematical Physics, Vol. 15, pp. 379-390, 2009.

[5] R. Adhikari & R. K. Agrawal, An Introductory Study on Time Series Modeling and Forecasting, Lambert Academic Publishing (LAP), Saarbrücken, Germany, 2013.

[6] R. Agarwal & M. V. Joshi, PNrule : A New Framework for Learning Classifier Models in Data Mining, Department of Computer Science, University of Minnesota, Report No. RC-21719 , 2000.

[7] D. K. Agarwal, Shrinkage estimator generalizations of proximal support vector machines, In Proceedings of the 8th International Conference Knowledge Discovery and Data Mining, pp. 173–182, 2002.

[8] C. C. Aggarwal, Data Classification : Algorithms and Applications, Taylor & Francis Group, LLC, 2015.

[9] C. C. Aggarwal, Data Mining : The Textbook, Springer International Publishing Switzerland 2015.

[10] M. H Aghdam, & Peyman Kabiri, Feature Selection for Intrusion Detection System Using Ant Colony Optimization, International Journal of Network Security, Vol.18(3), pp.420-432, 2016.

[11] R.Agrarwal, T. Imielinski, A. Swami, Mining association rules between sets of items in very large database, Proceedings of ACM SIGMOD conference, 1993.

[12] R. Agrawal & R. Srikant, Fast Algorithms for Mining Association Rules, In the proceedings of the 20th International Conference on Very Large Databases, Santiago, Chile, pp. 487–499, 1994.

[13] R. Agrawal & R. Srikant. Mining Sequential Patterns, In Proceedings of the 11th IEEE International Conference on Data Engineering, pp. 3–14, 1995.

[14] R. Agrawal, J. Gehrke, D. Gunopulos, & P. Raghavan, Automatic subspace clustering of high dimensional data for data mining applications, In Proceedings of the 1998 ACM SIGMOD international conference on Management of data (SIGMOD '98), pp. 94–105, 1998.

[15] I. Ahmad, A. Abdullah, A. Alghamdi, Towards the selection of best neural network system for intrusion detection, International Journal of the Physical Sciences Vol. 5(12), pp. 1830-1839, 2010.

[16] A. Ahmad, B. Shanmugam, N.B. Idris, G.N. Samy, Danger Theory Based Hybrid Intrusion Detection Systems for Cloud Computing, International Journal of Computer and Communication Engineering, Vol. 2(6), 2013.

[17] I. Ahmad, Feature Selection Using Particle Swarm Optimization in Intrusion Detection, International Journal of Distributed Sensor Networks , Vol. 11, 2015.

[18] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, J. McLeod, Danger Theory : The Link between AIS and IDS, In Proceedings of the ICARIS-2003, 2nd International Conference on Artificial Immune Systems, pp. 147-155, 2003.

[19] P. Akshaya, Intrusion Detection System Using Machine Learning Approach, International Journal of Engineering And Computer Science, Vol. 5(10), pp. 18249-18254, 2016.

[20] N. Aladenise, & B. Bouchon-Meunier, Acquisition de connaissances imparfaites : mise en évidence d'une fonction d'appartenance, Revue Internationale de Systémique, Vol. 11(1), pp. :109–127, 1997.

[21] P. Albuquerque, & A. Dupuis A, A parallel cellular ant colony algorithm for clustering and sorting, Fifth International Conference on Cellular Automata for Research and Industry(ACRI2002), pp. 220-230, 2002.

[22] E. Al Daoud, Intrusion Detection Using a New Particle Swarm Method and Support Vector Machines, World Academy of Science, Engeneering and Technology, Vol. 77, pp. 59-62 (2013)

[23] R. Al-Dhubhani, N. B. Idris, F & Saeed, A prototype for network intrusion detection system using danger theory, Jurnal Teknologi, Vol. 73(2), pp. 77-84, 2015.

[24] M. Aldwairi, Y. Khamayseh,M. Al-Masri, Application of artificial bee colony for intrusion detection systems, Journal of Security and communication networks, Vol. 8(16), pp. 2730–2740, 2015.

[25] O. Al-Jarrah Network Intrusion Detection System Using Neural Network Classification of Attack Behavior, Journal of Advances in Information Technology Vol. 6(1), pp. : 1-6, 2015.

[26] M. Amini, J. Rezaeenoor, & E. Hadavandi, The Application of Ensemble Classification Techniques in Network Intrusion Detection : a Review, 1 st National Conference on New Approaches in Computer Engineering and Information Retrieval, Islamic Azad University, Roudsar-Branch, 2013.

[27] J.P. Anderson, Computer Security Technology Planning Study, ESD-TR-73-51, Vol II, Electronic Systems Division, Air Force Systems Command, Hanscom Field, Bedford, MA 01730 (1972).

[28] J.P. Anderson. Computer security threat monitoring and surveillance, James P. Anderson Co., Fort Washington, PA ,1980.

[29] T. Andrysiak, L. Saganowski, M. Maszewski Network anomaly detection based on signal processing techniques, Image Processing & Communication, Vol. 18(01), pp.15-22, 2014.

[30] C. Anumba, Z. Ren, O.O. Ugwu, Agents and Multi-agent Systems in Construction, published by Spon Press, 2005.

[31] K. Aravinthan & M. Vanitha, Literature Survey on Clustering Algorithms, International Journal of Advance Research in Computer Science and Management Studies. Vol.3(4), pp. 447 – 453, 2015.

[32] M. Asaka, A. Taguchi, & S. Goto, The implementation of ida : An intrusion detection agent system, In Proeedings of the 11th Annual FIRST Conferene on Computer Security Incident Handling and Response, 1999.

[33] P. Asgharzadeh & S. Jamali, A survey on intrusion detection system based support vector machine algorithm, International Journal of Research In Computer Applications And Robotics(INRCAR), Vol.3(12), pp. 42-50, 2015.

[34] C. Azad, V.K. Jha, Data Mining in Intrusion Detection : A Comparative Study of Methods, Types and Data Sets, International Journal of Information Technology and Computer Science, Vol. 08, pp. 75-90, 2013.

[35] A. Azween, Cai Long Zheng, Improving Intrusion Detection using Genetic Linear Discriminant Analysis, International Journal of Intelligent Systems and Applications in Engineering, Vol. 3(1), pp. 34-39,2015.

[36] R. Bace, & P. Mell, Intrusion Detection Systems, National Institute of Standards and Technology Special Publication on Intrusion Detection Systems, pp. 1-51, 2001.

[37] C. Bae, W.-C. Yeh, M. A. M. Shukran, Y. Y. Chung, T.-J. Hsieh, A novel anomaly-network intrusion detection system using abc algorithms, International Journal of Innovative Computing, Information and Control, Vol. 8(12), pp. 8231–8248, 2012.

[38] J. Baek., G.J. McLachlan. Mixtures of common t-factor analyzers for clustering high-dimensional microarray data, Bioinformatics, Vol. 27(9), pp. : 1269-1276, 2011.

[39] A. Balon-Perin, Ensemble-based methods for intrusion detection, Master of Science in Computer Science, Norwegian University of Science and Technology, Department of Computer and Information Science, 2012.

[40] D. Barbara & S. Jajodia, Applications of data mining in computer security,6th volume of the Kluwer International Series on Advances in Information Security, 2002.

[41] D. Barbara, J. Couto, S. Jajodia, & N. Wu, ADAM : A testbed for exploring the use of data mining in intrusion detection, ACM SIGMOD Record 30, no. 4, pp. 15–24, 2001.

[42] D. Barbara, N. Wu & S. Jajodia Detecting Novel Network Intrusions Using Bayes Estimators, In Proceedings of the first SIAM International Conference on Data Mining (SDM'01),2001.

[43] P. Barford, J. Kline, D. Plonka, & A. Ron, A signal analysis of network traffic anomalies, Proceedings of the second ACM SIGCOMM Workshop on Internet measurment(Marseille, France), pp. 71-82, 2002.

[44] V. Barot, S.S. Chauhan & B. Patel. Feature Selection for Modeling Intrusion Detection. International Journal of Computer Network and Information Security (IJCNIS), Vol. 6(7), pp.56-62, 2014.

[45] T. Bass, Intrusion detection systems and multisensor data fusion, Communications of the ACM, Vol. 43(4),pp. 99–105,2000.

[46] T. Bayes, An essay towards solving a Problem in the Doctrine of Chances, Philossophical Transactions of the Royal Society of London, Vol. 53, pp. 370-418, 1764.

[47] N. BenAmor, S. Benferhat & Z. Elouedi, Naive Bayes vs. Decision Trees in Intrusion Detection Systems, Proc. of the ACM symposium on applied computing, pp. 420–424, 2004.

[48] G. Beni, & J. Wang, Swarm intelligence in cellular robotics systems, In : Proceedings of NATO Advanced Workshop on Robots and Biological System, pp. 703-712, 1989.

[49] P. Berezinski, B. Jasiul,& M. Szpyrka, An Entropy-Based Network Anomaly Detection Method, Entropy, Vol. 17(04), pp. 2367–2408, 2015.

[50] J. C. BEZDEK, Fuzzy models—what are they, and why ?, In IEEE Transactions on Fuzzy Systems, Vol. 1(1), Editorial, 1993.

[51] R Bhuyan, S Borah, A Survey of Some Density Based Clustering Techniques, National Conference on Advancements in Information, Computer and Communication(AICC'13), Jaipur, Rajasthan, India, March 2013.

[52] M. Bishop, Introduction to Computer Security, Prentice Hall PTR,2004.

[53] P. Biondi, Architecture expérimentale pour la détection d'intrusion dans un système informatique, Phd Thesis, École Nationale Supérieure des Télécommunications de Bretagne, 2001.

[54] H. Bock, Probabilistic aspects in cluster analysis, In O. Opitz editor, Conceptual and Numerical Analysis of Data, pp. 12–44, Augsburg, FRG, Springer-Verlag, 1989.

[55] Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia & S. Gombault, Intrusion Detection Using Principal Component Analysis, In Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics, 2003.

[56] Y. Bouzida & F. Cuppens, Neural networks vs. decision trees for intrusion detection, In IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM), Vol. 28, pp.29-37, 2006.

[57] K. Burbeck, & S. Nadjm-Tehrani, ADWICE - anomaly detection with real-time incremental clustering, In Park, C.-s., Chee, S. (eds.) ICISC 2004. LNCS, Vol. 3506, pp. 407-424. Springer, Heidelberg, 2005.

[58] P. Braun,& W. Rossak, Mobile Agents Basic Concepts, Mobility Models, and the Tracy Toolkit, Copublished by Morgan Kaufmann Publishers and dpunkt.verlag, 2005.

[59] L. Breiman, J. H. Friedman, R. A. Stone, & C. J. Olshen, Classification and regression trees, New York : Chapman and Hall, 1984.

[60] L. Breiman, Bagging predictors, Machine Learning Journal, Vol. 24(2), pp.123–140, 1996.

[61] L. Breiman, Random forests, Machine Learning Journal, Vol 45(1), pp. 5–32, 2001.

[62] S. M. Bridges & R. B. Vaughn RB, Fuzzy data mining and genetic algorithms applied to intrusion detection, In : Proceedings of the 23rd national information systems security conference, MA, USA, 2000.

[63] P. J. Brockwell & R. A. Davis, Time Series : Theory and Methods 2nd Edition, Springer Series in Statistics,1991.

[64] A. Bronstein, J. Das, M. Duro, R. Friedrich, G. Kleyner, M. Mueller, S. Singhal,& I. Cohen, Self-Aware Services : Using Bayesian Networks for Detecting Anomalies in Internet-Based Services, Integrated Network Management Proceedings, pp. 623-638, 2001.

[65] S. T. Brugger, Data Mining Methods for Network Intrusion Detection Technical report, University of Califorinia, Davis, 2004.

[66] P. Cabena ; P. Hadjinian ; R. Stadler ; J. Verhees & A.Zanasi, Discovering Data Mining : From Concept to Implementation, Prentice Hall, Upper Saddle River, NJ, 1998.

[67] B. P. Carlin Bayesian Methods for Data Analysis Third Edition, Chapman & Hall/CRC Taylor & Francis Group, 2009.

[68] J. Carmichael, J. George, & R. Julius, Finding natural clusters, Systematic Zoology, Vol. 17(2), pp. :144–150, 1968.

[69] L. N. De Castro, & J.I. Timmis, Artificial immune system : A new computational intelligence approach, pp. 76-79, Springer, 2002.

[70] G. Cauwenberghs, & T. Poggio, Incremental and decremental support vector machine learning, In Proceedings of the Advances in Neural Information Processing Systems, pp. 409–415, 2000.

[71] M. Cédric, Langage de description d'attaques pour la détection d'intrusion par corrélation d'événement ou d'alerte en environnement réseau hétérogène, thèse de doctorat, université de Rennes, 2003.

[72] K. J. Chabathula, C.D. Jaidhar, & M.A. A. Kumara, Comparative study of Principal Component Analysis based Intrusion Detection approach using machine learning algorithms, 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1-6, 2015.

[73] P. K. Chan, M. V. Mahoney, & M. H. Arshad, Learning Rules and Clusters for Anomaly Detection in Network Traffic, In Managing Cyber Threats : Issues, Approaches and Challenges, Edited by V. Kumar, J. Srivastava, A. Lazarevic Chapter , pp. 81-99, Springer Science+Business Media, Inc., 2005.

[74] H. C., Chang & C.C., Hsu. Using Topic Keyword Clusters for automatic Document Clustering,In proceedings of the _nth3 International Conference on Information Technology and Applications (ICITA'05), Kota Kinabalu, Sabah, 2005.

[75] S. Chaurasia , A. Jain, Ensemble Neural Network and K-NN Classifiers for Intrusion Detection, International Journal of Computer Science and Information Technologies, Vol. 5 (2) , pp. 2481-2485, 2014.

[76] S. Chebrolu, A. Abraham, & J. P. Thomas, Feature deduction and ensemble design of intrusion detection systems, Computers & Security, Vol. 24(4), pp. 295-307, 2005.

[77] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, & D. Freeman, AutoClass : A Bayesian Classification System, In Proceedings of the Fifth International Conference on Machine Learning, pp. 54-64, 1988.

[78] R. Chen, C. M. Liu, C. Chen, An Artificial Immune-Based Distributed Intrusion Detection Model for the Internet of Things, Advanced Materials Research, Vol. 366, pp. 165-168, 2012.

[79] S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, J. Rowe, S. Staniford-Chen, R. Yip, & D. Zerkle, The Design of GrIDS. A Graph-Based Intrusion Detection System, 1999, in Proceedings of the 19th National Information Systems Security Conference, 1999.

[80] D. M. Chickering, Learning Bayesian networks is NP complete, In Learning from Data : Artificial Intelligence and Statistics V, pp. 121–130. Springer-Verlag, 1996.

[81] A. Chittur, Model generation for an intrusion detection system using genetic algorithms, High School Honors Thesis, Ossining High School, 2005.

[82] J. Cannady, Artificial neural networks for misuse detection, In Proceedings of the 1998 National Information Systems Security Conference (NISSC'98), Arlington, VA, 1998.

[83] W. W. Cohen, Fast Effective Rule Induction, In A. Prieditis and S. Russell (Eds.), Proceedings 12th International Conference on Machine Learning, pages 115-123, 1995

[84] D. Cohn, Encyclopedia of machine learning, C. Sammut & G. I. Webb Eds., Springer Science+Business Media, LLC, 2011.

[85] A. Colorni, M. Dorigo,& V. Maniezzo, Distributed optimization by ant colonies, In proceedings of the European conference on artificial life, pp. 134–142, Elsevier, 1991.

[86] P. A. Cornillon, Éric Matzner-Løber, Régression Théorie et applications, Springer-Verlag France, Paris, 2007.

[87] C. Cortes, & V. Vapnik, Support-vector networks, Machine Learning, Vol. 20(3), pp. 273–297, 1995.

[88] M. Crosbie, E. Spafford, Applying Genetic Programming to Intrusion Detection, Proceedings of the AAAI Fall Symposium, 1995.

[89] M. Crosbie, B. Dole, T. Ellis, I. Krsul, & E. Spafford, IDIOT : Users Guide, Purdue University, West Lafayette, Technical Report : TR-96-050, 1996.

[90] R. Cunningham & R. Lippmann, Improving intrusion detection performance using keyword selection and neural networks, In Proceedings of the International Symposium on Recent Advances in Intrusion Detection, Purdue, IN, 1999.

[91] "DARPA Intrusion Detection Data set" `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`, 1999.

[92] D. Dasgupta, Artificial Immune Systems and Their Application, In Artificial Immune Systems and Their Applications, D. Dasgupta(ed.) Springer-Verglas, pp. 3-21, 1999.

[93] D. Dasgupta, Immunity-Based Intrusion Detection System : A General Framework, In Proceedings of the 22nd National Information Systems Security Conference, pp. 147-160, 1999

[94] D. Dasgupta, F. & Gongzalez, An immunity-based technique to characterize intrusions in computer network, IEEE transactions on evolutionary computing, Vol. 6(3), pp. 281-291, 2002.

[95] D. Dasgupta, F. Gonzalez, K. Yallapu, J. Gomez, R. Yarramsettii, CIDS : An agent based intrusion detection system, Computers & Security journal, Vol. 24(5), pp. 387–398 , 2005.

[96] M. Dash, & H. Liu, Feature selection for classiffication, Intelligent Data Analysis, Vol. 1, pp. 131-156, 1997.

[97] H. Debar, M. Dacier, M. Nassehi, & A. Wespi, Fixed vs Variable-Length Patterns for Detecting Suspicious Process Behavior, In Proceedings of the 5th European Symposium on Research in Computer Security, pp. 1-15, 1998.

[98] H. Debar, M. Dacier & A. Wespi, A revised taxonomy for intrusion-detection systems, Annales des Télécommunications, Vol. 55(7-8), pp. 361-378, 2000.

[99] H. Debar, D. Curry, B. Feinstein, RFC 4765 : The Intrusion Detection Message Exchange Format (IDMEF), 2007.

[100] A. J. Deepa,& V. Kavitha, A comprehenseive survey on approaches to intrusion detection system, Procedia Engineering, Vol. 38, pp. 2063 –2069, 2012.

[101] A. S. Deepthi, K. V. Rao, Anomaly Detection Using Principal Component Analysis, International Journal of Computer Science and Technology(IJCST) Vol. 5(4), pp. 124-126, 2014.

[102] P. Deheuvels, La fonction de dépendance empirique et ses propriétés : Un test non paramétrique d' indépendance, Bulletin de la classe des sciences, Académie Royale de Belgique. Vol. 65(6), pp.274-292, 1979.

[103] A.P. Dempster, N.M. Laird & Donald Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society, Series B (Methodological), Vol. 39(1), pp. 1–38, 1977.

[104] J. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, & L. Chretien, The dynamics of collective sorting : Robot-like ants and ant-like robots,In Proceedings of the First International Conference on Simulation of Adaptive Behaviour : From Animals to Animats, Vol. 1, pp. 356–365, 1991.

[105] D.E. Denning, An Intrusion-Detection Model, IEEE transactions on software engineering, SE-13(2), pp. 222-232,1987,

[106] O. Depren, M. Topallar, E. Anarim, and M.K. Ciliz, An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks, Expert systems with Applications, Vol(4), pp. 713–722, 2005.

[107] Md. F. Dewan , M. Zahidur Rahman, Chowdhury Mofizur Rahma, Adaptive Intrusion Detection based on Boosting and Nave Bayesian Classifier, International Journal of Computer Applications, Vol. 24(3), pp. 12-19, 2011.

[108] J. E. Dickerson, & J. A. Dickerson, Fuzzy network profiling for intrusion detection, In proceendings of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS),pp. 301–306, 2000.

[109] J. E. Dickerson, J. Juslin, O. Koukousoula, & J. A. Dickerson, Fuzzy intrusion detection. In IFSA World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) International Conference, Vol. 3, pp. 1506–1510. 2001.

[110] T. G. Dietterich, Machine learning research : Four current directions, the AI Magazine, Vol. 18(4), pp. 97–136, 1997.

[111] J. DiNardo, & J. L. Tobias, Nonparametric Density and Regression Estimation Journal of Economic Perspectives. Vol. 15(4), pp. 11-28, 2001.

[112] S. Ding, Z. Zhu, X. Zhang, An overview on semi-supervised support vector machine, In Neural Comput & Applications, pp.1-10, 2015.

[113] A. Dolgikh, T. Nykodym, V. Skormin, J. Antonakos, & M. Baimukhamedov, Colored Petri Nets as the Enabling Technology in Intrusion Detection Systems, In proceedings of MILCOM 2011 Military Communications Conference, pp. 1297-1301, 2011.

[114] S. S. Dongre & K. K. Wankhade, Intrusion Detection System Using New Ensemble Boosting Approach, International Journal of Modeling and Optimization, Vol. 2(4), pp. 488-492, 2012.

[115] C. Dowell & P. Ramstedt, The computerwatch data reduction tool, In 13th National Computer Security Conference, Washingthn DC, 1990.

[116] S. Dua and X. Du, Data Mining and Machine Learning in Cybersecurity Auerbach Publications, Taylor & Francis Group,ISBN-13 : 978-1-4398-3943-0, 2011.

[117] D. Dubois & H. Prade, Fuzzy set and systems : Theory and Applications, Series : MATHEMATICS IN SCIENCE AND ENGINEERING, ACADEMIC PRESS, New York, 1980.

[118] M. H. Dunham, Data mining introductory and advanced topics, Upper Saddle River, NJ : Pearson Education, Inc.2003.

[119] I. Dutt, S. Borah, & I. Maitra, Intrusion Detection System using Artificial Immune System, International Journal of Computer Applications, Vol. 144(12), pp. 19-22, 2016.

[120] R. C. Eberhart,& J. Kennedy, A new optimizer using particle swarm theory, In 6th international symposium on micro machine and human science, IEEE Service Center, Piscataway, 1995.

[121] H.F. Eid, A. Darwish, A.E. Hassanien, & A. Abraham, Principle components analysis and support vector machine based intrusion detection system, In 10th international conference on intelligent systems design and applications (ISDA), Cairo, Egypt, pp. 363-367, 2010.

[122] A. Elngar, Mohamed D. & F. Ghaleb A Real-Time Anomaly Network Intrusion Detection System with High Accuracy, International Journal of Information Sciences Letters, Vol. 2(02), pp. 49-56, 2013.

[123] S. Er, A. Bhandari & K. K. Saluja, Applying Genetic Algorithm in Intrusion Detection System : A Comprehensive Review, Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing, pp. 102-112, 2014

[124] Ethem Alpaydin Introduction to machine learning, Second edition, The MIT Press Cambridge, Massachusetts, London, Egland, 2010.

[125] E. Eskin, A. Arnold, M. Preraua, L. Portnoy, & S. J. Stolfo, A geometric framework for unsupervised anomaly detection : Detecting intrusions in unlabeled data. In D. Barbar and S. Jajodia (Eds.), Data Mining for Security Applications. Boston : Kluwer Academic Publishers, 2002.

[126] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, In Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining, pp. 226–231,1996.

[127] B.S. Everit & D. J. Hand Finite Mixture Distributions, Monographs on Statistics & Applied Probability,Chapman and Hall,1981.

[128] A. Fahad, N. Alshatri, Z. Tari, Member, A. Alamri, I. Khalil & A. Zomaya, A Survey of Clustering Algorithms for Big Data :Taxonomy & Empirical Analysis, IEEE Transactions on Emerging Topics in Computing, Vol. 2(3). pp.267–279, 2014.

[129] K. M. Faraoun, & A Boukelif, Securing network trafic using geneticaly evolved transformations. Malaysian Journal of Computer Science. Vol. 19(01), 2006.

[130] A. Farcomeni, L. Greco, Robust methods for data reduction, Taylor & Francis Group, LLC, 2015.

[131] M.D. Farid, N. Harbi, M.Z. Rahman, Combining Naïve Bayes and Decision Tree for Adaptive Intrusion Detection, International Journal of Network Security & Its Applications (IJNSA),Vol. 2(2), 2010.

[132] M. D. Farid, J. Darmont, & M.R. Zahidur, Attribute Weighting with Adaptive NBTree for Reducing False Positives in Intrusion Detection, International Journal of Computer Science and Information Security (IJCSIS), Vol. 8 (1), pp. 19-26, 2010.

[133] N. Farnaaz, M.A. Jabbar, Random Forest Modeling for Network Intrusion Detection System, Twelfth International Multi-Conference on Information Processing (IMCIP-2016), Procedia Computer Science, Vol. 89, pp. 213–217, 2016.

[134] U.M. Fayyad ; G. Piatetsky-Shapiro ; and P. Smyth. knowledge discovery and data mining : Towards a unifying Framework, In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Menlo Park, CA : AAAI Press, pp.82-88, 1996.

[135] S. Fenet, & S. Hassas, A distributed intrusion detection and response system based on mobile autonomous agents using social insects communication paradigm, In : Proceedings of the First International Workshop on Security of Mobile Multiagent Systems (SEMAS), pp. 41-58, 2001.

[136] G. Feng, O.L. Mangasarian, Semi-supervised support vector machines for unlabeled data classification, Optimization Methods Software, Vol.15, pp. 29–44, 2001.

[137] Y. Feng, Z. Wu, K. Wu, Z. Xiong, & Y. Zhou. An unsupervised anomaly intrusion detection algorithm based on swarm intelligence. In Proceedings of the International Conference on Machine Learning and Cybernetics, Vol. 7, pp. 3965–3969, 2005.

[138] Y. Feng, J. Zhong, C. Ye, and Z. Wu. Clustering based on self organizing ant colony networks with application to intrusion detection. In Proceedings of 6th International Conference on Intelligent Systems Design and Applications (ISDA '06), Vol. 6, pp. 3871–3875, 2006.

[139] Y. Feng, J. Zhong, Z. Xiong, C. xiao Ye, and K. gui Wu. Network anomaly detection based on dsom and aco clustering. In Advances in Neural Networks (ISNN 2007), Vol. 4492 of Lecture Notes in Computer Science, pp. 947–955. Springer Berlin / Heidelberg, 2007.

[140] R. A. Fisher, The use of multiple measurements in taxonomic problems. Annals Eugen., Vol. 7, pp. 179-188, 1936.

[141] D. Floreano & C. Mattiussi Bio-Inspired Artificial Intelligence Theories, Methods, and Technologies, Cambridge, Massachusetts, MIT Press, 2008.

[142] J. J. Flores, F. Calderon, A. Antolino,& J. M. Garcia, Network anomaly detection by continuous hidden markov models : An evolutionary programming approach, Intelligent Data Analysis Journal, Vol. 19(02), pp. 391-412, 2015.

[143] T.C. Fogarty, Varying the probability of mutation in the genetic algorithm. In Schaffer, J.D. (Ed.) Proceedings of the third Internation Conference on Genetic Algorithms, Morgan Kaufmann, Los ALtos, CA, pp. 104-109, 1989.

[144] N. Foukia, IDReAM : Intrusion Detection and Response executed with Agent Mobility, In : Proceedings of The International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05), pp. 264-270, 2005.

[145] S. Forrest, R. Smith, B. Javornik, & A. Perelson. Using genetic algorithms to explore pattern recognition in the immune system. Evolutionary Computation, Vol. 1(3), pp. 191–211, MIT Press, Cambridge, 1993.

[146] S. Forrest, A. S. Perelson, L. Allen, R. Cherukuri, Self-nonself discrimination in a computer, In Proceedings of the IEEE Symposium on Research in Security and Privacy, pp. 202-212, 1994.

[147] S. Forrest, S. A. Hofmeyr, A Somayaji, T. A. Longstaff, A sense of self for Unix processes, In Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, pp. 120-128, 1996.

[148] J. Frank, Artificial intelligence and intrusion detection : Current and future directions. In Proc. of the 17th National Computer Security Conference, Baltimore, MD. National Institute of Standards and Technology(NIST), 1994.

[149] L.Gacôgne, Éléments de logique floue, 1re édition, Hermès-Lavoisier, 1997.

[150] D.P. Gaikwad, R. C. Thool, Intrusion Detection System Using Bagging with Partial Decision TreeBase Classifier, In Proceedings of 4th International Conference on Advances in Computing, Communication and Control (ICAC3'15), Procedia Computer Science, Vol. 49, pp. 92-98, 2015.

[151] J. A. Gámez, J. L. Mateo, J. M. Puerta, Learning Bayesian networks by hill climbing : efficient methods based on progressive restriction of the neighborhood, Journal of Data Mining and Knowledge Discovery, Vol. 22(1), pp. 106-148, 2011.

[152] P. K. Ganesh & D.Devaraj, Intrusion detection using artificial neural network with reduced input features, ICTACT Journal on soft computing, Vol. 1(1), pp. 30-36, 2010.

[153] A. A. Ghorbani, W. Lu, M. Tavallaee, Network Intrusion Detection and Prevention : Concepts and Techniques, Advances in Information Security, Vol.47, Springer, 2010.

[154] A. Ghosh, & A. Schwartzbard, A Study in Using Neural Networks for Anomaly and Misuse Detection, USENIX Security Symp, Washington, D.C, USA, 23- 26 August 1999.

[155] A. Ghosh, A. Schwartzbard, & M. Schatz, Learning Program Behavior Profiles for Intrusion Detection, In Proceedings of the Workshop on Intrusion Detection and Network Monitoring, pp. 51-62, 1999

[156] J. W. de Godoy Stênico & L. L. Ling, Network Traffic Monitoring and Analysis, In. The State of the Art in Intrusion Prevention and Detection, Detection, Edited by Al-Sakib Khan Pathan, pp.23-46, Auerbach Publications, 2014.

[157] S. Goil, H. Nagesh & A. Choudhary, MAFIA : Efficient and scalable subspace clustering for very large data sets, Technical Report No. CPDC-TR-9906-010, Center for Parallel and Distributed Computing, Department of Electrical & Computer Engineering, Technological Institute, Northwestern University, 1999.

[158] B. G. Goodarzi, H. Jazayeri, S. Fateri, Intrusion Detection System in Computer Network Using, Hybrid Algorithms (SVM and ABC), Journal of Advances in Computer Research, Vol. 5(4), pp. 43-52, 2014.

[159] N. Gornitz, M. Braun, M. Kloft, Hidden Markov Anomaly Detection,In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015.

[160] Z. Gou, M. A. B. Ahmadon, S. Yamaguchi & B. B. Gupta, A Petri Net-based Framework of Intrusion Detection Systems IEEE 4^{th} Global Conference on Consumer Electronics (GCCE), Osaka, pp. 579-583, 2015.

[161] P. Grabusts & A. Borisov, Using grid-clustering methods in data classification. In PARELEC '02. Proceedings. International Conference on Parallel Computing in Electrical Engineering, pp. 425–426, 2002.

[162] R. Greiner, W. Zhou, X. Su, & B. Shen, Structural extension to logistic regression : Discriminative parameter learning of belief net classifiers, Machine Learning, Vol. 59(3), pp. 297–322, 2005.

[163] C. Grosan, A. Abraham, & M. Chis, Swarm Intelligence in Data Mining, Studies in Computational Intelligence (SCI), Vol. 34, pp. 1–20, 2006.

[164] Y. Gu, A. McCallum, D. Towsley, Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation, In Internet Measurement Conference, 2005.

[165] M. Gupta, S. K. Shrivastava, Intrusion Detection System based on SVM and Bee Colony, International Journal of Computer Applications, Vol. 111(10), pp. 27-32, 2015.

[166] G. Guojun, Data Clustering in C++ An Object-Oriented Approach, Data Mining and Knowledge Discovery Series, Chapman & Hall/CRC, 2011.

[167] G. Guojun, C. Ma & J. Wu, Data clustering : Therory, Algorithms and Applivations, ASA-SIAM Series on Statistics and Applied Probability, publised by SIAM, Society for Industrial and Applied Mathematics, 2007.

[168] J. Han, M. Kamber, & A.K.H. Tung, Spatial Clustering Methods in Data Mining : A Survey, In : Harvey J. Miller & Jawei Han (eds.), Geographic Data Mining and Knowledge Discovery. London, Taylor and Francis, 2001.

[169] J. Han ; M. Kamber & J. Pei Data Mining : Concepts and Techniques, 3^{rd} Edition, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publisher, 2012.

[170] L. K. Hansen, & P. Salamon, Neural Network Ensembles, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12(10), pp. 993-1001, 1990.

[171] P. K. Harmer, A distributed agent architecture of a computer virus immune system. Master's thesis, Air Force Institute of Technology, Air University, 2000.

[172] J. A. Hartigan, Clustering Algorithms, Wiley series in probability and mathematical statistics, John Wiley & Sons, Inc., 1975.

[173] Md. A Hasan, M. Nasser, Shamim Ahmad, Khademul Islam Molla, Feature Selection for Intrusion Detection Using Random Forest, Journal of Information Security, Vol. 7, pp. 129-140, 2016.

[174] J. Haweliya, & B. Nigam, Network Intrusion Detection using Semi Supervised Support Vector Machine, International Journal of Computer Applications , Vol.85(9), pp. 27-31, 2014.

[175] LT Heberlein, GV Dias, KN Levitt, B. Mukherjee, J.Wood, & D. Wolber, A network security monitor, In Proceedings of the Symposium on Research in Security and Privacy (Oakland, CA), pp. 296–304, 1990.

[176] I. M. Hegazy, T. Al-Arif, Z. T. Fayed, H. M., Faheem, A Multi-agent Based System for Intrusion Detection. IEEE Potentials journal Vol. 22(4), pp. 28–31, 2003.

[177] A. Hinneburg & D. A. Keim, Optimal grid-clustering : Towards breaking the curse of dimensionality in high-dimensional clustering, In Proceedings of the 25th International Conference on Very Large Databases(VLDB'99), pp. 506-517, 1999.

[178] Y. Ho, D. Frincke & D. Jr. Tobin, Planning, Petri Nets, and Intrusion Detection, In Proceedings of the 21st National Information Systems security Conference, Virginia, 2000.

[179] S. Hofmeyr & S. Forrest, Intrusion detection using sequences of system calls, Journal of Computer Security, Vol. 6, pp. 151–180, 1998.

[180] S. A. Hofmeyr & S. Forrest, Architecture for an artificial immune system, Computation Journal, Vol. 8(4), pp. 443–473, 2000.

[181] J. Holland, Adaptation In Natural And Artificial Systems, University of Michigan Press, 1975.

[182] K. Huang, I. King, & M. R. Lyu, Discriminative training of Bayesian Chow-Liu tree multinet classifiers, In Proceedings of the International Joint Conference on Neural Network, pp. 484–488, 2003.

[183] C.-T. Huang, S. Thareja, and Y.-J. Shin Wavelet-based Real Time Detection of Network Traffic Anomalies, International Journal of Network Security, Vol.6(3), pp. 309–320, 2008.

[184] E. Hüllermeier, Fuzzy methods in machine learning and data mining : Status and prospects, Fuzzy Set and Systems, Vol. 156(3), pp. 387-407, 2005.

[185] C. W. Hsu, C.-C. Chang, & C.-J.Lin, A practical guide to support vector classification, Technical report, version : 2016, National Taiwan University, `https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

[186] A. N. Huy & C. Deokjai, Application of Data Mining to Network Intrusion Detection : Classifier Selection Model. 11th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 399-408, 2008.

[187] 30. K. Hwang, M. Cai, Y. Chen, and M. Qin, Hybrid intrusion detection with weighted signature generation over anomalous internet episodes, IEEE Transactions on Dependable and Secure Computing (2007), 41–55.

[188] L. Hyafil, R. L. Rivest, Constructing optimal binary decision trees is NP-complete, Information Processing Letters, Vol.5(1), pp. 15-17, 1976.

[189] ISO/IEC 7498-2, Information Processing Systems—Open Systems Interconnection Reference Model—Part 2 : Security Architecture, 1989.

[190] ITU X.800, Security Architecture for Open Systems Interconnection for CCITT Applications, 1991.

[191] K. Jackson, D. DuBois, and C. Stallings, An expert system application for network intrusion detection, in Proceedings of the 14th National Computer Security Conference, 1991.

[192] A. K. Jain, R. C. Dubes, Algorithms for Clustering Data, Prentice Hall Advanced Reference Series, Computer Science,1988.

[193] C. Jain, & A. K. Saxena, General Study of Mobile Agent Based Intrusion Detection System (IDS), Journal of Computer and Communications,Vol. 4, pp. 93-98, 2016.

[194] H.S. Javitz, A. Valdez, T. Lunt, and M. Tyson, Next generation intrusion detection expert system (nides), Tech. Report SRI Technical Report A016, SRI International, 1993.

[195] E.T. Jaynes. Information theory and statistical mechanics. Physical Review Letters, 1963.

[196] T. Jebara, Machine learning discriminative and generative, published by Kluwer Academic Publishers, 2004.

[197] K. Jensen Coloured Petri Nets : Basic Concepts, Analysis Methods and Practical Use, Monographs in Theoretical Computer Science An EATCS Series, Vol. 3, Springer, 1997.

[198] F. V. Jensen, & T. D. Nielsen, Bayesian Networks and Decision Graphs,2nd edition, Information Science and Statistics Series, Springer Verlag, 2007.

[199] N. K. Jern, Towards a network theory of the immune system. Ann. Immunol. (Inst. Pasteur),125C :373–389, 1974.

[200] S.-Y. Ji, B.-K. Jeong, S. Choi, D. H. Jeong, A multi-level intrusion detection method for abnormal network behaviors, Journal of Network and Computer Applications Vol. 62, pp. 9-17, 2016.

[201] L. Jimenez & D. Landgrebe, High dimensional feature reduction via projection pursuit. School of Electrical and Computer Engineering, Purdue University, West Lafayette in 47907-1285, April 1995.

[202] G. H. John, R. Kohavi, and K. Peger. Irrelevant features and the subset selection problem, In Proceedings of the Eleventh International Conference on Machine Learning, pp. 121-129, 1994.

[203] I. T. Jolliffe, Principal Component Analysis, 2nd Edition, Springer Series in Statistics, Springer-Verlag, 488 pp., doi :10.1007/b98835, 2005.

[204] L. Jourdan, Méta-heuristique pour l'extraction de connaissance : application à la génomique, Thèse de doctorat de l'université des sciences et technologies de Lile, 2003.

[205] S. Juma, Z. Muda, M.A. Mohamed, Y. Warusia, machine learning techniques for intrusion detection system : a review, Journal of Theoretical and Applied Information Technology Vol.72(3), pp. 422-429, 2015.

[206] B. Kalaimathi,G. Annapoorani, N. Gayathri, Analysis of Intrusion Detection System Based on K-means Algorithm and Particle Swarm Optimization, International Journal of Engineering Research & Technology, Vol. 3(1), pp. 1857-1861, 2014.

[207] P. M. Kanade, & L. O. Hall, Fuzzy Ants as a Clustering Concept. In 22nd International Conference of the North American Fuzzy Information Processing Society, pp. 227-232, 2003.

[208] M. Kantardzic, Data Mining : Concepts, Models, Methods, and Algorithms, 2nd Edition, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2011.

[209] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[210] P. D. Karumanchi, Y. A. S. Prasad, N. Dhulipalla, Current Design Trends in Intrusion Detection System : A Review on Technologies Implemented by Researchers, International Journal of Computer Science Engineering and Technology (IJCSET), Vol. 2(04), pp. 1109-1112, 2012.

[211] J. Kennedy, R. Eberhart, Particle swarm optimization, In Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948, 1995.

[212] J. Kennedy, R. Eberhart, A discrete binary version of the particle swarm algorithm. In : Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 4104-4108, 1997.

[213] J. O. Kephart. A Biologically Inspired Immune System for Computers, In Artificial Life IV : Proceedings of the 4^{th} international workshop on the synthesis and simulation of living systems ,pp. 130-139, 1994.

[214] E. Kesavulu Reddy, Neural Network for Intrusion Detection and Its Applications, In Proceedings of the World Congress on Engineering, Vol. 2, 2013.

[215] L. Khan, M. Awad, M. & B. Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering, The VLDB Journal, Vol. 16, pp. 507-521,2007.

[216] A. Khobzaoui, M. Mesfioui, A.Yousfate, B. A. Bensaber, On Copulas-based Classification Method for Intrusion Detection, 5th IFIP International Conference on Computer Science and its application(CIIA'2015), 2015.

[217] A. Khobzaoui & A.Yousfate, Intrusion Detection with Multi-Connected Representation, Inernational Journal of Computer Network and Information Security, Vol8(1), pp. 35-42, 2016.

[218] J. Kim, & P. J. Bentley, An evaluation of negative selection in an artificial immune system for network intrusion detection, In Proceedings of the Genetice and Evolutionary Computation Conference (GECCO 2001), San Francisco, CA, 2001.

[219] G. Kitagawa, Introduction to Time Series Modeling, Monographs on Statistics and Applied Probability, Chapman and Hall/CRC, 2010.

[220] J. Kittler, Feature set search algorithms, Pattern Recognition and Signal Processing, Springer-Verlag, no 20, pp. 41-60, 1978

[221] S. Knerr, L. Personnaz, G. Dreyfus, Single layer learning revisited : A stepwise procedure for building and training a neural network, In Neurocomputing : Algorithms, architectures, and applications, F. Fogelman-Soulié, J. Hérault (eds.), NATO ASI Series, Vol. F68, pp. 41-50. Springer, Heidelberg, 1990.

[222] L. Koc, & A. D. Carswell, Network Intrusion Detection Using a HNB Binary Classifier, 17^{th} UKSIM-AMSS International Conference on Modelling and Simulation, pp. 81-85, 2015.

[223] T. Kohonen, Self-Organizing Maps. 3rd edition Springer, 2000.

[224] J. N. Kok, J. Koronacki R. Lopez de Mantaras, S.Matwin D. Mladenič & A. Skowron, Knowledge Discovery in Databases : PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Springer, 2007.

[225] C. Kolias, G. Kambourakis, M. Maragoudakis, Swarm intelligence in intrusion detection : A survey, Computers & Security journal, Vol. 30(8), pp. 625-642, 2011.

[226] I. Kononenko, Estimating Attributes, Analysis and Extensions of RELIEF, European Conference on Machine Learning, pp. 171–182, 1994

[227] V. B. Kosamkar, S. S. Chaudhari, Data Mining Algorithms for Intrusion Detection System : An Overview, International Conference in Recent Trends in Information Technology and Computer Science (ICRTITCS), 2012.

[228] M. Kudo & J. Sklansky, Comparaison of algorithms that select features for pattern classifiers, Pattern Recognition, Vol. 33(1), pp. 25–41, 2000.

[229] C. Kruegel,W. Robertson, F. Valeur, Bayesian event classification for intrusion detection, In Proceedings of the 19^{th} Annual Computer Security Applications Conference(ACSAC), pp. 14-23, 2003.

[230] C. Kruegel, F. Valeur, G. Vigna, Intrusion detection and correlation : Challenges and Solutions, Advances in Information Security, Vol 14, Springer, 2005.

[231] S. Kumar, & E. H. Spafford, An application of pattern matching in intrusion detection, Technical Report CSD-TR-94-013, Perdue University, 1994.

[232] S. Kumar & E. H. Spafford, A Pattern Matching Model for Misuse Intrusion Detection, In Proceedings of the 17^{th} National Computer Security Conference, pp. 11–21, 1994.

[233] S. Kumar & N. Sukumar, Multidensity Clustering Algorithm for Anomaly Detection Using KDD'99 Data set. A. Abraham et al. (Eds.) Springer, ACC 2011, Part I, CCIS 190, pp. 619-630, 2011.

[234] K. Labib & R. Vemuri, NSOM : a real-time Intrusion Detection System Using Self-Organizing Maps, Technical report, Dept. of Applied Science, University of California, Davis, 2002.

[235] C. Langin, &S. Rahimi, Soft computing in intrusion detection : the state of the art, Journal of Ambient Intelligence and Humanized Computing, Vol. 1(02), pp. 133–145, 2010.

[236] P. Laplace, Théorie Analytique des Probabilités, 3e M^{ME} V^{E} Courcier, Paris, 1820.

[237] D. T. Larose, Discovering knowledge in data : An Introduction to Data Mining, A JOHN WILEY & SONS, INC., PUBLICATION, 2005.

[238] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava, & V. Kumar, A comparative study of anomaly detection schemes in network intrusion detection. In Proceding of the 3rd SIAM Conference on Data Mining, 2003.

[239] W. Lee, & S. Stolfo, Data mining approaches for intrusion detection, In proceedings of the seventh USENIX security symposium (SECURITY'98). San Antonio, TX., 1998.

[240] W. Lee, S. J. Stolfo, A Framework for Constructing Features and Models for Intrusion Detection Systems, ACM Transactions on Information and System Security Vol. 3(4),pp. 227–261 2000.

[241] W. Lee, S. Stolfo, K. Mok, Adaptive intrusion detection : a datamining approach. Journal of Artificial Intelligence Review, Vol. 14(6), pp. 533–567 , 2000

[242] I.C. Lerman, T. Chantrel, Classification et analyse ordinale des données. Dunod, 1981.

[243] I. Levin, KDD-99 Classifier Learning Contest LLSoft's Results Overview, ACM SIGKDD Explorations Newsletter, Vol. 1(2), pp. 67-75, 2000.

[244] D. Li & J. S. Deogun, Applications of Fuzzy and Rough Set Theory in Data Mining, In Studies in Computational Intelligence, Vol. 255, pp. : 71-113, 2009.

[245] Y. Li, N. Wu, S. Jajodia, & X. S. Wang, Enhancing profiles for anomaly detection using time granularities, Journal of Computer Security archive, Vol. 10(1-2), pp. 137-157, 2002 .

[246] W. Li Using Genetic Algorithm for Network Intrusion Detection, Proceedings of the United States Department of Energy Cyber Security Group 2004 Training Conference, May 24-27, 2004, Kansas City, Kansas, USA.

[247] Y. Li, G. Yang, J. Xu,& B. Zhao, Anomaly detection for clustering algorithm based on particle swarm optimization, Journal of Jiangsu University of Science and Technology(Natural Science Edition), 2009.

[248] Z. Li, Y. Li, & L. Xu, Anomaly Intrusion Detection Method Based on K-Means Clustering Algorithm with Particle Swarm Optimization, Proceedings of the 2011 International Conference of Information Technology, Computer Engineering and Management Sciences, Vol. 2, pp. 157-161, 2011.

[249] W. K. Liao, Y Liu & A. Choudhary. A grid-based clustering algorithm using adaptive mesh refinement, In 7th Workshop on mining Scientific and Engineering Dataset of SIAM International Conference on Data Ming, 2004.

[250] C. Llorens, L. Levier, D. Valois & B. Morin, Tableaux de bord de la sécurité réseau, 3e édition, ÉDITIONS EYROLLES, 2010.

[251] N. P. Lin, Chung-I Chang, Nien-yi Jan, Hao-En Chueh, Hung-Jen Chen, & Wei-Hua Hao, An axis-shifted crossover-imaged clustering algorithm. WSEAS TRANSACTIONS on SYSTEMS, Vol. 7(3), pp. 175–184, 2008.

[252] H. Liu & H. Motoda Feature extraction, construction And selection : a data mining perspective, The kluwer international series In engineering and computer science, Springer Science+Business Media New York, 1998.

[253] H. Liu & H. Motoda, Computational Methods of Feature Selection, Chapman & Hall/CRC data mining and knowledge discovery series, Chapman & Hall/CRC, 2008.

[254] L. Liu, Y. Liu, MQPSO based on wavelet neural network for network anomaly detection, In : Proceedings of the 5^{th} International Conference on Wireless Communications, Networking and Mobile Computing (WiCom '09), pp. 1-5, 2009.

[255] Y. Liu, R. Ma, X. Lin, Network anomal detection wavelet neural network based on QPSO. Journal of Liaoning Technical University(Natural Science), 2009.

[256] C. Liu, J. Yang, Y. Zhang, R. Chen, & J. Zeng, Research on immunity-based intrusion detection technology for the internet of things, In Proceedings of the International Conference on Natural Computation (ICNC), Vol. 1, pp. 212-216, 2011.

[257] C. Liu, Y. Zhang, Z. Cai, J. Yang, L. Peng, Artificial Immunity-based Security Response Model for the Internet of Things, Journal of Computers, Vol. 8(12), pp. 3111-3118, 2013.

[258] M. Lorr, Cluster Analysis for Social Scientists. The Jossey-Bass Social and Behavioral Science Series. Jossey-Bass, San Francisco,Washington, London, 1983.

[259] W. Lu, I. Traore, Detecting New Forms of Network Intrusion Using Genetic Programming, Computational Intelligence, Vol. 20(3), pp. 475-494, 2004.

[260] W. Lu, & A. A. Ghorbani, Network Anomaly Detection Based on Wavelet Analysis, Journal on Advances in Signal Processing, 2008.

[261] E. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants, In Proceedings of the 3rd International Conference on Simulation of Adaptive Behaviour : From Animals to Animats, Vol. 3, pp. 599–508, 1994.

[262] T. F. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, P. G. Neumann H. S. Javitz, A. Valdes, and T. D. Garvey, A real time intrusion detection expert system (ides), Tech. report, SRI International, Menlo Park, CA, February 1992.

[263] J. Luo, Integrating fuzzy logic with data mining methods for intrusion detection, Master's thesis, Mississippi State Univ, 1999. ******

[264] M. Luo ; X. Li ; S. Xie, An Intrusion Detection Research Based on Spectral Clustering, 4th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1-4, 2008.

[265] J. Ma, X. Liu, S. Liu, A new intrusion detection method based on BPSO-SVM. In : Proceedings of the International Symposium on Computational Intelligence and Design, pp. 473-477, 2008

[266] R.,N, MABROUKEH & C. I. EZEIFE, A Taxonomy of Sequential Pattern Mining Algorithms, ACM Computing Surveys, Vol. 43(1), Article 3, November 2010.

[267] J. B MacQueen, Some methods for classification and analysis of multivariate observations. In : Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, 1967.

[268] O. Maimoun, L. Rokach, Data Mining and Knowledge Discovery Handbook, Second Edition, Springer-Verlag, 2010.

[269] E. H. Mamdani & S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Intern. I. of Man-Machine Studies, Vol. 7(1), pp. 1-13, 1975.

[270] H., Mannila ; P., Smyth ; & J.,D., Hand, Principles of Data Mining, MIT Press, 2001.

[271] A. Marshall, Copulas, marginals and joint distributions, in Distributions with Fixed Marginals and Related Topics, ed. by L. Rüschendorff, B. Schweizer, and M. Taylor, Institute of Mathematical Statistics, Hayward, CA. pp. 213-222, 1996.

[272] T. Marill & D. M. Green, On the effectiveness of receptors in recognition systems. IEEE transactions on Information Theory, Vol.9, pp. 11-17, 1963.

[273] J. A. Marin, D. Ragsdale, & J. Surdu, A hybrid approach to profile creation and intrusion detection. In Proc. of DARPA Information Survivability Conference and Exposition, Anaheim, CA. IEEE Computer Society, 2001.

[274] S. Masarat, S. Sharifian & H. Taheri, Modified parallel random forest for intrusion detection systems, The Journal of Supercomputing, Vol. 72(6), pp. 2235–2258, 2016.

[275] P. Matzinger, The danger model : A renewed sense of self, Science Vol. 296, pp. 301-305, 2002.

[276] Mayor, G. Suñer, J. and Torrens, J. : Sklar's theorem in finite settings. IEEE Transactions on Fuzzy Systems. Vol. 15(3), pp. 410-416, 2007.

[277] Fabrice MAZEROLLE, Statistique descriptive, Gualino éditeur, EJA – Paris – 2006.

[278] G. J. McLachlan, Classification and mixture ML approaches to cluster analysis, Handbook of Statistics, Vol. 2, pp. 199-208, 1982.

[279] L. Mè & V. Alanou, Détection d'intrusion dans un système informatique : méthodes et outils, TSI, Revue des sciences et technologies de l'information, Vol. 15(4), pp. 429-450, 1996.

[280] L. Me, GASSATA : A genetic algorithm as an alternative tool for security audit trails analysis. In Proc. of the International Symposium on Recent Advances in Intrusion Detection, 1998.

[281] A. Melek, 2004 Global Security Survey, Deloitte & Touche, 2004

[282] D. Michie, D. J. Spiegelhalter, C. C., Tayler, Machine Learning, Neural and Statistical Classification. Ellis Horwood Series in Artificial Intelligence (Upper Saddle River, NJ : Prentice Hall), 1994.

[283] E. Michailidis, S.K. Katsikas, E. Georgopoulos, Intrusion detection using evolutionary neural networks, In : Proceedings of the Panhellenic conference on informatics 2008 (PCI 2008), pp. 8-12, 2008.

[284] T. Mikami, & M. Wada M, Data visualization method for growing self-organizing networks with ant clustering algorithm, In 6th European Conference on Artificial Life (ECAL2001), pp. 623-626, 2001.

[285] B. P. Miller, D. Koski, C. P. Lee, V. Maganty, R. Murthy, A. Natarajan & J. Steidl, Fuzz Revisited : A Re-examination of the Reliability of UNIX Utilities and Services, Computer Sciences Department, University of Wisconsin, 1995.

[286] P. Miller, A. Inoue, Collaborative Intrusion Detection System, In 22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS 2003), pp. 519–524, 2003.

[287] B. Mirkin, Mathematical Classification and Clustering, Series : Non convex Optimization and Its Applications, 11e édition, 1996.

[288] T.M. Mitchell, Machine Learning. McGraw–Hill, 1997.

[289] N. Monmarché, & M. Slimane, & G. Venturini, AntClass : discovery of clusters in numeric data by an hybridization of an ant colony with the Kmeans algorithm, Technical Report Num 213, Laboratoire d'informatique, E3i, University of Tours 1999.

[290] V. H. Moraveji, Z. Muda & W. Yassin, Improving Intrusion Detection Using Genetic Algorithm, Information Technology Journal, Vol. 12(11), pp. 2167-2173, 2013.

[291] B. A. Mozzaquatro, R. P. de Azevedo, R. C. Nunes, A. de Jesus Kozakevicius, Anomaly-based Techniques for Web Attacks Detection, Journal of Applied Computing Research, Vol.1(2), pp. 111-120, 2011

[292] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, B., Schölkopf, An Introduction to Kernel-Based Learning Algorithms. IEEE Transactions on Neural Networks. Vol.(12), pp.181-201, 2001.

[293] S. Mukkamala, A. H. Sung, & A. Abraham, Identifying key variables for intrusion detection using soft computing, In Proceedings of 15th International Conference on Computer Communications, 2002.

[294] S. Mukkamala, A. H. Sung, & A. Abraham, Intrusion detection using an ensemble of intelligent paradigms, Journal of Network and Computer Applications, Vol. 28(2), pp. 167-182, 2005.

[295] S. Mukkamala, A. H. Sung, A Abraham, Hybrid Multi-agent Framework for Detection of Stealthy Probes, Applied Soft Computing, Vol. 7(3), pp. 631–641, 2007.

[296] K. P. Murphy, Machine Learning : A Probabilistic Perspective Massachusetts Institute of Technology press, 2012.

[297] P. Naïm, P.H. Wuillemin, Ph. Leray, O. Pourret, & A. Becker, Réseaux bayésiens, 3e édition, 2007.

[298] R. Nelsen,An Introduction to Copulas Second Edition, Springer, New York, 2006.

[299] H.A. Nguyen and D. Choi. « Application of Data Mining to Network Intrusion Detection : Classifier Selection Model. » Y. Ma, D.Choi, and S. Ata (Eds.), Springer : APNOMS 2008. LNCS 5297, pp. 399-408. ,2000.

[300] N. Nobelis, Un modèle de case-based reasoning pour la détection d'intrusion, Rapport de stage DEA réseau et Système Distribué, université Nice Sophia, France,2004.

[301] K. Noreen , B. S. Belhaouari A. Azween, A. Iftikhar, M. Hussain, A Review of Classification Approaches Using Support Vector Machine in Intrusion Detection, In Proceedings of Informatics Engineering and Information Science : International Conference(ICIEIS), pp. 24-34, Part III, 2011.

[302] S. Northcutt, J. Novak, D. McLachlan, Network Intrusion Detection : An Analyst's Handbook, 2nd edition, New Riders Publishing, 2000.

[303] A. K. Palit and D. Popovic Computational Intelligence in Time Series Forecasting : Theory and Engineering Applications, Springer-Verlag London,2005.

[304] M. Parimala, D. Lopez & N. C. Senthilkumar, A Survey on Density Based Clustering Algorithms for Mining Large Spatial Databases, In International Journal of Advanced Science and Technology, Volume-31, pp. 59-66, 2011.

[305] P. G. Majeed & S. Kumar, Genetic Algorithms in Intrusion Detection Systems : A Survey ,International Journal of Innovation and Applied Studies Vol. 5(3), pp. 233-240, 2014.

[306] R. Patel, A. Thakkar, A. Ganatra, A Survey and Comparative Analysis of Data Mining Techniques for Network Intrusion Detection Systems, International Journal of Soft Computing and Engineering, Vol. 2(1), pp. 265-271, 2012.

[307] P. Pawar,& D. Tiwari, Intrusion Detection System based on Particle Swarm Optimized Neural Network, International Journal of Digital Application & Contemporary Research, Vol. 4(11), 2016.

[308] J. Pearl, Probabilistic reasoning in intelligent systems : networks of plausible inference. Morgan Kaufmann Publishers Inc.,1988.

[309] K. Pearson, Contributions to the mathematical theory of evolution,Philosophical Transactions of the Royal Society of London,Vol.(185), pp. 71-110, 1894.

[310] S. Peddabachigari, A. Abraham & T. Juhnson, Intrusion Detection Systems Using Decision Trees and Support Vector Machines International Journal of Applied Science and Computations, USA, 2003.

[311] S. Peddabachigari, A. Abraham, C. Grosan, & J. Thomas, Modeling intrusion detection system using hybrid intelligent systems, Journal of Network and Computer Applications, Vol.30(1), pp. 114–132, 2007.

[312] J. Peng, C. Feng, and J. Rozenblit, A hybrid intrusion detection and visualization system, Proceedings of the 13th Annual IEEE International Symposium andWorkshop on Engineering of Computer Based Systems (ECBS'06), 2006, pp. 505–506.

[313] A. S. Perelson & G. F. Oster, Theoretical studies of clonal selection minimal antibody repertoire size and reliability of Self-Nonself discrimination, Journal of Theoretical Biology , Vol 81, pp. 645-670, 1979.

[314] M. S. Pervez & F. Md. Dewan, Literature Review of Feature Selection for Mining Tasks, International Journal of Computer Applications, Vol. 116(21),pp. 31-33, 2015.

[315] B. Pfahringer, Winning the KDD99 classification cup, bagged boosting, SIGKDD Explorations, Vol.1(2), pp. 65-66, 2000.

[316] J. Pieprzyk, T. Hardjono, J. Seberry, Fundamentals of Computer Security, Springer-Verlag Berlin Heidelberg New York, pp. 490-492, 2003.

[317] R. Di Pietro, L. V. Mancini, Intrusion Detection Systems, Springer Publishing Company, Incorporated, 2008.

[318] J. Pittman, Adaptive Splines and Genetic Algorithms. Journal of Computational & Graphical Statistics, Vol. 11(3), pp. 615-638, 2002.

[319] R. Polikar, Ensemble learning, In Ensemble Machine Learning : Methods and Application edited by Zhang Cha and Ma Yunqian, pp. 1-34, Springer, 2012.

[320] L. Portnoy, E. Eskin, & S. Stolfo, Intrusion detection with unlabeled data using clustering, In Proceedings of the ACM Workshop on Data Mining Applied to Security, 2001.

[321] T. H. Ptacek , T. N. Newsham, Insertion, evasion, and denial of service : Eluding network intrusion detection, Technical report, Secure Networks Inc., Alberta, Canada, T2R-OY6, January 1998.

[322] J. Pu, L. Xiao, Y. Li, X. Dong A Detection Method of Network Intrusion Based on SVM and Ant Colony Algorithm, National Conference on Information Technology and Computer Science (CITCS 2012), 2012.

[323] A. K. Pujari, Data Mining Techniques, Universities Press India, 2001.

[324] S. L. Pundir & Amrita, Feature selection using random forest in intrusion detection system, International Journal of Advances in Engineering & Technology, Vol. 6(3), pp. 1319-1324, 2013.

[325] A. Özgür & H. Erdem, A Review of KDD99 Dataset Usage in Intrusion Detection and Machine Learning between 2010 and 2015, PeerJ PrePrints journal, Vol.04, 2016.

[326] T. Özyera, R. Alhajja,& K. Barkera, Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening, Journal of Network and Computer Applications, Vol. 30(01), pp. 99–113, 2007.

[327] Q. Qian, J. Cai, & R. Zhang, Intrusion detection based on neural networks and Artificial Bee Colony algorithm, 13^{th},In International Conference on Computer and Information Science, pp. 257-262, 2014.

[328] C. Qiu, & J. Shan, Research on Intrusion Detection Algorithm Based on BP Neural Network, International Journal of Security and Its Applications, Vol. 9(4), pp. 247-258, 2015.

[329] J. D. Queiroz, L. F. R. da Costa Carmo, and L. Pirmez, Micael : An autonomous mobile agent system to protect new generation networked applications. In 2nd Annual Workshop on Recent Advances in Intrusion Detection, 1999.

[330] A. Quintela-del-Río, Estévez-Pérez, G. : Nonparametric Kernel Distribution Function Estimator with kerdiest : An R Package for Bandwidth Choice and Applications. Journal of Statistical Software, Vol. 50(8), 2012.

[331] S. Ohta, R. Kurebayashi & K. Kobayashi, Minimizing false positives of a decision tree classifier for intrusion detection on the internet, Journal of Network and Systems Management, Vol. 16(4), pp. 399-419, 2008.

[332] R. Oppliger, Security Technologies for the World Wide Web, 2nd ed, Artech House, Computer Security Series, 2002.

[333] R Core Team. R : A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. `http://www.R-project.org/`, 2015.

[334] Rossiter, D. G. : Tutorial : Using the R Environment for Statistical Computing An example with the Mercer & Hall wheat yield dataset. University of Twente, Faculty of Geo-Information Science & Earth Observation (ITC) Enschede (NL), 2014.

[335] V. Ramos, & A. Abraham, ANTIDS : Self-organized ant-based clustering model for intrusion detection system. In The 4^{th} IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05), pp. 977–986, 2005.

[336] P. Ravi Kiran Varmaa, V. Valli Kumarib,& S. Srinivas Kumarc Feature Selection Using Relative Fuzzy Entropy and Ant Colony Optimization Applied to Real-time Intrusion Detection System, International Conference on Computational Modelling and Security (CMS 2016), Procedia Computer Science Vol. 85, pp. 503–510, 2016.

[337] S. Rawat, & C.-S. Sastry, Network Intrusion Detection Using Wavelet Analysis, 7^{th} International Conference on Information Technology, Hyderabad, India, 2004.

[338] R. Rawat , & Anurag Jain , Review : Boosting Classifiers For Intrusion Detection, International Journal of Scientific & Engineering Research, Vol. 4(7), pp.1-5, 2013.

[339] C.R. Reeves, A genetic algorithm for flowshop sequencing, Computers & Operations research, Vol. 22, pp. 5-13, 1995.

[340] R. R. Reddy, B.Kavya, Y. Ramadevi, A Survey on SVM Classifiers for Intrusion Detection, International Journal of Computer Applications, Vol. 98(19), pp. 38-44, 2014.

[341] , C. Reynès Etude des Algorithmes génétiques et application aux données de protéomique, Life Sciences. Université Montpellier I, 2007, `https://tel.archives-ouvertes.fr/tel-00268927`

[342] B. C. Rhodes, J. A. Mahaffey, & J. D. Cannady, Multiple Self-Organizing Maps for Intrusion Detection, NIST National Information Systems Security Conference, 2000.

[343] A.M. Riyad, M.S Irfan Ahmed, An Ensemble Classification Approach for Intrusion Detection, International Journal of Computer Applications, Vol. 80(2), pp. 37-42, 2013.

[344] D. Ruth & M. L. P. Felciah, A Survey on Intrusion Detection System with Data Mining Techniques, IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1(3), 2014.

[345] J. Ryan, M. Lin, and R. Miikkulainen, Intrusion Detection with Neural Networks, AI Approaches to Fraud Detection and Risk Management, Papers from the 1997 AAAI Workshop, Providence, RI, pp. 72-79, 1997.

[346] M. Sabhnani & G. Serpen, Application of Machine Learning Algorithms to KDD Intrusion Detection Data set within Misuse Detection Context. In Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications (MLMTA), pp. 209-215, 2003.

[347] J. Sander, Generalized Density-Based Clustering for Spatial Data Mining, Dissertation im Fach Informatik an der Fakultät für Mathematik und Informatik der Ludwig-Maximilians-Universität München, September 1998.

[348] J. Sander, M. Ester, M., H. P. Kriegel, X. Xu, Density-Based Clustering in Spatial Databases : The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery, Vol. 2(2), pp 169-194, Kluwer Academic Publishers, 1998.

[349] K. P. Sankar K. & M.Pabitra, Pattern Recognition Algorithms for Data Mining, Scalability, Knowledge Discovery and Soft Granular Computing, Machine Intelligence Unit Indian Statistical Institute Calcutta, India, CRC Press LLC, 2004.

[350] J. Santiago-Paz, & D. Torres-Roman, On Entropy in Network Traffic Anomaly Detection, 2nd International Conference on Entropy and its application, 2015.

[351] H. Saxena, V. Richaariya,Intrusion Detection in KDD99 Dataset using SVM-PSO and Feature Reduction with Information Gain, International Journal of Computer Applications, Vol. 98(6), pp. 25-29, 2014.

[352] H. Saxena, & V. Richariya, Intrusion Detection System using K- means, PSO with SVM Classifier : A Survey, International Journal of Emerging Technology and Advanced Engineering, Vol. 4(02), pp. 653-657, 2014.

[353] S. Schockaert, M. De Cock, C. Cornelis, & E. E. Kerre E. E, Efficient clustering with fuzzy ants, In : Applied Computational Intelligence, World Scientific Press, pp. 195-200, 2004.

[354] B. Schölkopf, C.J.C. Burges, & V. N. Vapnik, Extracting support data for a given task. In Proceedings of First International Conference on Knowledge Discovery and Data Mining, 1995.

[355] B. Schölkopf, A. J. Smola, Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond, The MIT Press, 2002.

[356] E. H. Spafford, & D. Zamboni, Intrusion Detection Using Autonomous Agents. Computer Networks : The International Journal of Computer and Telecommunications Networking Vol. 34(4), pp. 547–570, 2000.

[357] M. Sebring et al., Expert systems in intrusion detection : A case study, in Proceedings of the 11th National Computer Security Conference, pp. 74-81,1988.

[358] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, S. Zhou, "Specification-based Anomaly Detection : A New Approach for Detecting Network Intrusions", Proceedings of the 9th ACM conference on Computer and Communications Security (CCS) 2002, Washington, DC, USA, pp. 265 - 274, 2002.

[359] G. Seni,& J. F. Elder, Ensemble Methods in Data Mining : Improving Accuracy Through Combining Predictions (Synthesis Lectures on Data Mining and Knowledge Discovery), Morgan & Claypool, 2010.

[360] G. H. Shah, C.K. Bhensdadia, A. P. Ganatra, An Empirical Evaluation of Density-Based Clustering Techniques, In International Journal of Soft Computing and Engineering (IJSCE), Vol. 2(1), pp. ,2012.

[361] A. Shaik, K. R. Nageswara, & J.A. Chandulal, Intrusion Detection System Methodologies Based on Data Analysis, International Journal of Computer Applications , Vol. 5(2), 2010.

[362] R. E. Shapire, Y. Freund, P. Bartlett, & W. Lee, Boosting the margin : A new explanation for the effectiveness of voting methods, Annals of Statistics, Vol. 26(5), pp. 1651–1686, 1998.

[363] R. E. Schapire, The Strength of Weak Learnability, Machine Learning, Vol. 5(2), pp. 197-227, 1990.

[364] S. K. Sharma, M. Manoria, Intrusion Detection using Hidden Markov Model, International Journal of Computer Applications, Vol. 115(4), pp. 35-38, 2015.

[365] S. Sheen & R. Rajesh, Network Intrusion Detection using Feature Selection and Decision tree classifier, IEEE Region 10th Conference, TENCON08, pp. 1–4, 2008.

[366] B. W. Silverman, Density Estimation for Statistics and Data Analysis. Published in Monographs on Statistics and Applied Probability, London : Chapman and Hall, 1986.

[367] C. Sinclair, P. Lyn, & S. Matzner, An Application of Machine Learning to Network Intrusion Detection, In Proceedings of the 15th Annual Computer Security Applications Confrence(ACSAC'99), pp. 371-377, 1999.

[368] S., Singh & S. Kandula, Argus - a distributed network-intrusion detection system, Undergraduate Thesis, Indian Institute of Technology, 2001.

[369] S. Singh, S. Silakari, Generalized Discriminant Analysis algorithm for feature reduction in Cyber Attack Detection System, International Journal of Computer Science and Information Security, Vol. 6(1), pp. 173-180,2009.

[370] A. Sklar, Fonction de répartition à n dimensions et leurs marges. Publ. Inst. Statist. Univ. Paris, Vol. 8, pp. 229-231, 1999.

[371] snow : Simple Network of Workstations, `http://cran.r-project.org/web/packages/snow/index.html`

[372] snowfall : Easier cluster computing (based on snow), `http://cran.r-project.org/web/packages/snowfall/index.html`

[373] A.Somayaji, S.Forrest, S.Hofmeyr & T. Longstaff, A sense of self for Unix processes, In. IEEE Symposium on Security and Privacy, pp. 120–128, 1996.

[374] A. Somayaji, S. Hofmeyr, & S. Forrest, Principles of a Computer Immune System, In proceedings of the 1997 workshop on new security paradigms, pp. 75-82, ACM press, 1998.

[375] P. Sornsuwit, & S. Jaiyen, Intrusion detection model based on ensemble learning for U2R and R2L attacks, 7th International Conference on Information Technology and Electrical Engineering (ICITEE), pp. 354-359, 2015.

[376] R., Srikant R. & R. Agrawal, Mining Sequential Patterns : Generalizations and Performance Improvements, In P. M. G. Apers, M. Bouzeghoub, G. Gardarin (eds), EDBT, Vol. 1057 of Lecture Notes in Computer Science, Springer, pp. 3-17, 1996.

[377] U. Stańczyk & L. C. Jain Feature Selection for Data and Pattern Recognition, Studies in Computational Intelligence, Vol. 584, Springer-Verlag Berlin Heidelberg, 2015.

[378] S. Staniford, J. A. Hoagland, & J. M. McAlerney , Practical automated detection of stealthy portscans. Journal of Computer Security Vol. 10(1-2), pp. 105–136, 2002.

[379] G. Stein, B. Chen, A.S Wu, K. A Hua, Decision tree classifier for network intrusion detection with GA-based feature selection, In Proceedings of the 43rd annual southeast regional conference ACM, Vol. 2, pp 136–141, 2005.

[380] P. Stoica, R. L. Moses Introduction to Spectral Analysis, Prentice-Hall, 1997.

[381] S. J. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan, & P. K. Chan, JAM : Java agents for meta-learning over distributed databases, In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pp. 74–81, AAAI Press, 1997.

[382] D. W. Stroock, An Introduction to Markov Processes, Graduate Texts in Mathematics 230, Springer-Verlag Berlin Heidelberg,2004.

[383] M. Sugeno, Industrial applications of fuzzy control, Elsevier Science Pub. Co., 1985.

[384] I. Syarif, & Ed Zaluska, A. Prugel-Bennett, G. Wills, Application of Bagging, Boosting and Stacking to Intrusion Detection, Machine Learning and Data Mining in Pattern Recognition, Vol. 7376 of the series Lecture Notes in Computer Science, pp. 593-602, 2012.

[385] D. G. Terrell, D. W., Scott, D. W., Variable kernel density estimation. Annals of Statistics. Vol. 20(3), pp. 1236-1265, 1992.

[386] W. J. Tian, & J. C. Liu, Network intrusion detection analysis with neural network and particle swarm optimization algorithm, In proceedings of the Chinese IEEE Control and Decision Conference (CCDC 2010), pp. 1749-1752, 2010.

[387] E. Tombini, H. Debar, L. Me, M. Ducasse, F. Telecom, & F. Caen, A serial combination of anomaly and misuse IDSes applied to HTTP traffic, In proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04), pp. 428–437, 2004.

[388] A. N. Toosi & M. Kahani, A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Computer Communications. Vol. 30, pp. 2201-2212, 2007.

[389] M. Tulloc, Microsoft Encyclopedia of Security, Published by Microsoft Press a division of Microsoft Corporation, 2003.

[390] C.-H. Tsang, & S. Kwong, Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction, In IEEE International Conference on Industrial Technology (ICIT '05), pp. 51–56, 2005.

[391] C.-H. Tsang, & S. Kwong, Ant colony clustering and feature extraction for anomaly intrusion detection, In Swarm Intelligence in Data Mining, Vol. 34 of Studies in Computational Intelligence, pp. 101–123, 2006.

[392] B. Uppalaiah, K. Anand, B. Narsimha, S. waraj, T. Bharat, Genetic Algorithm Approach to Intrusion Detection System, IJCST Vol.3(1), pp. 156-160, 2012.

[393] D. Urbani, P. Roussel-Ragot, L. Personnaz, & G. Dreyfus, The selection of neural models of non-linear dynamical systems by statistical tests, In Neural Networks for Signal Processing, Proceedings of the 1994 IEEE Workshop, 1994.

[394] J. Utans, J. Moody, Selecting Neural Network Architectures via the Prediction Risk : Application to Corporate Bond Rating Prediction, In Proceedings of the 1st International Conference on Artificial Intelligence Applications on Wall Street, IEEE Computer Soceity Press, Los Alamitos,CA, 1991.

[395] H. Vaccaro and G. Liepins, Detection of anomalous computer session activity, in Proceedings of the 1989 Symposium on Security and Privacy, (Oakland, CA), pp. 280-289, 1989

[396] A. Valdes, K. Skinner, Adaptive Model-based Monitoring for Cyber Attack Detection, In Proceedings of Recent Advances in Intrusion Detection (RAID 2000), Toulouse, France, pp. 80-92, 2000.

[397] V. N. Vapnik, The nature of statistical learning theory, Springer-Verlag, New York, NY, USA, 1995.

[398] V. N. Vapnik, A. Ya. Chervonenkis, Uniform convergence of frequencies of occurence of events to their probabilities, Soviet Mathematics Doklady Dokl. Akad. Nauk SSSR Vol 9(4), pp. 915–918, 1968.

[399] K. K. Vasan,& B. Surendiran, Dimensionality reduction using Principal Component Analysis for network intrusion detection, Perspectives in Science, Vol. 8, pp. 510—512, 2016.

[400] S. Vasanthi and S. Chandrasekar, A Study of Buffer Overflow Attack Detection Using Artifical Immune System Based Danger Theory, International Journal of Soft Computing, Vol. 10(1), pp. 1-5, 2015.

[401] E. Vegard, Machine learning for network based intrusion detection, A thesis submitted in partial fulfilment of the requirements of Bournemouth University for the degree of Doctor of Philosophy, 2010.

[402] M. Vella, M. Roper, & S. Terzis, Danger Theory and Intrusion Detection :Possibilities and Limitations of the Analogy, Artificial immune system,9th International Conference, ICARIS 2010, pp. 276-289, 2010.

learning techniques, International Journal of Computer Applications, Vol. 78(16), pp. :30–37, 2013.

[403] Y. Wang Statistical Techniques for Network Security : Modern Statistically-Based Intrusion Detection and Protection, Information Science Reference (an imprint of IGI Global),2009.

[404] J. Wang, X. Hong, R.-R. Ren, T.-H. Li, A Real-time Intrusion Detection System Based on PSO-SVM, Proceedings of the 2009 International Workshop on Information Security and Application (IWISA 2009), pp. 319-321, 2009.

Expert Systems with Applications, Vol. 37(9), pp. 6225-6232, 2010.

[405] Ke-Wei Wang,& Su-Juan Qin, A hybrid approach for anomaly detection using K-means and PSO, 2nd International Conference on Electronics, Network and Computer Engineering (ICENCE 2016), pp. 821-826, 2016.

[406] C. Warrender, S. Forrest, & B. Pearlmutter Detecting Intrusions Using System Calls : Alternative Data Models, In Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, pp. 133-145, 1999.

[407] C. S. Warnekar & G. Krishna, A heuristic clustering algorithm using union of overlapping pattern-cells. Pattern Recognition, Vol. 11(2),pp. 85–93, 1979.

[408] G. V. Welland, Beyond Wavelets, In series Studies in Computational Mathematics 10, Elsevier, 2003.

[409] A. W. Whitney, A direct method of nonparametric measurement selection, IEEE Trans. Comput., Vol. 20, pp. 1100-1103.

[410] M. J. Wierman, An Introduction to the Mathematics of Uncertainty, Edtion ed. Omaha, Nebraska : Center for Mathematics of Uncertainty, Inc., 2010.

[411] P. D. Williams, K. P. Anchor, J. L. Bebo, G. H. Gunsch, & G. D. Lamont, CDIS : Towards a computer immune system for detecting network intrusions, In Proceedings of the 4th Int'l Symp., Recent Advances in Intrusion Detection 2001, pp. 117–133, SpringerVerlag, Berlin, 2001.

[412] M. M. Williamson, Biologically Inspired Approaches to Computer Security, Technical Report HPL-2002-131, HP Laboratories Bristol,2002.

[413] R. Wirth, H. Jochen, CRISP-DM : Towards a Standard Process Model for Data Mining, In proceedings of the 4th International Conference on the Practical Application of Knowledge Discovery and Data Mining, pp. 29-39, 2000.

[414] D. Wishart, Mode analysis : A generalization of nearest neighbor which reduces chaining effects, In proceedings of the Colloquium in Numerical Taxonomy, A. J. Cole, edition, pp. 282–319, Academic Press, 1969.

[415] I. H. WITTEN , F. EIBE & M. A. HALL, Data Mining : practical Machine Learning Tools and Techniques, 3rd Edition, Morgan Kaufman Publishers, 2011.

[416] P. Wojtaszczyk, A mathematical introduction to wavelets, In London Mathematical Society student texts 37, Cambridge University Press, 1997.

[417] L. Xiao, Z. Shao, & G. Liu, K-means Algorithm Based on Particle Swarm Optimization Algorithm for Anomaly Intrusion Detection, 6th World Congress on Intelligent Control and Automation, Vol.2, pp. 5854-5858, 2006.

[418] P. Yadav & D. Singh, A Parallel Support Vector Machine for Network Intrusion Detection System, International Journal of Computer Applications Vol. 75(13), pp. 11-14, 2013.

[419] Z. Yanbin, Network intrusion detection system based on artificial immune, international journal of security and its applications, Vol. 9(09), pp. 359-370, 2015.

[420] S. Yang, M. Wang, & J. Licheng, A quantum particle swarm optimization, In Proceedings of the Congress on Evolutionary Computation(CEC2004), pp. 320-324, 2004.

[421] D. Yang & H. Qi. A Network Intrusion Detection Method using Independent Component Analysis, In the 19th International Conference on Pattern Recognition, 2008.

[422] N. Ye, A Markov Chain Model of Temporal Behavior for Anomaly Detection, In Proceedings of the 2000 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, pp. 171-174, 2000.

[423] D.Y. Yeung, & C. Chow, Parzen-window Network Intrusion Detectors, In 16th International Conference on Pattern Recognition, Quebec, Canada, pp. 11-15, 2002.

[424] B. Yu, E. Byres, & C. Howey, Monitoring Controller's DNA Sequence For System Security, ISA Emerging Technologies Conference, Instrumentation Systems and Automation Society, 2001.

[425] H. Yu, J. Yang, J. Han, Classifying large data sets using SVM with hierarchical clusters. In Proceedings of the SIGKDD 2003, pp. 306–315, 2003.

[426] Z. Yu, & J. J. P. Tsai, An efficient intrusion detection system using a boosting-based learning algorithm, International Journal of Computer Applications in Technology archive, Vol. 27(04), pp. 223-231, 2006.

[427] Z. Yu & J. J. P. Tsai, Intrusion Detection A Machine Learning Approach, In Electrical and Computer Engineering Series, Vol. 3, Imperial College Press, London, 2011.

[428] L. A. Zadeh, Fuzzy sets, Information and Control journal, Vol. 8(3), pp. : 338–353, 1965.

[429] M. Zekri – L. Souici & Meslati, Immunological Approach for Intrusion Detection, ARIMA Journal, Vol. 17, pp. 221-240, 2014

[430] Q. Zhang, A. Benveniste, Wavelet networks, IEEE Transactions on Neural Networks, Vol. 3(6), pp. :889-898, 1992.

[431] H. Zhang, L. Jiang, & J. Su, Hidden Naive Bayes, In Proceedings of the 20th national conference on Artificial intelligence, Vol. 2, pp. 919-924, 2005

[432] J. Zhang and M. Zulkernine, A hybrid network intrusion detection technique using random forests, The First International Conference on Availability, Reliability and Security (ARES'06), pp. 262–269, 2006.

[433] J. Zhang, M. Zulkernine, A. Haque, Random-Forests-Based Network Intrusion Detection Systems, IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews), Col. 38(5), pp. 649 - 659, 2008.

[434] H. Zhao, Intrusion Detection Ensemble Algorithm based on Bagging and Neighborhood Rough Set, International Journal of Security and Its Applications, Vol.7(5), pp.193-204, 2013.

[435] G. M. Ziegler, Lectures on Polytopes,In Graduate Texts in Mathematics 152 series, Updated 7th Printing of the 1st edition, Springer Science+Business Media New York 1995, Corrected and updated printing 2007.

[436] F. J. Von Zuben & L. N. De Castro, aiNet : An artificial Immune Network for Data analysis, In Data Mining : A Heuristic Approach, H.A. Abbass, R.A.Sarker, C.S. Newlton(Eds.), Idea Group publishing, USA, Chapter XII, pp.213-259, 2001.