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

Université Ferhat Abbas - Sétif -1

Thèse
Présentée à la Faculté des Sciences
Département Informatique

Pour l'Obtention du Diplôme de

DOCTORAT EN SCIENCES

Option : INFORMATIQUE

Par :

Maza Sofiane

Thème

Un système de vérification et de validation de la sécurité et l'intégration évolutive adaptative de la protection dans les systèmes d'informations avancés.

Soutenu le : 14 /04/2019 devant le jury composé de :

Président	Abdallah Khababa	Professeur	Université Sétif -1-
Directeur de thèse	Mohamed Touahria	Professeur	Université Sétif -1-
Examinateur	Abdelkrim Amirat	Professeur	Université Souk Ahras
Examinateur	Abdelhak Boubetra	Professeur	Université BBA

REMERCIEMENTS

Je tiens à exprimer mes remerciements à qui ne m'a jamais laissé perdu dans ce monde, qui m'a donné la volonté, la santé, et la patience pour terminer ce mémoire.

J'adresse mes sincères remerciements et mes grandes gratitudes tout d'abord à mon directeur de thèse TOUAHRIA Mohamed, Professeur à l'Université Ferhat Abbas- Sétif -1, pour m'avoir proposé ce sujet et pour son aide, ses encouragements, ses observations avisées, et ses précieux conseils tout au long de ce travail.

Je remercie les membres de jury d'avoir accepté de juger cette thèse. Monsieur Abdallah Khababa, Professeur à l'université Sétif -1-, monsieur Abdelhak Boubetra, Professeur à l'université BBA, monsieur Abdelkrim Amirat, Professeur Université Souk Ahras.

Je tiens à remercier tous ceux qui m'ont aidé, de près ou de loin, à contribuer la réalisation de ce travail.

DÉDICACES

Je dédie ce modeste travail :

- A mes très chers parents.
- A ma femme.
- A mes frères et sœurs.
- A tous mes amis et toutes les personnes qui m'ont aidé à réaliser ce travail.
- A la mémoire de BENABID Abdelhak.
- A mes chers amis : Djaafer, Adlan, et BOUFERROUM Rayad.

Résumé

L'Internet et les nouvelles technologies des réseaux informatiques exigent à l'entreprise d'assurer un haut degré de la sécurité et de protection au niveau des systèmes. La sécurité est devenue une nécessité importante. De ce fait, différents outils sont intégrés, comme les systèmes de détection d'intrusion (IDS : Intrusion Detection System). IDS est un composant très important dans l'infrastructure de sécurité. De plus, l'objectif principal de l'IDS est de détecter les différentes attaques et d'assurer la capacité de découvrir les nouvelles attaques pour accompagner leurs évolutions.

Concernant le grand nombre de connexions et le grand débit de données sur Internet, IDS a des difficultés de détection. De plus, les attributs non pertinents et redondants influencent sur la qualité de l'IDS plus précisément sur le taux de détection et le coût de traitement.

La sélection d'attributs (FS : Feature Selection) est une technique importante, ce qui aide à améliorer les performances de détection. La dissertation propose deux contributions principales. Dans la première contribution, nous proposons une nouvelle taxonomie pour les algorithmes de sélection d'attributs dans les systèmes de détection d'intrusion. Nous fournissons une classification avec une étude comparative entre différentes contributions en fonction de leurs techniques et résultats.

La deuxième contribution propose un algorithme d'estimation distribuée multi-objectif pour la sélection d'attributs des systèmes de détection d'intrusion. En fait, un nouvel algorithme multi-objectif de sélection d'attributs appelé 'MOEDAFS' (Multi-Objective Estimation of Distribution Algorithm for Feature Selection).

Le MOEDAFS est basé sur EDA (Estimation of Distribution Algorithm) et Information Mutuelle (IM). EDA est utilisée pour explorer l'espace de recherche et MI est intégré comme un modèle probabiliste pour guider la recherche en modélisant la redondance et les relations de pertinence entre les attributs. Par conséquent, nous proposons quatre modèles probabilistes pour MOEDAFS. MOEDAFS sélectionne les meilleurs sous-ensembles d'attributs qui ont une meilleure précision de détection avec le minimum nombre d'attributs sélectionnés. MOEDAFS utilise deux fonctions objectives : la minimisation du taux d'erreur de classification (ER : Error Rate) et le nombre d'attributs sélectionnés (NF : Number of Feature).

Afin de démontrer la performance de MOEDAFS, on a introduit deux comparaisons : une comparaison interne et une comparaison externe sur un ensemble de données de détection d'intrusion NSL-KDD. La comparaison interne est effectuée entre les quatre versions de MOEDAFS. La comparaison externe est faite par rapport à certains algorithmes de sélection

d’attributs déterministes, des algorithmes méta-heuristique, et multi-objectifs bien connus en littérature. Les résultats expérimentaux démontrent que MOEDAFS en plus efficace que les autres algorithmes de sélection d’attributs soit en termes de précision de classification ou en termes de taux de réduction.

Mots clés : Sécurité des systèmes d’information avancés, Détection d’intrusion, Sélection d’attributs, Optimisation multi-objectif, Estimation distribuée, et Information mutuelle.

Abstract

The Internet and new technologies of computer networks require to the company a high degree of the security and protection at the level of the systems. Security has become an important necessity, for that different tools are integrated as intrusion detection systems (IDS). IDS is very important components of the security infrastructure in any security policy. Moreover, the main purpose of IDS is to detect the different attacks and ensure the ability to discover the novel one for accompanying the attack's evolution.

Regarding to the huge number of connections and the large flow of data on the Internet, IDS has a difficulty to detect attacks. Moreover, irrelevant and redundant features influence on the quality of IDS precisely on the detection rate and processing cost.

Feature Selection (FS) is the important technique, which gives the issue for enhancing the performance of detection. The dissertation proposes two main contributions. In the first contribution, we propose a new taxonomy for feature selection algorithms in intrusion detection systems. We provide a classification with a comparative study between different contribution according to their techniques and results.

The second contribution proposes a distributed multi-objective algorithm of the feature selection for intrusion detection systems. In fact, we propose a new multi-objective algorithm for feature selection called 'MOEDAFS' (Multi-Objective Estimation of Distribution Algorithms (EDA) for Feature Selection).

MOEDAFS is based on EDA and Mutual Information (MI). EDA is used to explore the search space and MI is integrated as a probabilistic model to guide research by modelling redundancy and relevancy relationships between attributes. Therefore, we propose four probabilistic models for MOEDAFS. MOEDAFS selects the best feature subsets that have better detection accuracy with fewer selected features. MOEDAFS uses two objective functions: the minimization of the classification error rate (ER) and the number of selected features (NF).

In order to demonstrate the performance of MOEDAFS, we used two comparisons: an internal comparison and an external comparison on NSL-KDD intrusion detection data. The internal comparison is performed between the four versions of MOEDAFS. The external comparison is made against well-known feature selection algorithms: Deterministic, Metaheuristic, Multi-objective algorithms. Experimental results demonstrate that MOEDAFS outperforms other feature selection algorithms either in terms of classification accuracy or in terms of reduction rates.

Key words: Advanced Information System Security, Intrusion Detection, Attribute Selection, Multi-Objective Optimization, Distributed Estimation Algorithms, Mutual Information.

الملخص

الإنترنت والتقنيات الحديثة لشبكات الكمبيوترتفرض على الشركات تأمين درجة عالية من الأمان والحماية على مستوى الأنظمة. فقد أصبح الأمن ضرورة هامة داخل جميع الشركات. تعد IDS من المكونات المهمة جدًا للبنية الأساسية للأمن في أي سياسة أمنية لأي شركة. علاوة على ذلك، فإن الهدف الرئيسي من IDS هو اكتشاف الهجمات المختلفة وضمان القدرة على اكتشاف الفيروسات الجديدة لمرافقة تطور الهجمات.

فيما يتعلق بالعدد الهائل من الاتصالات والتدفق الكبير للبيانات على الإنترنت، تواجه IDS صعوبة في اكتشاف الهجمات في الوقت الفعلي للهجوم. علاوة على ذلك، تؤثر السمات غير الضرورية والمكررة على جودة IDS تحديدًا على معدل الكشف وتكلفة المعالجة.

اختيار الميزة (FS) هو الأسلوب الأنسب، الذي يعطي حل للمشكلة ويؤدي الى تحسين أداء الكشف لذلك تقترح الأطروحة مساهمتان رئيسيتان. في أول مساهمة، نقترح تصنيفًا جديدًا لخوارزميات تحديد السمة في أنظمة كشف التطفل. نحن نقدم تصنيفًا مع دراسة مقارنة بين المساهمات المختلفة وفقًا لتقنياتها ونتائجها. تقترح مساهمتنا الثانية خوارزمية متعددة الأهداف موزعة لاختيار سمات أنظمة كشف التطفل. في الواقع، نقترح خوارزمية متعددة الأهداف جديدة لاختيار السمة تسمى "MOEDAFS" (تقييم متعدد الأهداف لخوارزميات التوزيع (EDA) لاختيار الميزة).

يستند MOEDAFS على خوارزمية EDA والمعلومات المتبادلة (MI). يستخدم EDA لاستكشاف مساحة البحث ويتم دمج MI كنموذج احتمالي لتوجيه البحث عن طريق نمذجة علاقات التكرار والصلة بين السمات. لذلك، نقترح أربعة نماذج احتمالية لـ MOEDAFS. يحدد MOEDAFS أفضل مجموعات فرعية من السمات التي لها دقة اكتشاف أفضل مع عدد أقل من السمات المحددة. يستخدم MOEDAFS وظيفتين موضوعيتين: تقليل معدل الخطأ التصنيف (ER) وعدد من الميزات المحددة.

من أجل إظهار أداء MOEDAFS، استخدمنا مقارنتين: مقارنة داخلية ومقارنة خارجية على مجموعة من بيانات كشف التسلل من NSL-KDD. يتم إجراء المقارنة الداخلية بين الإصدارات الأربعة من MOEDAFS. يتم إجراء المقارنة الخارجية ضد خوارزميات تحديد ميزة معروفة: الخوارزميات Deterministic، Metaheuristic ، و متعددة الأهداف. تظهر النتائج التجريبية أن أداء MOEDAFS يتفوق على خوارزميات اختيار الخواص الأخرى سواء من حيث دقة التصنيف أو من حيث معدلات التخفيض.

الكلمات الدالة: أمن نظام المعلومات المتقدم، كشف التسلل، اختيار السمة، التحسين متعدد الأهداف، التقدير الموزع، المعلومات المتبادلة.

Table des matières

Chapitre 01 ***: Introduction générale***

Chapitre 02 ***: La sécurité des systèmes d'information avancés***

Chapitre 03 ***: La sélection des attributs pour les systèmes de détection d'intrusion***

Chapitre 04 ***: Algorithme à éstimation de distribution et l'optimisation multi-objectif***

Chapitre 05 ***: Algorithme à éstimation de distribution multi-objectif pour la sélection d'attributs***

Table des figures

Liste des tableaux

Introduction générale

Introduction générale

1. Contexte du travail

L'Internet et la technologie des réseaux informatiques sont considérés comme un monde virtuel qui offre aux personnes et aux entreprises des opportunités d'exercer leurs activités en tant que services, ce qui donne l'émergence de nouveaux systèmes d'information avancés (tel que : E-Commerce, E-Business, E-Service…etc.) qui ont besoin de la sécurité.

En outre, Internet est un environnement qui contient un grand flux de données représentant la vie privée des personnes et les transactions financières d'une part, et un environnement distribué, hétérogène, non administré par une seule entité et ne garantit aucune sécurité d'une autre part. L'influence sur les données d'échange concernant la *Confidentialité*, l'*Intégrité* et la *Disponibilité* (CID) représentent un impact dangereux sur les réseaux et les systèmes.

Face aux attaques croissantes du réseau, différents outils de sécurité ont été développés pour protéger les systèmes contre les attaques telles que : le pare-feu, chiffrement, antivirus, mécanisme d'authentification, système de prévention des intrusions (IPS : Intrution Prevention systems) et système de détection d'intrusion (IDS : Intrution Detection systems).

La détection d'intrusion est un processus essentiel qui est utilisé pour surveiller le système contre les menaces. Il empêche toute intrusion de causer des dommages dans le système. IDS est un composant très important dans l'infrastructure de sécurité. IDS sécurise le système contre les menaces en détectant toutes les intrusions dans les réseaux (N-IDS : Network-IDS) et les hôtes (H-IDS : Host).

Cependant, l'objectif principal de l'IDS est de garder l'adaptabilité pour détecter de nouvelles attaques. IDS utilise l'approche de détection à signatures et d'anomalies. Le premier utilise les signatures pour trouver des attaques, le second utilise des modèles statistiques et intelligents (apprentissage automatique) pour découvrir le comportement normal et anormal de connexion.

Les méthodes ont été proposées pour construire IDS, basées sur la détection d'anomalies. Ces méthodes ont été fondées sur des techniques de classification intelligentes utilisant des algorithmes d'intelligence artificielle pour découvrir entre le comportement normal et anormal de connexion. Le modèle classificateur dans IDS garantit la détection de nouvelles attaques et donne l'aspect de calcul intelligent au processus de détection. Chaque modèle de classificateur a son modèle, sa précision de détection et son taux d'erreur.

2. Position du problème

Presque tous les classificateurs qui ont été développés pour les IDS souffrent de faibles taux de détection d'attaques (DR : Detction Rate) et de fausses alarmes élevées (FA : False Alarm). En outre, ils ont encore des problèmes de complexité dans l'architecture des classificateurs et des coûts de traitement.

De plus, le haut degré de *classification*, de *complexité*, de *temps de calcul* et de *stockage* influence sur la qualité et la performance de la détection [1] [2]. Aussi, l'amélioration de la performance du classificateur et la réduction des coûts de traitement restent un défi majeur dans la détection d'intrusion.

Pour cela, la réduction de dimensionnalité de l'ensemble de données donne l'opportunité d'améliorer l'efficacité de la détection et d'éviter les problèmes généraux des frais de la classification. Au cours des dernières années, parmi les processus d'optimisation réussis qui ont été utilisés pour résoudre ces problèmes, est la sélection d'attributs (FS : Feature Selection).

FS [3] [4] a été créé pour réduire la dimension des données en sélectionnant les meilleurs attributs sans redondance (redondance minimum) et une pertinence maximale avec la haute performance du taux de précision (AR : Accuracy Rate). Ces objectifs nous donnent une bonne compréhension des données, évitent les problèmes de sur justement, et permettent de sélectionner le meilleur sous-ensemble des attributs qui ont une relation très pertinente entre eux et la classe cible pour améliorer la performance des IDS.

- De plus, des travaux ont été proposés dans ce domaine de la sélection des attributs pour l'IDS afin de les éclairer est besoin de proposer une nouvelle vision correspondant aux défis actuels. Aussi, une nouvelle étude (survey) et une nouvelle classification sont plus essentielles pour extraire les différentes techniques qui les utilisent pour améliorer les tendances futures. Une enquête avec une nouvelle taxonomie est très importante et constitue un défi majeur.

- Plusieurs travaux ont été proposés pour sélectionner les meilleurs sous-ensembles d'attributs pour IDS, basés sur différentes techniques comme : les algorithmes déterministes et méta-heuristiques. De nouvelles approches ont été proposées pour en déduire les attributs importants pour l'IDS, mais le taux de faux positifs (False positif) est toujours supérieur au taux de détection. Ils souffrent également de la difficulté de sélectionner des sous-ensembles qui ont en même temps, un plus petit nombre d'attributs et un taux de détection plus élevé.

- Quand le nombre d'attributs dans l'ensemble de données est très grand, nous ne pouvons pas essayer tous les sous-ensembles possibles d'attributs en raison du grand espace de recherche

(2^n -1 sous-ensemble d'attributs ; n : nombre d'attributs), il correspond un problème d'optimisation NP-Hard. Ainsi, le but principal est de réduire le nombre d'attributs pour sélectionner le meilleur sous-ensemble qui a le nombre minimum d'attributs et un minimum taux d'erreur. Ces deux objectifs signifient que le FS devient un problème d'optimisation multi-objectif (MOO) car un seul critère ne peut pas bien évaluer les attributs sélectionnés pour tous la base de connaissance. Lorsque, la transformation de FS pour la détection d'intrusion vers un problème MOO, nous bénéficions de trouver plusieurs solutions non dominantes (Pareto Optimale Solutions) et chacune d'elles représente la solution optimale du grand espace de recherche.

- Garantir un processus de calcul intelligent pour le problème de la détection d'intrusion, il soit encore actuellement a besoin de plus de recherche sur les meilleurs sous-ensembles d'attributs. Pour cela, nous nous concentrons sur la sélection d'attributs dans la détection d'intrusion en tant que problème d'Optimisation Multi-Objectif (OMO). Il est très important de proposer un nouvel algorithme basé sur des multi-objectifs pour sélectionner les meilleurs sous-ensembles d'attributs avec des performances de détection élevées.

3. Contribution

Dans cette thèse, nous décrivons un aperçu de la plupart des techniques qui ont été proposées dans la recherche de la sélection des attributs pour les systèmes de détection d'intrusion en examinant les contributions existantes. Nous présentons aussi les derniers algorithmes FS bien connus pour l'IDS, développés pour sélectionner les meilleurs sous-ensembles d'attributs.

Par conséquent, une proposition d'une nouvelle taxonomie est primordiale pour comprendre les progrès de la recherche et identifier les tendances futures et les défis existants. Nous fournissons une mappe sur les recherches pour comprendre et construire l'état actuel de la sélection des attributs dans l'IDS par classification et étude comparative.

Un nouvel algorithme de sélection d'attributs est proposé pour la détection d'intrusion comme une nouvelle approche pour rechercher les meilleurs sous-ensembles d'attributs pour IDS. L'algorithme MOEDAFS (Multi-Objective Estimation of Distribution Algorithms (EDA) for Feature selection) est considéré comme une méthode de sélection d'attributs hybride basée sur l'algorithme évolutif multi-objectif EDA et l'information mutuelle (IM) pour bénéficier à la fois des méthodes de Filtre et Wrapper. Nous intégrons la méthode de filtrage MI dans EDA qui est basée sur le modèle Wrapper.

L'objectif principal de MOEDAFS est de sélectionner les meilleurs sous-ensembles d'attributs qui représentent les solutions non dominées (Pareto Optimal Solutions) pour la détection d'intrusion. Ces sous-ensembles d'attributs garantissent un taux de précision de classification

plus élevé et un plus petit nombre d'attributs. Pour cela, deux fonctions objectives sont utilisées, basées sur le taux d'erreur de classification (ER : Error Rete) et le nombre d'attributs (NF : Number of Feature). MOEDAFS est soutenu par un modèle probabiliste basé sur IM, IMC (Conditionnel MI), CO-Inf (Co-Information) et II (Interaction Information). Un modèle probabiliste guide la recherche MOEDAFS en estimant la probabilité de chaque attribut dans chaque itération.

Le modèle probabiliste représente et transforme les relations de pertinence de l'attribut f_i avec d'autres attributs f_j et les classes cibles C à une probabilité. Selon ces probabilités, MOEDAFS génère la nouvelle population candidatée pour une nouvelle itération.

Au total, quatre nouveaux modèles probabilistes pour MOEDAFS sont présentés. Chacun de ces modèles probabilistes à sa stratégie et son support mathématique. Cette combinaison donne à MOEDAFS la possibilité d'explorer tout l'espace de recherche avec un calcul intelligent et de sélectionner les solutions non dominées (meilleurs sous-ensembles d'attributs).

4. Organisation du mémoire

Cette thèse est organisée en cinq chapitres :

Le chapitre 2 introduit le contexte dans lequel se situent nos travaux de recherche. Il présente dans un premier temps la sécurité des systèmes d'information avancés, et les différents aspects, ses importances et ses difficultés. Il aborde dans un deuxième temps les mécanismes de la détection et de la sécurité.

Le chapitre 3 présente une vue globale sur les recherches et les approches proposées dans le domaine de la sélection des attributs pour les systèmes de détection d'intrusion. Une nouvelle taxonomie de classification est proposée pour les algorithmes de la sélection d'attributs pour les IDS. Le chapitre est achevé par l'introduction d'une nouvelle classification avec une étude comparative.

Le chapitre 4 présente les techniques utilisées dans l'algorithme MOEDAFS pour la sélection des attributs dans l'IDS. Ces techniques concernes l'algorithme d'estimation distribué (EDA : Estimation of Distribution Algorithms), les différents aspects de l'information mutuelle IM, IMC (Conditionnel MI), CO-Inf (Co-Information) et II (Interaction Information). Nous présentons les aspects de l'optimisation multi-objective (MOO), et les différents algorithmes d'élitiste connus pour le choix de Pareto set.

Le chapitre 5 est consacré à la deuxième proposition, celle de l'algorithme MOEDAFS. Nous présentons les étapes et les techniques utilisées. On exposera l'aspect réalisation de l'algorithme avec un ensemble d'interfaces et de tests. La discussion et critiques des résultats est

représentée avec une comparaison interne entre les versions de MOEDAFS et externe avec des différents algorithmes déterministes, méta-heuristique et multi-objectif existants.

La thèse termine par une conclusion générale et des perspectives future.

Chapitre 02

La sécurité des systèmes d'information avancés

1. Introduction

Durant ces dernières années, les gouvernements donnent une grande importance pour sécuriser leurs systèmes contre le piratage, causés par les individus ou les organisations ou même les autres gouvernements. Ce qui donne une autre dimension à la sécurité des systèmes. Les attaques contre les systèmes sont devenues une arme utilisée comme toutes les autres armes qui causent la paralysie dans les services de la société comme l'énergie, transport, et même les centres nucléaires avec un impact financière de millions de dollars.

Donc, la sécurité dans les entreprises devenu un besoin incontestable. Avec les systèmes d'information avancés (comme : E-commerce, E-busniss, E-service, et Big date…etc.) qui manipulent différentes transactions financières, la vie privée, et l'ensemble des données échangés entre les compagnies, oblige les concepteurs à intégrer la sécurité comme une étape dans le développement des systèmes ou plutôt d'être dans tous les cycles de développement. Par conséquent, l'amélioration des outils de sécurité soit de conception, ou de protection est inévitable et une responsabilité pour protéger les systèmes et les gouvernements.

Dans ce chapitre, nous décrivons les notions de l'ingénierie de la sécurité et l'ingénierie des besoins de sécurité. Ensuite, nous définissons les différentes propriétés de sécurité qui doivent être dans n'importe quel système. Après, nous allons décrire la sécurité des systèmes d'information avancés. Nous définissons les outils pratiques de protection qui apparaissent dans les mécanismes de sécurité (nous donnons les mécanismes optimaux, nécessaires pour protéger un système contre les différents risques). Nous situerons l'emplacement de ces outils de protection au sein d'un réseau afin d'aider à présenter l'architecture des réseaux sécurisés.

2. Ingénierie de sécurité

Au cours des années 2000, la définition de l'ingénierie de la sécurité souffre de la confusion dans la clarté de la terminologie des concepts et l'objectif global. En même temps les développeurs prennent la sécurité comme une dernière étape dans le développement des systèmes par l'intégration d'une couche de mécanisme de sécurité. Yeun-Hee [5] propose une définition selon les nouveaux besoins et défis comme : *"un ensemble de méthodologies et de technologies pour garantir le développement, et une exploitation rapide et à moindre coût (maintenance)*

des systèmes de sécurité[1] avec une haute qualité de fonctionnement en appliquant des technologies de cryptographie, des technologies de sécurité d'information[2] et du génie logiciel ".
La nouvelle vue de développement de la sécurité selon [5] [6] [7] [8] [9] [10] [11], est de prendre les aspects de sécurité dans toutes les étapes de cycle de vie du développement d'un système. C'est-à-dire de développer une variété de techniques pour assurer l'intégration de la sécurité dans toutes les phases du cycle de vie de développement d'un système. Il prend en compte la sécurité depuis l'analyse des besoins, la conception, l'implémentation, test, et la phase d'entretien.

Parmi les outils de conception de la sécurité dans tous les cycles de développement on cite : Misuse case [12] [13] [14] [15], Secure Tropos [12] [16] [17] [18] [19], UMLsec [8] [9] [10] [11] [20], SecureUML [21] [22], et Abuse frame [6] [7] [12] [23] [24]. Les travaux [21] [25] [26] [27] [28] essayent de proposer des processus, méthodes, et des méthodologies pour organiser et comment intégrer la sécurité dans tout le cycle de développement des systèmes pour être plus sécurisé. L'évaluation de ces méthodes est basée sur quelques principes de coût et de performances.

Selon [5] qui situent quelques recommandations pour évaluer et orienter les développeurs comme :

- Réduction des coûts et la durée du développement dans les produits de sécurité.
- Maximisation de la qualité du développement d'un produit de sécurité (sécurité, facilite d'utilisation, maintenance, …etc.).
- garantit le niveau et la fonction de sécurité dans la construction du système de sécurité.
- Obtention d'une sécurité maximale avec coût minimal pour construction du système de sécurité.

3. Ingénierie du besoin de sécurité

Selon Yeun-Hee jei et al [5], l'ingénierie du besoin de sécurité comprend l'analyse de l'environnement de sécurité, la sélection des objets les fonctions de sécurité, et les spécifications de SFRS (Security Functional Requirement Specification) / PP (Protection Profil) / ST (Security Target) qui sont comparables à celles de l'étude de faisabilité, de la spécification et de l'analyse des exigences, respectivement dans le processus d'ingénierie des exigences. Haley et al

[1] Systèmes de sécurité : ce sont les systèmes d'information développés par l'utilisation des produits de sécurité comme : IDS, firewall, anti-virus…etc.
[2] Technologies de la sécurité d'information : ce sont les protocoles de chiffrement, les algorithmes de chiffrement…etc.

[29] proposent un Framework pour l'ingénierie des exigences (besoins) de sécurité (SREF : Security Requirement Engineering Framework).

SREF [29, 30] suit 4 étapes comme suit :

1. Identifier les exigences fonctionnelles.
2. Identifier les objectifs de sécurité (Identifier les assets, Générer une description des menaces, appliquer les principes de gestion (séparation des tâches, fonctions, ..)).
3. Identifier les exigences de sécurité (contraintes sur un ou plusieurs objectifs de sécurité. Les exigences de sécurité sont notées textuellement).
4. Construire des arguments de satisfaction (montrer que le système peut satisfaire aux exigences de sécurité).

L'ingénierie du besoin de sécurité est utilisée pour spécifier l'analyse des besoins de sécurité, l'analyse d'environnement de la sécurité, les fonctions, les techniques et les outils nécessaires pour accomplir tous les objectifs de sécurité. Il contient tous les aspects aidant à la réalisation de la sécurité d'un système, à réaliser selon l'étude de faisabilité et le cahier de charges [5] [8].

4. Propriétés de la sécurité

Les propriétés de la sécurité représentent les principes ou les critères qui doivent être dans n'importe quel système de sécurité. C'est-à-dire il faut assurer ces contraintes pour dire qu'un système ou un autre contient un niveau de sécurité bien définit. Les trois grands principes dans tous les programmes de sécurité sont la *Disponibilité*, *l'Intégrité* et la *Confidentialité*. On parle de l'AIC ou *CIA* [31] (Availability, Integrity, and Confidentiality).

Tous les contrôles, mécanismes et sauvegardes de sécurité sont mis en œuvre pour assurer un ou plusieurs de ces principes de protection. Les pirates qui organisent les attaques contre les systèmes utilisent un ensemble des vulnérabilités pour essayer de détruire ces trois principes. La figure 2.1 illustre les principes de sécurité.

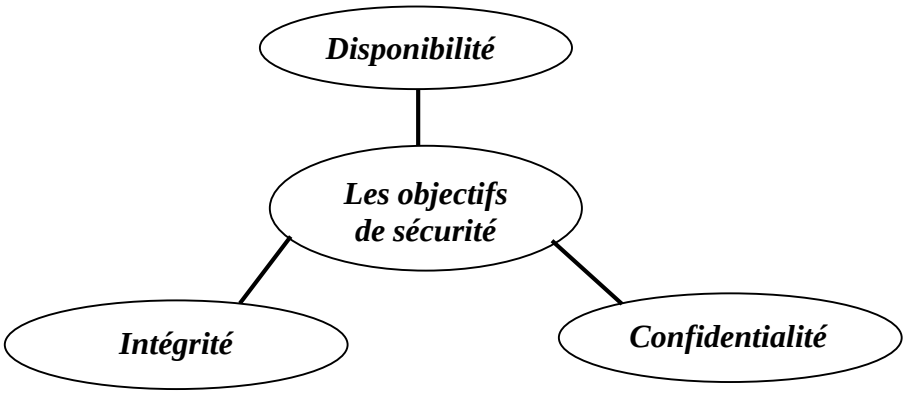

Figure 2.1 : Principes de sécurité. [31]

4.1 Disponibilité

La disponibilité [31] [32], garantit la fiabilité et l'accès rapide aux données et aux ressources par les personnes autorisées selon leurs privilèges d'utilisation. Les périphériques réseaux, les ordinateurs et les applications doivent offrir des fonctionnalités adéquates pour fonctionner de manière stable avec un niveau de performance acceptable. Les mesures de sécurité doivent être capables de maintenir rapidement les perturbations. Les mécanismes de protection nécessaires doivent être en place pour protéger contre les menaces internes et externes susceptibles d'affecter la disponibilité et la productivité de tous les composants du système.

Déni de service (DoS : Denial-of-service) est parmi les attaques utilisées pour perturber la disponibilité des fichiers, de la transmission, et du flux de trafic. Pour se protéger contre ces attaques, les pare-feu, et les systèmes de détection d'intrusion (IDS) avec les serveurs proxy qui sont implémentés dans le réseau afin de diminuer les attaques orientées vers la disponibilité.

4.2 Intégrité

Le service d'intégrité [31] [32] est garanti lorsque l'assurance de l'exactitude et de la fiabilité des informations et des systèmes est fournie et que toute modification non autorisée est empêchée. L'intégrité permet d'assurer que les informations n'ont pas été modifiées pendent la manipulation ou la transmission. Les mécanismes de communication doivent fonctionner correctement pour maintenir, traiter et déplacer les données vers les destinations prévues sans modification inattendue. Les systèmes qui imposent et fournissent cet attribut de sécurité garantissent que les attaquants, ou les erreurs commises par les utilisateurs, ne dysfonctionnent pas l'intégrité des systèmes ou des données.

4.3 Confidentialité

La confidentialité [31] [32] préserve le secret de l'information qui permet l'accès à l'information pour les utilisateurs autorisés et éviter toute divulgation d'informations en direction de ceux qui ne sont pas autorisés à les connaître ou à les utiliser. La confidentialité assure que chaque fonction de traitement des données est hors la divulgation non autorisée.

Le niveau de confidentialité devrait être préservé dans les données résidant sur les systèmes et les périphériques du réseau, au fur et à mesure de leur transmission et une fois qu'elles ont atteint leur destination. La confidentialité assure la confidentialité des fichiers, de la transmission, et du flux de trafic. Le chiffrement des données, l'établissement de dispositifs de garde-barrière, un contrôle d'accès strict et la classification des données peuvent améliorer la confidentialité des informations.

5. Sécurité des systèmes d'information avancés

La sécurité des systèmes d'information avancés a besoin plus que l'implémentation des mécanismes de sécurité au sein des réseaux de l'entreprise, mais une étude approfondie qui touche tous les éléments du système. La sécurité est un processus dans le but est d'assurer le fonctionnement fiable et garantit les principes de sécurité (AIS). Parmi les processus les plus fiables pour maintenir la sécurité, on cite le travail de [33] qui vise à donner une vision globale sur la sécurité des systèmes d'information. L'auteur du livre [33] propose cinq phases pour établir un système d'informations qui sont :

L'inspection : est le processus qui consiste à déterminer l'état courant et à évaluer le niveau approprié de sécurité. C'est dans cette phase qu'on évalue les besoins en sécurité de l'organisation autant que son niveau actuel de préparation.

La protection : est le processus d'anticipation qui vise à créer un environnement aussi sécurisé que possible. Dans cette phase, dix aspects fondamentaux de la sécurité des informations et les problèmes qui sont examinés.

La détection : est le processus réactif par lequel ils déterminent les activités inappropriées et alertent les personnes responsables. La détection est indispensable pour tout ce qui ne peut pas être protégé ou prévu.

La réaction : est le processus de réponse à un incident de sécurité. Cette phase met l'accent sur la façon de réagir à un incident de sécurité pour en minimiser l'impact.

La réflexion : est le processus de suivi nécessaire pour évaluer qualitativement l'implémentation des mesures de sécurité. Ces procédures de débriefing sont nécessaires pour l'organisation afin qu'elle puisse enrichir son expérience à partir de chaque incident.
La figure 2.2 illustre les composants essentiales dans chaque étape.

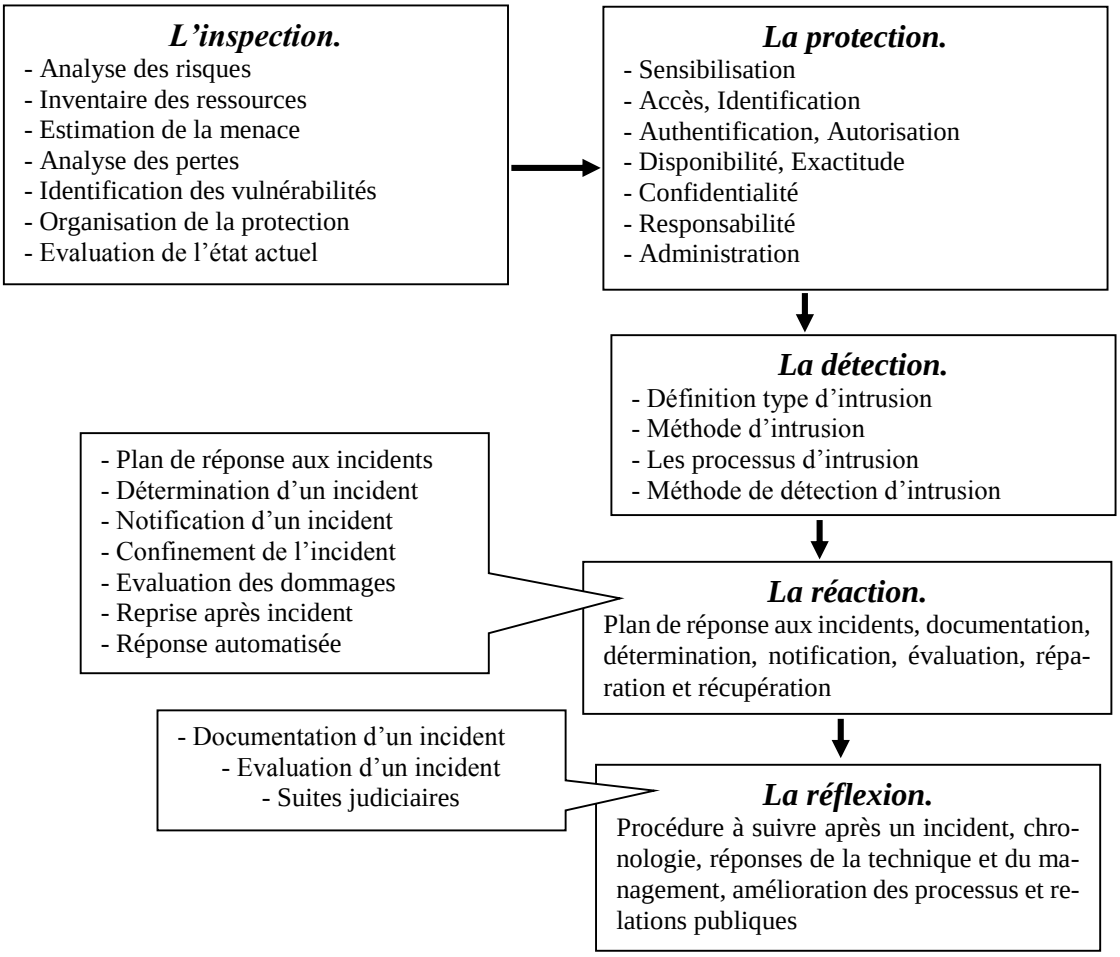

Figure 2.2 : Gestion de la sécurité dans un système d'information.

6. Types des attaques

Dans cette section, on présente les différents types d'attaques utilisés par les pirates. Selon [32] qui classifiait les attaques en quatre catégories principales (les attaques : d'accès, de modification, le déni de service, et la répudiation) :

- ***Les attaques d'accès*** : ces attaques sont connues par leur signature d'accès à des données non autorisées. Ils attaquent la confidentialité des systèmes. Parmi les attaques les plus utilisées sont : Snooping, Ecoute, et Interception.
- ***Les attaques de modification*** : ces types des attaques tentent de modifier les informations du système dans tous les types de modification (modification, insertion, suppression). Ce type d'attaque est dirigé contre l'intégrité de l'information.

- ***Les attaques de déni de service*** : les attaques par déni de service sont connues par le nom de Dos (*DoS : Denial of service*). Ce sont des attaques qui menacent la disponibilité de l'information et des ressources qui rendent impossible leur utilisation par les utilisateurs légitimes. Les attaques DoS cherchent à déni l'accès à l'information, aux applications, aux systèmes, et aux communications. Ce sont les attaques les plus utilisées par les pirates.
- ***Les attaques de répudiation*** : Ces types d'attaques tentent de donner une fausse vue sur le déroulement du système (négation d'un évènement, ou une transition s'est réellement produits). Ce type des attaques est dirigé contre la responsabilité et la réputation de l'entreprise.

7. Mécanismes de protection

Dans cette section, on présente les composants essentiels pour construire l'infrastructure de la sécurité dans les systèmes d'information avancés. Selon la politique de la sécurité utilisée par les systèmes, les mécanismes de la protection restent les mêmes avec quelques modifications au niveau des versions ou de la configuration qui est reliée à l'architecture et la manier de l'organisation des systèmes. Les mécanismes de sécurité sont les éléments de protection contre les incidents de piratage et les attaques.

Dans cette section, on présente les différents mécanismes de la sécurité d'une façon ordinaire simple avec les définitions sans détailles jusqu'au mode d'implémentation car chaque système à sa propre situation. On cite la cryptographie, Firewall, serveur Proxy, IDS & IPS, les VPN, et les Contrôle d'accès.

7.1 Cryptographie

La cryptographie [31] [32] [34] est la science à dissimuler l'information (écriture et lecture des messages codés) dans une forme illisible pour les individus non autorisés, lisible et utilisable pour les personnes autorisées. Le mécanisme de cryptographie utilise un algorithme avec une clé pour chiffrer un texte clair (fichiers, documents,…etc.) à un texte chiffrer. L'opération inverse faite par la personne autorisée par un algorithme et une clé de déchiffrement en texte clair. Le cryptosystème utilise un algorithme de chiffrement (qui détermine la simplicité ou la complexité du processus de chiffrement), des clés et les composants logiciels et protocoles nécessaires. La cryptographie assure les trois principes de sécurité CIA.

A la base des travaux [32] [34] le tableau 2.1 et la figure 2.3 présentent l'ensemble de la terminologie et le principe de base du chiffrement.

Tableau 2.1 : Terminologie du chiffrement**.**

Terminologie	Définition
Texte en clair	L'information dans sa forme originale (Plaintext).
Texte chiffré	L'information après l'exécution de l'algorithme de chiffrement.
Algorithme	Fonction mathématique de chiffrement.
Clé	Ensemble des données d'entrée de l'algorithme pour faire l'opération de déchiffrement.
Chiffrement	Processus de transfert de texte clair en texte chiffré.
Déchiffrement	Processus inverse de chiffrement.
Cryptographie	Art de dissimuler l'information en employant le chiffrement.
Cryptographe	Personne qui pratique la cryptographie.
Cryptanalyse	Art d'analyser les algorithmes cryptographiques pour chercher les points faibles.

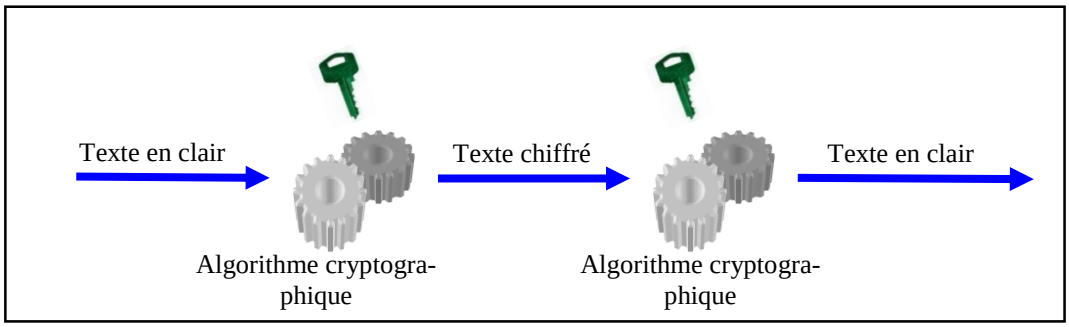

Figure 2.3 : Principe de base du chiffrement. [32]

Dans ce qui suit, on présente le cryptage symétrique (à clé secrète), le cryptage asymétrique (à clé publique), les fonctions de hachage non réversibles, et les signatures numériques.

7.1.1 Cryptage symétrique

Le cryptage symétrique (clé secrète) [31] [32] [34], ce type de cryptage utilise la même clé et le même algorithme pour le chiffrement et le déchiffrement des messages. La clé a donc une double fonctionnalité, en ce sens qu'elle peut exécuter à la fois les processus de cryptage et de décryptage.

La figure 2.4 présente le principe de base de cryptage symétrique. Le cryptage symétrique est beaucoup plus rapide (moins de calcul) que les systèmes asymétriques et difficiles à rompre si la taille de clé est grande. Cependant, il nécessite un mécanisme sécurisé pour livrer les clés et le nombre de clés augmente avec le nombre d'individus, ce qui rend la gestion des clés très difficile.

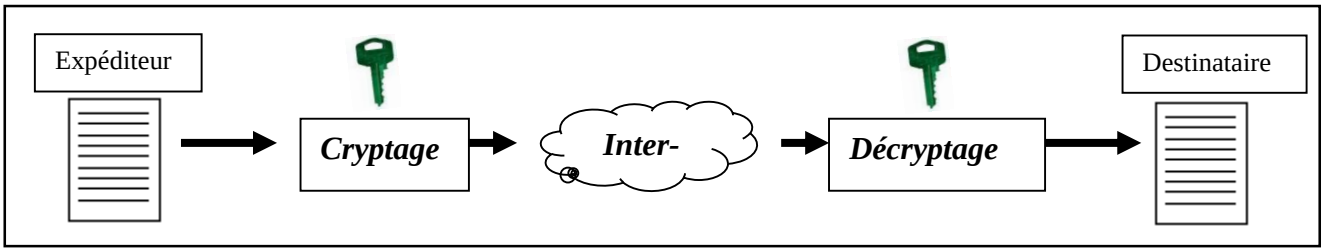

Figure 2.4 : Cryptage à clé secrète. [32] [34]

Les algorithmes utilisés dans le cryptage symétrique sont nombreux, on cite :

- DES (Data Encryption Standard) et 3DES ou triple DES.
- Blowfish, Skipjack, CAST-128, et GOST(256 bits)
- Advanced Encryption Standard (AES), RC-4 (River Cipher 4) RC-5, RC-6, et IDEA (International Data Encryption Algorithme).

7.1.2 Cryptage Asymétrique

Le cryptage asymétrique (clé publique) [31] [32] [34] est différent du cryptage symétrique dans le nombre des clés utilisées. C'est-à-dire, chaque entité possède des clés différentes ou des clés asymétriques (une clé publique et une clé privée). Si un message est crypté par une clé, l'autre clé est nécessaire pour décrypter le message. La clé publique peut être connue de tous et la clé privée doit être connue et utilisée uniquement par le propriétaire.

Les clés publique et privée d'un cryptosystème asymétrique sont liées mathématiquement, mais si quelqu'un obtient la clé publique d'une autre personne, elle ne devrait pas être en mesure de déterminer la clé privée correspondante. La figure 2.5 illustre le mécanisme de chiffrement asymétrique.

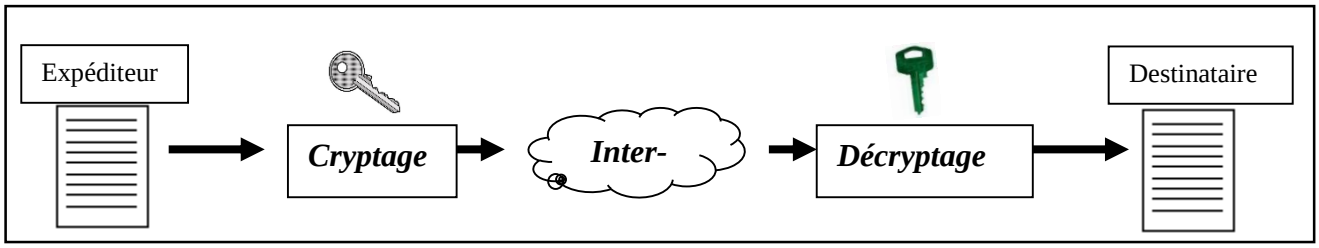

Figure 2.5 : Chiffrement à clé publique. [31] [32]

Parmi les avantages du cryptage asymétrique : il communique de façon sécurisée, il emploie des clés différentes pour crypter et décrypter les données. Parmi les avantages des algorithmes de clés asymétriques, on cite les exemples suivants : Meilleure distribution des clés que les systèmes symétriques, et meilleure évolutivité que les systèmes symétriques. D'autres côtés les inconvénients sont : fonctionnent beaucoup plus lentement que les systèmes symétriques, et des tâches intensives en mathématiques.

Les algorithmes les plus utilisés dans le cryptage asymétrique, on cite :

- Elliptic curve cryptosystem (ECC).
- RSA (Rivest-Shamir-Adleman), El Gamal, Diffie-Hellman.
- Digital Signature Algorithm (DSA) et Merkle-Hellman Knapsack.

7.1.3 Les fonctions de hachage

Une fonction de hachage [32] [34] est une fonction de condensation d'un message de longueur aléatoire à un code de longueur fixe. Ce code est utilisé comme une empreinte pour le document quand doit envoyer. La fonction de hachage assure l'intégrité des données c'est-à-dire en peut vérifier que les données n'ont pas été changé pendant la transmission. Les fonctions de hachage les plus connues sont :

- L'algorithme SHA (Secure Hash Algorithm).
- L'algorithme MD4 (Message Digest 4).
- L'algorithme MD5 (Message Digest 5).

7.1.4 Les signatures numériques

Une signature numérique [32] [34] est une technique qui assure l'authentification de l'information. C'est une combinaison entre le cryptage à clé publique et la fonction de hachage. Les signatures numériques n'assurent pas la confidentialité du contenu d'un message. Les algorithmes de signature numérique à clé publique les plus connus sont RSA (Rivest-Shamir-Adleman) et DSS (Digital Signatue Standard).

7.2 Firewalls & Proxy

Dans cette section, on décrit deux types de pare-feu et la différence entre eux qui sont : Firewalls et les serveurs Proxy.

7.2.1 Firewalls

Un pare-feu [31] [32] [34] est un dispositif pour contrôler le trafic (contrôle d'accès du réseau) des paquets entrants et sortants du réseau et les décisions de routage. Les pare-feu sont utilisés pour restreindre l'accès à un réseau depuis un autre réseau. Il est décrit comme un "point d'arrêt" dans le réseau car toutes les communications doit passer par là, et c'est là que le trafic est inspecté et restreint. Selon les règles mises dans le pare-feu qui décide d'accepter ou rejeter

divers types de trafic. Les règles de contrôles sont reliées à des services de réseau et de configuration pour autoriser le trafic en fonction du service, de l'adresse IP de la source ou de la destination, ou l'ID de l'utilisateur.

Les caractéristiques les plus utilisées pour l'analyse des paquets sont : La direction de trafic, l'origine du trafic, l'adresse IP, les numéros de port, l'authentification, et le contenu applicatif. La figure 2.6 présente un exemple sur le déploiement de pare-feu. Selon les critères de filtrage et d'emplacement, il y a plusieurs classifications pour les pare-feu. En ce qui concerne l'emplacement, il y a les pare-feu au niveau d'application (proxy) et les pare-feu au niveau du réseau. Dans le côté de la fonction de filtrage, il y a les pare-feu filtreurs de paquets, circuits, et applicative.

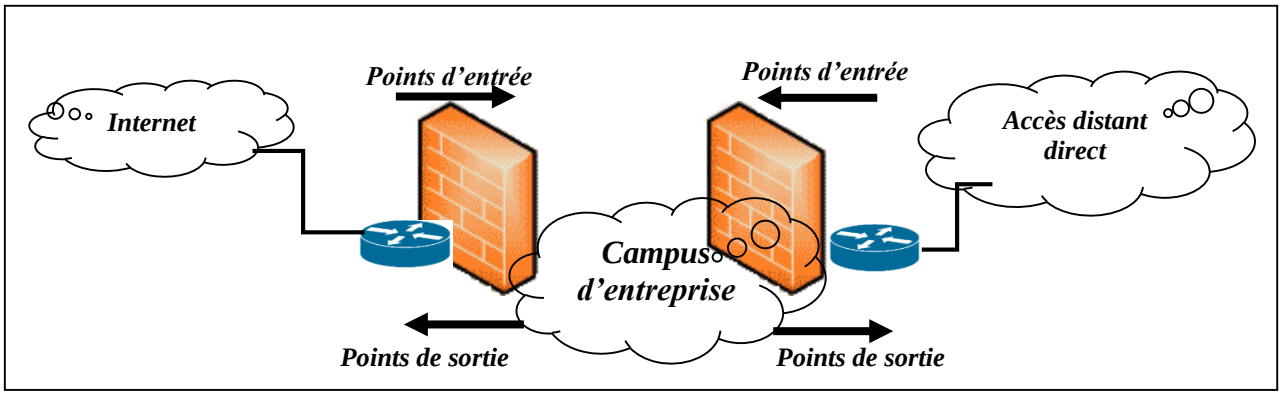

Figure 2.6 : Déploiement de firewall [34].

7.2.1 Proxy Firewall

Un Proxy [31] est un firewall qui intercepte et inspecte les messages avant de les livrer aux destinataires prévus. Il accepte les messages entrants ou sortants d'un réseau, les inspecte à la recherche d'informations malveillantes, et lorsqu'il décide que les messages sont corrects, les transmet à l'ordinateur de destination. Le Proxy protège les systèmes contre les attaques par leur mécanisme qui sépare le trafic de l'extérieur de connecter directement vers la destination sans contrôle et traitement des paquets.

7.3 IDS &IPS

Dans cette section, on décrit deux types de détection d'intrusion et la différence entre eux qui sont : l'IDS et les IPS (Intrusion Prevention Systems).

7.3.1 IDS

Dans les systèmes de détection d'intrusion (IDS : Intrution Detection systems) [31] [32] la détection d'intrusion est le processus de détecter un événement suspect ou tout type de comportement «Anormal» sur un ordinateur, un réseau ou une infrastructure de télécommunication. Les IDS peuvent être configurés pour surveiller les attaques, analyser les journaux d'audit, mettre fin à une connexion, alerter un administrateur en cas d'attaques, exposer les techniques d'un pirate informatique, indiquer les vulnérabilités qu'il faut traiter et éventuellement aider à localiser les pirates.

Les trois composants communs entre les différents IDS sont : *Capteurs (Sensors), analyseurs (analyzers), et interfaces administrateurs (Administrator interfaces).* Les capteurs recueillent le trafic et les données d'activité des utilisateurs et les envoient à un analyseur qui recherche une activité suspecte. L'analyseur décide si c'est une activité normale ou non. Dans le cas où l'activité est anormale, il envoie une alerte à l'interface de l'administrateur.

Les types des IDS sont classés selon plusieurs catégories en fonction de l'emplacement (Réseaux ou Host) et la base de détection. Les deux types N-IDS (Network-IDS) et H-IDS (Host-IDS) appartiennent à la catégorie de l'emplacement. Les N-IDS sont des IDS installés dans le réseau pour surveiller tout le trafic et communication effectués au sein de l'entreprise. Les H-IDS sont des IDS installés au niveau des ordinateurs, des stations de travail et / ou des serveurs individuels afin de détecter toute activité inappropriée ou anormale.

D'autre part, la catégorie des IDS est basée sur la méthode de détection. Il y a deux types, selon qui utilisent les bases de signature (Knowledge/Signature-Based Intrusion Detection) et selon qui utilise le comportement anormal (Anomaly-based Intrusion Detection) de l'utilisateur pour détecter les activités suspectes. Les IDS à base de signatures utilisent des bases de données contiennant les scénarios des attaques connues auparavant (Pattern matching, Stateful matching). Les IDS à base de comportement surveillent les actions des utilisateurs et décident leurs types de connexion (normale ou anormale). Pour faire la déférence sont utilisées des bases statistiques d'anomalie, protocole, trafic, et les bases de règles ou d'heuristiques.

7.3.2 IPS

Les systèmes de prévention d'intrusion [31], sont des technologies préventives et proactives, considérées comme une extension de la technologie IDS. L'objectif des IPS est de détecter les activités anormales avant de causer une menace ou un impact sur le système. La technologie IPS «basée sur le contenu», c'est-à-dire qu'elle prend des décisions concernant ce qui est

malveillant ou non, en fonction de l'analyse de protocole ou des capacités de correspondance de signature.

7.4 VPN

Un Réseau Privé Virtuel (VPN : Virtual Private Network) [31] [32], est un système qui permet de faire une connexion privée sécurisée entre plusieurs utilisateurs par l'intermédiaire d'un réseau public. Les objectifs des VPN dotent l'entreprise d'un réseau privé avec tous les critères de l'intégrité, et la confidentialité avec le moindre coût. Ce qui donne un ensemble des avantages au système comme : l'information reste inchangée au niveau de l'entreprise, et l'échange d'informations entre les sites distants.

Pour arriver à ces objectifs, les VPN doivent être contenus de nombreux caractéristiques comme : le chiffrement de trafic, l'authentification des sites distants, et utilisations des protocoles de sécurité, avec la connexion de type "point à point". Les deux catégories le plus connu des VPN sont : les VPN site à site et les VPN utilisateur à site. Le VPN le plus utilisé est PPTP (Point-to-Point Tunneling Protocol), IPSec et L2TP (Layer 2 Tunneling Protocol). PPTP est devenu très populaire lorsque Microsoft l'a inclus dans ses produits Windows.

La figure 2.7 présente un exemple sur les VPN et fournit un lien virtuel dédié entre deux entités sur un réseau public.

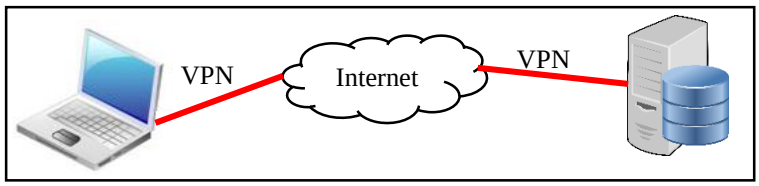

Figure 2.7 : VPN [31].

7.5 Contrôles d'accès

Les contrôles d'accès [31] [35] consistent à gérer chaque demande de ressources et de données et déterminer si la demande doit être accordée ou refusée. D'une autre manière, les contrôles d'accès sont des fonctions de sécurité qui contrôlent la manière dont les utilisateurs et les systèmes communiquent et interagissent avec d'autres systèmes et ressources. Les contrôles peuvent être de nature technique, physique ou administrative.

Le contrôle d'accès est un terme large qui couvre plusieurs types de mécanismes qui appliquent des fonctionnalités de contrôle d'accès sur les systèmes informatiques, les réseaux et

les informations. Pour simplifier ce mécanisme, les étapes d'*identification* et l'*authentification* et l'*autorisation*, ce sont les trois étapes nécessaires dans chaque type de contrôle d'accès.

Les politiques de contrôle d'accès peuvent être regroupées en trois classes principales : *Contrôle d'Accès Discrétionnaire (DAC : Discretionary Access Control), Contrôle d'Accès Obligatoire (MAC : Mandatory Access Control), Contrôle d'Accès base-rôle (RBAC : Role-based Access Control).*

Le tableau 2.2 ci-dessous explique les principes de ces trois classes [35].

Tableau 2.2 : Principes de contrôle d'accès.

Contrôle d'accès	Principe
DAC : Discretionary Access Control	Les politiques (basées sur les autorisations) contrôlent l'accès en fonction de l'identité du demandeur et des règles d'accès indiquant ce que les demandeurs sont autorisés à faire (ou non).
MAC : Mandatory Access Control	Les politiques contrôlent l'accès en fonction de la réglementation prescrite par une autorité centrale.
RBAC : Role-based Access Control	Les stratégies contrôlent l'accès en fonction des rôles des utilisateurs dans le système et des règles indiquant les accès autorisés aux utilisateurs dans des rôles donnés.

Les stratégies Discretionary et Role-based sont généralement associées à (ou incluent) une stratégie administrative qui définit qui peut spécifier les autorisations / règles régissant le contrôle d'accès. ***RADIUS*** (*Remote Authentication Dial-In User Service*), ***Kerberos*** (*systèmes d'authentification*), et ***TACACS*** (*Terminal Access Controller Access Control System*) sont les contrôles d'accès les plus utilisés.

8. Conclusion

La sécurité des systèmes d'information avancés est devenue un domaine vaste qui cherche à trouver des *méthodes, processus, protocoles, et des plateformes* pour sécuriser les systèmes et les réseaux informatiques. La sécurité doit être intégrée dans tous les cycles de développement et garder toujours l'évolution, la rénovation, et la maintenance des exigences et les besoins des entreprises.

Dans ce chapitre, nous avons décrit les points de vue sur les notions de l'ingénierie de sécurité et l'ingénierie des besoins de sécurité. Ensuite, nous avons défini les différentes propriétés de sécurité qui doit être assurée dans chaque système. Après, nous avons présenté la sécurité au niveau des systèmes d'information avancés. Enfin, nous avons étudié quelques mécanismes de protection disponibles pour assurer la protection d'un système avec les utilisations de ces mécanismes et des exemples sur leur manipulation

Chapitre 03

La sélection des attributs pour les systèmes de détection d'intrusion

1. Introduction

Les IDS sont utilisés pour protéger les systèmes contre les virus et les attaques survenues par les pirates. Les IDS dans leurs processus de protection manipulent une grande masse de données pour vérifier toutes les connexions entrantes et sortantes du système, ces données sont assignées à des facteurs. Ces facteurs sont connus comme des variables descriptives qui sont nommées par les attributs. Il est difficile de détecter des menaces en temps réel au niveau de connexion à cause de tous ces attributs qui sont reliés à cette vérification.

Les IDS se basent sur les modules de classification pour faire la différence entre la connexion normale et les attaques, et à cause de la masse de données, les IDS sont confrontés à des problèmes au niveau du taux et la vitesse de détection. Bien que les solutions proposées et l'amélioration de l'efficacité des modèles de classification, mais le problème persiste. D'où, la sélection d'attributs (FS : Feature Selection) qui donne un autre souffle pour traiter les problèmes des IDS.

FS tente de diminuer la dimension de données par la sélection des attributs principaux qui peut facilement faire la différence entre la connexion normale et les attaques, et éliminer les attributs qui contiennent l'aspect de la redondance dans l'ensemble de données.

Dans ce chapitre, nous décrivons un aperçu de la plupart des techniques qui ont été proposées dans la recherche sur la sélection des attributs en examinant les contributions existantes. Nous présentons les derniers algorithmes de la sélection connus pour l'IDS qui sont développés pour sélectionner les meilleurs sous-ensembles d'attributs. Nous présenterons une nouvelle taxonomie avec différents travaux de la sélection sur l'IDS, classés en cinq catégories. Nous fournissons une carte pour comprendre et construire l'état actuel de la sélection dans l'IDS par une classification et l'étude comparative. Une enquête sera présentée pour comprendre les progrès de la recherche et identifier les tendances futures et les défis existants.

2. La sélection des attributs

2.1 Définition

La sélection d'attributs (FS) [3] [4] est un processus d'optimisation de prétraitement visant à réduire la dimensionnalité de l'ensemble de données en sélectionnant les attributs intéressants sans redondance et avec pertinence. Ces attributs représentent le (s) meilleur (s) sous-ensemble (s) qui assurent [4] [36] [37] [38] :

- L'augmentation des performances du modèle de classification (Taux de précision), et éviter les frais généraux de problème de classification.
- La Réduction de temps de calcul et les besoins de stockage.
- La compréhension des données sur le bruit et éviter les problèmes de sur-ajustement.

La définition mathématique de la sélection d'attributs est présentée comme 6-uplet [132] :
FS = { D, F, C, S, fs, E }, où: D est un ensemble de données D= { $i_1,i_2,...,i_m$} avec m instances, F est un ensemble d'attributs F = { $f_1,f_2,...,f_n$} avec n nombre d'attributs. C est la classe cible C, C= {$c_1,c_2,...,c_k$} avec k l'étiquette des classes cibles. S (espace de recherche) est une partition de l'ensemble F qui contient tous les sous-ensembles que nous pouvons construire par F, S= { $S_1,S_2,...,S_l$ } (l= 2^n - 1: problème d'optimisation NP-Hard) avec S_i = { $f_j,f_k,...,f_l$ } ($1 \leq j \neq k \neq l \leq n$), E mesure d'évaluation, la fonction fs représente le processus de sélection d'attributs:

$$f_S : F \rightarrow S.$$

En outre, mesure d'évaluation (E) est utilisée par fs pour évaluer les sous-ensembles d'attributs et fournit des mesures qui en déterminent leur qualité. Mesure d'évaluation est divisée en cinq types selon [39] qui sont : *La distance, l'information, la dépendance, la cohérence et le taux d'erreur du classificateur*.

2.2 Processus et mécanismes

Selon [3] [40], la figure 3.1 illustre le processus le plus populaire utilisé pour le FS. La génération de sous-ensembles est la méthode de recherche qui découvre l'espace de recherche pour sélectionner le ou les meilleurs sous-ensembles. L'évaluation de sous-ensemble utilise les mesures d'évaluation pour évaluer chaque sous-ensemble selon des critères bien définis. Alors que la validation de résultats est l'étape de valider les sous-ensembles sélectionnés. Cette étape utilise les algorithmes de classification pour décider de son efficacité. Les mesures d'évaluation (étape d'évaluation) et la méthode de recherche (génération de sous-ensembles) sont les étapes les plus importantes dans la sélection d'attributs. Elles déterminent l'efficacité du ou des sous-ensembles d'attributs sélectionnés.

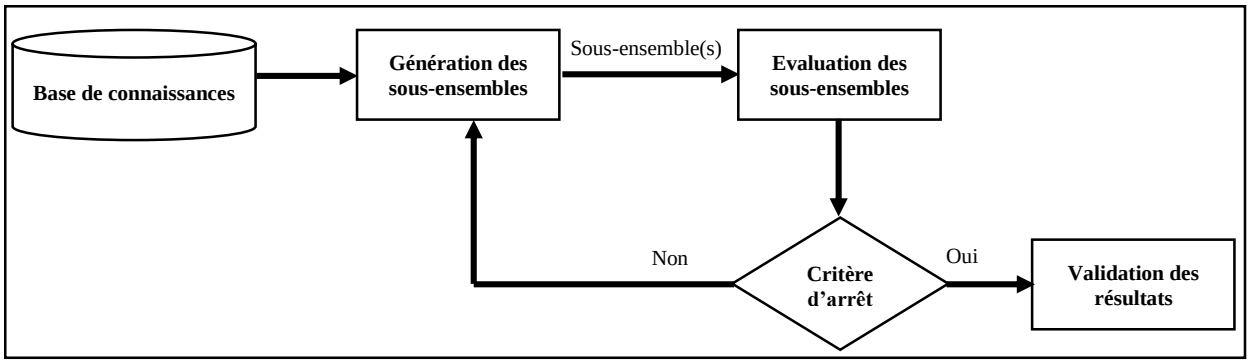

Figure 3.1 : Processus de la sélection d'attributs [3].

Le processus FS suit un mécanisme de détection qui se concentre sur trois catégories telles que [132] : la sélection Incrémentale, Décrémentale et Aléatoire.

Dans le mécanisme incrémental, l'ensemble d'attributs qui représentent les meilleurs attributs démarre vide. À chaque itération, le processus ajoute un nouvel attribut à l'ensemble (un par un). La sélection du nouvel attribut dépend des mesures d'évaluation. À la fin du processus, l'ensemble contient les meilleurs attributs qui représentent les meilleurs attributs sélectionnés pendant les itérations.

Dans le mécanisme décrémental, l'ensemble des meilleurs attributs commence complet avec tous les attributs. Lorsque le processus de sélection est terminé, l'ensemble d'attributs contient uniquement les meilleurs attributs qui restent après avoir supprimé les autres un par un dans chaque itération. Le mécanisme décrémental utilise les mesures d'évaluation pour sélectionner l'attribut qu'on doit supprimer.

Le mécanisme Aléatoire, dans chaque itération sélectionne un groupe d'attributs comme sous-ensemble et l'évalue avec des mesures d'évaluation. À la fin du processus, il obtient le meilleur sous-ensemble entre tous les sous-ensembles sélectionnés. La manière de sélectionner le groupe d'attributs dépend des techniques utilisées par l'approche proposée.

2.3 Type

Selon la dépendance avec des algorithmes d'apprentissage / classification, la sélection d'attributs a été classée en trois groupes [3] [42] [41] : Wrapper, Filter et l'approche Hybride.

Wrapper approche [4] [38] [43] utilise les algorithmes d'apprentissage / classification pour sélectionner les meilleurs attributs comme un outil d'évaluation. Il utilise le taux de précision (taux d'erreur) comme une rétroaction pour déterminer l'efficacité des sous-ensembles d'attributs.

L'approche par filtre [38] [39] [43] utilise les données d'apprentissage statistique comme mesure pour évaluer les attributs indépendamment de l'algorithme d'apprentissage / classification.

Cependant, l'approche Wrapper et du Filtre présente certains avantages et inconvénients. Les méthodes wrapper atteignent une meilleure performance de classification que la méthode de filtrage, mais elles sont plus coûteuses en calcul [43] [44]. Dans le cas contraire, l'approche Filter a été assignée lorsque le traitement des données a des dimensions élevées [4]. Par conséquent, l'approche Hybride est une combinaison entre les méthodes de Wrapper et de Filtre, qui a émergé pour couvrir ces inconvénients et bénéficier de ces avantages.

3. Les bases de connaissances et d'évaluation de la performance

3.1 Bases de connaissances

Différents ensembles de données publiques et privées sont intégrés en tant que données de référence pour l'évaluation IDS. Les bases de connaissances privées et auto construites sont générées pour éviter les bases de données d'apprentissages incompletes, mais toujours inaccessibles et difficiles de garantir leur efficacité.

Les bases de connaissances publiques utilisent des séquences d'appel du système de comportement de l'utilisateur et une source de données de trafic réseau. DRAPA98-99, KDDcup99, et NSL-KDD sont les célèbres benchmarks publics qui sont basés sur le réseau de trafic et sont considérés comme des données utiles pour évaluer l'IDS.

La base de connaissance DRAPA-Lincoln [45] a été créé par le Lincoln Laboratory du MIT. DRAPA-Lincoln a deux versions (DRAPA98 et DRAPA99). DRAPA98 est une collection de 300 instances de 38 attaques entre les données d'apprentissage et de tests.
Alors que la base de connaissance DRAPA99 contient 200 instances de 58 attaques. De plus, KDDcup99 [46] est considéré comme une dérivation de DRAPA98 qui intègre les paquets TCP individuels dans les connexions TCP.

KDDcup99 est une collection de cinq millions d'enregistrements de connexions provenant de sept semaines de trafic réseau. KDDcup99 est distribué en deux versions, l'ensemble de données complet avec 4898431 enregistrements et le sous-ensemble de 10% avec 494307 enregistrements.

La nouvelle version de KDDcup99 est NSL-KDD [47] [48] qui est considérée comme une version révisée. La plupart des avantages de NSL-KDD sont concentrés sur le nombre d'enregistrements d'apprentissage et l'ensemble d'enregistrements de tests qui minimisent le niveau de difficulté. À NSL-KDD, ils suppriment tous les enregistrements redondants dans l'ensemble d'apprentissage et les enregistrements en double dans l'ensemble de tests dont l'ensemble de données a un numéro d'enregistrement raisonnable.

Les deux ensembles de données (KDDcup99 et NSL-KDD) ont les mêmes problèmes concernant la représentation réelle du réseau [49]. Cependant, les deux ensembles de données ont toujours été utilisés par la plupart des chercheurs comme un ensemble de données expérimentales. Tandis que, KDDcup99 est dérivé de DRAPA98 et NSL-KDD est obtenu en supprimant les instances redondantes et en double de KDDcup99. Nous nous concentrons sur KDDcup99 et NSL-KDD pour décrire les différents attributs qui contiennent 41 attributs (32 attributs continus et 9 attributs nominaux) avec la classe cible. Les 41 attributs ont été classés en quatre catégories (*de base, de contenu et de trafic* (*trafic basé sur le temps et trafic basé sur l'hôte*)).

La description de tous les attributs de KDDcup99 et de NSL-KDD est présentée dans le tableau 3.1.

Table 3.1. KDDcup99 / NSL-KDD description.

Num	Nom	Type		Description
1	duration	Continu	Catégorie 1	Longueur de la connexion.
2	Protocol_type	Nominal		Protocole de connexion.
3	service	Nominal		Service de destination.
4	flag	Nominal		Indicateur d'état de la connexion.
5	Src_bytes	Continu		Octets envoyés de la source à la destination
6	Dst_bytes	Continu		Octets envoyés de destination à source
7	land	Nominal		1 si est depuis / vers le même hôte / port; 0 sinon
8	Wrong_fragment	Continu		Nombre de mauvais fragments
9	urgent	Continu		Nombre de paquets urgents
10	hot	Continu	Catégorie 2	Nombre d'indicateurs chauds
11	Num_failed_logins	Continu		Nombre d'échecs de connexion lors des tentatives
12	Logged_in	Nominal		1 si connecté avec succès; 0 sinon
13	Num_compromised	Continu		Nombre de conditions compromises
14	Root_shell	Nominal		1 si la coquille racine est obtenue; 0 sinon
15	Su_attempted	Nominal		1 si la commande su root a été tentée; 0 sinon
16	Num_root	Continu		Nombre d'accès root
17	Num_file_creations	Continu		Nombre d'opérations de création de fichiers
18	Num_shells	Continu		Nombre d'invites de shell
19	Num_access_files	Continu		Nombre d'opérations sur les fichiers de contrôle d'accès
20	Num_outbound_cmds	Continu		Nombre de commandes sortantes dans une session ftp
21	Is_hot_login	Nominal		1 si la connexion appartient à la liste critique; 0 sinon
22	Is_guest_login	Nominal		1 si la connexion est une connexion invité; 0 sinon

23	count	Continu	Catégorie 3	Nombre de connexions au même hôte que la connexion actuelle au cours des deux dernières secondes
24	Srv_count	Continu		Nombre de connexions au même service que la connexion actuelle au cours des deux dernières secondes
25	Serror_rate	Continu		% de connexions ayant des erreurs SYN (connexions de même hôte)
26	Srv_serror_rate	Continu		% de connexions ayant des erreurs SYN (connexions de même service)
27	Rerror_rate	Continu		% de connexions ayant des erreurs REJ (connexions de même hôte)
28	Srv_rerror_rate	Continu		% de connexions ayant des erreurs REJ (connexions de même service)
29	Same_srv_rate	Continu		% de connexions au même service (mêmes connexions de service)
30	Diff_srv_rate	Continu		% de connexions à différents services
31	Srv_diff_host_rate	Continu		% de connexions à des hôtes différents (connexions de même service)
32	dst host count	Continu	Catégorie 4	% Nombre de connexions ayant le même hôte de destination
33	Dst_host_srv_count	Continu		% Nombre de connexions ayant le même hôte de destination et utilisant le même service
34	Dst_host_same_srv_rate	Continu		% de connexions ayant le même hôte de destination et utilisant le même service
35	Dst_host_diff_srv_rate	Continu		% de services différents sur l'hôte actuel
36	Dst_host_same_src_port_ rate	Continu		% de connexions à l'hôte actuel ayant le même port
37	Dst_host_srv_diff_host_rate	Continu		% de connexions au même service provenant de différents hôtes
38	Dst_host_serror_rate	Continu		% de connexions à l'hôte en cours qui ont une erreur SO
39	Dst_host_srv_serror_rate	Continu		% de connexions à l'hôte actuel et au service spécifié qui ont une erreur SO
40	Dst_host_rerror_rate	Continu		% de connexions à l'hôte actuel avec une erreur RST
41	Dst_host_srv_rerror_rate	Continu		% de connexions à l'hôte actuel et au service spécifié avec une erreur RST

Chaque instance de KDDcup99 et NSL-KDD est classée par des connexions normales ou attaques. KDDcup99 a 24 types d'attaques qui ont été classées en 4 classes avec la classe normale, à savoir : *DOS* (Refusé de Service), *Probe*, *U2R* (Utilisateur à Racine), et *R2L* (Romote à Local). Ces cinq classes définissent le type de connexion dans chaque instance de KDDcup99. Le tableau 3.2 donne les détails statistiques sur les 5 classes de KDDcup99.

Tableau 3.2. Détail de KDDcup99.

	Normale	**DOS**	**Probe**	**R2L**	**U2R**	**Total**
KDD	972780	3883370	41102	1126	52	4898430
10% KDDcup99 data set	97564	391458	4107	1126	52	494307

L'étape de prétraitement est très importante pour préparer l'ensemble de données pour l'expérimentation avant toute étude ou construction du modèle. Les opérations de prétraitement

sont effectuées pour éliminer tous les problèmes concernant *la taille*, les informations i*ncomplètes* et les *duplications* dans les enregistrements.

Pour cela, différentes techniques ont été intégrées en utilisant des techniques de *discrétisation*, de *discrimination*, de *réduction* et de *normalisation* pour préparer l'ensemble de données aux étapes d'apprentissage et de tests.

3.2 Évaluation des performances

Les performances IDS sont mesurées par des métriques d'évaluation. Selon la matrice de confusion (Tableau 3.3), nous montrons que la plupart des mesures de performance sont utilisées pour évaluer l'efficacité de tout IDS.

Table 3.3. Confusion Matrix.

		Predicale class	
		Normale	**Attacks**
Actual class	Normale	True Positive (TP)	False Negative (FN)
	Attacks	False Positive (FP)	True Negative (TN)

Voici sept évaluations de performance :

- Taux d'erreur (ER : Error Rate) = (FN + FP) / (TP + TN + FN + FP).
- Taux de précision (AR : Accuracy Rate) = (TN + TP) / (TP + TN + FN + FP).
- Taux de détection (DR : Detection Rate) = Rappel = TP / (TP + FN).
- Faux positif (FPR False Positive Rate) = FP / (FP + TN).
- Précision = (TP) / (TP + FP).
- F-mesure = (β2 + 1) (Precision.Recall) / (β2. Précision + Rappel) ; où β = 1.
- Complexité temporelle = Temps prit par un algorithme pour accomplir ses tâches (pour sélectionner le sous-ensemble d'attributs).

4. Taxonomie des algorithmes de la sélection d'attributs

Dans cette section, nous proposons une nouvelle taxonomie de classification pour les algorithmes de sélection d'attributs. Dans cette taxonomie, nous présentons les caractéristiques de chaque algorithme en fonction de sa technique de sélection, le mécanisme de sélection, le type de la solution de sous-ensemble mono/multi-solution, la base de connaissance utilisée, l'approche de sélection, aspects mono ou multi-objectifs et les types des algorithmes de classification pour évaluer leurs performances. La figure 3.2 illustre la taxonomie globale des algorithmes de sélection d'attributs.

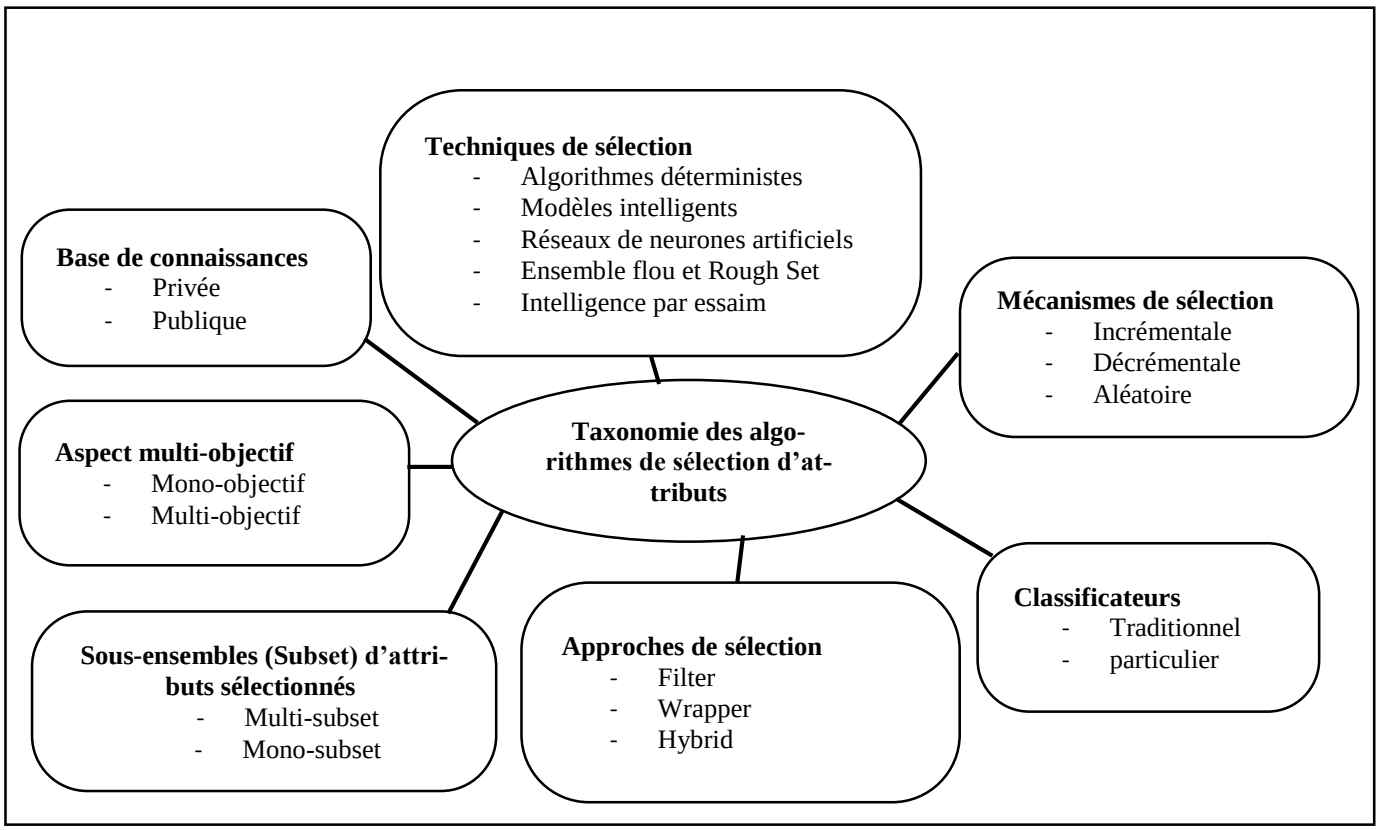

Figure. 3.2 Taxonomie des algorithmes de sélection d'attributs.[132]

Nous présentons les algorithmes développés dans la sélection d'attributs pour la détection d'intrusion en fonction de cette taxonomie. Les algorithmes sont classés en cinq approches selon les techniques qui ont été intégrées : Algorithmes Déterministes, Modèles Intelligents, Réseaux Neuronaux Artificiels, Ensemble Flou et Rough set et Essaim Intelligent.

Nous spécifions l'approche de sélection des algorithmes entre Filtre, Wrapper ou Hybride. Les algorithmes de sélection d'attributs utilisant une approche par Filtre ont une complexité temporelle inférieure à celle de Wrapper car un Filtre est basé sur des mesures de distance, d'information, de dépendance et de cohérence au lieu de Warpper basé sur le taux d'erreur.

De plus, nous intégrons l'aspect de solution unique et multi-solution dans cette classification, cela signifie que l'algorithme a un sous-ensemble d'attributs comme une solution ou un sous-ensemble multiple. Un autre aspect est intégré dans cette classification qui est l'approche mono ou multi-objective qui a été utilisée dans les algorithmes. D'autre part, l'algorithme utilise ou non le Trade-off (Pareto set) dans leur processus de sélection.

Dans chaque section de la classification, nous représentons les algorithmes de classifications traditionnelles telles que DT, SVM, KNN, MLP et BN qui sont utilisés par les étapes d'*évaluation* ou de *validation* pour confirmer les performances des sous-ensembles d'attributs.

En outre, des d'algorithmes développent leurs propre classificateurs particuliers pour assurer l'évaluation des sous-ensembles d'attributs. Dans chaque méthode, on définit les techniques qui sont utilisées avec les résultats obtenus. Une table de résultats qui spécifie le *nombre différent de sous-ensembles* sélectionnés, le *nombre d'attributs* dans le sous-ensemble (s), et le *taux de précision* (Moyenne et Max).

4.1 Algorithmes déterministes

Dans cette section, on présente les algorithmes qui ont été basés sur des mesures *statistiques*, *probabilistes* et de *similarité* pour sélectionner les meilleurs attributs sans redondance et avec pertinence. Parmi ces mesures, on cite l'information mutuelle (IM), l'entropie (E), le coefficient de corrélation (CC), le Chi-Squared, le One-R, le Relief-F et le Gain Ratio, ..., etc. La stratégie a été supportée dans ces algorithmes soit incrémentale, soit Décrémentale. La recherche d'attributs et des résultats des algorithmes est présentée dans les tableaux 3.4 et 3.5.

Table 3.4. Algorithmes déterministes de sélection d'attributs. [132]

Algorithmes	Techniques	Mécanisme	Unique ou Multi-solution	Dataset	Approche	Description
[39]	Informations mutuelles	Incrémental	Multi	KDDcup99	Filter	Analyse de dépendance et de corrélation
[50]	Informations mutuelles	Incrémental	Multi	KDDcup99	Filter	Utilisation de Least Square SVM
[51]	Corrélation et cohérence basées sur FS avec algorithme INTERACT	Incrémental	Multi	KDDcup99	Filter	Combinaison entre multi techniques. (Discrétiseur, filtres et classificateurs)
[52]	Régression des moindres carrés, indice de compression des informations maximales et coefficient de corrélation.	Incrémental	Multi	KDDcup99	Filter	Étude de comparaison entre les techniques de classement d'attributs
[53]	Coefficient de corrélation de Pearson	Incrémental	Unique	NSL-KDD	Filter	Utilisation de la corrélation entre les attributs, puis corrélation entre les attributs sélectionnées et les classes cibles.
[54]	Sélection d'attributs génériques (GFS)	Incrémental	Multi	KDDcup99	Filter	Utilisation de multi SVM classifier.

[55]	Génération d'attributs visualisés	Aléatoire	Multi		Filter	Technique de génération.
[56]	Ratio de gain d'information	Incrémental	Unique	KDDcup99	Filter	Utilisation de SVM classifier.
[4]	Information mutuelle et algorithme de recherche gravitationnelle binaire (BGSA) avec SVM.	Aléatoire	Unique	NSL-KDD	Hybrid	MI intégré dans BGSA avec le classificateur SVM.
[57]	Chi-square et SVM Multi Class	Aléatoire	Unique	NSL-KDD	Hybrid	Utilisation de multi-class SVM classifier
[58]	Chi-square and BN modifié.	Décrémental	Unique	NSL-KDD	Hybrid	Utiliser LDA pour supprimer les attributs nuisible.
[59]	Sélection d'attributs avec corrélation	Incrémental	Unique	NSL-KDD	Filter	Utilisation de BN classifier
[60]	Chi-square, One-R, Relief-F, Information Gain, Gain Ratio, et Symmetrical Uncertainty.	Incrémental	Unique	NSL-KDD	Filter	Étude de comparaison entre les techniques de classement d'attributs et différentes combinaisons de classificateurs. Sous-ensemble d'attributs non-mentionné.

Tableau 3.5. Résultats des algorithmes de sélection d'attributs déterministes. [132]

Algorithmes	**Nombre des sous-ensembles d'attributs**	**Nombre d'attributs dans les sous-ensembles**	**Moyenne**	**Max**
[39]	4	10, 10, 9, et 9	77.6	82.99
[50]	10	6, 15, 13, 7, 8, 36, 10, 4 (R2L), 10, et 3 (U2R)	96.27	97.77
[51]	9	6, 6, 7, 7, 6, 7, 7, 16, et 7	77.02	94.86
[52]	3	10, 20, et 30	81.83	98.14
[53]	1	17	/	99.1
[54]	6	13, 13, 15, 15, 20, et 29	99.4	99.94
[55]	2	4,16	93.73	94.35
[56]	1	10	/	93.34
[4]	1	5	/	88.36
[57]	1	31	/	98
[58]	1	22	/	96.8
[59]	1	12	/	65.43(U2R)
[60]	1	(no mentionné)	/	/

Dans cette section, la plupart des recherches supportent le mécanisme incrémental avec l'approche par Filtre. Selon les mesures de la sélection utilisées (la mesure de classement, Ranking), les algorithmes déterministes sont obligés de suivre le mécanisme incrémental avec un moyen d'arriver aux meilleurs attributs. Ils sélectionnent le premier attribut qui obtient une meilleure valeur de critère et, selon cet attribut et cette mesure d'évaluation, ils sélectionnent l'attribut suivant jusqu'à ce qu'ils obtiennent l'ensemble final qui représente le meilleur sous-

ensemble d'attribut. Des mesures ont été utilisées dans cette section sont basées sur des mesures de classement, mais, ils n'ont pas empêché d'atteindre un bon résultat.

4.2 Modèles intelligents

Dans cette section, on présente différents algorithmes basés sur des techniques d'intelligence artificielle (IA). Nous nous concentrons sur SVM, DT, BN, k-means, Cuttlefish, Système Immunitaire Artificiel (IAS) et l'algorithme évolutif qui est spécifié sur les algorithmes génétiques (GA). Les tableaux 3.6 et 3.7 montrent les algorithmes de cette section avec les résultats obtenus. La plupart de ces recherches sont considérées comme des approches Wrapper ou hybrides. D'autre part, ils ont été fusionnés entre les techniques de séléction pour créer une nouvelle approches par exemple SVM et DT, K-means et SVM…etc. La stratégie a été suivie par ces algorithmes basés sur une sélection Aléatoire.

Ils utilisent des techniques de réduction avec des algorithmes intelligents pour obtenir les meilleurs attributs en minimum de temps. Parmi les techniques de réduction utilisées avec les approches de sélection d'attributs dans cette section, on site : PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), ICA (Independent Component Analysis) and GPC (Genetic Principal Component).

Tableau 3.6 Les modèles intelligents comportent des algorithmes de sélection. [132]

Algorithmes	**Techniques**	**Mécanisme**	**Unique ou Multi-solution**	**Dataset**	**Approche**	**Description**
[43]	Algorithme Génétique et fuzzy rule **(Multi-objective).**	Aléatoire	Unique	KDDcup99	Wrapper	Utilisation multi-objective technique.
[61]	DT (CART), SVM-wrapper, Markov-blanket, Generic Feature Selection (GFS).	Aléatoire	Multi	KDDcup99	Hybrid	Étude de comparaison entre différentes méthodes Wrapper et GFS.
[62]	SVM, DT et Simulated Annealing	Aléatoire	Unique	KDDcup99	Wrapper	Hybrid approach
[63]	SVM et GPC (Genetic Principal component).	Aléatoire	Multi	KDDcup99	Hybrid	utilisation GPC et PCA pour réduire la dimension.
[64]	Hybrid Bat algorithme et SVM.	Aléatoire	Unique	NSL-KDD	Hybrid	Hybrid approach comparé avec PSO-SVM.
[65]	Hierarchical clustering method, information Mutual et DT.	Aléatoire	Multi	KDDcup99	Hybrid	Agglomerative hierarchical clustering avec DT et information Mutuel.
[66]	Système Immunitaire Artificiel	Aléatoire	Unique	KDDcup99	Wrapper	Comparé avec ANN.
[67]	K-means clustering algorithme et SVM	Aléatoire	Multi	KDDcup99	Hybrid	Utilisation du radial basis kernel function (RBF) pour SVM comme un modèle de

						classification. Les attributs sélectionnés sont différentes pour chaque classe d'attaque.
[68]	K-means clustering algorithme	Aléatoire	Multi	NSL-KDD	Wrapper	Utilisation MLP pour la classification.
[69]	Consistency based feature selection , SVM, et LPBoost	Aléatoire	Unique	NSL-KDD	Hybrid	Modèle de fusion.
[70]	Hypergraph-Genetic algorithm et SVM. **(Multi-objective)**	Aléatoire	Unique	NSL-KDD	Hybrid	Utilisation d'une fonction objective pondérée (Trade-off) entre le taux de détection maximum et le taux de fausse alarme minimum.
[71]	Vote algorithme et Gain d'Information	Aléatoire	Unique	NSL-KDD	Hybrid	Estimer le seuil de portée de l'intrusion. Utilisant différents classificateurs : DT, Meta Pagging, RandomTree, REPTree, AdaBoostM1, DecisionStump, et BN.
[72]	Algorithme Génetique et logistic regression	Aléatoire	Multi	KDDcup99	Wrapper	En utilisant différents classificateurs de type DT.

Tableau 3.7 Résultats des algorithmes de sélection d'attributs des modèles intelligents. [132]

Algorithmes	**Nombre des sous-ensembles**	**Nombre d'attributs dans les sous-ensembles**	**Moyenne**	**Max**
[43]	1	25	/	92.76
[61]	5	17, 18, 17, 12, 4, et 5	98.22	99.6
[62]	1	23	/	99.96
[63]	2	10 et 12	99.95	99.96
[64]	1	23	/	99.28
[65]	7	8, 10, 9, 11, 13, 12, et 14	93.35	93.8
[66]	1	21	/	99.1
[67]	5	41, 30, 26, 29, et 35	91.02	/
[68]	20	Entre 16 to 26	96.93	99.73
[69]	1	10	/	96.2
[70]	1	35	/	96.72
[71]	1	8	/	99.81
[72]	8	18, 15, 20, 17, 16, 18, 22, et 18	99.34	99.5

4.3 Réseaux de neurones artificiels

Les réseaux neuronaux artificiels (RNA) (ANN : Artificial Neural Networks) [1] [54] [73] sont une technique qui s'inspirent des neurones du cerveau humain. Ce sont des neurones interconnectés que chaque neurone représente une unité de traitement. Ces collections d'unités de traitement ont la capacité d'apprendre à résoudre des problèmes. Il existe différents types d'ANN qui sont classés en différentes catégories selon l'apprentissage supervisé ou non supervisé, et l'architecture Feed-forward ou récurrente.

Différents travaux de recherche ont été proposés sur la sélection d'attributs pour l'IDS qui intègre l'ANN dans leurs solutions. Les tableaux 3.8 et 3.9 représentent les différents algorithmes basés sur l'ANN pour résoudre le problème de la sélection d'attributs pour IDS.

Table 3.8. ANN feature selection algorithms. [132]

Algorithmes	Techniques	Mécanisme	Unique ou Multi-solution	Dataset	Approche	Description
[74]	Arbre neuronal flexible hybride	Aléatoire	Unique	DRAPA98	Wrapper	En utilisant un algorithme évolutif et des paramètres par PSO. Pour chaque classe a son sous-ensemble d'attributs.
[73]	Rétro-propagation réseau neuronal.	Aléatoire	Unique	KDDcup99	Wrapper	Utiliser ICA (Independent Component Analysis) pour éliminer les entrées non-significatives et / ou inutiles.
[75]	Réseau neuronal Rétro-propagation et algorithme génétique.	Aléatoire	Unique	KDDcup99	Wrapper	Processus de classification multi-classes
[1]	Hierarchical self-organizing maps. **(Multi-objective)**	Aléatoire	Multi	KDDcup99	Wrapper	Approche multi-objectif
[76]	Hybridation du réseau neuronal et K-Means Clustering.	Aléatoire	Unique	NSL-KDD	Hybrid	Utiliser PCA pour réduire la complexité de calcul.
[77]	Réseau neuronal artificiel	Incrémental	Unique	KDDcup99	Hybrid	Utiliser le gain d'information et la corrélation.

ANN est intégrée dans le processus de sélection d'attributs, tel qu'un classificateur dans l'étape de validation, mais différents travaux utilisés l'ANN comme un algorithme de sélection d'attributs. Les réseaux de neurones de rétro-propagation et les cartes d'auto-organisation sont les plus utilisés dans la zone FS. ANN est utilisé avec d'autres techniques comme GA et K-means, pour cela, la plupart des algorithmes de ANN utilisent un mécanisme aléatoire avec des approches Wrapper et Hybrides.

Tableau 3.9. Résultats des algorithmes de sélection d'attributs ANN. [132]

Algorithmes	Nombre des sous-ensembles	Nombre d'attributs dans les sous-ensembles	Moyenne	Max
[74]	5	4, 13, 12, 8, et 10	99.02	99.75
[73]	1	8	/	99.5
[75]	23 (pour chaque attaque son sous-ensemble)	Entre 10 to 17	58.39	86
[1]	5	22, 29, 25, 25, et 29	98.13	99.12
[76]	1	23	/	97.63
[77]	1	25	/	97.91

4.4 Ensemble flou et Rough set

Fuzzy Logic (FL) [78] [79] et Rough Set (RS) [80] [81] sont deux techniques d'intelligence artificielle qui sont utilisées pour résoudre les problèmes des ensembles de données incertains, incohérents et incomplets. La logique floue est une extension de la logique classique et de la théorie des ensembles. Il a la capacité de définir des règles de décision en utilisant des concepts vagues pour résoudre les différents problèmes réels. En outre, Rough set est un cadre formel qui est utile pour les connaissances découvertes et analyser les données pour les problèmes NP-Hard.

FS est parmi les domaines qui ont été intéressés à intégrer Fuzzy Logic et Rough Set. Les tableaux 3.10 et 3.11 illustrent les travaux récents qui ont été développés dans la sélection d'attributs pour l'IDS en utilisant FL et RS.

Tableau 3.10. Algorithmes de sélection d'attributs de l'ensemble Fuzzy & Rough set.[132]

Algorithmes	Techniques	Mécanisme	Unique or Multi-solution	Dataset	Approche	Description
[80]	Rough set and fuzzy	Aléatoire	Unique	KDDcup99	Wrapper	Approche combinée
[82]	Méthode de classification floue multicritère	Aléatoire	Unique	KDDcup99	Wrapper	Combiné avec une sélection d'attributs gourmands.
[83]	Langage de contrôle flou	Aléatoire	Multi	KDDcup99	Wrapper	ntégration de la selection d'attributs basée sur Entropie.
[81]	Rough set et NetFlow/IPFIX	Aléatoire	Multi	KDDcup99	Wrapper	utilisation de KNN comme un classifier.
[84]	Rough set et Hypergraph Technique	Aléatoire	Unique	KDDcup99	Filter	Utilisation de la linéarité transversale minimale et de la linéarité des sommets pour l'identification du sous-ensemble d'attributs optimal.

Tableau 3.11 : Résultats d'algorithmes de sélection d'attributs Fuzzy & Rough set.[132]

Algorithmes	Nombre des sous-ensembles	Nombre d'attributs dans les sous-ensembles	Moyenne	Max
[80]	4 (pour chaque attaque son sous-ensemble)	5, 5, 4, et 4	94.15	99.75
[82]	1	11	/	99.96
[83]	3	14, 17, et 21	99.24	99.66
[81]	6	11, 16, 16, 16, 16, et 17	90.33	98
[84]	1	23	/	96.63

FL et Rough Set ont été principalement intégrés dans la sélection d'attributs pour obtenir l'attribut minimal de toutes les combinaisons possibles. Les recherches qui utilisent le FL et RS sont associées sous l'approche de l'enveloppe et le mécanisme de sélection aléatoire. Les travaux effectués dans FL et RS pour la sélection d'attributs sont fusionnée avec d'autres techniques comme une approche pour plus de précision ou comme un algorithme de classification pour évaluer leur efficacité.

4.5 Intelligence par essaim

Dans cette section, nous présentons la technique des essaims (SI : Swarm Intelligence) [40] [85], qui est une technique d'intelligence artificielle. Elle est inspirée du comportement émergent des insectes sociaux et des essaims. SI est basé sur les individus qui interagissent entre eux et les environnements pour optimiser les objectifs par une recherche collaborative.

Elle est utilisée pour résoudre les problèmes complexes en appliquant une intelligence collective sophistiquée. Chaque individu représente une solution potentielle et tous présentent la population de solutions. Deux techniques célèbres de SI sont présentées dans cette section, qui sont : l'optimisation des colonies de fourmis (ACO : Ant Colony Optimization) et l'optimisation des essaims de particules (PSO : Particle Swarm Optimization). L'ACO et le PSO ont été utilisées pour résoudre le problème de la sélection d'attributs pour IDS.

4.5.1 Optimisation des colonies de fourmis

ACO [40] [85] [86] est une inspiration du comportement réel des fourmis, qui veulent trouver le plus court chemin entre la colonie et les sources de nourriture. Chaque individu de la population est présenté par les fourmis avec ses phéromones.

ACO est basé sur leur phéromone et leurs fourmis pour trouver la solution optimale dans l'espace de recherche. Les ACO ont été appliqués pour résoudre les problèmes d'optimisation discrets. En outre, il est utilisé dans la sélection d'attributs pour l'IDS avec une approche intéressante, mais il est encore limité.

Dans cette section, on présente trois approches d'amélioration qui sont [86] [87] [88]. Les tableaux 3.12 et 3.13 illustrent les différentes approches proposées par la colonie de fourmis pour la sélection d'attributs et leurs résultats.

Tableau 3.12. Algorithmes de sélection d'attributs ACO. [132]

Algorithmes	Techniques	Mécanisme	Unique ou Multi-solution	Dataset	Approche	Description
[87]	Optimisation des colonies de fourmis	Aléatoire	Unique	KDDcup99	Wrapper	Utilisation de SVM pour la détection.
[86]	Optimisation des colonies de fourmis, K-means et SVM	Décremental	Multi	KDDcup99	Wrapper	Combinaison entre plusieurs techniques.
[88]	ACO et fuzzy entropy	Aléatoire	Unique	Privé	Wrapper	Combinaison d'approaches.

Tableau 3.13 : Résultats des algorithmes de sélection d'attributs ACO. [132]

Algorithmes	Nombre des sous-ensembles	Nombre d'attributs dans les sous-ensembles	Moyenne	Max
[87]	1	32	/	97.76
[86]	4	10, 10, 10, et 19	96.78	98.62
[88]	1	13	/	99.69

4.5.2 Optimisation par essaim de particules

PSO [40] est parmi les techniques utilisées dans la sélection d'attributs pour l'IDS. Elle est inspirée de la simulation du comportement social des oiseaux. PSO a été utilisé pour résoudre le problème global d'optimisation non linéaire avec des contraintes. PSO est basé sur la fonction fitness, la vitesse et la position de chaque particule pour obtenir la solution optimale.

En PSO, chaque solution partielle (individuelle) est codée par un vecteur. PSO à deux versions, qui sont : discrètes et continues selon le problème, le type de donnée et la population. Sur la base des mécanismes qui ont été intégrés dans PSO, différentes recherches ont été proposées pour rechercher les meilleurs sous-ensembles d'attributs pour le système de détection d'intrusion.

Les tableaux 3.14 et 3.15 montrent les travaux ont été basés sur les PSO et leurs résultats.

Tableau 3.14 : Algorithmes de sélection d'attributs PSO. [132]

Algorithmes	Techniques	Mécanisme	Unique ou Multi-solution	Dataset	Approche	Description
[89]	PSO et rough set.	Aléatoire	Unique	KDDcup99	Wrapper	Rough-DPSO algorithme entre RS and PSO
[90]	PSO. **(Multi-objective)**	Aléatoire	Unique	KDDcup99	Wrapper	**Approche Multi-objective.**
[91]	PSO et Random forest.	Aléatoire	Multi	KDDcup99	Wrapper	**Approche Multi-objective.**
[2]	PSO et les algorithmes Génétique	Aléatoire	Multi	NSL-KDD	Hybrid	Utilisation de PCA.
[92]	Multi-objective PSO. **(Multi-objective)**	Aléatoire	Unique	KDDcup99	Wrapper	Faire face aux attaques en temps réel
[93]	PSO avec trois algorithmes de classification	Aléatoire	Unique	NSL-KDD	Wrapper	Utilisation de trois types d'arbre de décision (CART, Random forest, et C4,5).
[94]	PSO et Bat algorithme	Aléatoire	Multi	NSL-KDD	Wrapper	Etude comparative sur l'intelligence d'essaim basée sur FS. En utilisant deux versions du Bat algorithme (BAL et BAE).

Tableau 3.15 : Résultats des algorithmes de sélection d'attributs PSO.[132]

Algorithmes	Nombre des sous-ensembles	Nombre d'attributs dans les sous-ensembles	Moyenne	Max
[89]	1	6	93.40	95.35
[90]	1	6	/	94.15
[91]	9 pour PROB attaque	7, 13, 20, 15, 13, 30, 17, 14, et 3	85	100
[2]	2	8 et 10	98.8	99.4
[92]	1	11	/	98
[93]	3	11, 9, et 6	99.47	99.8
[94]	20	Entre 13 et 22	93.63	97.17

On a remarqué que les travaux ACO et PSO ont été intégrés dans Wrapper avec des approches hybrides. Ils utilisent un mécanisme incrémental ou aléatoire dans leurs algorithmes de sélection d'attributs. ACO et PSO ont besoin de techniques d'orientation pour guider leur exploration de l'espace de recherche. Pour cela, ACO et PSO sont combinés avec d'autres techniques pour orienter leur recherche dans l'espace de recherche. Des classificateurs de déflection ont été utilisés avec ACO et PSO pour calculer les performances de chaque sous-ensemble d'attributs.

5. Conclusion

Les entreprises utilisent les IDS pour garder leurs systèmes dans toute leur fiabilité et hors de la main des pirates et les différentes attaques. L'amélioration des IDS est une politique incontestable par rapport à l'évolution des menaces et les techniques de piratage intelligent. FS

est une solution pour améliorer le *taux*, le *temps de détection*, et *simplifier* la complexité des modèles de classification.

Dans ce chapitre, nous avons représenté un aperçu de la sélection d'attributs pour les systèmes de détection d'intrusion. Nous avons exploré les contributions dans l'application de la FS au problème de l'IDS par un aperçu des différents travaux. Une nouvelle taxonomie a été proposée, qui portait sur les techniques de sélection, les mécanismes de sélection, la solution unique ou multiple de sous-ensemble, les bases de connaissances utilisées, le type d'approche et l'aspect mono ou multi-objectif. Les différents algorithmes de sélection d'attributs sont classés en cinq classes en fonction de leurs techniques, qui sont utilisées dans chaque œuvre. Nous avons présenté leurs caractéristiques en fonction de la nouvelle taxonomie, et nous avons illustré leurs résultats obtenus en montrant leur nombre de sous-ensembles, le nombre d'attributs dans le sous-ensemble et le taux de précision (Moyenne, Max)). Nous avons considéré cette étude exhaustive de recherche comme une carte pour comprendre l'état actuel et les défis futurs.

Chapitre 04

Algorithme à estimation de distribution et l'optimisation multi-objectif

1. Introduction

L'optimisation par méta-heuristiques est devenue un sujet de recherche vital ces dernières années. L'optimisation fait référence à la recherche de la meilleure solution possible à un problème en fonction d'un ensemble de limitations (ou de contraintes). Lorsqu'on traite un seul objectif à optimiser, nous cherchons à trouver la meilleure solution possible ou du moins une bonne approximation.

Cependant, lors de la conception de modèles d'optimisation pour un problème, c'est souvent le cas qu'il n'y en a pas un mais plusieurs objectifs que nous aimerions optimiser. En fait, c'est le cas où ces objectifs sont en conflit les uns avec les autres. Ces problèmes avec deux ou plusieurs fonctions objectives sont appelés "multi-objectif" et nécessitent des outils mathématiques et différents algorithmiques de ceux adoptés pour résoudre des problèmes d'optimisation à multi objectif.

En fait, même la notion de «changement d'optimalité» lorsqu'il s'agit de problème d'optimisation multi-objectif. En outre, de nombreux algorithmes évolutifs (EA) tels que GA, Evolution Différentielle (DE) et Estimation de l'Algorithme de Distribution (EDA) ont été intégrés dans le problème d'optimisation multi-objectif (MOEAs: Algorithms Evolutifs Multi-Objectives).

En fonction de leur nature basée sur la population, les EA sont appropriés pour MOO et sont capables de trouver et d'approximer les solutions non dominantes PS (Pareto Set) en une seule série (ensemble de Pareto, (PF : Front de Pareto)). MOEAs ont été utilisés dans plusieurs domaines en raison de leur robesteness que d'autres techniques, en particulier dans les problèmes multimodaux, les problèmes de nombreux objectifs, des problèmes MOO dynamiques.

EDA (Estimation of Distribution Algorithms) est un nouvel EA (evolutionary algorithms) qui est un algorithme d'optimisation stochastique basé sur un modèle probabiliste à la place des opérateurs de croisement et de mutation. EDA a une meilleure crédibilité pour résoudre le problème d'optimisation en termes de taille et de complexité. Les EDA sont basées sur les solutions candidates de la population, dans chaque génération, évaluent la solution avec la fonction de condition pour en extraire le meilleur, et utiliser le modèle de probabilité sur ces solutions pour échantillonner la nouvelle génération.

Par conséquent, avec un bon modèle probabiliste qui représente les relations entre les variables, EDA a le pouvoir de nous donner une exploration efficace de l'espace de recherche.

Dans ce chapitre, nous discutons sur l'optimisation multi-objectif avec les différents aspects de dominances et de Pareto Set. Nous présentons les algorithmes d'estimation distribuée.

Nous citons les différents algorithmes élitistes qui sont utilisés pour la sélection des solutions non dominées. Par la suite, nous présentons une vue globale sur les techniques de l'information mutuelle utilisées dans le domaine de sélection avec les différents types utilisés comme IMC, Co-information et II Interaction Information.

2. Optimisation multi-objectif

Selon [95] [96] [97] [98] quand, on cherche les meilleures solutions pour le problème qui minimise ou maximise plusieurs fonctions d'objectif de conflit avec contrainte. On trouve des solutions multiples (de plus, il n'y a pas de définition précise de la solution optimale). Par conséquent, on devrait comparer entre ces solutions pour extraire la meilleure, parce que, on ne peut pas optimiser tous les objectifs simultanément sans tombe dans des conflits entre eux. On cherche un compromis optimal entre les objectifs de conflit (les meilleures solutions de compromis, les solutions optimales de Pareto).

Le problème d'optimisation multi-objectif (MOO) est représenté mathématiquement selon [96] [98] [99] tels que :

$$minimise\ F(x) = [f_1(x), f_2(x), \dots, f_k(x)]^{\mathrm{T}}$$

$$\text{Avec : } g_i(x) \leq 0, i = 1, 2, \dots, m.$$

$$h_i(x) \leq 0, i = 1, 2, \dots, l.$$

Où x est le vecteur des variables de décision, f_i (x) est une fonction de x, K est le nombre de fonctions objectives à minimiser, g_i (x) et h_i (x) sont les fonctions de contrainte du problème. m et l sont des nombres entiersOù x est le vecteur des variables de décision, f_i (x) est une fonction de x, K est le nombre de fonctions objectives à minimiser, g_i (x) et h_i (x) sont les fonctions de contrainte du problème, m et l sont des nombres entiers

2.1 Définitions

Selon [95] [96] [97] [98] [100] multi objectif contient déférents définitions et vocabulaire qui il faut d'être présenté comme suit :

- ***Fonction objective***

C'est le nom donné à la fonction *f* (on l'appelle encore **fonction de coût** ou **critère d'optimisation**). C'est cette fonction que l'algorithme d'optimisation va devoir "optimiser" (trouver un optimum). [100]

- ***Variables de décision***

Elles sont regroupées dans le vecteur $\vec{x}$. C'est en faisant varier ce vecteur que l'on recherche un optimum de la fonction *f*. [100]

- ***Minimum global***

Un "point" $\vec{x}^{*}$ est un minimum global de la fonction *f* si on a : $f(\vec{x}^{*}) < f(\vec{x})$ quel que soit $\vec{x}$ tel que $\vec{x}^{*} \neq \vec{x}$. [100]

En optimisation multi-objectif, la qualité d'une solution est expliquée en termes de compromis entre des objectifs contradictoires.
Soit y et z deux solutions du problème de minimisation de l'objectif K ci-dessus. Si des conditions suivant est rempli, on peut dire que y domine z ou y vaut mieux que z :

$$\forall\, i : f_i(y) \leq f_i(z) \;\; et \; \exists\, j : f_j(y) < f_j(z)$$

Pour en savoir plus sur MOO, nous introduisons la définition de la *Dominance*, la *solution optimale de Pareto* (solutions de compromis), *l'ensemble optimal de Pareto* et le *front de Pareto* [96] [97] [98] :

- ***Relation de dominance***

Un vecteur $u = (u_1, \dots, u_k)^{\mathrm{T}}$ est dit dominer un autre vecteur $v = (v_1, \dots, v_k)^{\mathrm{T}}$ (on note par $u \prec v$). Si seulement si : $\forall\, i \in \{1, \dots, k\},\ u_i \leq v_i \ \wedge\ \exists\, i \in \{1, \dots, k\} :\ u_i < v_i$.

- ***Pareto Optimal solution***

Une solution $x \in \Omega$ est appelée Pareto Optimal Solution si seulement si $\nexists\, y \in \Omega$ pour que $F(y) \prec F(x)$.

- ***Pareto Set (PS)***

Pareto Set est l'ensemble de tous les Pareto optimales solutions avec :

$$PS = \{x \in \Omega \mid \nexists y \in \Omega\, , F(x) \prec F(y)\}.$$

- ***Pareto Front***

Pareto Front (PF) est l'image de PS dans l'espace objectif avec : $PF = \{F(x) \mid x \in PS\}$

2.2 Les algorithmes d'optimisation multi-objectif

Dans cette section, nous présentons les algorithmes multi-objectifs évolutionnaires les plus typiques [101] qui ont été utilisés pour classer et mettre à jour les solutions dans MOO. Ces algorithmes sont basés sur le Pareto. Selon [102] [103], les algorithmes sont classés dans deux catégories : élitistes et non élitistes. Ce qui fait la différence entre eux est la population archive. Les algorithmes non élitistes sont caractérisés par l'absence de la population archive par contre

les algorithmes élitistes contiennent la population secondaire (archive) pour sauvegarder les meilleures solutions trouvées pendant toutes les itérations. Selon [103] les algorithmes non élitistes sont caractérisés par la difficulté de maintenir la diversité sur la frontière de Pareto, la convergence vers la frontière de Pareto est lente, et ne conserve pas les individus Pareto-optimaux.

Les algorithmes élitistes sont spécifiés par l'utilisation d'une population externe (archive), préfèrent les solutions non dominées, et utilisation des techniques de formation de niches, partitionnement en cluster, et maillage. A la base des travaux effectués dans [102] [103] [131], le tableau 4.1 illustre les différents algorithmes élitistes et non élitistes avec leur description.

Tableau 4.1 : Algorithmes élitistes et non élitistes avec leur description.

Algorithmes	Type	Description
MOGA [104] (Multiple Objective Genetic Algorithm)	Non élitiste	Chaque individu de la population est rangé en fonction du nombre d'individus qui le dominent.
NSGA [105] (Non dominated Sorting Genetic Algorithm)	Non élitiste	Le calcul du fitness s'effectue en séparant la population en plusieurs groupes en fonction du degré de domination au sens de Pareto de chaque individu
NPGA [106] (Niched Pareto Genetic Algorithm)	Non élitiste	Sélection, tournoi de dominance sur un sous ensemble de la population.
SPEA [107] (Strength Pareto Evolutionary Algorithm)	Elitiste	Pour chaque individu de l'ensemble Pareto-optimal. Le fitness est proportionnel au nombre d'individus de la population qu'il domine. Dépendance par rapport à la taille de l'archive.
PAES [108] (Pareto Archived Evolution Strategy)	Elitiste	Non basée sur une population, elle n'utilise qu'un seul individu à la fois pour la recherche des solutions. Elle utilise une technique de *crowding* basée sur un découpage en **hypercubes** de l'espace des objectifs
PESA [109] (Pareto Envelope based Selection Algorithm)	Elitiste	Basée sur les AGs. Reprend le principe de crowding développé dans PAES (Mesure d'encombrement d'une zone de l'espace, Utilisée lors de la sélection et de la mise à jour de l'archive)
NSGAII [110] (Non dominated Sorting Genetic Algorithm II)	Elitiste	Amélioration de NSGA. Sélection, tournoi classique et préférence en fonction du degré d'encombrement de l'espace.
PESA II [111] (Pareto Envelope based Selection Algorithm II)	Elitiste	Amélioration de PESA. Cette méthode se montre plus efficace pour repartir les solutions sur la frontière de Pareto.
SPEAII [112] (Strength Pareto Evolutionary Algorithm II)	Elitiste	Amélioration de SPEA

Ici, nous nous intéressons à l'algorithme FNS (Fast Non-dominate Sorting) et à la Crowding Distance de NSGAII pour les impliquer dans l'approche proposée. L'algorithme FNS divise la population P de solutions en un certain nombre de classes distinctes Aj de la façon suivante : Tous les non-dominés de P appartiennent à l'ensemble A_1 ; alors tous les éléments non dominés de $P_{-\{A1\}}$ sont placés dans l'ensemble A_2 et ainsi de suite jusqu'à ce qu'ils règlent l'ensemble de la population. Au final, NSGAII sélectionne les solutions A_1 comme une nouvelle population

3. Estimation de l'algorithme de distribution

EDA (Estimation des Algorithmes de Distribution [95] [113] est un nouvel algorithme évolutionnaire (EA : Evolutionary Algorithms) qui est un algorithme d'optimisation stochastique basée sur un modèle probabiliste à la place des opérateurs de croisement et de mutation. L'EDA appartient à EA, utilise le modèle de probabilité (réseau bayésien, interaction multi variée... etc.) pour générer de nouvelles solutions candidates. Comme EDA a été hérité d'EA, il partage les mêmes caractéristiques que les algorithmes évolutionnaires multi-objectifs (MOEA) tels que l'utilisation de critères d'évaluation telle que les fonctions de remise en forme.

EDA a la capacité de trouver une solution approximative, multiple et non dominée PS (Pareto Set) en un seul passage [98] [101]. EDA a le pouvoir d'explorer efficacement des espaces de recherche [113] [114] [115]. EDA à une grande crédibilité en résolvant le problème d'optimisation concernant la taille et la complexité. Le problème d'optimisation pour EDA est envisagé comme des techniques d'optimisation stochastiques. L'algorithme 1 montre la procédure de base de l'EDA typique basée sur [95] [113]

Algorithm 1 : EDA Typical Algorithm

Input: Objective Function f
Output: Better Solutions P_t

1: Generate the initial population P_0
2: $t \leftarrow 0$
3: **while** (Note done) **do**
4: Select the promising solution (S_t) according to f from P_t
5: Build the probabilistic model (PM_t) from S_t
6: Sample from (PM_t) to generate the new candidate solution NP_t
7: Incorporate NP_t into P_t
8: $t \leftarrow t + 1$
9: **end while**

Les EDA sont basées sur les solutions candidates de la population. A chaque itération, EDA évalue les solutions par une fonction de remise en forme pour en extraire la meilleure et utilise le modèle probabiliste sur ces solutions pour échantillonner la nouvelle population pour une nouvelle génération. Le processus est répété jusqu'à ce que la condition d'arrêt soit remplie.

Au final, EDA obtient les meilleures solutions. Ainsi, avec un bon modèle probabiliste qui représente la structure complexe des relations entre les variables, nous obtient des solutions optimales avec un nombre moins élevé d'itérations.

Les choses les plus importantes que nous devons prendre en compte lorsque nous utilisons l'EDA sont le modèle probabiliste, une méthode d'apprentissage, une méthode d'échantillonnage et une méthode de remplacement. Parce qu'ils définissent la manière dont l'EDA donne la

solution à tout problème. Il y a beaucoup de techniques qui ont été utilisées pour construire l'EDA spécifiquement le modèle probabiliste. Hauschild et al [113] divisé en fonction de la relation entre les variables en trois catégories (Univariate, Tree-based model, et Multivariate).

EDA a été considéré parmi les techniques qui sont utilisées pour résoudre le problème d'optimisation pour un objectif, en outre, c'est une bonne technique pour MOO. Il y a plusieurs recherches utilisées un EDA multi-objectif pour résoudre leurs problèmes tels que : [95] [115] [116].

4. Information mutuelle

Mesure de l'information comme l'entropie et l'information mutuelle est très importante dans la théorie de l'information pour mesurer la corrélation entre les variables. L'entropie (H) a été considérée comme une mesure de l'incertitude des variables aléatoires. De plus, l'IM a été utilisé pour mesurer la corrélation entre les variables [118].

L'IM a été défini pour mesurer la quantité d'informations et la dépendance entre les variables aléatoires. C'est une mesure de la pertinence et de la quantité de connaissances que contient la variable aléatoire X à propos d'une autre variable Y [50] [117] [118].

L'entropie a été définie mathématiquement par [118] [119] comme : X variable aléatoire discrète avec alphabet Ω et la distribution de probabilité de X est $p(x) = pr\{X = x\}$, $x \in \Omega$ l'entropie de X est :

$$H(x) = -\sum_{x\in\Omega} p(x)\log p(x)$$

Pour deux variables aléatoire (X, Y) avec une distribution conjointe $p(x,y)$, l'entropie conjointe H (X,Y) est [118] :

$$H(X,Y) = -\sum_{x\in\Omega}\sum_{y\in\gamma} p(x,y)\log p(x,y)$$

Avec

$$H(X,Y) = -H(X) + H(Y/X)$$
$$H(X,Y) = -H(Y) + H(X/Y)$$

Mathématiquement, l'information mutuelle entre deux variables aléatoires X, Y avec leur fonction de masse de probabilité $p(x,y)$ et la fonction de masse de probabilité marginale $p(x), p(y)$ est définie comme [118] [120] [121] :

$$IM(X;Y) = \sum_{x\in\Omega}\sum_{y\in\gamma} P(x,y)\log P(x,y)/P(x)P(y)$$

L'autre définition de l'entropie relative entre la distribution conjointe et la distribution du produit $P\ (x)$, $P\ (y)$ tels que :

$$IM(X;Y) = H(X) - H(Y/X)$$
$$IM(X;Y) = H(Y) - H(X/Y)$$
$$IM(X;Y) = H(X) + H(Y) - H(Y,X)$$
$$IM(X;X) = H(X)$$
$$IM(X;Y) = IM(Y;X)$$

Mesure d'information multi variée (MIM : Mutual Information Multivariate) [121] [122] a été nécessaire plus que l'information à deux variable pour évaluer et distinguer la relation importante de synergie et de redondance entre les variables. Pour cela, nous clarifions les définitions de les informations d'interaction (II : Interaction Information), Co-information (Co-Inf : Co-Information) et information mutuelle conditionnelle (CMI : Conditionnal Mutual Information) parce que dans les littératures ont fournis une mesure distincte qui donne un résultat différent et n'a pas été cohérent.

D'après [118] [120] [121] [123] le CMI des variables aléatoires X et Y étant donné Z est :

$$IM(X;Y\ /Z) = \sum_{x\in\Omega}\sum_{y\in\gamma}\sum_{z\in\Psi} P(x,y,z)\log p(z)p(x,y,z)\ /p(x,z)p(y,z)$$
$$= \sum_{x\in\Omega}\sum_{y\in\gamma}\sum_{z\in\Psi} P(x,y,z)\log p(x,y\ /z)\ /p(x\ /z)p(y\ /z)$$
$$IM(X;Y\ /Z) = H(X\ /Z) - H(X\ /Y,Z)$$

Selon Srinivasa [124] :

$$IM(X_1;\ X_2;\ldots;\ X_n) = IM(X_1;\ X_2;\ldots;\ X_{n-1}) - IM(X_1;\ X_2;\ldots;\ X_{n-1}\ /X_n)$$

Pour cela

$$IM(X_1;\ X_2;\ X_3) = IM(X_1;\ X_2) - IM(X_1;\ X_2\ /X_3)$$

Pour les trois variables on a :

$$IM(X;\ Y;\ Z) = \left[\sum_{x\in\Omega}\sum_{y\in\gamma}\sum_{z\in\Psi} P(x,y,z)\log p(z)p(x,y,z)\ /p(x,z)p(y,z)\right]$$
$$-\sum_{x\in\Omega}\sum_{y\in\gamma} P(x,y)\log P(x,y)\ /P(x)P(y)$$

Les informations d'interaction (II) introduites par [121] [122] [123] sont définies comme suit :

$$II(X;Y;Z) = IM(X;Y\ /Z) - IM(X;Y) = IM(X;Z\ /Y) - IM(X;Z)$$
$$= IM(Z;Y\ /X) - IM(Z;Y)$$
$$II(X;Y;Z) = IM(X;Y;Z) - [IM(X;Z) + IM(Y;Z)]$$

Co-information selon [121] [125] est :

$$Co - Inf(X;Y;Z) = IM(X;Y) - IM(X;Y/Z)$$
$$= IM(X;Z) + IM(Y;Z) - IM(X,Y;Z)$$

Co-Inf est égal à II, lorsque le nombre d'entités est pair et que Co-Inf est égal au négatif II lorsque les entités sont impaires

5. Informations mutuelles avec sélection d'attributs

Dans cette section, on représente les différents travaux de sélection d'attributs qui ont intégré IM dans leur processus de sélection. Avant de citer ces travaux, nous mentionnons que nous utilisons la même signification de la section 2.1 Définition (Chapitre 3) avec $f_{i\neq j}$ le nouvel attribut quand doit évaluer et ajouter dans le (s) meilleur (s) ensemble (s) d'attributs S, s'il atteint les valeurs de sélection et f_s est l'ensemble d'attributs qui sont évalués avant f_i et qui appartiennent à S (le sous-ensemble sélectionné).

- Qu et al [39] proposent deux métriques $Q_C\ (f_i, f_j)$ et $e\ (s)$ pour mesurer la pertinence et la fonction de corrélation. Ces nouvelles métriques sont basées sur l'information mutuelle et l'entropie, qui ont obtenu un bon taux de détection.

$$Q_C\ (f_i, f_j) = \frac{\left(IM(Y;f_i) + IM(Y;f_j) - IM(Y;f_i,f_j)\right)}{H(C)}$$

$$e(S) = \left(\sum_j \frac{IM(C;f_i)}{H(C)}\right) - \sum_{i,j} Q_C(f_i, f_j)$$

- Ding et al [117] proposent mRMR qui utilise une fonction basée sur une redondance minimale- avec une pertinence maximale :

$$IM(C;f_i) - (1/|s|)\sum_{f_s \in s} IM(f_c;f_i)$$

- Amiri et al [50] proposent deux algorithmes pour la sélection d'attributs. le premier a été reposé sur la sélection d'attributs basée sur la corrélation linéaire (LCFS : Linear Correlation Feature Selection) et le deauxième basé sur l'information mutuelle modifiée (MMIFS : Modified Mutual Information Feature Selection) qui est calculée par la fonction suivante :

$$IM\ (C;f_i) - (\frac{\beta}{|s|})\sum_{f_i \in s} IM\ (f_s;f_i)$$

Cette fonction a été utilisée pour évaluer plusieurs attributs pour sélectionner le meilleur d'entre eux. Ils remplissent leur fonction sur l'ensemble de données KDD99.

- Estévez et al [126] proposent un nouvel algorithme basé sur IM nommé NMIFS (Normalized Mutual Information Feature Selection). Il est basé sur la fonction d'évaluation comme :

$$NI(f_i; f_s) = IM\ (f_s; f_i)/min\{H(f_i), H(f_s)\}$$

$$G = IM(C; f_i) - \left(\frac{1}{|s|}\right) \sum_{f_s \in S} NI(f_i, f_s)$$

- Battiti [127] propose MIFS (Mutual Information Feature Selection) pour rechercher les meilleurs attributs. Le MIFS utilise deux formules $IM(C;\ fi)$ et $IM(f_s;\ fi)$ pour sélectionner la fonction suivante, telle que :

$$IM(C;\ fi) - \beta \sum_{f_s \in S} IM(f_s;\ fi)$$

- Kwak et al [120] présentent l'algorithme MIFS-U (2002) (Mutual Information Feature Selection- Uniforme) basé sur IMC :

$$IM(C; f_i; f_s) = IM(C;\ fs) + IM(C;\ fi\ /f_s)$$

Quand $IM(C;\ fs)$ est commun à tous les attributs, elles se concentrent sur $IM(C;\ fi\ /f_s)$. Ils utilisent la fonction suivante pour construire la fonction finale de sélection :

$$IM(C;\ fi\ /f_s) = IM(C;\ fi) - \{IM(f_s;\ fi) - IM(fs;\ fi\ /C)\}$$

La fonction finale de MIFS-U a été utilisée pour la sélection est :

$$IM(C;\ fi) - \beta \sum_{f_s \in S} (IM(f_C;\ fi)\ /H(f_s))\ IM(f_s;\ fi)$$

- Aussi Foithong et al [119] développent un algorithme de sélection d'attributs basé sur IM et la mesure Rough Set. La fonction d'évaluation de IM a été utilisée est basée sur le IMC de [120] pour démontrer leur fonction d'évaluation telle que :

$$G =\ IM(C; f_i) - (H(f_i/C)/H(f_i)) \sum_{f_s \in S} R(f_i; f_s)$$

$$R(f_i; f_s) = \frac{IM(C; f_i)\ IM(f_s; f_i)}{(H(f_s)H(f_i))}$$

- Qin et al [37] présentent CMIFS algorithme (Conditional Mutual Information Feature Selection) pour la sélection des attributs qui a été basée sur la relation entre l'information

d'interaction et l'information conditionnelle. CMIFS se concentre à la fois sur l'information de traction et la redondance. La fonction a été utilisée par CMIFS est également basée sur la fonction de [119] pour démontrer leur fonction qui est :

$$IM(C; f_i\ /f_1) - IM(f_i; f_s\ /f_1) + IM(f_i; f_s\ /C)$$

- Kumar et al [128] proposent d'utiliser le IMC comme une fonction d'évaluation basée sur la fonction [120] telle que :

$$IM(C;\ fi) - \beta \sum_{f_s \in s} (IM(f_s;\ fi) + \gamma \sum_{f_s \in s} IM(f_i; f_s\ /C)$$

Kumar et al [128] n'ont pas intégrés la fonction dans aucun algorithme spécifié pour la sélection d'attributs (Filtre, Wrapper et Hybride). Dans le cas contraire, ils n'ont pas proposés un processus qui inclut cette fonction pour assurer sa fonctionnalité (comment il fonctionne pour extraire ses performances). Le problème dans [128] est qu'ils ne sont pas fait des tests et d'implémentation avec n'importe quel benchmark (ensemble de données) pour améliorer son efficacité.

Les inconvénients dans MIFS, MIFS-U, et [128] qui sont listés dans [126] et [129] sont :

- Favoriser la fonctionnalité non pertinente que le pertinent.
- Utiliser le paramètre $\beta\ et\ \gamma$ qui pose le problème de la valeur optimale. En outre, lorsque $\beta\ =\ 1$, le MIFS fonctionne mieux.
- Baser sur l'hypothèse que le conditionnement par la classe C ne modifie pas le rapport de l'entropie de f_s et du IM entre f_s et f_i, ce qui n'est valable que pour les distributions de probabilités uniformes.

Selon tous les travaux de MIFS, MIFS-U, MMFIS, CMIFS, mRMR, NMIFS, et [119] on présente quelques recommandations qui doivent être intégrées dans n'importe quel algorithme de sélection d'attributs utilise le MI :

- Préciser la dépendance et la redondance entre les attributs avec la variable de décision pour éviter le risque de sélectionner les attributs non pertinents que pertinent.
- Intégrer les informations d'interaction II et l'IMC entre les attributs pour donner la bonne signification entre le groupe d'attributs et leur classe cible C en ce qui concerne la dépendance, la redondance et la pertinence.
- Inutile d'intégrer les paramètres comme β et γ dans la fonction d'évaluation avec IM. Les paramètres permettent à l'utilisateur d'influencer sur la fonction d'évaluation et le processus de recherche. Dans tous les cas selon ces paramètres, petit ou grand, donnent une autre signification aux fonctions d'évaluation. L'algorithme de sélection d'attributs

tend à se rapprocher d'un côté de la fonction plutôt que de l'autre. Ils favorisent un côté de la fonction qui favorise certains attributs plutôt que l'autre.

- Le plus grand inconvénient de ces travaux est de suivre un processus incrémental pour sélectionner un meilleur sous-ensemble d'attributs (solution unique). Ce processus de sélection impose un schéma de recherche pour arriver au meilleur sous-ensemble d'attributs qui est basé sur le premier attribut sélectionné et la fonction d'évaluation. Par conséquent, il n'est pas raisonnable de laisser le premier attribut sélectionné guide la recherche sans parcourir tout l'espace de recherche et se combiner entre les attributs pour extraire tous les meilleurs sous-ensembles d'attributs possibles.
- Sélectionner et évaluer les attributs par groupes. Nous choisissons un sous-ensemble d'attributs à la place d'un seul attribut pour l'évaluation, afin d'améliorer la performance des solutions et leur évolution. Par conséquent, nous trouvons le meilleur sous-ensemble d'attributs au lieu d'une solution. De cette manière, nous respectons toutes les relations entre attributs, parcourons tout l'espace de recherche, et sélectionnons les meilleures solutions (sous-ensembles) sans oublier les solutions extrêmes.

6. Conclusion

Dans ce chapitre, nous avons présenté le problème de l'optimisation multi-objective. Nous avons défini les différents concepts de dominance, Pareto Set, Pareto optimale solution, Pareto Front, les algorithmes élitistes et non élitistes. Ensuite, nous avons donné une vue globale sur l'algorithme d'estimation distribué, l'algorithme typique et les différents avantages par rapport aux autres algorithmes évolutionnaires. IM est parmi les sections de ce chapitre que nous avons présenté. Nous avons donné une vue globale sur les techniques utilisées au niveau de IM, IMC, II, et Co-information pour la sélection des attributs. Enfin, nous avons présenté les différents travaux basés sur l'approche Filtre à base de MI et leurs inconvénients avec des recommandations pour la proposition des nouveaux algorithmes.

Chapitre 05

Algorithme à estimation de distribution multi-objectif pour la Sélection d'attributs.

1. Introduction

Dans ce chapitre, nous présentons l'approche proposée MOEDAFS (MultiObjective Estimation of distribution algorithm for Feature Selection). Nous illustrons le processus global de MOEDAFS. Ensuite, nous donnerons les étapes de l'algorithme MOEDAFS. Apres, nous dériverons tous les détails de chaque étape et toutes les techniques utilisées. Nous présentons le support mathématique des quatre modèles probabilistes utilisés dans MOEDAFS qui correspondent aux quatre versions de MOEDAFS.

Afin de démontrer les performances de MOEDAFS, une étude comparative est conçue par comparaison interne et externe sur un ensemble de données NSL-KDD. Une comparaison interne est effectuée entre les quatre versions de MOEDAFS. Dans cette comparaison interne, nous utilisons 5 types des algorithmes de classification. La comparaison externe est organisée en fonction de certains algorithmes de sélection d'attributs déterministes, méta-heuristiques et multi-objectif bien connus, dotés d'une solution unique ou des solutions multiples. A la fin de ce chapitre une conclusion sur les différents aspects de MOEDAFS.

2. Processus MOEDAFS

MOEDAFS est un algorithme hybride de sélection d'attributs multi-objectifs. L'objectif de MOEDAFS est de trouver de meilleurs sous-ensembles d'attributs assurant le plus grand taux de précision de classification (AR) et un plus petit nombre d'attributs (NF). MOEDAFS est basé sur EDA et MI pour :

- Explorer l'espace de recherche.
- Evaluer chaque sous-ensemble d'attributs.
- Sélectionner les solutions non dominées.
- Générer la nouvelle population candidate.

Le MOEDAFS suit un processus intelligent pour sélectionner les meilleurs sous-ensembles d'attributs. Il sélectionne un groupe de sous-ensembles d'attributs comme une population initiale au lieu d'un processus incrémental. Cette population initiale est évaluée et évoluée jusqu'à arriver aux meilleurs sous-ensembles d'attributs. Dans chaque itération, MOEDAFS sélectionne les solutions non dominées qui représentent les meilleurs sous-ensembles d'attributs sélectionnés dans toutes les itérations précédentes.

Le MOEDAFS utilise deux fonctions objectives : *le minimum taux d'erreur de classification* (ER) et *le minimum nombre d'attributs* (NF) (taux de réduction). Les modèles probabilistes MI sont intégrés pour guider la recherche en estimant la probabilité de chaque attribut f_i qui est

utilisée pour générer la nouvelle population candidate. La probabilité de l'attribut f_i représente la modélisation des relations pertinentes entre l'attribut f_i et les autres attributs f_j avec la classe cible C. La figure 5.1 présente le processus global de MOEDAFS.

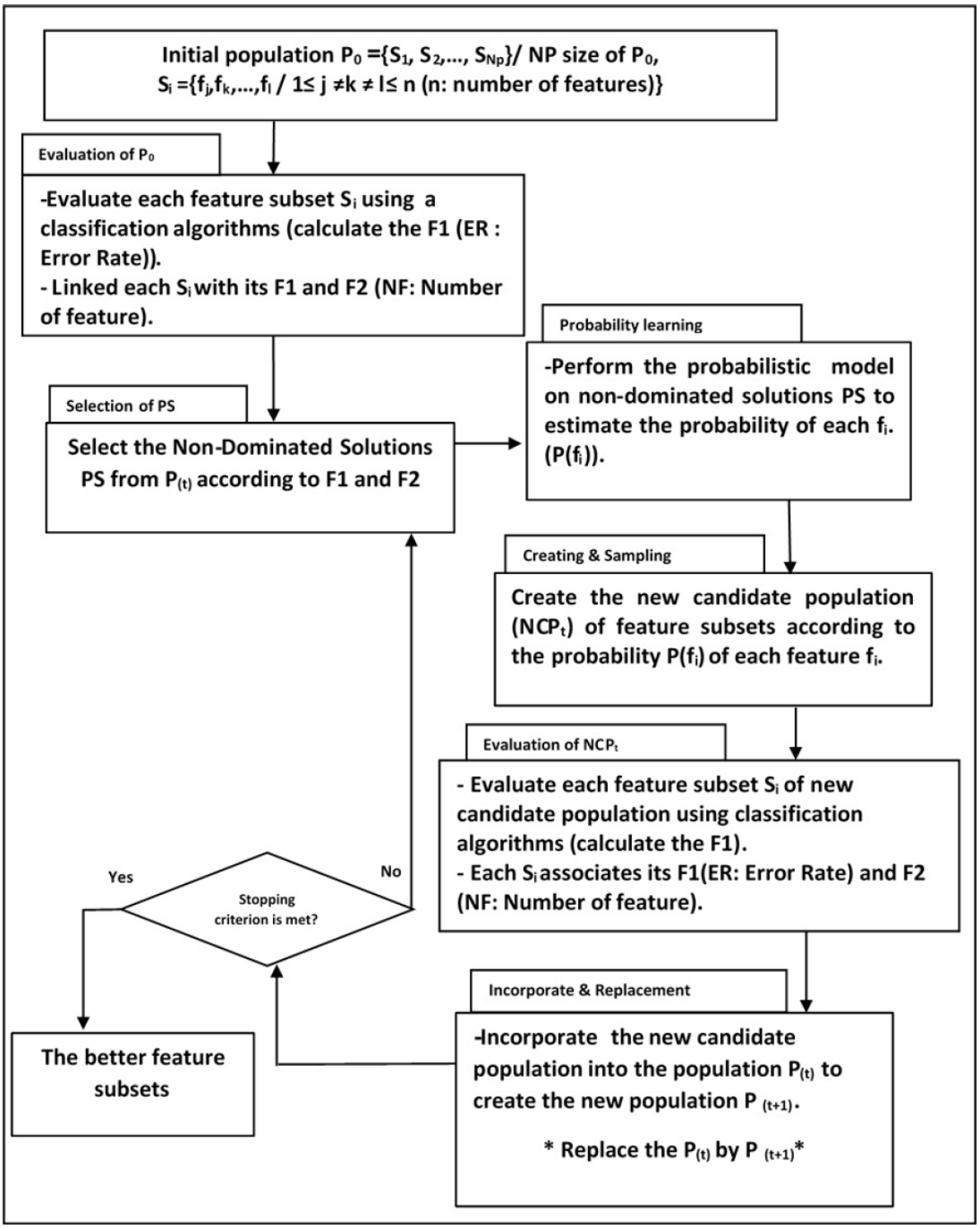

Figure 5.1 : Processus globale de MOEDAFS

MOEDAFS commance par l'étape de population initiale (Section 4) qui contient les sous-ensembles d'attributs initiales (la population initiale est créée par une fonction intelligente au lieu de la fonction aléatoire ce qui donne une faveur au niveau de l'état initial (Voir section 4)).

Ensuite, l'étape d'évaluation (Section 5) évalue chaque sous-ensemble S_i de la population P_t pour détecter le taux d'erreur (ER) et le nombre d'attributs (NF). On associe les ER et NF pour chaque sous-ensemble S_i comme deux fonctions de fitness (F1, F2).

Après cela, l'étape de sélection (Section 6) est effectuée selon (F1, F2) pour sélectionner les solutions non dominées PS de la population P_t.

L'étape d'apprentissage probabiliste (Section 7) utilise un modèle probabiliste sur les solutions non dominées PS pour estimer la probabilité de chaque attribut.

Selon la probabilité de chaque attribut f_i on crée la nouvelle population candidatée (Section 8). Cette nouvelle population candidate est évaluée et incorporée dans la population P_t pour créer la nouvelle population P_{t+1} qui représente les meilleurs sous-ensembles d'attributs entre la population P_t et la nouvelle population candidate.

La population P_t est remplacée par la population P_{t+1} (Section 9). Ensuite, on répète les étapes jusqu'à ce que les itérations maximales ou la condition d'arrêt soient satisfaites. Enfin, MOEDAFS a les meilleurs sous-ensembles d'attributs qui représentent les meilleurs sous-ensembles de toutes les générations de MOEDAFS.

Les étapes de MOEDAFS sont présentées avec leur section comme suit :

Etape 1 : Seeding (Section 4)

- Création de l'état initial.

Etape 2 : Evaluation de P_0 *(Section 5)*

- Évaluer chaque sous-ensemble S_i de P_0 par des algorithmes de classification.
- Calculer le nombre d'attribut NF pour chaque sous-ensemble S_i.
- Associer ER (F1) et NF (F2) pour chaque sous-ensemble.

Etape 3 : Sélection (Section 6)

- Sélectionner les solutions non dominées (PS) dans la population P_t selon F1, F2 en utilisant l'algorithme 2 (section 6).

Etape 4 : Apprentissage probabiliste (Section 7)

- Calculer la probabilité de chaque entité f_i en fonction des sous-ensembles de PS en utilisant l'un des quatre modèles probabilistes.

Etape 5 : Création et échantillonnage (Section 8)

- Créer la nouvelle population candidate en fonction de la probabilité de chaque f_i : P (f_i).

Etape 6 : Evaluation (Section 5)

- Évaluer chaque sous-ensemble S_i de la nouvelle population candidate par des algorithmes de classification.
- Calculer le nombre d'attributs NF pour chaque sous-ensemble.
- Associer ER (F1) et NF (F2) pour chaque sous-ensemble.
- Donner la valeur de F1, F2 pour chaque sous-ensemble.

Etape 7 : Incorporer et remplacement (Section 9)

- Incorporer la nouvelle population candidate dans P (t) pour générer la nouvelle population P_{t+1}.
- Remplacer P_t par P_{t+1}.

Si les itérations maximales ou la condition d'arrêt sont remplies, arrêtez l'algorithme avec les meilleurs sous-ensembles d'attributs (PS), sinon passez à l'étape 3.

3. Codage de population de solutions

Dans cette section, nous présentons la codification des solutions (population) pour MOEDAFS. La population de solutions P_t est construite par un ensemble de chromosomes, chaque chromosome C_i représentant le sous-ensemble d'attributs S_i. Le type de chromosome est binaire $\in [0,1]$ avec une taille de 41 qui présente les 41 attributs de NSL-KDD (base de connaissance de l'expérimention) (Section 3.1, chapitre 3). Chaque fonction f_j dans le chromosome C_i est modélisée par 1 ou 0 selon son apparence dans le sous-ensemble d'attributs S_i ou non. La table 5.1 illustre le modèle de population P_t.

Tableau 5.1 : Le modèle de population P_t.

	Attributs					
	f_1	f_2	…	f_j	…	f_{41}
sous-ensemble s_1 (chromosome c_1)	0	1	…	1		1
sous-ensemble s_2 (chromosome c_2)	1	0	….	0		0
……	…	…	…	…	…	…
sous-ensemble s_i (chromosome c_i)	1	1	…	0		1
……						
sous-ensemble s_{np} (chromosome c_{np})	1	0	…	1		0
	f_1 [01…1]	f_2 [10…0]	…	f_j [10…1]	…	f_{np} [10…0]

$P_t = \{c_1, c_2, ..., c_i, ..., c_{NP}\}$ où c_i est le chromosome correspondant au sous-ensemble d'attributs S_i, NP taille de la population. Sous-ensemble $S_i = \{f_j, ..., f_k, ..., f_l\}$ $(1 \leq j \neq k \neq l \leq n)$ où chaque sous-ensemble S_i contient sa taille (nombre d'attributs).

Le vecteur f_j [1..np] de l'attribut f_j présente l'apparence de f_j dans chaque sous-ensemble d'attribut (chromosome) de la population P_t tel que:

$$\begin{cases} f_j\ [\mathrm{i}] = 1\ ;\ \text{apparaît dans } C_i \\ f_j\ [\mathrm{i}] = 0\ ;\ \text{n'apparaît pas dans } C_i \end{cases}$$

Le numéro d'apparence de 1 dans le vecteur f_j [1..np] décrit le numéro d'apparence (AN) de l'attribut f_j dans tous les sous-ensembles de la population. La valeur d'AN de l'attribut f_j est calculée à partir de la probabilité de f_j (Section 7, chapitre 5).

Pour générer une population de MOEDAFS, on crée le vecteur de chaque attribut f_j [1..np]. Pour cela, nous effectuons les étapes suivantes pour chaque attribut f_j dans F :

- Calculer la probabilité de l'attribut f_j (P (f_j)).
- Déterminer la valeur du numéro d'apparence (AN) de l'attribut f_j dans tous les sous-ensembles d'attribut de la nouvelle population, tels que :

$$AN_{\mathrm{fj}} = P(f_j) * np\ ;\ np\ la\ taill\ de\ population.$$

Nous supposons que la probabilité de f_j est P (f_j) = 0.5 ; La taille de la population est np = 20 sous-ensembles (chromosomes).

L'AN de l'attribut f_j est 10. Donc, 10 sous-ensembles au hasard de la nouvelle population contiennent la fonction f_j.

- Créez le vecteur f_j [1..np] en fonction de AN_{fj}. La valeur de AN_{fj} est égale à l'AN de 1 dans le vecteur f_j [1..np] . La distribution de 1 dans le vecteur f_j [1..np] est aléatoire.

4. Population initiale

Dans l'étape initiale de MOEDAFS, on crée la population initiale P_0 qui représente les sous-ensembles d'attributs initiaux de MOEDAFS.

Pour créer la population initiale, on utilise la probabilité initiale de chaque attribut f_j. La probabilité initiale de chaque attribut f_j dans la population initiale est basée sur la relation de pertinence entre attribut f_j et la classe cible C en utilisant l'information mutuelle (IM (f_j, C)). La probabilité d'attribut f_j est calculée comme suit :

$$P\left(f_j\right) = 1/1 + e^{-IM(f_j,C)}.$$

La figure 5.2 montre les attributs de NSL-KDD entre 3 à 6, 12, 23 à 26 et 29 à 39 qui ont plus de 50 % de probabilité, ce qui signifie que chaque vecteur f_j [1..np] de ces attributs a aléatoirement plus de 50 % du nombre 1 dans leurs vecteurs. Sinon, ces fonctionnalités apparaissent de manière aléatoire dans les sous-ensembles d'attributs initiaux plus de 50 %. Pour le reste

d'attributs, ils ont seulement 50 % de probabilité dans l'état initial. On calcule le vecteur f_j [1..np] de chaque attribut f_j en fonction de AN_{fj} en utilisant la probabilité initiale P (f_j). Cette distribution représente la population initiale de toutes les versions de MOEDAFS.

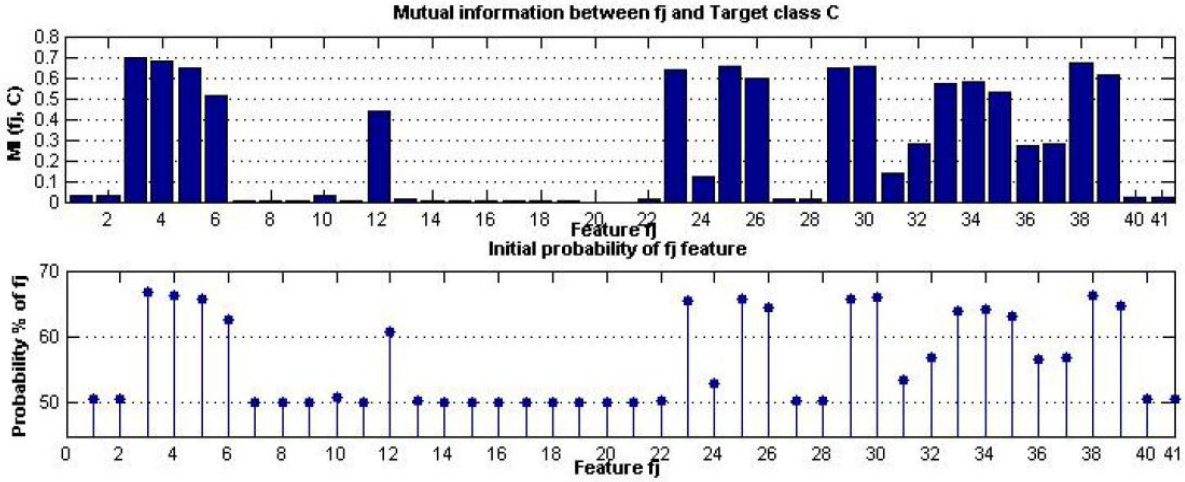

Figure 5.2 : Etat initiale.

5. Évaluation de wrapper de sous-ensembles d'attributs sélectionnés

Les sous-ensembles d'attributs de la population P_t et la nouvelle population candidate dans MOEDAFS sont évalués à l'aide de deux fonctions fitness :

Minimisation du taux d'erreur (ER) et Minimisation du nombre d'attributs (NF) (taux de réduction).

$$\begin{cases} F1\ (Si) = Error\ Rate\ (ER)(S_i) = \dfrac{(FN + FP)}{TP + TN + FN + FP} \\ F2\ (S_i) = Reduction\ Rate\ (NF{:}\ Number\ of\ features)(S_i) = \dfrac{|S_i|}{|F|} \end{cases}$$

Où FP, FN, TP et TN se trouvent sur la table de matrice de confusion tableau 3.3 (Chapitre 3, Section 3.2). L'arbre de décision (DT) est utilisé comme un algorithme de classification par l'étape d'évaluation pour estimer le taux d'erreur pour chaque sous-ensemble d'attributs.

6. Sélection de solutions non dominées

Dans cette étape, on sélectionne les sous-ensembles d'attributs prometteurs qui représentent les solutions non dominées PS de la population P_t qui est basée sur l'équation (2.1 définition, chapitre 4 (relation de dominance), on définit l'algorithme 2 pour sélectionner les solutions non dominées (meilleurs sous-ensembles d'attributs).

Algorithm 2 Non-dominated solutions selection algorithm

Objective: Find PS (better solutions s_i)

Input: $P_t = \{s_1, s_2, ..., s_i, ..., s_{np}\}$: of Solutions, where each solution s_i contains its $\{F1, F2\}$

Output: PS

```
for each solution s_i of P_t do
    State = 0;
    for each solution s_j of P_t do                          ▷ i ≠ j
        if ∀k ∈ [1..2] : F_k(s_j) ≤ F_k(s_i) ∧ ∃l ∈ [1..2] : F_l(s_j) < F_l(s_i) then
            State = 1;
        end if
    end for
    if State = 0 then
        ADD (s_i) in PS;
    end if
end for
```

Les solutions non dominées PS sont utilisées comme un noyau pour générer la nouvelle population candidate. Dans chaque itération, on génère la nouvelle population candidatée à partir des meilleurs sous-ensembles d'attributs sélectionnés dans toutes les générations précédentes. Ce processus de génération assure l'échantillonnage de nouveaux sous-ensembles de meilleurs attributs.

7. MI-Modèles probabilistes

Dans cette section, on présente les modèles probabilistes qui sont utilisés pour estimer la probabilité de chaque attribut $f_j \in F$ dans chaque itération de MOEDAFS. Chaque probabilité de f_j correspond à la modélisation des relations de pertinence et de redondance avec les autres fonctions f_k et la classe cible C.

Ces relations sont conçues à partir des relations internes et externes. Les relations internes de f_j modélisent la relation entre f_j et les autres attributs f_k avec la classe cible C du même sous-ensemble S_i. Les relations externes présentent la relation de f_j avec d'autres sous-ensembles de population $P_{\_\{Si\}}$ sur lesquels f_j apparaît. Pour cela, on propose quatre nouveaux modèles probabilistes basés sur le II, le Co-Inf et l'IMC pour représenter l'aspect formel des relations internes et externes de chaque f_j. Ces modèles sont utilisés dans MOEDAFS, chaque modèle est intégré dans MOEDAFS représente une version de MOEDAFS (MOEDAFS-One, MOEDAFS-Two, MOEDAFS-Three et MOEDAFS-Four).

Avant de présenter les quatre modèles probabilistes, on donne les termes significatifs utilisés : $f_{j \neq k}$ présente l'attribut dont on veut calculer sa probabilité, f_k attributs partagent le même

sous-ensemble avec f_j. C : classes cibles ; Q est le nombre d'attributs qui partagent le même sous-ensemble avec f_j; M: nombre de sous-ensembles d'attribut S_i dans lesquels f_j apparaît.

7.1 Modèle One (MOEDAFS-One)

Dans ce premier modèle, nous nous sommes basés sur le II et le Co-Inf pour approximer les relations entre les attributs. MOEDAFS-One repose sur le modèle probabiliste One. La présentation de probabilité mathématique de chaque f_j du modèle un est illustrée comme suit :

$$Mean = (\frac{1}{M}) \sum_{l=1}^{M} \left[IM(f_j; C) - \sum_{k=1}^{Q} II(C; f_j ; f_k) \right]$$

$$= (\frac{1}{M}) \sum_{l=1}^{M} \left[IM(f_j; C) - \sum_{k=1}^{Q} Co - Inf(C; f_j; f_k) \right]$$

$$= \left(\frac{1}{M}\right) \sum_{l=1}^{M} \left[IM(f_j; C) - \sum_{k=1}^{Q} IM(f_j; f_k) + \sum_{k=1}^{Q} IM\left(f_j; \frac{f_k}{C}\right) \right] \dots (1)$$

$$P\,(f\mathrm{j}) = 1/1 + e^{-Mean}$$

7.2 Modèle Two (MOEDAFS-Two)

Le modèle deux est basé sur l'information d'interaction II et le négatif de Co-Inf entre C, f_j et f_k. Le modèle deux est utilisé par MOEDAFS-Two comme un modèle probabiliste. La probabilité de chaque attribut f_j est présentée mathématiquement comme suit :

$$Mean = (\frac{1}{M}) \sum_{l=1}^{M} \left[\sum_{k=1}^{Q} II(C; f_j ; f_k) \right]$$

$$= (\frac{1}{M}) \sum_{l=1}^{M} \left[- \sum_{k=1}^{Q} Co - Inf(C; f_j; f_k) \right]$$

$$= (\frac{1}{M}) \sum_{l=1}^{M} \left[\sum_{k=1}^{Q} IM(f_j; f_k/C) - \sum_{k=1}^{Q} IM(f_j; f_k) \right] \dots (2)$$

$$P\,(f\mathrm{j}) = 1/1 + e^{-Mean}$$

7.3 Model Three (MOEDAFS-Three)

MOEDAFS-Three est basé sur le modèle trois comme un modèle probabiliste. Le troisième modèle représente une autre vision du calcul de la probabilité de f_j. Dans ce modèle, on intéresse aux relations d'intersection entre CMI de f_j, f_k étant donné C et f_j, C étant donné f_k pour extraire la valeur réelle de f_j. Par conséquent, on sélectionne la valeur minimale entre différentes relations d'intersection de chaque sous-ensemble S_i où f_j apparaît. La présentation mathématique pour chaque probabilité de f_j est illustrée comme suit :

$$MIN = Min\ (S_1, S_2, \dots, S_i, \dots S_M) \text{ Tel que}$$

$$S_i = \left[\sum_{k=1}^{Q} IM(f_j; f_k\ /C\ ) - \sum_{k=1}^{Q} IM(f_j; C\ /f_k\ ) \right] \dots (3)$$

$$P\ (f\text{j}) = 1/1 + e^{-MIN}$$

7.4 Modèle Four (MOEDAFS-Four)

Le dernier modèle de probabilité représente le modèle probabiliste de MOEDAFS-Four. Pour calculer la valeur de probabilité de chaque f_j, on fait l'intersection entre IMC de (f_j, f_k étant donné C), (f_j, C étant donné f_k) et (f_k, C étant donné f_j) pour extraire le degré réel de la relation. Le quatrième modèle est présenté comme suit :

$$Mean = (\frac{1}{M}) \sum_{l=1}^{M} \left[IM(f_j; f_k\ /C) - \sum_{k=1}^{Q} IM(f_j; C\ /f_k) - \sum_{k=1}^{Q} IM(C; f_k\ /f_j) \right] \dots (4)$$

$$P\ (f\text{j}) = 1/1 + e^{-Mean}$$

8. Génération de la nouvelle population candidate

La génération de la nouvelle population candidate est basée sur les solutions non dominées PS de la population P_t et l'un des quatre modèles probabilistes. P_t représente les meilleurs sous-ensembles d'attributs sélectionnés dans toutes les générations précédentes.

Pour générer la nouvelle population candidate, on exécute l'un des quatre modèles probabilistes sur le PS afin de calculer la valeur de probabilité de chaque attribut f_j. Ensuite, on détermine la valeur du numéro d'apparence (AN) de l'attribut f_j dans tous les sous-ensembles d'attribut en fonction de cette probabilité. Après cela, on crée le vecteur f_j [1..np] de l'attribut f_j qui représente l'apparence de f_j dans chaque sous-ensemble d'attributs de la nouvelle population. Pour

calculer AN de f_j et le vecteur f_j [1..np], on suive les mêmes étapes pour générer une nouvelle population dans la section 5. On utilise la valeur de probabilité de chaque attribut f_j qui est calculée par des modèles probabilistes. La figure 3.5 montre le processus de génération de la nouvelle population candidate.

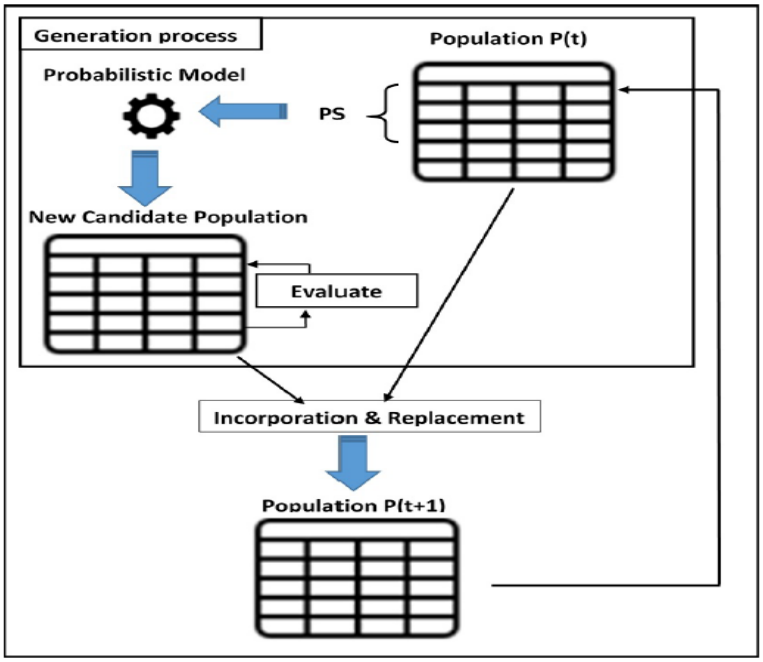

Figure 5.3 : Génération de population**.**

9. Population de remplacement

Dans cette étape, on remplace l'ancien P_t par le nouveau P_{t+1} pour la prochaine itération. Ce nouveau est un enrichissement de l'ancien P_t par des nouvelles solutions. Dans la figure 5.3, on expose le processus de mise à jour de P_t par P_{t+1} que la population P_{t+1} est le résultat de l'incorporation entre la population P_t et la nouvelle population candidate. Le P_{t+1} contient les meilleurs sous-ensembles d'attributs de P_t et de la nouvelle population candidate. On élimine les solutions de foule (Crowd Solutions) dans P_t et ajoutons de nouvelles solutions meilleures depuis la nouvelle population candidate.

De plus, on classe les solutions de P_t et de nouvelle population candidatée en fonction de F1 et F2. Ensuite, on incorpore les meilleures solutions dans la population P_{t+1}. On intégre l'algorithme FNS de NSGAII et Crowding Distance pour créer la population P_{t+1}. Ainsi, P_{t+1} ne contient que les meilleures solutions de toutes les générations. P_{t+1} est considéré comme un nouveau P_t pour une nouvelle itération contenant le nouveau PS pour la prochaine génération.

10. Résultats expérimentaux

10.1 Étude de comparaison

Dans cette section, on présente le résultat de MOEDAFS qui est effectué sur le NSL-KDD pour sélectionner les meilleurs sous-ensembles d'attributs. Cette comparaison est divisée en une comparaison interne et externe. La comparaison interne est exécutée entre les quatre versions de MOEDAFS qui utilisent respectivement les quatre modèles probabilistes. Une comparaison externe est effectuée entre MOEDAFS et des travaux récents bien connus (algorithmes de sélection d'attributs déterministes, méta-heuristiques et multi-objectifs qui ont un seul sous-ensemble, et multiples sous-ensembles solutions).

Les expériences ont été réalisées en utilisant un PC avec Intel Core i5 2,67 GHZ et 4 Go de RAM fonctionnant sous Windows 7. Les codes expérimentaux sont écrits sur Matlab R2010 et R Studio 0,99.491.

10.2 Pré-traitement

Avant de lancer l'implémentation, il est nécessaire de réaliser l'étape de pré-traitement sur l'ensemble de données expérimentales NSL-KDD. On effectue le processus suivant :

1. Grouper les attaques dans les cinq classes DOS, Probe, U2R, R2L et la connexion normale.
2. Supprimer les instances en double.
3. Transformer les données (Convertir l'entité nominale en une valeur numérique de la fonction de mappage simple [0, catégories-1]).
4. Discrétisation : discrétiser les fonctionnalités continues qui ont une grande valeur.

Selon les étapes de prétraitement, on diminue le nombre d'instance de NSL-KDD. La table 5.2 présente la nouvelle taille de l'ensemble de données après l'étape de prétraitement. On prend 80 % de NSL-KDD comme une base de traitement (Training set) et 20 % comme une base de test (test set). La taille de la population P_t dans chaque version de MOEDAFS est de 20 chromosomes. Pour le nombre d'itérations de MOEDAFS, nous prenons T = 100.

Tableau 5.2 : Prétraitement**.**

	Nombre total d'instances
NSL-KDD entier	125965
NSL-KDD Après le prétraitement	**103395**
80 % de NSL-KDD comme une base de traitement	**82716**
20 % comme une base de test	**20679**

10.2 Comparaison interne

Dans cette section, on présente les résultats expérimentaux d'une comparaison interne de MOEDAFS. Cette comparaison est effectuée entre les quatre versions de MOEDAFS respectivement (MOEDAFS-One, MOEDAFS-Two, MOEDAFS-Three et MOEDAFS-Four). Après leur implémentation sur NSL-KDD pour sélectionner les meilleurs sous-ensembles d'attributs (solutions non dominées PS), la figure 5.4 montre les résultats des quatre versions de MOEDAFS par le Front Pareto, qui se situe sur l'ER (axe horizontal) et NF (axe vertical). Front Pareto présente l'image de PS de chaque version de MOEDAFS.

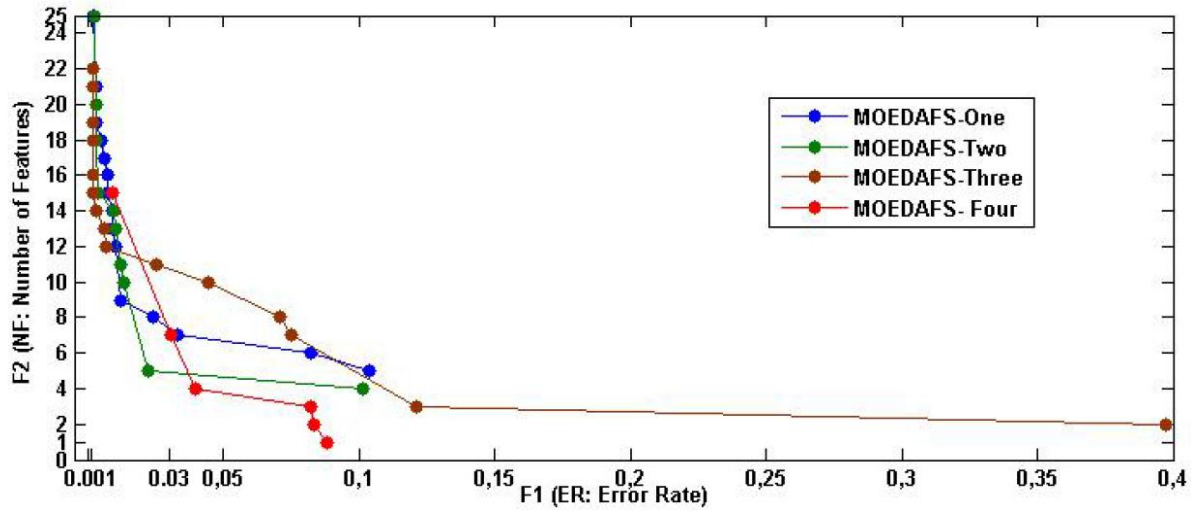

Figure 5.4. Pareto Front de versions MOEDAFS

Selon la figure 5.4, les quatre versions de MOEDAFS montrent à première vue une variété de combinaisons de sous-ensembles d'attributs. Il existe une compétition entre les uns et les autres en ce qui concerne les sous-ensembles d'attributs sélectionnés. En examinant les fronts de Pareto des versions de MOEDAFS, on peut remarquer une similitude de solutions, différents sous-ensembles d'attributs obtenus peuvent avoir le même nombre d'attributs mais, en fait, ils sont complètement différents entre eux dans le type d'attributs sélectionné.

Après le processus de traitement, les sous-ensembles d'attributs sélectionnés dans les quatre versions de MOEDAFS sont évalués sur une base du test (Test Set) pour obtenir le taux

de précision de la classification dans la base de test. Les tables 5.3, 5.4, 5.5, et 5.6 affichent les résultats qu'on associe pour chaque sous-ensemble : leurs *attributs sélectionner*, le *nombre d'attributs sélectionner*, la *précision de la classification* (Accuracy Rate : AR) dans la base de *tests* et la base de *traitement* avec les trois autres mesures d'évaluation.

Selon la figure 5.4, tableaux 5.3, 5.4, 5.5, et 5.6, on confirme la diversité des solutions sélectionnées par les versions MOEDAFS. Dans toutes les versions de MOEDAFS, le nombre d'attributs utilisés dans le processus de détection a été réduit. MOEDAFS-One a sélectionné environ 60 % (25 attributs) à 12 % (5 attributs) des attributs disponibles, et MOEDAFS-Two sélectionné seulement environ 60 % (25 attributs) à 9 % (4 attributs) des attributs disponibles.

D'autre part, MOEDAFS-Three a réduit le nombre d'attributs autour de 53 % (22 attributs) à 4,8 % (2 attributs) des attributs disponibles, tandis que MOEDAFS-Four a diminué le NF entre 36,5 % (15 attributs) à 2,4 % (1 attribut) des attributs disponibles. Selon la figure 5.4 et les tables 5.3, 5.4, 5.5, et 5.6 nous confirmons la diversité et l'efficacité des solutions sélectionnées par les versions de MOEDAFS.

Table 5.3 : PS de MOEDAFS-One.

Feature Subset	NF	Testset Accuracy (%)	Trainingset Accuracy(%)	Precision	Recall	F-measure	MCC
9 20 21 24 34	5	89,58	89,01	1	0,7676	0,8685	0,7970
2 8 10 16 22 23	6	91,85	91,68	1	0,8201	0,9011	0,8427
2 14 24 27 28 29 34	7	96,69	96,93	1	0,9321	0,9649	0,9395
2 8 9 13 18 35 39 40	8	97,61	97,67	1	0,9486	0,9736	0,9539
2 6 13 17 27 29 35 36 37	9	98,80	99,09	1	0,9798	0,9898	0,9817
2 6 9 10 21 23 24 27 31 34 36 41	12	99,00	99,30	1	0,9845	0,9922	0,9859
5 8 9 11 17 19 22 23 24 33 36 39 41	13	99,10	99,38	1	0,9861	0,9930	0,9874
1 2 3 7 13 15 17 19 24 28 36 39 40 41	14	99,14	99,38	1	0,9861	0,9930	0,9875
1 2 3 8 11 13 17 22 23 27 28 29 35 39 40	15	99,26	99,41	1	0,9869	0,9934	0,9881
1 2 3 8 10 16 18 19 22 28 35 36 38 39 40 41	16	99,34	99,46	1	0,9880	0,9940	0,9891
1 2 8 10 13 14 15 21 23 24 27 34 36 37 39 40 41	17	99,39	99,59	1	0,9909	0,9954	0,9918
2 6 8 11 13 15 17 21 22 23 24 28 33 34 36 39 40 41	18	99,46	99,63	1	0,9918	0,9959	0,9926
2 3 4 5 7 9 13 16 22 23 25 27 28 29 30 33 35 37 39	19	99,66	99,81	1	0,9958	0,9979	0,9963
1 5 6 7 8 12 14 15 16 18 19 24 25 26 30 31 32 33 36 40	20	99,72	99,84	1	0,9963	0,9982	0,9967
2 3 4 5 6 7 8 10 15 17 18 19 22 23 24 25 26 28 30 36 39	21	99,75	99,84	1	0,9964	0,9982	0,9968
1 2 4 5 6 7 8 9 12 16 17 18 21 23 24 25 27 28 30 31 34 36 37 38 40	25	99,79	99,89	1	0,9975	0,9988	0,9977
Moyenne	**14**	**98,01**	**98,12**				

Table 5.4 : PS de MOEDAFS-Two.

Feature Subset	NF	Testset Accuracy (%)	Trainingset Accuracy(%)	Precision	Recall	F-measure	MCC
7 14 29 31	4	89,87	88,48	1	0,7473	0,8554	0,7853
5 10 17 35 37	5	97,79	97,95	1	0,9549	0,9769	0,9593
2 7 10 20 27 32 33 36 39 40	10	98,66	98,99	1	0,9776	0,9887	0,9798
2 6 8 16 19 20 22 33 36 39 40	11	98,79	98,95	1	0,9767	0,9882	0,9790
1 2 13 20 24 28 31 32 33 36 39 40 41	13	98,98	99,44	1	0,9876	0,9937	0,9888
3 8 11 13 17 19 20 21 23 24 34 39 40 41	14	99,06	99,24	1	0,9830	0,9914	0,9846
2 4 5 6 7 8 14 19 21 27 34 37 38 40 41	15	99,66	99,77	1	0,9949	0,9974	0,9954
2 3 5 7 8 10 17 18 19 20 23 24 25 26 27 34 35 37	18	99,70	99,83	1	0,9962	0,9981	0,9966
1 5 6 7 8 12 14 15 16 18 19 24 25 26 30 31 32 33 36 40	20	99,72	99,84	1	0,9963	0,9982	0,9967
1 3 4 5 8 9 12 13 15 16 18 20 22 25 26 27 28 31 32 35 36 37 38 40 41	25	99,76	99,89	1	0,9976	0,9988	0,9978
Moyenne	**13,5**	**98,20**	**98,24**				

Table 5.5 : PS de MOEDAFS-Three.

Feature Subset	NF	Testset Accuracy (%)	Trainingset Accuracy(%)	Precision	Recall	F-measure	MCC
11 28	2	60,26	58,87	1	0,1283	0,2275	0,2686
24 31 38	3	87,88	87,51	1	0,7302	0,8440	0,7696
4 7 16 18 20 32 41	7	92,51	91,48	1	0,8163	0,8989	0,8393
2 8 9 12 13 16 31 40	8	92,90	92,36	1	0,8433	0,9150	0,8566
2 9 11 13 16 22 24 27 34 40	10	95,58	95,92	1	0,9111	0,9535	0,9204
8 13 14 19 26 27 31 33 37 40 41	11	97,48	97,13	1	0,9361	0,9670	0,9432
2 3 4 5 14 15 16 22 23 28 34 37	12	99,35	99,58	1	0,9907	0,9953	0,9916
2 5 6 9 14 15 18 19 24 34 37 38 40	13	99,41	99,58	1	0,9905	0,9952	0,9914
2 3 5 11 12 14 21 23 29 31 32 35 38 41	14	99,70	99,78	1	0,9950	0,9975	0,9955
1 3 4 5 7 10 12 15 17 18 25 30 34 36 38	15	99,79	99,87	1	0,9972	0,9986	0,9974
1 3 5 6 11 12 17 20 21 22 23 30 34 36 39 40	16	99,80	99,89	1	0,9976	0,9988	0,9978
1 2 3 4 5 6 8 12 23 24 27 31 32 33 36 37 38 39	18	99,81	99,89	1	0,9975	0,9987	0,9977
1 2 3 4 5 6 7 8 15 16 17 18 21 23 24 26 32 35 40	19	99,82	99,89	1	0,9976	0,9988	0,9979
1 3 4 5 6 9 11 12 15 17 18 20 22 26 28 30 32 33 36 37 38	21	99,82	99,91	1	0,9979	0,9990	0,9981
1 2 3 5 7 8 9 11 18 19 22 23 24 25 26 27 28 30 31 32 36 40	22	99,83	99,88	1	0,9972	0,9986	0,9975
Moyenne	**12,73**	**94,93**	**94,77**				

Table 5.6 : PS de MOEDAFS-Four.

Feature Subset	NF	Testset Accuracy (%)	Trainingset Accuracy(%)	Precision	Recall	F-measure	MCC
30	1	91,21	89,92	1	0,7815	0,8773	0,8114
6 36	2	91,64	90,59	1	0,7999	0,8888	0,8242
8 17 30	3	91,78	90,60	1	0,7962	0,8865	0,8234
5 21 32 36	4	96,04	96,08	1	0,9148	0,9555	0,9235
5 7 11 15 29 32 36	7	96,96	97,04	1	0,9347	0,9663	0,9416
1 3 10 11 13 14 17 18 26 27 28 30 36 37 40	15	99,11	99,34	1	0,9854	0,9927	0,9868
Moyenne	**5,33**	**94,46**	**93,93**				

En comparant les quatre versions de MOEDAFS, on peut voir que MOEDAFS-One et MOEDAFS-Three contient respectivement 16 et 15 sous-ensembles d'attributs (solutions non dominées), mais MOEDAFS-Two et MOEDAFS-Four respectivement contiennent 10 et 6 sous-ensembles. Cependant, MOEDAFS-Two et MOEDAFS-Four contenaient les sous-ensembles d'attributs les plus petits avec 4 attributs et jusqu'à un seul attribut, respectivement, que MOEDAFS-One et MOEDAFS-Three qui sélectionnaient respectivement des sous-ensembles avec 5 et 2 attributs.

Selon la figure 5.4 et les tables 5.3 et 5.4 en comparant MOEDAFS-One et MOEDAFS-Two, on peut voir que les fronts de Pareto sont similaires, mais le nombre d'attributs dans les sous-ensembles sélectionnés dans MOEDAFS-Two est presque plus petit avec un taux de précision (AR) plus élevé que MOEDAFS-One.

Par exemple dans MOEDAFS-One, le plus petit sous-ensemble sélectionné a 5 attributs avec 89.58 % taux de précision, contrairement à MOEDAFS-Two qui a sélectionné 2 sous-ensembles avec 4 et 5 attributs et (89.87 % et 97.79 % respectivement) de taux de précision (AR). La moyenne de taux de la précision (AR) dans la base teste et de traitement MOEDAFS-Two est meilleur que celle de MOEDAFS-One.

Le nombre moyen d'attributs sélectionnés dans des sous-ensembles de MOEDAFS-Two (13 attributs) est meilleur que MOEDAFS-One (14 attributs). On dit que les solutions non dominées de MOEDAFS-One sont dominées par les solutions de MOEDAFS-Two.

L'explication de cette domination est concentrée dans le modèle probabiliste utilisé par les deux versions pour estimer la probabilité de chaque attribut. Le modèle probabiliste de MOEDAFS-Two est l'amélioration du modèle probabiliste de MOEDAFS-One ce qui rend la performance d'un MOEDAFS est plus élevé. Selon les équations (1) et (2) de MOEDAFS-One et MOEDAFS-two respectivement, on peut voir l'absence du côté IM (C; f_j) dans (2) contrairement à (1) .

Cette élimination permet à MOEDAFS-Two de converger vers les solutions ayant le plus petit nombre d'attributs et un taux de précision (AR) plus élevé que MOEDAFS-One. Donc, quand on garde juste les deux parties de l'équation (2) qui est l'intersection entre CMI de f_j, f_k étant donné C (IM (f_j; f_k / C)) et IM (f_j; f_k), MOEDAFS-Two tente d'explorer plus efficacement l'espace de recherche et de trouver de meilleurs résultats que MOEDAFS-One.

Comparant MOEDAFS-Three et MOEDAFS-Four, d'après la figure 5.4 et les tables 5.5 et 5.6, on peut voir que le front de Pareto MOEDAFS-Three est dominé par le front de Pareto de MOEDAFS-Four. MOEDAFS-Four a sélectionné une meilleure solution de sous-ensembles que MOEDAFS-Three, soit dans le nombre d'attributs dans les sous-ensembles, soit dans le taux de précision (AR).

Par exemple, MOEDAFS-Three a sélectionné 3 sous-ensembles plus petits avec 2, 3 et 7 attributs avec seulement 60.26 % , 87.88 % et 92.51 % de taux de précision respectivement, tandis que MOEDAFS-Four a sélectionné quatre sous-ensembles avec seulement 1 , 2, 3 et 4 attributs avec respectivement 91,21 % , 91,64 % , 91,78 % et 96,04 % de taux de précision. En outre, on peut remarquer que le troisième sous-ensemble dans MOEDAFS-Trois contient 7 attributs avec 92,51 % de taux de précision, tandis que le cinquième sous-ensemble de MOEDAFS-Four avec le même nombre d'attributs (7) atteint 98,96 % du taux de précision.
La moyenne de taux de précision de la base de test de MOEDAFS-Three est égale à celle de MOEDAFS-Four, mais lorsque on concentre sur la performance du sous-ensemble d'attributs, on trouve la moyenne du nombre de sous-ensembles sélectionnés dans MOEDAFS-Four est 5 au lieu de 12 dans MOEDAFS-Trois.

De plus, la qualité des sous-ensembles d'attributs sélectionnés dans MOEDAFS-Four est meilleure que celle de MOEDAFS-Three, que ce soit dans le nombre d'attributs dans les sous-ensembles ou dans le taux de précision (AR). Ainsi, on peut dire que MOEDAFS-Four domine les solutions de MOEDAFS-Three.

Le modèle probabiliste utilisé explique la domination de MOEDAFS-Four. Dans MOEDAFS-Three, on a utilisé l'équation (3) basée sur la valeur minimale entre différentes relations d'intersection de chaque sous-ensemble S_i où f_j apparaît. Alors que MOEDAFS-Four utilise l'équation (4) qui est une intersection entre IM (f_j; f_k / C) , IM (f_j; C /f_k) et IM (C; f_k / f_j). L'équation (4) permet au MOEDAFS-Four d'équilibrer ses capacités globales, locales et de converger vers les meilleures solutions que MOEDAFS-Three.

Afin de confirmer les performances de classification des sous-ensembles sélectionnés par les quatre versions de MOEDAFS, ils sont évalués par quatre autres algorithmes de classification qui sont : SVM (Support Vector Machine), KNN (K = 5, pour les 5 classes cibles de NSL-

KDD) (K-Plus proches voisins), BN (Naive Bayes), MLP (Réseaux Neuronaux Perceptron Multilayer), et DT (Arbre de Décision). La classification est faite sur l'ensemble NSL-KDD. Les tableaux (5.7, 5.8, 5.9, et 5.10) affichent les résultats de la classification de tous les sous-ensembles d'attributs qui sont focalisés sur le NF, le taux de précision (AR), et la précision moyenne de chaque sous-ensemble sur ces classificateurs.

Tableau 5.7 : MOEDAFS-One avec cinq algorithmes de classification.

Feature subset	NF	NB	MLP	SVM	KNN	DT	Moyenne
		Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)
9 20 21 24 34	5	87,69	89,26	87,24	90,47	89,84	88,9
2 8 10 16 22 23	6	88,59	91,27	88,6	91,96	91,9	90,464
2 14 24 27 28 29 34	7	89,38	95,79	91,86	97,67	97,05	94,35
2 8 9 13 18 35 39 40	8	91,84	96,15	91,43	98,26	97,65	95,066
2 6 13 17 27 29 35 36 37	9	94,11	97,09	94,05	99,54	98,95	96,748
2 6 9 10 21 23 24 27 31 34 36 41	12	93,58	97,76	94,52	99,62	99,14	96,924
5 8 9 11 17 19 22 23 24 33 36 39 41	13	95,38	98,11	95,68	99,73	99,35	97,65
1 2 3 7 13 15 17 19 24 28 36 39 40 41	14	95,63	98,3	94,71	99,67	99,31	97,524
1 2 3 8 11 13 17 22 23 27 28 29 35 39 40	15	95,78	97,38	95,5	99,74	99,3	97,54
1 2 3 8 10 16 18 19 22 28 35 36 38 39 40 41	16	95,43	98,05	92,83	99,68	99,39	97,076
1 2 8 10 13 14 15 21 23 24 27 34 36 37 39 40 41	17	94,8	98,95	96,37	99,88	99,43	97,886
2 6 8 11 13 15 17 21 22 23 24 28 33 34 36 39 40 41	18	93,18	98,99	96,32	99,91	99,47	97,574
2 3 4 5 7 9 13 16 22 23 25 27 28 29 30 33 35 37 39	19	95,1	99,32	96,35	99,62	99,73	98,024
1 5 6 7 8 12 14 15 16 18 19 24 25 26 30 31 32 33 36 40	20	95,27	98,8	96,21	99,98	99,7	97,992
2 3 4 5 6 7 8 10 15 17 18 19 22 23 24 25 26 28 30 36 39	21	94,84	99,25	96,88	99,98	99,76	98,142
1 2 4 5 6 7 8 9 12 16 17 18 21 23 24 25 27 28 30 31 34 36 37 38 40	25	94,56	99,43	97,66	99,99	99,8	98,288

Tableau 5.8 : MOEDAFS-Two avec cinq algorithmes de classification.

Feature Subset	NF	NB	MLP	SVM	KNN	DT	Moyenne
		Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)
7 14 29 31	4	89,58	89,05	88,82	90,03	89,86	89,468
5 10 17 35 37	5	95,36	95,94	92,5	98,22	97,83	95,97
2 7 10 20 27 32 33 36 39 40	10	93,96	96,98	93,92	99,51	98,97	96,668
2 6 8 16 19 20 22 33 36 39 40	11	94,61	97,77	93,72	99,45	98,96	96,902
1 2 13 20 24 28 31 32 33 36 39 40 41	13	93,37	98	95,82	99,77	99,28	97,248
3 8 11 13 17 19 20 21 23 24 34 39 40 41	14	93,03	97,66	94,64	99,64	99,15	96,824
2 4 5 6 7 8 14 19 21 27 34 37 38 40 41	15	95,81	99,01	95,14	99,9	99,72	97,916
2 3 5 7 8 10 17 18 19 20 23 24 25 26 27 34 35 37	18	96,1	99,12	96,24	99,99	99,8	98,25
1 5 6 7 8 12 14 15 16 18 19 24 25 26 30 31 32 33 36 40	20	95,27	98,8	96,21	99,98	99,7	97,992
1 3 4 5 8 9 12 13 15 16 18 20 22 25 26 27 28 31 32 35 36 37 38 40 41	25	96,51	99,48	96,93	99,99	99,75	98,532

Tableau 5.9 : MOEDAFS-Three avec cinq algorithmes de classification.

Feature Subset	NF	NB	MLP	SVM	KNN	DT	Moyenne
		Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)	Accuracy (%)
11 28	2	60,22	60,28	60,09	60,32	60,22	60,23
24 31 38	3	86,25	85,7	84,9	88,4	88,06	86,66
4 7 16 18 20 32 41	7	91,53	92,12	91,2	93,01	92,64	92,10
2 8 9 12 13 16 31 40	8	91,12	92,45	87,51	93,11	92,83	91,40
2 9 11 13 16 22 24 27 34 40	10	98,18	92,95	90,32	96,76	95,71	94,78
8 13 14 19 26 27 31 33 37 40 41	11	91,18	96,61	94,15	98,13	97,6	95,53
2 3 4 5 14 15 16 22 23 28 34 37	12	94,64	98,6	94,19	99,86	99,54	97,37
2 5 6 9 14 15 18 19 24 34 37 38 40	13	96,11	98,73	95,21	99,75	99,46	97,85
2 3 5 11 12 14 21 23 29 31 32 35 38 41	14	96,53	99,09	94,75	99,96	99,71	98,01
1 3 4 5 7 10 12 15 17 18 25 30 34 36 38	15	96,82	99,15	94,38	99,96	99,77	98,02
1 3 5 6 11 12 17 20 21 22 23 30 34 36 39 40	16	96,32	99,1	96,26	99,89	99,82	98,28
1 2 3 4 5 6 8 12 23 24 27 31 32 33 36 37 38 39	18	95,1	98,85	97,13	99,31	99,78	98,03
1 2 3 4 5 6 7 8 15 16 17 18 21 23 24 26 32 35 40	19	96,43	99,41	96,91	99,99	99,83	98,51
1 3 4 5 6 9 11 12 15 17 18 20 22 26 28 30 32 33 36 37 38	21	96,14	98,72	95,84	99,98	99,83	98,10
1 2 3 5 7 8 9 11 18 19 22 23 24 25 26 27 28 30 31 32 36 40	22	95,92	99,15	96,01	99,99	99,78	98,17

Tableau 5.10 : MOEDAFS-Four avec cinq algorithmes de classification.

Feature Subset	NF	NB	MLP	SVM	KNN	DT	Moyenne
		Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)
30	1	91,05	88,75	89,96	91,30	91,23	90,46
6 36	2	88,18	91,43	89,21	91,92	91,61	90,47
8 17 30	3	91,63	90,81	91,19	91,88	91,80	91,46
5 21 32 36	4	92,55	94,67	91,73	96,51	96,05	94,30
5 7 11 15 29 32 36	7	94,52	95,59	91,88	97,50	96,97	95,29
1 3 10 11 13 14 17 18 26 27 28 30 36 37 40	15	96,07	97,34	94,09	99,64	99,33	97,29

Selon les tableaux 5.7, 5.8, 5.9, et 5.10, les résultats suggèrent que MOEDAFS sélectionne le mieux sous-ensemble d'attributs caractérisés par le plus petit nombre d'attributs et un taux de précision de classification plus élevé. Dans tous les sous-ensembles, les quatre versions de MOEDAFS ont obtenu un meilleur taux de précision pour tous les classificateurs, ce qui présente une plus grande stabilité des solutions dans MOEDAFS.

Dans chaque solution dans les quatre versions MOEDAFS, ils ont obtenu dans chaque classificateur presque le même taux de précision, ce qui reflète la précision et la qualité des solutions.

10.3 Comparaison externe

Dans cette section, on compare les résultats de performance de MOEDAFS aux travaux récents bien connus dans la sélection d'attributs pour la détection d'intrusion. Dans tous les algorithmes impliqués dans la comparaison, NSL-KDD est utilisé comme un ensemble de données pour l'expérimentation. Ces algorithmes appartiennent à différentes techniques qui sont : *les algorithmes déterministes, les algorithmes méta-heuristiques* et *les algorithmes multi-objectifs*. La comparaison externe est divisée en deux parties.

Dans la première partie, on compare les MOEDAFS avec des algorithmes qui sont un sous-ensemble d'attributs comme une solution. Dans la deuxième partie, on compare MOEDAFS avec les travaux qui ont des sous-ensembles de solutions comme multi-solution.

10.3.1 Comparaison MOEDAFS vs algorithmes avec une seule solution

Dans cette partie, on compare les quatre versions des résultats expérimentaux de MOEDAFS avec les algorithmes récents qui ont obtenu une seule solution. On choisit onze algorithmes qui sont basés sur différentes techniques. Le tableau 5.12 montre la comparaison des résultats qui se concentre sur le nombre d'attributs sélectionnés par les approches et le taux de précision maximal (Taux de détection = AR) qui sont atteints.

Tableau 5.11 : Comparaison MOEDAFS vs algorithmes avec une seule solution.

Algorithmes	**NS**	**NF**	**Max Accuracy %**
[4] Mutual Information and Binary gravitational search algorithm (BGSA).	1	5	88.36
[53] Pearson correlation coefficient.	1	17	99.1
[64] Hybrid Bat algoritm and SVM.	1	23	99.28
[58] Chi-square and modified BN.	1	22	96.8
[57] Chi-square and multi class SVM.	1	31	98
[59] Correlation feature selection 1 12 65,43 (U2R)	1	12	65.43 (U2R)
[92] Multi-objective PSO	1	11	98
[69] Consistency based feature selection, SVM, and LPBoost.	1	10	96.3
[71] Vote algorithm with Information Gain	1	8	99.81
[70] Hypergraph-Genetic algorithm and SVM (Multi-objective)	1	35	96.72
[76] Hybridization of Neural Network and K-MeansClustering	1	23	97.63
MOEDAFS-One MOEDAFS-Tow MOEDAFS-Three MOEDAFS-Four	**(16) PS** **(10) PS** **(15) PS** **(6) PS**	**14 (Average of 5 to 25)** **13 (Average of 4 to 25)** **12 (Average of 2 to 22)** **5 (Average of 1 to 15)**	**99,79** **99,76** **99,83** **99,11**
NS: Number of Subsets; NF: Number of Features			

Dans tous les cas, les quatre versions de MOEDAFS ont obtenu le meilleur taux de précision et le plus petit nombre d'attributs. Quand, on compare les travaux [53] [64] [58] [57] [70] [76] contre seulement les résultats moyens de MOEDAFS-One et MOEDAFS-Two, on peut voir le nombre moyen d'attributs dans les sous-ensembles de MOEDAFS-One et MOEDAFS-Two sont plus petits et le taux de la précision de la classification est plus élevé.

De plus, l'approche [59] a atteint 12 attributs dans le sous-ensemble avec un taux de précision de 65 % juste pour (U2R), alors que MOEDAFS-Three a obtenu 12 attributs comme une moyenne dans les sous-ensembles d'attributs sélectionnés avec 99.89 % taux de précision.

D'autre part, en comparant les travaux [4] [92] [69] contre MOEDAFS-Three et MOEDAFS-Four, on peut remarquer que MOEDAFS-Four a sélectionné en moyenne 5 attributs dans les sous-ensembles qui sont meilleur que [4, 92, 69] avec 5, 11 et 10 attributs dans le sous-ensemble, respectivement.

En outre, MOEDAFS-Three et MOEDAFS-Four ont obtenu un meilleur taux de précision moyen de classification (99,83 % et 99,11 % respectivement) que [4] [92] [69] avec (88,36 %, 98 %, et 96,3 % respectivement). En comparant MOEDAFS-Four avec [71], MOEDAFS-Four a sélectionné 5 attributs dans le sous-ensemble comme une moyenne de nombre d'attributs au lieu de [71] avec 8 attributs. Alors que le taux de précision entre eux est presque le même.

Basé sur la comparaison dans le tableau 5.11, on conclut que la méthode proposée MOEDAFS est meilleure que d'autres algorithmes récents qui propose une seule solution dans le nombre de sous-ensembles obtenus et le taux de précision. De plus, la comparaison se fait uniquement avec la moyenne des résultats obtenus dans les versions de MOEDAFS qui confirment leur efficacité.

10.3.2 Comparaison MOEDAFS vs algorithmes avec des multi-solutions

Dans cette comparaison, on évalue les performances des quatre versions de MOEDAFS par rapport aux résultats d'autres algorithmes [2] [68] [1] [94] [81] qui ont des multi-solutions (sous-ensembles). Le tableau 5.13 montre les résultats de la comparaison qui sont focalisés sur NS (nombre de sous-ensembles sélectionnés), NF (nombre d'attributs), et le taux de précision (Mean et Max).

Tableau 5.13 : Comparaison MOEDAFS vs algorithmes avec multi-solution.

Algorithmes	**NS**	**NF**	**Accuracy (Mean & Max)%**
[2] PSO and Genetic Algorithm.	2	8 et 10	98.8 – 99.4
[68] K-means clustering algorithm .	20	16 vers 26	96.93 – 99.73
[1] Hierarchical self-organising maps (Multi-objective).	5	22, 29, 25,25, et 29	98,13 - 99,12.
[94] PSO and Bat algorithm.	16	13 vers 22	93,63 - 97,1
[81] Rough set and NetFlow/IPFIX.	6	11, 16, 16,16, 16,17	90.33 - 98
MOEDAFS-One	**(16)**	**5 vers 25**	**98.01 – 99.79**
MOEDAFS-Tow	**(10)**	**4 vers 25**	**98.2 – 99.76**
MOEDAFS-Three	**(15)**	**2 vers 22**	**94.93 – 99.83**
MOEDAFS-Four	**(6)**	**1 vers 15**	**94.46 – 99.11**
NS : Number of Subsets; NF: Number of Features			

Comme indiqué dans le tableau 5.13, dans tous les cas, les quatre versions de MOEDAFS ont obtenu le plus petit nombre d'attributs dans les sous-ensembles que les autres approches. Quatre versions de MOEDAFS atteignent 5, 4, 3, 2 et même un seul attribut dans le sous-ensemble avec un meilleur taux de précision, tandis que dans les approches de comparaison, le sous-ensemble d'attributs le plus petit est contenu seulement 8 attributs dans le travail [81].

Concernant le taux de précision (Mean et Max), il confirme l'efficacité des quatre versions de MOEDAFS. La meilleure précision Max a appartenu au MOEDAFS et le taux de précision moyen de MOEDAFS-One et MOEDAFS-Two a atteint la meilleure précision moyenne avec 98,01 % et 98,2%.

Cette comparaison montre la haute performance des résultats MOEDAFS concernant le nombre d'attributs dans les sous-ensembles et le taux de précision par rapport à d'autres recherches de multi-solution.

11. conclusion

Dans ce chapitre, un nouvel algorithme de sélection d'attributs multi-objectifs hybrides a été proposé pour les systèmes de détection d'intrusion. L'algorithme MOEDAFS est basé sur l'approche multi-objectif évolutive EDA et MI pour sélectionner les meilleurs sous-ensembles d'attributs pour l'IDS.

Chaque sous-ensemble d'attributs a été sélectionné par MOEDAFS avec une meilleure performance de classification (taux d'erreur) et un plus petit nombre d'attributs comme des solutions non dominées.

Quatre modèles probabilistes ont été proposés et intégrés dans MOEDAFS pour guider la recherche en calculant la probabilité de chaque attribut. Cette probabilité représente les degrés

de la relation de pertinence pour l'attribut f_i avec d'autres attributs et classes cibles. Selon les quatre modèles probabilistes, quatre versions de MOEDAFS ont été proposées.

Les résultats expérimentaux montrent que les quatre versions de MOEDAFS sélectionnent avec succès un ensemble de sous-ensembles d'attributs mieux que les différentes recherches qui ont une solution simple ou multiple. Cette efficacité est due à la nature de la stratégie de recherche effectuée dans MODAFS.

MOEDAFS utilise différents mécanismes qui sont basés sur : l'algorithme évolutionnaire EDA, des modèles probabilistes et les techniques évolutives multi-objectifs pour maintenir la diversité des solutions avec meilleure qualité et à la fois en condition de nombre d'attributs minimum et de taux de précision élevé (AR).

À chaque itération, MOEDAFS utilise le PS et un modèle probabiliste pour mener la recherche d'exploration, où le PS stocke les meilleures solutions sélectionnées dans toutes les générations précédentes. Plus précisément, MOEDAFS est différent des autres MOEA qui, à chaque itération, utilisent les solutions non dominantes stockées dans le modèle PS et le model probabiliste en tant que leader potentiel pour maintenir la diversité des solutions et éviter la stagnation dans les optimaux locaux.

Lorsqu'on exécute le modèle probabiliste dans PS, on modélise les relations entre les attributs et la classe cible des différentes solutions pour générer une nouvelle population (Solutions) pour une nouvelle itération. Ce mécanisme aide l'algorithme à explorer l'espace des solutions, tente de l'orienter vers les meilleures nouvelles solutions et évite en outre les problèmes de convergence prématurée pour conserver la diversité dans les itérations futures. MOEDAFS utilise le mécanisme de classement et de remplacement pour mettre à jour le nouveau PS, qui est une intégration entre l'ancienne et les solutions de nouvelle itération.

Ce mécanisme sera plus utile pour mettre à jour et maintenir le PS de génération à une autre. MOEDAFS filtre certaines foules solutions (Crowd solutions) qui peut susceptibles limité la capacité d'exploration de l'algorithme. Cette élimination donne au MOEDAFS la possibilité d'explorer plus efficacement l'espace de recherche et d'équilibrer ses capacités de recherche globales et locales pour trouver les meilleurs résultats.

MOEDAFS utilise différentes techniques dans son algorithme pour garantir la diversité des solutions, évité de converger vers des optima locaux et obtenir de meilleures solutions.

De ce fait, MOEDAS surpasse les autres algorithmes (les algorithmes de sélection d'attributs déterministes, méta-heuristiques et multi-objectifs.) à solution simple ou multiple.

La supériorité de l'algorithme MOEDAFS apparaît en *termes de diversité de solutions* (*sous-ensembles d'attributs*), en *termes de nombre d'attributs sélectionnés* dans les sous-ensembles, et en *termes de taux de précision* de la classification.

Conclusion Générale et Perspectives

Conclusion Générale et perspectives

Dans cette thèse, nous avons représenté un aperçu global sur la sélection d'attributs pour les systèmes de détection d'intrusion. Nous avons exploré des différents travaux et des contributions dans l'application de la FS (Feature Selection) au problème de l'IDS (Intrusion Detection System). IDS est un outil important dans toute infrastructure de sécurité, il est utilisé pour protéger un système contre les attaques. En outre, il souffre de la complexité de calcul, du temps de réponse et des exigences de stockage.

FS est parmi l'étape de prétraitement, qui cherche à diminuer le degré de ces problèmes en réduisant le nombre d'attributs. FS sélectionne le (s) sous-ensemble (s) des meilleurs attributs, ce qui évite les problèmes de classification.

Une nouvelle taxonomie a été proposée pour construire une mappe qui guide les nouvelles recherches et donner une vue vers le futur. La taxonomie est basée sur les *techniques de sélection*, les *mécanismes de sélection*, *unique ou multiple solution de sous-ensemble*, les *différentes bases de connaissances* utilisées dans l'expérimentation, le *type d'approche* et *l'aspect mono ou multi-objectif* utilisé.

Les différents algorithmes de sélection d'attributs sont classés en cinq classes en fonction de leurs techniques, qui sont utilisées dans chaque œuvre. Nous avons présenté leurs caractéristiques en fonction de la nouvelle taxonomie, et nous avons illustré leurs résultats obtenus en montrant leur nombre de sous-ensembles, le nombre d'attributs dans le sous-ensemble et le taux de précision (AR : Accuracy Rate) (Moyenne, Max)). Nous avons considéré cette étude comme une carte pour comprendre l'état actuel et les défis futurs.

Un nouvel algorithme de sélection d'attributs multi-objectifs hybrides a été proposé pour les systèmes de détection d'intrusion. L'algorithme MOEDAFS (Multi-Objective Estimation of Distribution Algorithms (EDA) for Feature selection) basé sur l'approche multi-objectif évolutive EDA (Estimation of Distribution Algorithms) et IM (Information Mutuelle) pour sélectionner les meilleurs sous-ensembles d'attributs pour l'IDS.

Chaque sous-ensemble d'attributs a été sélectionné par MOEDAFS avec une meilleure performance de classification (taux d'erreur) et le plus petit nombre d'attributs comme des solutions non dominées.

Quatre modèles probabilistes ont été proposés et intégrés dans MOEDAFS pour guider la recherche en calculant la probabilité de chaque attribut. Cette probabilité représente le degré de pertinence de la relation d'un attribut f_i avec d'autres attributs f_j et classes cibles. Selon les quatre modèles probabilistes, nous avons proposé quatre versions de MOEDAFS.

Des résultats expérimentaux ont montré que les versions de MOEDAFS sont appliquées avec succès dans NSL-KDD, ce qui démontre la capacité de rechercher dans l'espace de solution avec des performances plus élevées des solutions non dominées trouvées.

En examinant les fronts de Pareto obtenus par les quatre versions de MOEDAFS, on peut confirmer la *compétitivité*, la *diversité* et *l'efficacité des solutions*. A cela s'ajoute, la haute performance de MOEDAFS effectuée par la comparaison externe. MOEDAFS est plus performant que les algorithmes de sélection d'attribut *déterministes, méta-heuristiques* et *multi-objectifs.*

Les recherches récentes bien connues dans le domaine de la sélection d'attributs pour les IDS sont dominées par les résultats des quatre versions de MOEDAFS.

La supériorité de l'algorithme MOEDAFS apparaît en termes :

- *De diversité de solutions (sous-ensembles d'attributs).*
- *De nombre d'attributs dans le sous-ensemble*.
- *De taux de précision de la classification*.

Les principales performances de l'algorithme MOEDAFS sont obtenues par la stratégie de recherche utilisée pour explorer l'espace de recherche. Trois facteurs importants sont à l'origine de l'efficacité de la stratégie d'exploration de l'algorithme MOEDAFS :

- *la coopération entre l'algorithme EDA et MI.*
- *les modèles probabilistes intégrés pour modéliser les attributs pertinents et guider la stratégie de recherche vers les meilleures solutions.*
- *l'approche multi-objective qui nous donne le compromis entre les solutions*.

Dans les travaux futurs, nous projetterons : 1. d'exécuter MOEDAFS pour d'autres bases de connaissance. 2. Utiliser MOEDAFS pour la sélection d'attributs dans la classification non-supervisé. 3. Intégrer des techniques d'évaluation des filtres pour estimer la qualité des sous-ensembles d'attributs au lieu des algorithmes de classification. 4. D'un autre côté, nous prévoyons d'étendre MOEDAFS à des autres problèmes multi-objectifs.

Bibliographie

Bibliographie

[1] : De la Hoz, E., de la Hoz, E., Ortiz, A., Ortega, J., Martínez-Álvarez. A, *"Feature selection by multi-objective optimisation : Application to network anomaly detection by hierarchical self-organising maps"*, Knowledge-Based Systems, Vol. 71, PP. 322-338. 2014.

[2] : Ahmad. I, *"Feature selection using particle swarm optimization in intrusion detection "*, International Journal of Distributed Sensor Networks, Vol. 11, No. (10), PP. 806954. 2015.

[3] : Liu, H., Yu. L, *"Toward integrating feature selection algorithms for classification and clustering"*, IEEE Transactions on knowledge and data engineering, Vol. 17, No. (4), PP. 491-502. 2005.

[4] : Bostani, H., Sheikhan. M, *"Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems"*, Soft computing, Vol. 21, No. (9), PP. 2307-2324. 2017.

[5] : Yeun-Hee jei, Ick-Whan Bae, Sung-ja Choi and Gang-soo Lee, *"An Inforamtion Security Engineering Paradigm for Overcoming Infrmation Security Crisis"*, 2006 International Conference on Hybrid Information Technology (ICHIT'06), pages 453-461, IEEE Computer Society Press.

[6]: Lunching Lin, Bashar Nuseibeh, Darrel Ince, Michael Jackson, Jonthan Moffett, *"Introdusing Abuse Frame for Analysing Security Requirement"*, Requirements engineering conference, 2003. Proceedings 11th IEEE International, 8-12 sept. 2003, pages: 371-372, IEEE computer society press.

[7]: L. Lin, B. A. Nuseibeh, D. C. Ince, M. Jackson, J. D. Moffett, *"Analysing Security Threats and vulnerabilities using Abuse Frames"*, technical Report NO° 2003/10, the open university, 2003.

[8] : Jan Jurjens, *"Using UMLsec and Goal Tree for Secure Systems Development"*, Symposium of applied computing (SAC2002), Madrid, Spain, March, 10-14, 2002, pages 1026-1030, ACM Press.

[9]: Jan Jurjens, *"UMLsec: Extending UML for Secure Systems Development"*, UML 2002, Dresden, Sept 30, Oct 4, 2002, LNCS 2460, pp. 412-425, Springer-Verlag.

[10]: Jan Jurjens, Pasha Shabalin, *"Tool for Secure System Development with UML"*, International journal on software tools for Technology transfer (STTT), Octobre 2007, pages 527-544, Springer-Varlag 2007.

[11]: Jan Jurjens, *"Secure System Development with UML"*, Springer-Verlag, 2004.

[12]: "*MOSTRO Del.5, WP 3 Methodologies for Organization and Security Analysis*", university of Trento April 27, 2006.

[13]: Nicolas Mayer, *"Model-Based Management of Information System Risk"*, Doctoral thesis in computer science, Numer, Belguim 2009, ISBN: 978-2-87037-640-9 © presses universitaires de Numer 2009. Available at : www.cases.lu.

[14]: Raimundas Matulevicius, Nicolas Mayer, and Patrick Heymans,*"Alignment of Misuse cases With Security Risk management"*, in proceeding of the ARES 2008 symposium on requirements engineering for information security (SREIS 2008), PP: 1397-1404, IEEE Computer Society press 2008.

[15]: Ida Hogganvik, *"A Graphical Approach to Security Risk Analysis"*, series of dissertations submitted to the Faculty Mathematics and Natural Science, University of Oslo. No.662, ISSN: 1501-7710, October 2007.

[16]: P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. "*Modeling Security Requirements through Ownership, Permission and Delegation*". In *Proceedings of the 13th IEEE International Requirements Engineering Conference*, pages 167–176. IEEE Computer Society Press, 2005.

[17]: Mostro Del.6, WP3, *"Whole Methodology Specification"*, University of Trento, October 10, 2007.

[18]: http://sistar.disi.unitn.it/Main_Page . Available 04/10/2018.

[19]: http://www.troposproject.eu/ . Available 04/10/2018.

[20]: Bastian Best, Jan Jurjens and Bashar Nuseibeh, *"Model-Based Security Engineering of Distributed Information systems Using UMLsec"*, 29th International Conference On Software engineering (ICSE'07), IEEE Computer Society Press 2007.

[21]: Jordi Gabot, Nicola Zannone, *"Towards an Integrated Framework for Model-Driven Security Engineering"*, Modelling security workshop in association with Models'08, Toulouse, France, 28th September 2008.

[22]: Trosten Lodderstedt, David Basin, and Jurgen Doser, *"SecureUML: A UML-Based Modelling Language for Model-Driven Security "*, proceeding in UML'02: proceedings of the 5th international conference on the unified modelling language (2002), Dresden, Germany, September 30- October 4, 2002, PP: 426-441, LNCS 2460, Springer-verlag 2002.

[23]: Luncheng Lin, Bashar Nuseibeh, Darrel Ince, Michael Jackson, *"Using Abuse Frames to Bound the Scope of Security Problems"*, proceeding of the 12th IEEE international requirements engineering conference (RE'04), PP: 354-355, IEEE computer society press, 2004.

[24]: Luncheng Lin, Bashar Nuseibeh, Darrel Ince, *"Using Abuse Frames to Bound The Scope of Security problems "*, proceedings of the third international workshop requirements for high assurance systems (RHAS 2004), in conjunction with 12th IEEE international requirements engineering conference. Sept 6, 2004, Kyoto, Japan.

[25] : Terry Bernstein & Anish B.Bhimain & Eugene Schultz & Carol A.Siegel, "*Sécurité Internet pour l'entreprise*", France, paris 1997 ISBN : 2-84180-133-0, Wiley Press. 1997.

[26]: Raimunidas Matulevicius, Nicoals Mayer, Haralambos Mouratidis, Eric Dubois, Patrick Heymans, and Nicolas Genon, *"Adapting SecureTropos for Security Risk Management in The Early Phases Of Information Systems Development"*, CAiSE 2008, LNCS 5074, PP: 541-555, 2008, Springer-verlag 2008.

[27]: Nicoals Mayer, Patrick Heymans, and Raimunidas Matulevicius, *"Design Of a Modelling Language for Information System Security Risk Management"*, In proceeding of the 1st International conference on research challenges in information séance (RCIS 2007). PP: 121-131.

[28] : Sofiane MAZA, *"Une méthodologie de développement sécurisé des systèmes d'information avancés"*, mémoire de magister en Informatique, Université de Université Mohamed KHIDER - BISKRA, 14 Juillet 2010.

[29] : Haley, C. B., Laney, R., Moffett, J. D., & Nuseibeh, B. *"Security Requirements Engineering: A Framework for Representation and Analysis"*. IEEE Transactions on Software Engineering, 34(1), 133–153. IEEE 2008.

[30] : Ilham Maskani, Jaouad Boutahar and Souhaïl El Ghazi El Houssaïni, *"Analysis of Security Requirements Engineering : Towards a Comprehensive Approach"* International Journal of Advanced Computer Science and Applications (ijacsa), 7(11), 2016.

[31]: Shon Harris, *"CISSP Certification ALL-in-one Exam Guide, 6th Edition"*, Publisher: Tata Graw-Hill, ISBN: 978-0-07-178173-2, 2013.

[32] : Eric Maiwald, *"Sécurité des réseaux"*, Publié par Campus Press, ISBN : 2-7440-1240-8 , paris 2001.

[33] : Donald L.Pipkin, "S*écurité des systèmes d'information*", France paris 2000. ISBN : 2-7440-0948-2, CampusPress.

[34] : Merike Kaeo, *"Sécurité des réseaus"*, ISBN : 2-7440-0850-8, Campus Press, France, 2000.

[35]: Pierangela Samarati, and Sabrina Capitani de Vimercati, *"Access Control: Policies, Models, and Mechanisms"*, FOSAD 2000, LNCS 2171, PP: 137-196, 2001, Springer-verlag 2001.

[36] : Luo, B., Xia. J, *"A novel intrusion detection system based on feature generation with visualization strategy, " Expert Systems with Applications*, Vol. 41, No. (9), PP. 4139-4147. 2014.

[37] : Qin, Z., Feng, C., Wang, Y., Li. F, *"Conditional Mutual Information-Based Feature Selection Analyzing for Synergy and Redundancy, " Etri Journal*, Vol. 33, No. (2), PP. 210-218, 2011.

[38] : Xue, B., Cervante, L., Shang, L., Browne, W.N., Zhang. M, *"A multi-objective particle swarm optimisation for filter-based feature selection in classification problems," Connection Science*, Vol. 24, No. (2-3), PP. 91-116, 2012.

[39] : Qu, G., Hariri, S., Yousif. M, *"A new dependency and correlation analysis for features", IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. (9), PP. 1199-1207 ,2005.

[40] : Xue .B, *"Particle swarm optimisation for feature selection in classification,"* A thesis submitted to the Victoria University of Wellington in fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science. Victoria University of Wellington 2014.

[41] : Salappa, A., Doumpos, M., Zopounidis. C, *"Feature selection algorithms in classification problems: An experimental evaluation", Optimisation Methods and Software*, Vol. 22, No. (1), PP. 199-212, 2007.

[42] : Chen, Y., Li, Y., Cheng, X.-Q., Guo. L, *"Survey and taxonomy of feature selection algorithms in intrusion detection system, " In : International Conference on Information Security and Cryptology 2006*, pp. 153-167. Springer.

[43] : Tsang. C.-H, Kwong. S, Wang. H, *"Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection, " Pattern Recognition*, Vol. 40, No. (9), PP. 2373-2391, 2007.

[44] : Xue, B., Qin, A.K., Zhang. M, *"An archive based particle swarm optimisation for feature selection in classification, " In :* Evolutionary Computation (CEC), 2014 IEEE Congress on 2014, pp. 3119-3126.

[45]: *The drapa dataset*. 1998. Available: https://www.ll.mit.edu/ideval/data/; [accessed 2017-01-15].

[46] : *The kdd cup 1999 dataset*. 1999. Available : http://kdd.ics.uci.edu/databases/kddcup99/kddcup99 ; [accessed 2017-01-15].

[47] : *The nsl-kdd dataset*. 2009. Available : http://nsl.cs.unb.ca/nsl-kdd/ ; [accessed 2017-01-15].

[48] : Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A, *"A detailed analysis of the KDD CUP 99 data set, "* In: Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on 2009, pp. 1-6.

[49] : McHugh. J, *"Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory,"* ACM Transactions on Information and System Security (TISSEC), Vol. 3, No. (4), PP. 262-294, 2000.

[50] :Amiri, F., Yousefi, M.R., Lucas, C., Shakery, A., Yazdani. N, *"Mutual information-based feature selection for intrusion detection systems,"* Journal of Network and Computer Applications, Vol. 34, No. (4), PP. 1184-1199, 2011.

[51] : Bolon-Canedo, V., Sanchez-Marono, N., Alonso-Betanzos. A, *"Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset, " Expert Systems with Applications*, Vol. 38, No. (5), PP. 5947-5957, 2011.

[52] : Parsazad, S., Saboori, E., Allahyar. A, *"Fast feature reduction in intrusion detection datasets, "* In : MIPRO, 2012 Proceedings of the 35th International Convention 2012, pp. 1023-1029. IEEE.

[53] : Eid, H.F., Hassanien, A.E., Kim, T.-h., Banerjee. S, *"Linear correlation-based feature selection for network intrusion detection model, " In: Advances in Security of Information and Communication Networks*. pp. 240-248. Springer, 2013.

[54] : Le Thi, H.A., Le, A.V., Vo, X.T., Zidna. A, *"A filter based feature selection approach in msvm using dca and its application in network intrusion detection, "* In: Asian Conference on Intelligent Information and Database Systems 2014, pp. 403-413. Springer.

[55] : Luo, B., Xia. J, *"A novel intrusion detection system based on feature generation with visualization strategy, "* Expert Systems with Applications, Vol. 41, No. (9), PP. 4139-4147, 2014.

[56] : Balakrishnan, S., Venkatalakshmi, K., Kannan. A, *"Intrusion detection system using Feature selection and Classification technique, "* International Journal of Computer Science and Application (IJCSA) Vol. 3, No. (4), November 2014, 2014.

[57] Thaseen, I.S., Kumar, C.A, *" Intrusion detection model using fusion of chi-square feature selection and multi class SVM, " Journal of* King Saud University-Computer and Information Sciences, Vol. 29, No. (4), PP. 462-472, 2017.

[58] : Thaseen, I.S., Kumar, C.A, *"Intrusion Detection Model Using Chi Square Feature Selection and Modified Naïve Bayes Classifier, "* In: Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (*ISBCC–16') 2016*, pp. 81-91. Springer.

[59] : Bahl, S., Sharma, S.K, *"A minimal subset of features using correlation feature selection model for intrusion detection system, "* In: Proceedings of the Second International Conference on Computer and Communication Technologies 2016, pp. 337-346. Springer, 2016.

[60] : Panigrahi, A., Patra, M.R, *" Performance Evaluation of Rule Learning Classifiers in Anomaly Based Intrusion Detection, "* In: Computational Intelligence in Data Mining. Vol 2. pp. 97-108. Springer, 2016.

[61] : Nguyen, H.T., Petrović, S., Franke. K, *"A comparison of feature-selection methods for intrusion detection, "* In : International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security 2010, pp. 242-255. Springer.

[62] : Lin, S.-W., Ying, K.-C., Lee, C.-Y., Lee, Z.-J, "An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection, " Applied Soft Computing, Vol. 12, No. (10), PP. 3285-3290, 2012.

[63] : Ahmad, I., Hussain, M., Alghamdi, A., Alelaiwi. A, "Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components, " Neural computing and applications, Vol. 24, No. (7-8), PP. 1671-1682, 2014.

[64] : Laamari, M.A., Kamel. N, *"A hybrid bat based feature selection approach for intrusion detection, "* In : Bio-Inspired Computing-Theories *and Applications*. pp. 230-238. Springer, 2014.

[65] : Song, J., Zhu, Z., Price. C, *"Feature grouping for intrusion detection system based on hierarchical clustering, "* In : International Conference on Availability, Reliability, and Security 2014, pp. 270-280. Springer, 2014.

[66] : Yin, C., Ma, L., Feng. L, *"Towards accurate intrusion detection based on improved clonal selection algorithm, "* Multimedia Tools and Applications, Vol. 76, No. (19), PP. 19397-19410 , 2017.

[67] : Ravale, U., Marathe, N., Padiya. P, *"Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function, "* Procedia Computer Science, Vol. 45, PP. 428-435, 2015.

[68] : Kang, S.-H., Kim, K.J, "A feature selection approach to find optimal feature subsets for the network intrusion detection system, " Cluster Computing, Vol. 19, No. (1), PP. 325-333, 2016.

[69] : Thaseen, I.S., Kumar, C.A, *"An integrated intrusion detection model using consistency based feature selection and LPBoost, "* In : Green Engineering and Technologies (IC-GET), 2016 Online International Conference on 2016, pp. 1-6. IEEE. 2016.

[70] : Raman, M.G., Somu, N., Kirthivasan, K., Liscano, R., Sriram, V.S, *"An efficient intrusion detection system based on hypergraph-Genetic algorithm for parameter optimization and feature selection in support vector machine, "* Knowledge-Based Systems, Vol. 134, PP. 1-12, 2017.

[71] : Aljawarneh, S., Aldwairi, M., Yassein, M.B, *"Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, "* Journal of Computational Science, 2017.

[72] : Khammassi, C., Krichen. S, *"A GA-LR wrapper approach for feature selection in network intrusion detection, "* Computers & Security, Vol. 70, PP. 255-277, 2017.

[73] : Sun, N.-Q., Li. Y, *"Intrusion detection based on back-propagation neural network and feature selection mechanism, " In :* International Conference on Future Generation Information Technology 2009, pp. 151-159. Springer.

[74] : Chen, Y., Abraham, A., Yang. J, *"Feature selection and intrusion detection using hybrid flexible neural tree, "* In : International Symposium on Neural Networks 2005, pp. 439-444. Springer.

[75] : Subbulakshmi, T., Ramamoorthi, A., Shalinie, S.M, *"Feature Selection and Classification of Intrusions Using Genetic Algorithm and Neural Networks, " In :* Recent Trends in Networks and Communications. pp. 223-234. Springer, 2010.

[76] : Biswas, N.A., Shah, F.M., Tammi, W.M., Chakraborty. S, *"FP-ANK: An improvised intrusion detection system with hybridization of neural network and K-means clustering over feature selection by PCA, "* In: Computer and Information Technology (ICCIT), 2015 18th International Conference on 2015, pp. 317-322. IEEE.

[77] : Manzoor, I., Kumar. N, *"A feature reduced intrusion detection system using ANN classifier, "* Expert Systems with Applications, Vol. 88, PP. 249-257, 2017.

[78] : Reardon, B.J, *"Fuzzy logic versus niched Pareto multiobjective genetic algorithm optimization, "* Modelling and Simulation in Materials Science and Engineering, Vol. 6, No. (6), PP. 717, 1998.

[79] : El Ougli. A, *"Intégration des techniques floues à la synthèse de contrôleurs adaptatifs, "* (2009).

[80] : Muthurajkumar, S., Kulothungan, K., Vijayalakshmi, M., Jaisankar, N., Kannan. A, *"A Rough Set based feature Selection Algorithm for Effective Intrusion Detection in Cloud Mode, "* In : Proceedings of the international conference on advances in communication, network, and computing 2013, pp. 8-13.

[81] : Beer, F., Bühler. U, "Feature selection for flow-based intrusion detection using Rough Set Theory, " In : Networking, Sensing and Control (ICNSC), 2017 IEEE 14th International Conference on 2017, pp. 617-624. IEEE.

[82] : El-Alfy, E.-S.M., Al-Obeidat, F.N, *"A multicriterion fuzzy classification method with greedy attribute selection for anomaly-based intrusion detection, "* Procedia Computer Science, Vol. 34, PP. 55-62, 2014.

[83] : Ramakrishnan, S., Devaraju. S, *"Attack's feature selection-based network intrusion detection system using fuzzy control language, "* International Journal of Fuzzy Systems, Vol. 19, No. (2), PP. 316-328, 2017.

[84] : Raman, M.G., Kirthivasan, K., Sriram, V.S, *"Development of Rough Set–Hypergraph Technique for Key Feature Identification in Intrusion Detection Systems, "* Computers & Electrical Engineering, Vol. 59, PP. 189-200, 2017.

[85] : Wu. S.X., Banzhaf. W, *"The use of computational intelligence in intrusion detection systems: A review, "* Applied soft computing, Vol. 10, No. (1), PP. 1-35, 2010.

[86] : Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., Dai. K, *"An efficient intrusion detection system based on support vector machines and gradually feature removal method, "* Expert Systems with Applications, Vol. 39, No. (1), PP. 424-430, 2012.

[87] : Gao, H.-H., Yang, H.-H., Wang, X.-Y, *"Ant colony optimization based network intrusion feature selection and detection, "* In : Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on 2005, pp. 3871-3875. IEEE.

[88] : Varma, P.R.K., Kumari, V.V., Kumar, S.S, *"Feature Selection Using Relative Fuzzy Entropy and Ant Colony Optimization Applied to Real-time Intrusion Detection System,"* Procedia Computer Science, Vol. 85, PP. 503-510, 2016.

[89] : Zainal, A., Maarof, M.A., Shamsuddin, S.M, *"Feature selection using Rough-DPSO in anomaly intrusion detection, "* In : International Conference on Computational Science and Its Applications 2007, pp. 512-524. Springer.

[90] : Zhou, L.-H., Liu, Y.-H., Chen, G.-L, *"A feature selection algorithm to intrusion detection based on cloud model and multi-objective particle swarm optimization," In :* Computational Intelligence and Design (ISCID), 2011 Fourth International Symposium on 2011, pp. 182-185. IEEE.

[91] : Malik, A.J., Khan, F.A, *"A Hybrid Technique Using Multi-objective Particle Swarm Optimization and Random Forests for PROBE Attacks Detection in a Network,"* In : Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on 2013, pp. 2473-2478. IEEE.

[92] : Sujitha, B., Kavitha. V, *"Layered Approach For Intrusion Detection Using Multiobjective Particle Swarm Optimization,"* International Journal of Applied Engineering Research, Vol. 10, No. (12), PP. 31999-32014 , 2015.

[93] : Tama, B.A., Rhee, K.H, *"A combination of PSO-based feature selection and tree-based classifiers ensemble for intrusion detection systems," In : Advances in Computer Science and Ubiquitous Computing*. pp. 489-495. Springer, 2015.

[94] : Enache, A.-C., Sgârciu, V., Togan. M, *"Comparative Study on Feature Selection Methods Rooted in Swarm Intelligence for Intrusion Detection,"* In : Control Systems and Computer Science (CSCS), 2017 21st International Conference on 2017, pp. 239-244. IEEE.

[95] : Karshenas, H., Santana, R., Bielza, C., Larranaga, P.: *"Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables"*. IEEE Transactions on Evolutionary Computation 18(4), 519-542, 2014.

[96] : Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., & Coello, C. A. C. (2014). *"A Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part I"*. IEEE Transactions on Evolutionary Computation, 18(1), 4–19, 2014.

[97] : Xue, B., Zhang, M., & Browne, W. N. *"Particle Swarm Optimization for Feature Selection in Classification : A Multi-Objective Approach"*. IEEE Transactions on Cybernetics, 43(6), 1656–1671, 2013.

[98] : Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., & Zhang, Q. (2011). *"Multiobjective evolutionary algorithms: A survey of the state of the art"*. Swarm and Evolutionary Computation, 1(1), 32–49, 2011.

[99] : Xue, B., Cervante, L., Shang, L., Browne, W.N., Zhang, M.: *"A multi-objective particle swarm optimisation for filter-based feature selection in classification problems"*. Connection Science 24(2-3), 91-116, 2012.

[100] : Collette, Y., and P. Siarry. *"Optimisation Multiobjectif. Algorithmes (Paris) "*. Eyrolles, ISBN : 2212111681, 2002.

[101] : Xue, B.: *"Particle Swarm Optimisation for Feature Selection in Classification"*. Doctorat thesis in Victoria University of Wellington, 2014.

[102] : Coello Coello, Carlos, Lamont, Gary B., van Veldhuizen, David A. *"Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition"*, Edition 2nd, ISBN 978-0-387-33254-3, Springer, New York, 2007.

[103] : Alain Berro,*"Algorithme évolutionnaire pour l'optimisation multiobjectif"*, Séminaire du 4 novembre 2008 - LAAS. http://www.laas.fr/files/MOGISA/sem_Alain-Berro.pdf (5/10/2018).

[104] : Fonseca, C.M. and Fleming, P.J. *"Multiobjective genetic algorithms"*. In IEEE Colloquium on `Genetic Algorithms for Control Systems Engineering' (Digest No. 1993/130), 28 May 1993. 1993. London, UK : IEEE.

[105] : Srinivas, N., & Deb, K. *"Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms"*. Evolutionary Computation, 2(3), 221–248, 1994.

[106] : Jerey Horn, Nicholas Nafpliotis, and David E. Goldberg ,*"A Niched Pareto Genetic Algorithm for Multiob jective Optimization"*. ICEC '94 c 1994 IEEE (pp. 82-87).

[107] : Eckart Zitzler and Lothar Thiele, *"An Evolutionary Algorithm for Multiobjective Optimization : The Strength Pareto Approach"*, TIK-Report n° 43, 1998.

[108] : Knowles, J., Corne, D. Jerey Horn, Nicholas Nafpliotis, and David E. Goldberg, *"The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation"*. In : Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on 1999, pp. 98-105. IEEE.

[109] : Corne, D. W., Knowles, J. D., & Oates, M. J. (2000). *"The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization"*. Lecture Notes in Computer Science, 839–848, 2000.

[110] : Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: *"A fast and elitist multiobjective genetic algorithm: NSGA-II"*. IEEE transactions on evolutionary computation 6(2), 182-197, 2002.

[111] : PESA II .: *"Region-based Selection in Evolutionary Multiobjective Optimization"*. D.W.Corne and al, in Procedings of the Genetic and Evolutionary Computation Conference (GECCO'2001), p. 283-290, San Francisco, California, 2001.

[112] : Zitzler, E., Laumanns, M., Thiele, L.: *"SPEA2: Improving the strength Pareto evolutionary algorithm"*, 2001.

[113] Hauschild, M., Pelikan, M.: *"An introduction and survey of estimation of distribution algorithms"*. Swarm and Evolutionary Computation 1(3), 111-128, 2011.

[114] : Wang, L., Fang, C., Mu, C.-D., Liu, M.: *"A Pareto-archived estimation-of-distribution algorithm for multiobjective resource-constrained project scheduling problem"*. IEEE Transactions on Engineering Management 60(3), 617-626, 2013.

[115] : Zhang, Q., Zhou, A., Jin, Y.: *"RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm"*. IEEE Transactions on Evolutionary Computation 12(1), 41-63, 2008.

[116] : Zhou, A., Zhang, Q., & Jin, Y. *"Approximating the set of pareto optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm"*. IEEE Transactions on Evolutionary Computation, 13 , 1167-1189, 2009.

[117] : Ding, C., Peng, H.: *"Minimum redundancy feature selection from microarray gene expression data"*. Journal of bioinformatics and computational biology 3(02), 185-205, 2005.

[118] : Cover, T.M., Thomas, J.A.: *"Entropy, relative entropy and mutual information"*. Elements of information theory 2, 1-55, 1991.

[119] : Foithong, S., Pinngern, O., Attachoo, B.: *"Feature subset selection wrapper based on mutual information and rough sets"*. Expert Systems with Applications 39(1), 574-584, 2012.

[120] : Kwak, N., Choi, C.-H.: *"Input feature selection for classification problems"*. IEEE Transactions on Neural Networks 13(1), 143-159, 2002.

[121] : Timme, N., Alford, W., Flecker, B., Beggs, J.M.: *Multivariate information measures: an experimentalist's perspective*. arXiv preprint arXiv:1111.6857, 2011.

[122] : McGill, W.: *"Multivariate information transmission"*. Transactions of the IRE Professional Group on Information Theory 4(4), 93-111, 1954.

[123] : Van de Cruys, T.: *"Two multivariate generalizations of pointwise mutual information"*. In: Proceedings of the Workshop on Distributional Semantics and Compositionality 2011, pp. 16-20. Association for Computational Linguistics, 2011.

[124] : Srinivasa, S. *"A review on multivariate mutual information"*. Univ. of Notre Dame, Notre Dame, Indiana, 2, 1-6, 2005.

[125] : Bell, A. J. *"The co-information lattice"*. In Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation: ICA. Citeseer volume 2003.

[126] : Estévez, P. A., Tesmer, M., Perez, C. A., & Zurada, J. M. (2009). *"Normalized mutual information feature selection"*. IEEE Transactions on Neural Networks, 20 , 189-201. 2009.

[127] : Battiti, R. (1994). *"Using mutual information for selecting features in supervised neural net learning"*. IEEE Transactions on neural networks, 5, 537-550, 1994.

[128] : Kumar, G., & Kumar, K. *"A novel evaluation function for feature selection based upon information theory"*. In Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on (pp. 000395-000399). IEEE.

[129] : Sotoca, J. M., & Pla, F. (2010). *"Supervised feature selection by clustering using conditional mutual information-based distances"*. Pattern Recognition, 43, 2068-2081.

[130] : Howe, D. *"Information System Security Engineering : Cornerstone to the Future,"* Proceedings of the 15th National Computer Security Conference, Baltimore, MD, Vol. 1, October 15, 1992. pp. 244-251.

[131] : Joshua D. Knowles, David W. Corne, and Martin J. Oates, *"The Pareto-Envelope based Selection Algorithm for Multiobjective Optimization"*, In Proceedings of the Sixth International Conference.

[132] : Sofiane MAZA, Mohamed Touahria, *"Feature Selection Algorithms in Intrusion Detection System: A Survey"*, KSII Transactions on Internet and Information Systems Journa, ISSN : 1976-7277, VOL. 12, NO. 10, Oct. 2018.