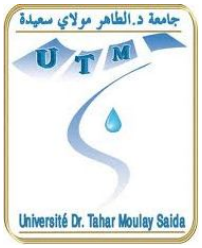
الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Dr. Tahar Moulay SAIDA — جامعة د الطاهر مولاي سعيدة
Faculté : Technologie — كلية التكنولوجيا
Département : Informatique — قسم : الإعلام الآلي



MEMOIRE DE MASTER

Option : MICR

THEME

La Swarm Intelligence Dans La detection D'intrusion : Application au Loups Sauvages

Présenté par :

- KHORSI ASSIA

Encadreur :

-HAMOU Reda Mohamed

Promotion : Juin 2018

Remerciements

En premier lieu, je remercie DIEU de m'avoir aidé et donner la force et la volonté pour achever ce modeste travail. Par la suite ce travail a été réalisé sous la direction du monsieur ***HAMOU Reda Mohamed*** qu'il trouve ici ma profonde reconnaissance et mes sincères remerciements, pour ses encouragements, son aide ses conseils précieux et aussi ses idées pour la réalisation de ce mémoire.

Je tiens aussi à remercier, les membres du jury qui ont accepté de juger ce travail. Je remercie aussi l'ensemble des enseignants ayant intervenu aux cours de notre première année de post-graduation, trouvent ici l'expression de ma gratitude. Et enfin, je tiens à remercier mes chères parents, mes frères et mon mari pour leurs encouragements, leurs aides et leur grande patience avec moi.

Dédicace

Je dédie ce mémoire
Au meilleur des pères KADA
A ma très chère maman
Qu'ils trouvent en moi la source de leur fierté a qui je dois tout, je vous aime énormément.
A mon neveu IYAD HAYTHAM et ma nièce ASSIA
ma plus grande source de bonheur
A mes frères : Omar, Sofiane, Oussama, Ilies pour leur tendresse.
ainsi qu'a ma belle sœur, mes tantes et mes cousines pour leur soutien moral.
A mon mari qui m'a beaucoup soutenu
A ma deuxieme famille "BENAISSA"
A toute ma famille ainsi qu'à mes amies et A tous ceux qui me sont chers.

Khorsi Assia

ملخص

مع التطور في مجال الكمبيوتر والتكنولوجيا، أصبح أمن شبكات الكمبيوتر معرض للعديد من المخاطر مثل الهجمات الخبيثة والعديد من البرامج المفسدة لها.

لهذا ارتأى الباحثون على ضرورة حماية هاته الشبكات، وجعل هذه النظم سليمة ومحصنة من الفيروسات والاختراقات.

بغية الكشف عن أي محاولة انتهاك سياسة الأمن، قد برزت أنظمة الرصد الدائم أو بما يعرف أنظمة كشف التسلل.

هي عبارة عن أنظمة كشف الاختراقات (كشف الهجمات والبرامج الخبيثة والدخيلة).

أنظمة كشف التسلل قد أصبحت منتشرة على نطاق واسع جداً في النظم وأخذت مكاناً هاما في تصميم استراتيجية الأمن والحماية.

وتحقيقا لهذه الغاية وهي غاية الحماية، هدفنا إيجاد نظام كشف تسلل محترف وجيد.

وعليه فقد اقترحنا نموذجين لصنع نموذج إرشادي بيومعلوماتي لبناء نظام لكشف التسلل باستخدام طريقة تصنيف وتعمل على مبدأ مستنبط من العمل الجماعي للذئاب

بحيث أنّ عمل نظامي الحماية يعتمد على انتقاء أحسن فرد .

هذا النموذج يهدف إلى حل المشاكل التي تسببها أنظمة كشف التسلل التقليدية،

ويسعي إلى الكشف الجيد للاختراقات، والبحث عن سياسة الأمن الجيدة.

الكلمات الرئيسية :

شبكات الكمبيوتر، الأمن، نظام كشف التسلل، الذئاب، السرب ، نموذج إرشادي ، بيومعلوماتي.

Abstract

With the development in the field of computer and technological, computer networks are increasingly likely to be the target of various disturbances, contrary to their security, such as congestion, malicious access and attacks. For this purpose, it is necessary to make these systems on safety and immune from viruses and intrusions.
In order to detect any attempted violation of security policy, a permanent monitoring and regular systems may be implemented : these are the Detection of Intrusions (IDS) systems.
Intrusion detection systems have become very widely deployed in information systems and they won an important place in the design of the security strategy.
Our goal is to create an intelligent intrusion detection system.
In this spirit, we propose a model that is a heuristic bio-inspiration model for building an intrusion detection system using a method of classification based on the **social wolves** .
the goals of this models to solve the problems caused by conventional intrusion detection systems and seeks to good detection of intrusions, and good security.
Key words : computer networks, security, detection intrusions system, IDS, meta-heuristic, bio-inspiration,wild wolves.

Résumé

Avec le développement dans le domaine informatique et technologique, Les réseaux informatiques sont de plus en plus susceptibles d'être la cible de dérèglements divers, à l'encontre de leur sécurité, tels que les congestions, les accès malveillants et les attaques. A cet effet, il est nécessaire de rendre ces systèmes on sécurité et Immunitaire contre les virus et les intrusions.
Afin de détecter toute tentative de violation de la politique de sécurité, une surveillance permanente ou régulière des systèmes peut être mise en place : ce sont les Systèmes de Détection d'Intrusions (IDS).
Les systèmes de détection d'intrusions sont devenus très largement déployés dans les systèmes d'informations et ils ont gagné une place importante dans la conception de la stratégie de sécurité.
A cet effet, notre objectif s'inscrit dans le cadre d'une détection d'intrusions distribuée et intelligente.
Dans cet esprit, nous proposons des modèles méta-heuristique bio-inspiré pour construire un système de détection d'intrusions en utilisant des méthodes de classification basé sur **Les loups social** .
Ces modèles visent à résoudre les problèmes induits par des systèmes de détection d'intrusions classique et cherche à la bonne détection des intrusions, et la bonne sécurité.
Mots clés : réseaux informatiques, Sécurité, système de détection d'intrusions, IDS, méta-heuristique, bio-inspiré, loups sauvages.

Table Des matières

Table des matières

Table Des figures

Table des figures

Liste des Tableaux

Liste des tableaux

Introduction Générale

1 Introduction Générale

1.1 Introduction

Les réseaux et les systèmes informatiques sont devenus des outils indispensables au fonctionnement des entreprises et même pour les gens dans leur vie quotidienne. Ils sont aujourd'hui déployés dans tous les secteurs professionnels : la banque, les assurances, la médecine ou encore le domaine militaire. Initialement isolés les uns des autres ces réseaux sont à présent interconnecter et le nombre de points d'accès ne cessent de croître. Ce développement phénoménal s'accompagne naturellement de l'accroissement du nombre d'utilisateurs.qui ne sont pas forcément pleins de bonnes intentions vis-à-vis de ces systèmes informatiques. Ils peuvent exploiter les vulnérabilités des réseaux et les systèmes pour essayer d'accéder à des informations sensibles dans le but de les lire, les modifier ou les détruire, portant atteinte au bon fonctionnement du système.
La sécurité des systèmes informatiques vise à protéger l'accès et la manipulation des données et les ressources d'un système par des mécanismes d'authentification, d'autorisation, de contrôle d'accès, etc.
Le déploiement des ordinateurs et des réseaux a considérablement augmenté les risques causés par les attaques sur les systèmes informatiques qui deviennent un réel problème pour les entreprises et les organisations. Les attaques les plus récentes profitent des failles de sécurité des services ou systèmes informatiques qui sont plus vulnérables. Pour pallier ce problème, des nouvelles approches appelées Systèmes de Détection d'intrusions (SDI) ont fait leur apparition. Ils ont pour objectif de détecter des comportements malveillants.

1.2 Problématique

L'importance de sécurité des systèmes informatiques motive les angles divers de la recherche dont l'objective est de fournir de nouvelles solutions prometteuses qui ne pourraient être assurées par des méthodes classiques. Les systèmes de détection d'intrusions sont l'une de ces solutions qui permettent la détection des utilisations non autorisées et les anomalies, les mauvaises utilisations et les abus dans un système informatique par les utilisateurs externes ainsi que ses utilisateurs internes. Le défi dans le domaine de la sécurité informatique et plus précisément dans les systèmes de détection d'intrusions est de pouvoir déterminer la différence entre un fonctionnement normal et un fonctionnement anomalie. Cependant, les systèmes et les réseaux à protéger sont devenus de plus en plus complexes et larges ainsi que la nature des intrusions courantes et futures nous incite à développer des outils de défense automatiques et surtout adaptatifs. Une solution prometteuse est d'utilisé les systèmes inspiré de la biologie se sont les systèmes bio-inspiré. Le but de ces systèmes ces d'augmenté la détection d'intrusion réel, alors on doit améliorer les systèmes de détection d'intrusion on se basant sur le modèle bio-inspiré pour la détection des anomalies et des éléments dangereux .

1.3 Objectif du travail

l'objectif de notre travail est de voir le lien entre la sécurité informatique et les modèles bio-inspiré pour le but de détecter les intrusions informatique . Dans ce travail nous avons utilisé un modèle bio-inspiré qui est inspiré des loups sauvages ,ce modèle est un nouveau modèle que nous avons proposé au cours de mes études sur les loups et avec des discussion avec mon encadreur , et après la lecture des travaux qui sont réalisés dans ce domaine, pour but de faire détecter les intrusions ou avec un caractère précis bien détecter les dangers et donner une meilleur protection des réseaux et des systèmes informatique.

1.4 Organisation du mémoire

Ce travaille est composé de cinq chapitres.

Après l'introduction générale on trouve le deuxième chapitre de la recherche d'information en commençant par définir la recherche d'information et le SRI et après on définie Les modèles de la recherche d'information et l' évaluation des résultats d'un SRI. .

Dans le troisième chapitre on définie quesqu'un système de détection d'intrusion on commence par les intrusions , après on focalise sur la détection d'intrusion et on conclue par un coup d'œil sur des systèmes de détection d'intrusion .

Dans le quatrième chapitre on parle des méta-heuristique en générale et l'avènement de ces méthodes et on parle sur les méthodes bio-inspiré et a la fin on conclue avec la définition du fonctionnement biologique des loups sauvages et le modèle artificiel. .

Dans le cinquième chapitre on parle de l'implémentation de notre modèle ,d'abord on définit notre corpus de donnée intitulé NSL-KDD , ensuite on définit notre modèle et on applique notre modèle sur le corpus et on discute sur les résultats et les tests.

Enfin ,on conclue notre mémoire avec une conclusion générale et un coup d'œil sur les perspectives de ce travail.

CHAPITRE 2 :

Recherche D'information

2 Recherche D'information

2.1 Introduction

La recherche d'information est un domaine historiquement lié aux sciences de l'information et à la bibliothéconomie. Elle traite l'information dans la manière de l'organiser et de la façon de la sélectionner, elle peut être définie comme une activité qui dans le but de répondre à une question vise à localiser et à traiter une ou plusieurs informations au sein d'un environnement documentaire complexe.
Le traitement de cet environnement ne peut pas être effectué manuellement et donc l'objectif de la recherche d'information est d'extraire les informations pertinentes vis-à-vis d'une requête pour un utilisateur donné à travers l'utilisation d'un ensemble de programmes informatiques appelés systèmes de recherche d'information.
Dans ce chapitre, nous allons définir les concepts de base de la recherche d'information et les systèmes de recherche d'information (SRI).

2.2 Bref historique de la RI

La RI n'est pas un domaine récent :

— 1940 : Avec la naissance des ordinateurs, la RI se concentrait sur les applications dans des bibliothèques. Depuis le début de ces études, la notion de pertinence a toujours été un objet.[**1**]

— 1950 : Début de petites expérimentations en utilisant des petites collections de documents (références bibliographiques). Le modèle utilisé est le modèle booléen.

— 1960-1970 : Expérimentations plus larges ont été menées. On a développé une méthodologie d'évaluation du système qui est aussi utilisée maintenant dans d'autres domaines (des corpus de test ont été conçus pour évaluer des systèmes différents).

— 1970 : Développement du système SMART. Les travaux sur ce système a été dirigéspar G. Salton. [**1**]

2.3 Définition de la recherche d'information et le SRI

La recherche d'information (RI) est un ensemble de techniques et d'outils informatiques dont la finalité initiale était bibliographique : il s'agissait d'aider les usagers à trouver dans des fonds documentaires les références concernant un thème particulier.
En effet, «un Système de Recherche d'Informations (SRI) est un système informatique qui permet de retourner à partir d'un ensemble de documents, ceux dont le contenu correspond le mieux à un besoin en informations d'un utilisateur, exprimé à l'aide d'une requête.» [**2**]
Un SRI inclut différentes fonctionnalités permettant de gérer, stocker, interroger,

rechercher, sélectionner et représenter la grande masse d'information souhaitée. Le schéma suivant résume la liaison entre indexation et recherche d'information ainsi que la place des référentiels dans ce contexte.

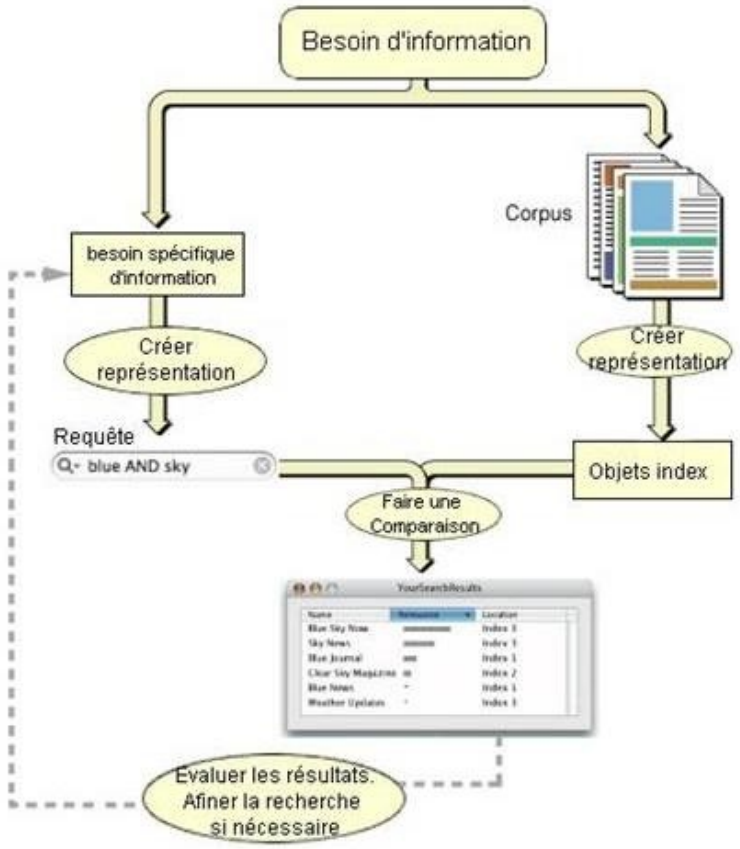


FIGURE 1: Schéma du système de Recherche d'Information

Ce schéma peut être lu de deux manières. Soit en partant de l'indexation avec le corpus indexé, soit en partant de la recherche avec le besoin spécifique d'information.
Un SRI se compose évidement d'un moteur de recherche qui permet de récolter l'information que l'on recherche. Mais si le moteur de recherche est un élément important d'un SRI, il y a d'autres composants sont impératifs. Un SRI peut notamment posséder un système de gestion des documents qui permet d'ajouter des documents. Il peut également avoir des modules d'importation et d'exportation qui permettent l'échange avec d'autres bases documentaires.

2.4 Les modèles de la recherche d'information

Modeles		
MANUEL	AUTOMATIQUE	ADAPTIF
BOOLEEN	VECTORIEL	PROBABILISTES

TABLE 1: modèles de la recherche d'information

2.4.1 Le modèle booléen

Le modèle « exact match » est le modèle le plus commun, il définit une requête est une expression booléenne sur les termes des documents et un document est sélectionné si et seulement si il satisfait l'expression booléenne , Il y a d'autres modèles (non booléens) utilisent un formalisme booléen pour les requêtes (traité de façon déférente) .
Ce modèle simple s'appuie sur l'algèbre booléenne et les opérations ensemblistes correspondantes, Son implantation est efficace grâce aux listes inverses.
Une liste inverse associe à chaque terme descripteur les documents qu'il décrit.

Les opérations booléennes peuvent alors interprétées sous forme d'opérations ensemblistes sur les listes de documents associées aux termes de la requête : ET s'obtient par l'intersection des listes inverses associées à chacun des termes, s'obtient par l'union, s'obtient en prenant le complémentaire.[**3**]

Ce modèle ne permet donc pas de distinguer un document qui comprend quelques termes de la requête de celui qui n'en contient aucun. L'information sur le nombre de termes communs entre requête et descriptions de documents est perdue.

Un autre inconvénient du modèle booléen est la difficulté pour un utilisateur de formuler une requête booléenne exprimant exactement son besoin : l'utilisation de ET et de OU pour un utilisateur ne correspond pas à l'interprétation qu'en fait un ordinateur cet inconvénient tend à être comblé par les langages graphiques de formulation de requête et la construction automatique de requêtes booléennes à partir de mots-clés.
Le modèle booléen reste cependant assez fréquemment utilisé pour effectuer une présélection parmi les documents comme dans les systèmes SMART ou SIRE dans ces situations, le connecteur logique ET dans les requêtes est interprété comme un OU (inclusif).[**3**]

2.4.2 Le Modèle Probabiliste

Ce modèle comprend :

— **Le modèle probabiliste général** Le modèle de recherche probabiliste utilise un modèle mathématique basé sur la théorie de la probabilité. Le processus de recherche se traduit par calcul de proche en proche, du degré ou probabilité de pertinence d'un document relativement à une requête.[**4**]

— **Le modèle de réseau de document ou d'inférence (Document Network) :** Le réseau de document comprend les nœuds de document (un pour chaque document dans la collection), les nœuds de représentation de texte et les nœuds de représentation de concepts. Les nœuds de représentation de texte sont à l'intersection des deux niveaux de représentation. Ils synthétisent l'information sur la manière dont le document est représenté, notamment lorsqu'il s'agit de documents non textuels, tels le son et la vidéo. Un document peut avoir déférentes représentations. Les nœuds représentant les concepts décrivent les déférents concepts identifiés dans le texte des documents et des requêtes.[**4**]

— **Modèle probabiliste de pertinence :**Le modèle probabiliste de pertinence est une méthode probabiliste de représentation du contenu d'un document, proposée en 1976 par Robertson et Jones1. Elle est utilisée en recherche d'information pour exprimer une estimation de la probabilité de pertinence d'un document par rapport à une requête, et ainsi classer une liste de documents dans l'ordre décroissant d'utilité probable pour l'utilisateur. L'une des applications directes de ce modèle est la méthode de pondération Okapi BM25, considérée comme l'une des plus performantes dans le domaine.[**4**]

2.4.3 Le Modèle Vectoriel

Un modèle vectoriel (parfois nommé sémantique vectorielle) est une méthode algébrique de représentation d'un document visant à rendre compte de sémantique, proposé par Gerard Salton dans les années 1970. Elle est utilisée en recherche d'information, notamment pour la recherche documentaire, la classification ou le filtrage de données. Ce modèle concernait originellement les documents textuels et a été étendu depuis à d'autres types de contenus. Le premier exemple d'emploi de ce modèle est le système SMART.[**5**]

— **Problématique :**Le modèle vectoriel est une représentation mathématique du contenu d'un document, selon une approche algébrique .l'ensemble de représentation des documents est un vocabulaire comprenant des termes d'indexation. Ceux-ci sont typiquement les mots les plus significatifs du corpus considéré : noms communs, noms propres, adjectifs.. Éventuellement ils peuvent être des constructions plus élaborées comme des expressions ou des entités sémantiques). À chaque élément du vocabulaire est associé un index unique arbitraire. Chaque contenu est ainsi représenté par un vecteur v, dont la dimension correspond à la taille du vocabulaire. Chaque élément vi du vecteur v consiste en un poids associé au terme d'indice i et à l'échantillon de texte. Un exemple simple est d'identifier vi au nombre d'occurrences du terme i dans l'échantillon de texte. La composante du vecteur représente donc le poids du mot dans le document. L'un des schémas de pondération les plus usités est le TF-IDF.[**1**]

— **Proximité entre documents :**Étant donnée une représentation vectorielle d'un corpus de documents, on peut introduire une notion d'espace vectoriel sur l'espace des documents en langage naturel. On en arrive à la notion mathématique de proximité entre documents.

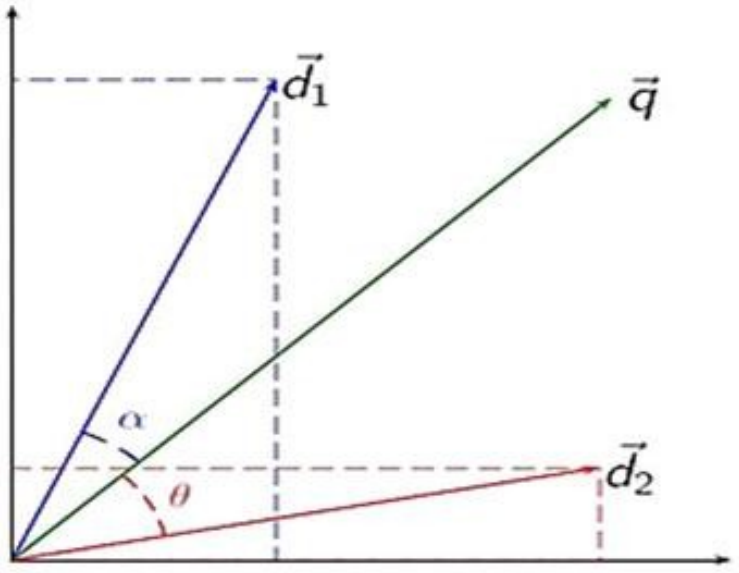


FIGURE 2: Représentation de deux documents et d'une requête dans un espace vectoriel.[**5**]

En introduisant des mesures de similarité adaptées, on peut quantifier la proximité sémantique entre déférents documents. Les mesures de similarité sont choisies en fonction de l'application. Une mesure très utilisée est la similarité cosinus, qui consiste à quantifier la similarité entre deux documents en calculant le cosinus entre leurs vecteurs. La proximité d'une requête à un document sera ainsi donnée par :

$$cos_\alpha = \frac{d_1 * q}{\|d_1\|\|q\|}$$

En conservant le cosinus, nous exprimons bien une similarité. En particulier, une valeur nulle indique que la requête est strictement orthogonale au document. Physiquement, cela traduit l'absence de mots en commun entre q et d1. De plus, cette mesure n'est pas sensible à la norme des vecteurs, donc ne tient pas compte de la longueur des documents.

— **Applications**Parmi les applications existantes, on peut citer :

1. **la catégorisation :** regrouper automatiquement des documents dans des catégories prédéfinies.
2. **la classification :** étant donné un ensemble de documents, déterminer automatiquement les catégories qui permettront de séparer les

documents de la meilleure façon possible (catégorisation non supervisée).

3. **la recherche documentaire :**trouver les documents qui répondent le mieux à une requête (ce que fait un moteur de recherche) ; la requête de l'utilisateur est considérée comme un document, traduite en vecteur, et comparée aux vecteurs contenus dans le corpus des documents indexés.
4. **Le filtrage :** classer à la volée des documents dans des catégories prédéfinies (par exemple, identifier un spam sur la base d'un nombre suspect d'occurrence du mot « pénis » dans un mail et l'envoyer automatiquement à la corbeille).

— **Avantages et inconvénients :**Le modèle vectoriel est relativement simple à appréhender (algèbre linéaire) et est facile à implémenter. Il permet de retrouver assez efficacement des documents dans un corpus non structuré (recherche d'information), son efficacité dépendant pour une grande part à la qualité de la représentation (vocabulaire et schéma de pondération). La représentation vectorielle permet aussi une mise en correspondance des documents avec une requête imparfaite.[**5**]
Il comporte également plusieurs limitations qui furent, pour certaines, corrigés par des affinements du modèle. En particulier, ce modèle suppose que les termes représentatifs sont indépendants. Ainsi, dans un texte, l'ordre des mots n'est pas pris en compte. Dans sa version la plus simple, il ne prend pas non plus en compte les synonymes ou la morphologie des contenus Avec cette approche :

1. Seule la présence ou l'absence de termes est porteuse d'information.
2. Aucune analyse linguistique n'est utilisée, ni aucune notion de distances entre les mots.
3. Les documents sont représentés en "sacs de mots".

De nombreuses solutions ont été proposées dans la littérature pour coder les composantes des vecteurs, c'est-à-dire pour attribuer un poids à chaque terme.
Historiquement, le plus connu de ces codages s'appelé tf.idf, et donne parfois son nom à l'approche vectorielle ; ce codage signifie : term frequency * inverse document frequency.

— **Pondération TF.Idf :** Le terme Tf*Idf désigne un ensemble de pondérations et de sélections de termes. Tf =term frequency (importance du terme pour un document).Idf =Inverted document frequency (on mesure si le terme est discriminant).Les termes importants dans un document doivent avoir un poids fort.

1. **Le facteur Tf (sac de mots, bag of words) :**
   — Tenir compte de la fréquence d'un terme dans le document.
   — Plus un terme est fréquent dans un document plus il est important dans la description de ce document.[**1**]

2. **Le facteur IDF (Inverse Document Frequency) la fréquence du terme dans la collection :**
   — Tenir compte du nombre de documents contenant un terme donné.
   — un terme apparaissant dans tous les documents n'est pas important.

   Avec : Idf =Log(N/ni), où N est la taille de la collection, et ni le nombre de documents contenant le terme ti.

   On déduit la formule classique suivante :

   Wij = freq(ti, dj) * log(taille corpus / docfreq(t))

   Où bien :

   Wij = tfij * log(taille corpus / dfi)

   Où :

   — Wij est le poids du terme ti dans le document Di.
   — tfij est donnée dans la matrice précédente (fréquence du terme ti dans le document Di).
   — taille corpus = nombre de documents du corpus (collection).
   — dfi = fréquence documentaire de ti, le nombre de documents contenant le terme i (Le document apparaît au moins une fois).

— **Mesure de similarité :**Cette mesure correspond au cosinus de l'angle formé par les vecteurs dans l'espace multidimensionnel.

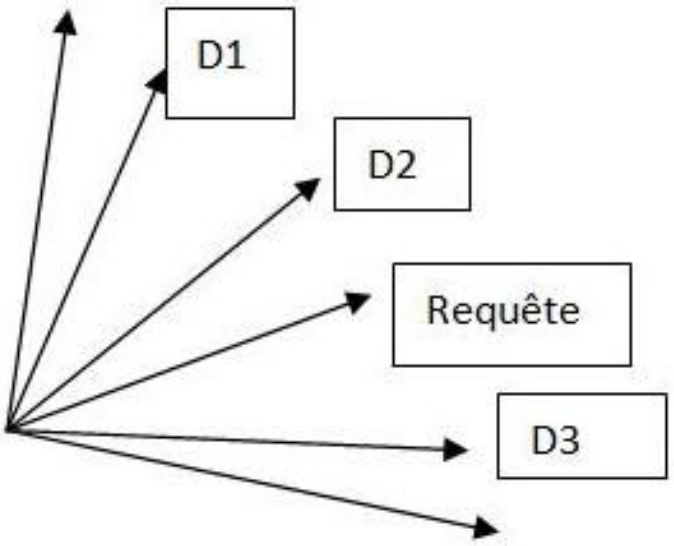


FIGURE 3: Exemple de calcul

Le document D2 est le proche de la requête.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

— **Distance :**Distance entre un document et une requête.

$$Sim(Q, Di) = \frac{\sum_j W_{qj} \cdot d_{ij}}{\sqrt{\sum_j (d_{ij})^2 \cdot \sum_j (w_{qj})^2}}$$

Sim (Q, Di) = similitude entre la requête Q et le document Di

— dij = poids du terme Tj dans le document Di

— wqj = poids du terme Tj dans la requête Q

2.5 Evaluation en Recherche d'Information

L'évaluation est un problème crucial et qui revient régulièrement sous les feux de l'actualité en RI, En effet, les évolutions technologiques remettent en cause les établis dans ce domaine pour les SRI non interactifs.
On peut évaluer les SRI selon plusieurs métriques, d'une façon générale, tout SRI a deux objectifs principaux : retrouver tous les documents pertinents, et rejeter tous les documents non pertinents. Ces objectifs sont évalués par les mesures de rappel et précision comme illustré dans la figure 4.

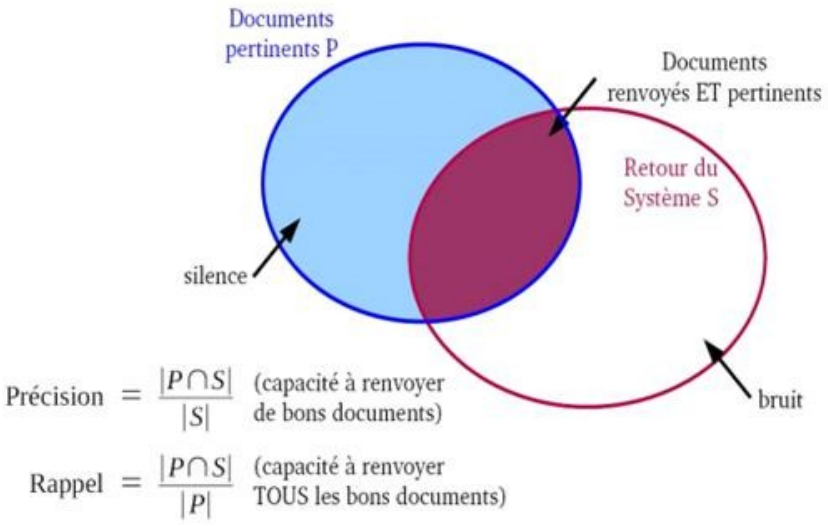


FIGURE 4: Les mesures de précision et rappel

— **Précision :**Elle mesure la proportion de documents pertinents relativement à l'ensemble des documents restitués par le système
— **Rappel :** Il mesure la proportion de documents pertinents restitués par le système relativement à l'ensemble des documents pertinents contenus dans la base documentaire.[**6**]

— **La mesure F :** La moyenne harmonique F combine le rappel et la précision en un nombre compris entre 0 et 1.

2.5.1 Rappel et précision

On mesure l'efficacité d'une technique de recherche d'informations en utilisant deux mesures distinctes. Prenons comme scénario un système de recherche d'informations qui, à la suite d'une requête avec le mot pomme, retourne une liste de 60 documents. Sur 100 documents traitant du fruit « pomme », il en fournit 50, mais les 10 documents restants portent plutôt sur la compagnie Pomme et Fils qui vend des tournevis.
La précision donne le pourcentage de réponses correctes. Dans le cas de ce scénario, le pourcentage de réponses correctes est 50/60 ou 83 pour cent.
Le rappel donne le pourcentage des réponses correctes qui sont données. Dans ce cas précis, le rappel est de 50%.[**6**]
En pratique, il est facile de fournir un système avec un rappel de 100 pour cent : il suffit de retourner la liste de tous les documents. Il est aussi facile d'obtenir une précision qui se rapproche de 100 pour cent : il suffit de retourner aussi peu de documents possibles, sauf un ou deux documents dont on est certain de la pertinence.
En pratique, on cherche un bon compromis entre le rappel et la pertinence. Afin d'évaluer un système, on fait souvent un graphique du rappel par rapport à la pertinence (ou vice versa).

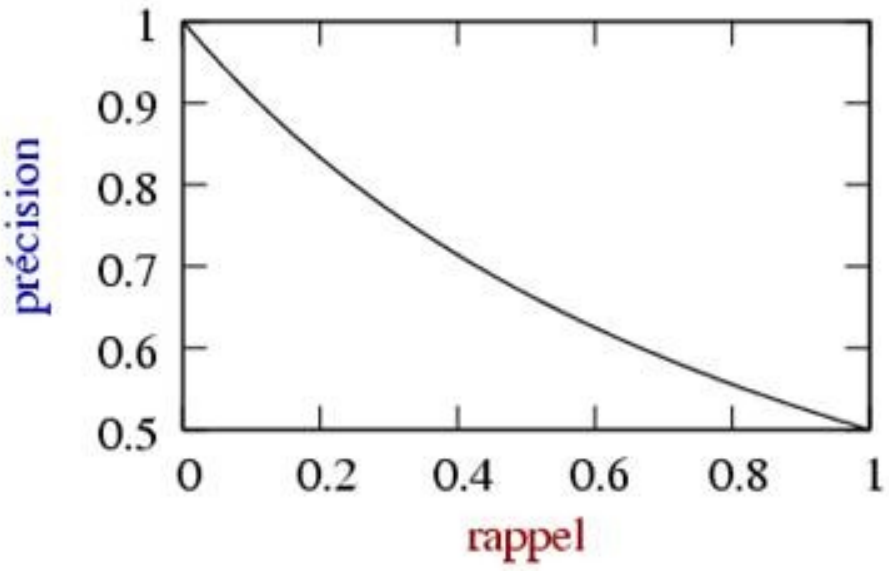


FIGURE 5: Graphique du rappel par rapport à la pertinence

Dans certaines applications, le rappel est beaucoup plus important que la précision. Par exemple, lorsqu'il s'agit de trouver les courriels qui ne sont pas des pourriels, il est très important de trouver tous les courriels qui ne sont pas des

pourriels ; il est cependant moins grave que certains pourriels survivent au filtrage.

Le contraire est parfois vrai. Supposons qu'on doive attribuer à des documents des mots-clés pour faciliter la recherche. On peut mesurer la qualité d'exécution de cette tâche en fonction du rappel (est-ce qu'on a trouvé tous les mots-clés qui s'appliquent ?) et de la précision (est-ce que tous les mots-clés attribués sont pertinents ?). Dans ce cas, la précision n'est pas très importante, mais on souhaite que tous les documents puissent être traités.
Comment choisir le meilleur compromis lorsque la précision et le rappel sont pratiquement d'égale importance ? Une des méthodes utilisées est de maximiser la moyenne harmonique de la précision et du rappel :

$$\frac{r+p}{2rp}$$

On appelle cette moyenne le score F.

2.6 Évaluation des résultats d'un SRI

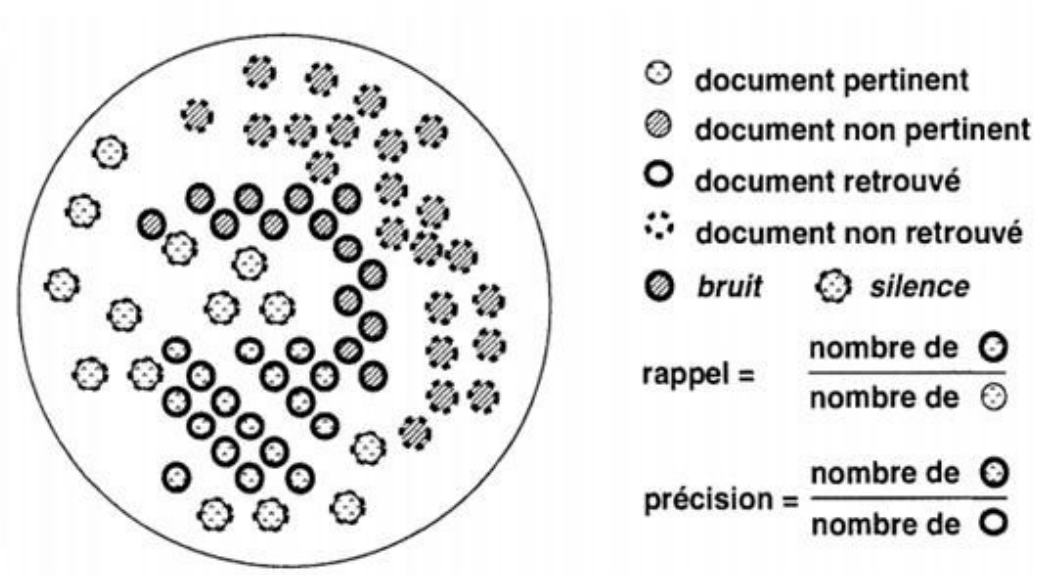


FIGURE 6: Rappel , précision , silence et bruit en RI

2.6.1 Pertinence

La pertinence est un concept abstrait souvent utilisé en recherche d'information. Ce concept recouvre des notions différentes selon que l'on se place du point de vue du système ou de l'utilisateur. Du point de vue du système, la pertinence est la correspondance dans le contexte entre l'énoncé d'un besoin d'information

(une requête) et un document, c'est à dire le point auquel le document couvre matière de l'énoncé du besoin . Le problème est d'anticiper, lors de la conception du système, tous les besoins auxquels le SRI devra répondre. Ceci est bien évidemment impossible a priori, puisque l'apparition d'un système informatisé fait en général naître de nouveaux besoins. Lorsque les documents passent par une phase intermédiaire de représentation de leur contenu, comment déterminer à quel point un document traite d'un sujet particulier et comment le refléter lors de l'indexation ? Faut-il d'ailleurs refléter ces degrés de pertinence lors de l'indexation ? il est possible de dire qu'un document traite d'un sujet s'il contient une information sur ce sujet. Mais on peut simplement dire qu'un document a un rapport avec un concept, qu'il traite d'un sujet à un certain degré c'est ce que van Rijsbergen désigne par le néologisme aboutness que nous traduisons imparfaitement par à-propos.
Du de vue de l'utilisateur, la pertinence dépend de l'utilité de chaque document que lui présente le SRI , Ainsi, un document peut être pertinent du point de vue du système pour la catégorie de sujet dont il traite, mais il peut ne pas être pertinent pour un utilisateur qui est déjà retrouvé auparavant d'autres documents qui couvrent le sujet.[**1**]
L'utilité d'un document pour l'utilisateur ne peut être mesurée qu' à travers les jugements qu'il émet lorsque le SRI le lui présente.
La pertinence est mesurée par une mesure des similarités sur l'espace vectoriel de représentation produit scalaire :

$$s(d, q) = \sum_{i=1}^{n} d_i * q_i$$

avec des valeurs binaires de présence/absence dans les vecteurs la taille de l'intersection entre q et D

2.6.2 Les domaines d'application de RI

La RI est un domaine vaste qui se situe dans les frontières de plusieurs disciplines tel que :

1. Classification /catégorisation (clustering),Question-réponses (Query answering)
2. Filtrage d'information (filtering/recommendation)
3. Méta-moteurs (data-fusion, Meta-search)
4. Résumé automatique (Summarization)
5. Croisement de langues (cross language)
6. Fouille de textes (Text mining)

2.7 Conclusion

Dans ce chapitre nous avons présenté les principales notions et concepts de la recherche d'information.
A travers les différentes sections que nous avons présentées nous concluons que la recherche d'information s'attache à définir des modèles et des systèmes afin de faciliter l'accès à un ensemble de documents se trouvant dans des bases documentaires.
Le but est de permettre aux utilisateurs de retrouver les documents dont le contenu répond à leur besoin en information, il s'agit donc de retourner l'ensemble de documents pertinents.

CHAPITRE 3 :

Détection d'intrusion

3 Détection d'intrusion

3.1 Introduction

L'ordinateur se faisant chaque jour plus présent dans nos vies quotidiennes, la question de la sécurité informatique prend elle aussi de l'importance. Avec l'essor d'internet et du « tout connecté », nous dépendons de plus en plus de la fiabilité de nos appareils, sans même parfois nous en rendre compte. alors grâce a ces différent problème sécurité des systèmes de détection d'intrusions ont été apparu.

3.2 Intrusion

Une intrusion est toute utilisation d'un système informatique à des fins autres que celles prévues, généralement dues à l'acquisition de privilèges de façon illégitime. L'intrus est généralement vu comme une personne étrangère au système informatique qui a réussi à en prendre le contrôle, mais les statistiques montrent que les utilisations abusives (du détournement de ressources à l'espionnage industriel) proviennent le plus fréquemment de personnes internes ayant déjà un accès au système.[**7**]
Une intrusion dans un système informatique est aussi définie par Heady et al. comme :[**8**]
« N'importe quel ensemble d'actions essayant de compromettre l'intégrité, la confidentialité ou l'accessibilité d'une ressource ».
En dépit de différentes formes d'intrusions, elles peuvent être regroupées dans deux classes :[**9**]
-Les intrusions connues : Ces intrusions sont des attaques bien définies qui généralement exploitent des failles connues du système cible.
-Les intrusions inconnues ou anomalies : Ces intrusions sont considérées comme des déviations du profil normal d'un système. Elles sont détectées des qu'il est observé un comportement anormal du système.

3.3 Protection

Nous venons de voir qu'il existe un grand nombre d'attaques connues. Une première idée est de vouloir lutter contre celles déjà existantes. Pour cela, on utilise un IDS qui surveille l'état de l'ensemble du système à protéger. Il existe plusieurs manières pour différencier les types d'IDS. Ainsi, certains s'implémentent de manière software alors que d'autres le sont en hardware. Une autre différenciation se fait sur ce que l'IDS regarde. Il peut être HIDS (Host-based Intrusion Detection System), NIDS (Network-based Intrusion Detection System) ou un mélange des deux types.[**10**]

3.4 Système de détection d'intrusion

3.4.1 Détection d'intrusions

La détection d'intrusions consiste à analyser les informations collectées par les mécanismes d'audit de sécurité, à la recherche d'éventuelles attaques.[**7**]
C'est la capacité à identifier les individus utilisant un système informatique sans autorisation (cracker) et identifier ceux qui ont un accès légitime au système mais qui abusent de leurs privilèges (menace interne).[**11**]
On dit qu'une intrusion a eu lieu quand une victime vient d'enregistrer des pertes au sens large, ou des conséquences relatives à l'attaque. Ces attaques sont motivées par la présence de vulnérabilités dans le système, qui sont exploitées par les intrus pour atteindre leurs objectifs. D'une manière formelle une attaque est une action conduite par un ou plusieurs intrus, contre une ou plusieurs victimes, tout en ayant un objectif à atteindre. Cette action est une série d'événements qui occasionne des conséquences sur la sécurité du système.

3.4.2 Définition d'un IDS

IDS signifie Intrusion Detection System (Système de détection d'intrusions). Il s'agit d'un logiciel permettant de surveiller l'activité d'un réseau ou d'un hôte donné, afin de détecter toute tentative d'intrusion et éventuellement de réagir à cette tentative.[**10**]
Debar [**13**] simplifie le système de détection d'intrusions dans un détecteur qui analyse les informations en provenance du système surveillé (voir figure 7).

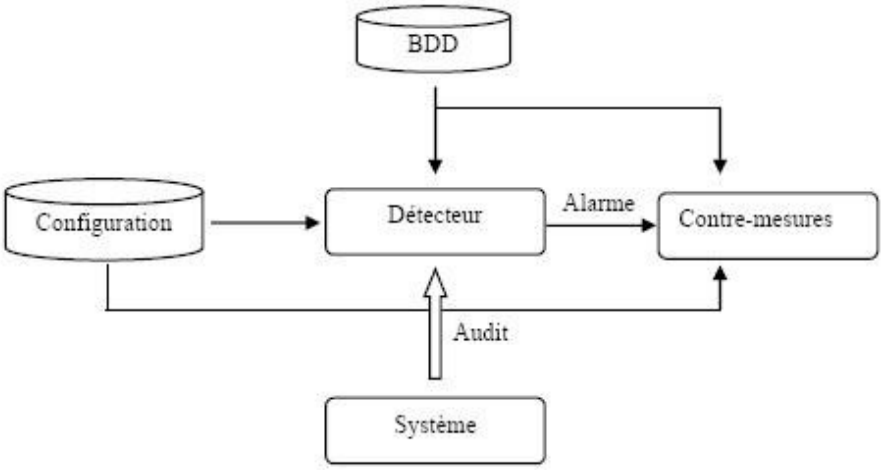


FIGURE 7: Modèle simplifié d'un système de détection d'intrusions

Le détecteur analyse trois types d'informations : les informations de long terme relatives aux techniques utilisées dans la détection (Base de données de signatures), les informations de configuration qui déterminent l'état courant du système, et les

informations d'audit qui décrivent les événements survenus dans le système.
Philip définit trois critères pour évaluer l'efficacité des systèmes de détection d'intrusions.[**14**]
-L'exactitude : On parle de l'exactitude quand les systèmes de détection d'intrusions déclarent comme malicieux une activité légale.
-La performance : La performance de système de détection d'intrusions est le taux de traitement des événements. Si ce taux est faible, la détection en temps réel est donc impossible.
-La complétude : On parle de la complétude quand le système de détection d'intrusions rate la détection d'une attaque. Ce critère est le plus difficile parce qu'il est impossible d'avoir une connaissance globale sur les attaques.

Debbar a rajouté également les deux critères suivants :[**13**]
-La tolérance aux fautes : Le système de détection d'intrusions doit lui-même résister aux attaques, particulièrement au déni de service. Ceci est important parce que plusieurs systèmes de détection d'intrusions s'exécutent sur des matériels ou logiciels connus vulnérables aux attaques.
-La réaction à temps : Le système de détection d'intrusions doit s'exécuter et propager les résultats de l'analyse le plus tôt possible, pour permettre à l'officier de sécurité de réagir avant que des graves dommages n'aient lieu. Ceci implique plus qu'un calcul de performances, parce qu'il ne s'agit pas seulement de temps de traitement des événements, mais aussi de temps nécessaire pour la propagation et la réaction à cet événement.

3.4.3 Caractéristiques des systèmes de détection d'intrusions

Un système de détection d'intrusions se doit de présenter les caractéristiques suivantes :[**11**]
- Être en mesure d'effectuer une surveillance permanente et d'émettre une alarme en cas de détection .
- Fournir suffisamment d'informations pour réparer le système et déterminer l'étendu des dommages et la responsabilité de l'intrus .
- Être modulable et configurable pour s'adapter aux plates-formes et aux architectures réseaux .
-Être en mesure d'assurer sa propre défense, comme supporter que tout ou partie du système soit hors service .
- Avoir un faible taux de faux positifs .
- Être en mesure de tirer les leçons de son expérience et être fréquemment mis à jour avec de nouvelles signatures d'attaques .
- Être en mesure de gérer les informations apportées par chacune des différentes machines et discuter avec chacune d'entre elles .
- Être capable d'apporter une réponse automatique en cas d'attaques, mêmes coordonnées ou distribuées .
- Être en mesure de travailler avec d'autres outils, et notamment ceux de diagnostic de sécurité du système .

- Être en mesure de retrouver les premiers évènements de corruption pour réparer correctement le système d'informations .
- Ne pas créer de vulnérabilités supplémentaires .
- Surveiller l'administrateur système.

Lorsque le nombre de systèmes à superviser augmente et que, par conséquent, les attaques potentielles augmentent également, nous devons, alors, attendre de système de détection d'intrusions les caractéristiques suivantes :
- Il doit être capable de superviser un nombre important de stations tout en fournissant des résultats de manière rapide et précise .
- Il doit fournir « un service minimum de crise » c'est à dire que si certains composants de système de détection d'intrusions cessent de fonctionner, les autres composants doivent être affectés le moins possible par cet état de dégradation .
- Il doit autoriser des reconfigurations et des installations de patchs d'une manière dynamique. Si un grand nombre de stations est supervisé, il devient pratiquement impossible de redémarrer le système de détection d'intrusions sur tous les hôtes lorsque l'on doit effectuer un changement.

3.4.4 Classification des systèmes de détection d'intrusions

Pour classifier les systèmes de détection d'intrusions, on peut se baser sur plusieurs variables. La principale différence retenue est l'approche utilisée, qui peut être soit comportementale, soit par scénarios. Nous verrons ensuite d'autres paramètres permettant de classer les différents systèmes de détection d'intrusions (voir figure 8)[**15**]

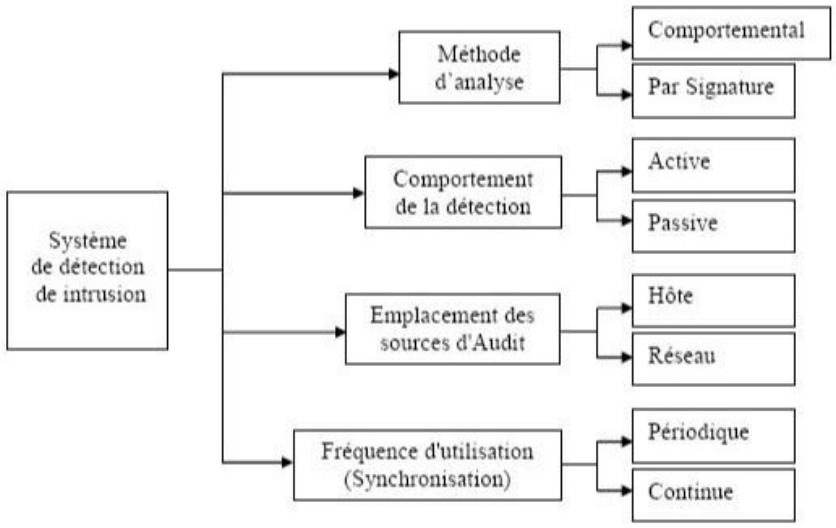


FIGURE 8: Taxonomie des systèmes de détection d'intrusions

3.4.4.1 La méthode de détection

Deux approches ont été proposées à ce jour l'approche comportementale (anomaly

detection) et l'approche par scénario (misuse detection ou knowledge based detection).
La première se base sur l'hypothèse que l'on peut définir un comportement «normal» de l'utilisateur et que toute déviation par rapport à celui-ci est potentiellement suspecte.
La seconde s'appuie sur la connaissance des techniques employées par les attaquants : on en tire des scénarii d'attaque et on recherche dans les traces d'audit leur éventuelle survenue.**[15]**

3.4.4.1.1 Approche comportementale

Une approche, proposée par Anderson**[16]** puis reprise et étendue par Denning**[17]** , consiste à utiliser des méthodes basées sur l'hypothèse selon laquelle l'exploitation d'une vulnérabilité du système implique un usage anormal de celui-ci.**[18]**
La détection d'anomalies consiste à définir, dans une première phase, un certain comportement du système, des utilisateurs, des applications, etc. considéré comme «normal» , dans une seconde phase, on observe l'entité ainsi modélisée et tout écart par rapport au comportement de référence est signalé comme étant suspect.**[19]**
Cette approche recouvre en fait deux problèmes distincts : la définition du comportement «normal» (souvent appelé profil) d'une part, la spécification des critères permettant d'évaluer le comportement observé par rapport à ce profil d'autre part.
Les différentes approches de détection d'anomalies se distinguent essentiellement par le choix des entités modélisées dans le profil et l'interprétation qui est faite des divergences par rapport à ce profil.**[15][18][19]**

-Approche Comportementale

Cette approche présente des avantages très importants. On peut citer entre autres, cet avantage qui n'est pas des moindres et qui caractérise ce type d'approche portant capacité à détecter de nouvelles attaques.
Cependant, l'approche comportementale souffre de quelques défauts intrinsèques :**[18][19]**
- Le choix des différents paramètres du modèle statistique est assez délicat et soumis à l'expérience de l'officier de sécurité .
- En cas de profonde modification de l'environnement du système cible, le modèle statistique déclenche un flot ininterrompu d'alarmes, du moins pendant une période transitoire .
- Un utilisateur peut changer lentement de comportement dans le but d'habituer le système à un comportement intrusif .
- Il est difficile de dire si les observations faites pour un utilisateur particulier correspondent à des activités que l'on voudrait prohiber .
- Pour un utilisateur au comportement erratique, toute activité est « normale ». Une attaque par déguisement sur son compte ne pourra pas être détectée .
- Il n'y a pas de prise en compte des tentatives de collusion entre utilisateurs, alors même que cet aspect est très important, notamment dans le cas des réseaux.

3.4.4.1.2 Approche par scénarios

Le problème de la détection d'intrusions est également couramment approché d'une façon radicalement différente, en visant à détecter des signes de scénario d'attaques connues. Le principe commun à toutes les techniques de cette classe

consiste à utiliser une base de données, contenant des spécifications de scénario d'attaques (on parle de signatures d'attaques et de base de signatures).[**19**]
Le détecteur d'intrusions confronte le comportement observé du système à cette base et lève une alerte si ce comportement correspond à l'une des signatures.
La terminologie « approche par scénarios » vient du fait que l'on s'appuie sur la connaissance des techniques utilisées par les attaquants pour déduire des scénarios typiques.[**7**][**18**][**19**]
Chacune de ces deux approches présente des avantages et des inconvénients.
-Approche par scénarios ?
Ce type de détecteur d'intrusions nécessite une maintenance active : puisque par nature il ne peut détecter que les attaques dont les signatures sont dans sa base, cette base doit être régulièrement (sans doute quotidiennement) mise à jour en fonction de la découverte de nouvelles attaques. Aucune nouvelle attaque ne peut par définition être détectée, ce qui implique un taux plus élevé de faux négatifs. Le problème se pose essentiellement pour les attaques très récentes, dont les signatures n'ont pas encore pu être incluses dans la base. Il y a donc un besoin permanent de veille technologique et de maintenance, ce qui engendre un coût global d'utilisation élevé.
La construction de cette base représente ainsi un problème à part entière et un système de détection d'intrusions de ce type doit s'accompagner d'outils efficaces de maintenance de la base.
De manière générale, les détecteurs de scénarii se montrent fiables pour signaler les attaques référencées dans la base. Théoriquement, leur taux de faux positifs devrait rester très faible, car par définition une alerte n'est levée que dans le cas où la signature d'une attaque est observée. Cependant, pour des raisons de performance, les signatures sont souvent trop simples. Peuvent donc leur correspondent des actions tout a fait légitimes. Le taux de faux positif reste donc élevé avec les outils existant aujourd'hui.
De plus, une éventuelle connaissance de la base de signatures (particulièrement dans le cas des patterns) permet en principe à l'attaquant de construire précisément un scénario non détectable. Cela ne fait que renforcer encore l'exigence de maintenir régulièrement la base.
Ce type de détecteur reste assez facile à mettre en oeuvre, ne nécessitant pas de phase d'apprentissage (ce qui élimine le risque de sur-apprentissage ou de déformation volontaire du profil).[**19**]
Il semble donc indispensable d'hybrider l'approche comportementale avec l'approche par scénarios de manière à profiter des avantages de l'une et de l'autre.

3.4.4.2 Le comportement de la détection (réponse)

Le comportement de la détection décrit la réponse du système de détection d'intrusions à une attaque, elle est qualifiée d'active, si le détecteur réagit activement par des actions correctives, ou proactives (changer les règles de filtrage de Firewall des connexions TCP, ou encore attaquer l'attaquant, etc.). Si le système de détection d'intrusions génère simplement des alarmes (afficher un message sur l'écran, générer un son spécifique, envoi d'un email, archivage dans un fichier ou dans une base de donnée, etc.), la réponse est qualifiée de passive.[**7**]

3.4.4.2.1 Les réponses actives

Les réponses actives des systèmes de détection d'intrusions sont des actions automatisées prises quand certains types d'intrusions sont détectés.
Il y a trois catégories de réponses actives :

-Rassembler des informations additionnelles

Il est très important de rassembler des informations additionnelles sur une attaque afin de l'identifier avec précision. Chacun de nous a fait probablement l'équivalent de cela une fois réveillée par un bruit étrange pendant la nuit. La première chose à faire dans une telle situation est d'écouter d'avantage, recherchant l'information additionnelle qui nous permet de décider si on doit agir ou non. Dans le cas des systèmes de détection d'intrusions, cela se traduira par l'exigence d'analyse des informations additionnelles, faire de corrélations, ou bien communiquer avec d'autres types de systèmes de détection d'intrusions installés sur le réseau.

-Changer l'environnement

Une autre réponse active doit stopper une attaque en progression ensuite bloquer l'accès de l'attaquant. Typiquement les systèmes de détection d'intrusions n'ont pas les capacités de bloquer l'accès d'une personne spécifique, mais ils peuvent uniquement rompre des connexions ou bloquer certains paquets spécifiques en s'appuyant sur les mécanismes des protocoles Internet. Parmi ces actions on trouve :
- L'envoi des paquets TCP de type Reset ou des paquets ICMP au système de l'attaquant pour arrêter la connexion.
- La configuration des routeurs et des Firewalls pour bloquer les paquets provenant des adresses IP de l'attaquant.
- La configuration des routeurs et des Firewalls pour bloquer les paquets selon le numéro de port, le protocole, ou le service utilisé par l'attaquant.

-Agir contre l'intrus

La première option dans la réponse active est d'agir contre l'intrus.
En effet, la forme la plus agressive de cette réponse implique le lancement des contres attaques ou d'essayer d'obtenir activement les informations sur l'hôte ou l'emplacement de l'attaquant.
La première question concernant le choix de cette option même avec beaucoup d'attention est : « est ce que notre action peut être illégale ?». Beaucoup d'attaquants emploient de fausses adresses de réseau quand ils attaquent des sites internet ou de torts causés aux utilisateurs innocents. Donc, il faut prendre les actions avec plus de prudence.

3.4.4.2.2 Les réponses passives

Les réponses passives des systèmes de détection d'intrusions fournissent l'information nécessaire aux administrateurs réseau et aux responsables de la sécurité pour les aider à prendre des mesures basées sur cette information. Beaucoup de systèmes de détection d'intrusions se fondent seulement sur des réponses passives dont les principales sont :

-L'alarme

Les alarmes sont produites par les systèmes de détection d'intrusions pour infor-

mer les administrateurs réseau quand des attaques sont détectées. La forme la plus connue est d'afficher un message d'alerte concernant des informations détaillées de l'intrusion détectée sur la console du responsable de la sécurité réseau. Une autre option très utile consiste à envoyer ces alertes au téléphone du responsable, on peut aussi envoyer des emails, ou générer des alertes sonores.

-SNMP Trap

Certains systèmes de détection d'intrusions sont conçus pour produire des alertes et envoyer les rapports aux systèmes de gestion du réseau (network management system). Ils utilisent le protocole SNMP (Sample Network Management Protocol), qui est un protocole dédié à la gestion du réseau.

-L'archivage

L'archivage permet aux analystes de faire des analyses approfondies, et de faire des corrélations avec l'historique dont ils disposent concernant les événements qui se sont produits auparavant.

3.4.4.3 L'emplacement des sources d'audits

La manière la plus connue pour classifier les systèmes de détection d'intrusions est de les grouper par sources d'informations (sondes). Certains systèmes de détection d'intrusions analysent des paquets capturés à partir du réseau, en plaçant des sniffers sur les différents segments du réseau local. D'autres systèmes de détection d'intrusions analysent des informations produites par le système d'exploitation ou par des applications pour la recherche d'intrusions.[**7**][**11**]

3.4.4.3.1 NIDS (Network-Based IDS)

Le rôle essentiel d'un NIDS est l'analyse et l'interprétation des paquets circulant sur ce réseau.[**20**]

L'implantation d'un NIDS sur un réseau se fait de la façon suivante : des capteurs sont placés aux endroits stratégiques du réseau et génèrent des alertes s'ils détectent une attaque. Ces alertes sont envoyées à une console sécurisée, qui les analyse et les traite éventuellement. Cette console est généralement située sur un réseau isolé, qui relie uniquement les capteurs et la console.[**12**]

Les capteurs placés sur le réseau sont placés en mode furtif (ou stealth mode), de façon à être invisibles aux autres machines. Pour cela, leur carte réseau est configurée en mode « promiscuous », c'est à dire le mode dans lequel la carte réseau lit l'ensemble du trafic, de plus aucune adresse IP n'est configurée.[**10**]

Les avantages des NIDSs sont les suivants :

- Ils peuvent être complètement cachés sur le réseau, donc un attaquant ne saura pas qu'il est contrôlé.
- Un système NIDS unique peut être employé pour contrôler le trafic d'un grand nombre de systèmes cibles potentiels.
- Il peut capturer le contenu de tous les paquets envoyés à un système cible.

Les inconvénients des NIDSs :

-Ils ne peuvent donner d'alarmes que si le trafic correspond aux règles ou aux signatures préconfigurées.

- Ils peuvent manquer le trafic intéressant si le trafic est important sur la bande passante ou si des routes altérées sont utilisées.

- Il ne peut pas déterminer si une attaque a réussie.
- Il ne peut pas examiner le trafic chiffré.
- Il faut des configurations spéciales sur les réseaux commutés pour que le NIDS puisse voir tout le trafic.

3.4.4.3.2 HIDS (Host-Based System)

Les systèmes de détection d'intrusions basés sur l'hôte ou HIDSs analysent exclusivement l'information concernant cet hôte. Comme ils n'ont pas à contrôler le trafic du réseau mais « seulement » les activités d'un hôte ils se montrent habituellement plus précis sur les types d'attaques subies.[**12**][**20**]
De plus, l'impact sur la machine concernée est sensible immédiatement, par exemple dans le cas d'une attaque réussie par un utilisateur. Ces systèmes de détection d'intrusions utilisent deux types de sources pour fournir une information sur l'activité de la machine : les logs et les traces d'audit du système d'exploitation. Chacun a ses avantages : les traces d'audit sont plus précises et détaillées et fournissent une meilleure information alors que les logs qui ne fournissent que l'information essentielle sont plus petits. Ces derniers peuvent être mieux contrôlés et analysés en raison de leur taille, mais certaines attaques peuvent passer inaperçues, alors qu'elles sont détectables par une analyse des traces d'audit.[**12**][**20**]
Les HIDSs sont en général placés sur des machines sensibles, susceptibles de subir des attaques et possédant des données sensibles pour l'entreprise. Les serveurs web et applicatifs peuvent notamment être protégés par un HIDS.
Ce type de système de détection d'intrusions présente un certain nombre d'avantages :.[**12**][**20**]
- Il est possible de constater immédiatement l'impact d'une attaque et donc de mieux réagir.
- Il est possible d'observer les activités se déroulant sur l'hôte avec précision et d'optimiser le système en fonction des activités observées.
- Ils permettent de détecter plus facilement les attaques de type « Cheval de Troie », alors que ce type d'attaque est difficilement détectable par un NIDS.
- Ils permettent également de détecter des attaques impossibles à détecter avec un NIDS, car elles font partie de trafic crypté.
Néanmoins, ce type de système de détection d'intrusions possède également des inconvénients :.[**12**][**20**]
- Il peut être identifié et mis hors service par un attaquant.
- Il ne peut donner l'alerte que si les entrées des journaux d'événements ou les appels au système correspondent à des signatures ou des règles préconfigurées.
- Sensible aux attaques de type Déni de Service.
- Ils sont assez gourmands en CPU et peuvent parfois altérer les performances de la machine hôte.

3.4.4.3.3 IDS d'application

Les systèmes de détection d'intrusions basés sur les applications sont un sous-groupe des HIDSs. Ils contrôlent l'interaction entre un utilisateur et un programme en ajoutant des fichiers de log afin de fournir de plus amples informations sur les activités d'une application particulière. Puisque on opère entre un utilisateur et

un programme, il est facile de filtrer tout comportement notable. Ils se situent au niveau de la communication entre un utilisateur et l'application surveillée.[**20**]
L'avantage de ce système de détection d'intrusions est qu'il lui est possible de détecter et d'empêcher des commandes particulières dont l'utilisateur pourrait se servir avec le programme et de surveiller chaque transaction entre l'utilisateur et l'application. De plus, les données sont décodées dans un contexte connu, leur analyse est donc plus fine et précise.[**12**]
Par contre, du fait que ce système de détection d'intrusions n'agit pas au niveau du noyau, la sécurité assurée est plus faible, notamment en ce qui concerne les attaques de type « Cheval de Troie ». De plus, les fichiers de log générés par ce type de système de détection d'intrusions sont des cibles faciles pour les attaquants et ne sont pas aussi sûrs.[**12**]
Ce type des systèmes de détection d'intrusions est utile pour surveiller l'activité d'une application très sensible, mais son utilisation s'effectue en général en association avec un HIDS. Il faudra dans ce cas contrôler le taux d'utilisation CPU des systèmes de détection d'intrusions afin de ne pas compromettre les performances de la machine.[**12**]

3.4.4.3.4 IDS hybrides

Les systèmes de détection d'intrusions hybrides rassemblent les caractéristiques de plusieurs systèmes de détection d'intrusions différents. En pratique, on ne retrouve que la combinaison de NIDS et HIDS. Ils permettent, en un seul outil de surveiller le réseau et l'hôte. Les sondes sont placées dans des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes les sondes remontent alors les alertes à une machine qui va centraliser, agréger, et lier les informations d'origines multiples.[**11**]

3.4.4.4 La fréquence d'utilisation (La synchronisation)

La synchronisation se rapporte au temps écoulé entre les événements qui sont surveillés et l'analyse de ces événements. Elle est réalisée en : temps réel ou différé.
-En temps différé (Périodique)
Dans cette classe, le flux d'informations émanant des points de surveillance vers les détecteurs n'est pas continu. En effet, l'information est traitée dans un mode semblable au principe « emmagasiner et expédier » : Cette approche est employée surtout dans les HIDSs qui scrutent les logs du système d'exploitation dans des intervalles de temps réguliers.
-En temps réel
Les systèmes de détection d'intrusions en temps réel traitent des flux continus d'informations à partir des différentes sources d'informations. C'est la technique prédominante de synchronisation pour les NIDSs, qui récoltent l'information du trafic réseau. Par conséquent les systèmes de détection d'intrusions peuvent prendre des actions pour affecter la progression d'une attaque détectée.

3.4.5 Les imperfections des systèmes de détection d'intrusions actuels

Avec la croissance rapide de l'Internet, les incidents de la sécurité ont été augmentés. En outre, la technologie s'est développée en une approche complexe comme les attaques coordonnées et les attaques coopératives. Sous ces circonstances, il y a un grand besoin pour des outils logiciels qui peuvent automatiquement détecter une variété d'intrusions. En tant qu'un portier important d'un réseau, les systèmes de détection d'intrusions doivent avoir la capacité de détecter et de défendre des intrusions plus proactivement dans un bref délai. Cependant, les systèmes de détections d'intrusions de nos jours présentent quelques imperfections.[**21**]
1. Mis à part les inconvénients déjà évoqués et qui concernent les approches de détection d'intrusions, beaucoup de systèmes de détection d'intrusions existants (NIDS et HIDS) sont faits d'un seul bloc ou module qui se charge de toute l'analyse. Ils réalisent leur collecte de données ainsi que leur analyse en utilisant une architecture centralisée. Ces systèmes monolithiques exigent beaucoup de données d'audit, ce qui consomme beaucoup de ressources de la machine à protéger, pose des problèmes de mise à jour et font de ce point central d'analyse une cible d'attaque propice (Si un intrus parvient à le faire compromettre alors la totalité du réseau se retrouve sans protection).[**22**][**23**]
En somme les majeurs problèmes issus de cette architecture sont :
- Il est généralement difficile de mettre à jour des profils ou de signatures d'attaques. De plus, les systèmes de détection d'intrusions demande de plus en plus de compétence à celui qui administre le système de sécurité.
- La flexibilité du réseau est limitée : Le traitement de toutes les informations sur une seule station implique des limites sur la taille du réseau à observer. Au-delà de cette limite, l'analyseur central devient incapable de gérer le flot d'informations. La collecte de données peut également engendrer des problèmes lors du trafic excessif sur le réseau.
- Il est difficile de mettre à jour ces systèmes de détection d'intrusions : Les changements et l'ajout de possibilités sont habituellement effectués en éditant un fichier de configuration et cela en ajoutant une entrée dans une table ou en installant un nouveau module. L'IDS nécessite habituellement d'être redémarré afin de prendre en compte ces changements.
- L'analyse des données du réseau peut être imparfaite : réaliser la collecte de données d'un réseau ailleurs que sur la station destinée à les recevoir peut offrir à des intrus la possibilité d'attaques dites d'insertion ou d'évasion. Ceux-ci se servent des failles dans les piles de protocoles du réseau de différents centres serveurs pour dissimuler des attaques ou des dénis de service.
2. Même si le système de détection d'intrusions n'est pas monolithique, la majorité des systèmes de détections d'intrusions actuels sont moins que parfaits. Leurs imperfections sont :[**21**][**22**]
- La plupart des systèmes de détection d'intrusions détectent des attaques dans toute l'entreprise en analysant l'information d'un seul hôte, ou d'une seule interface réseau, à beaucoup d'emplacements dans le réseau. Les composants de système de détection d'intrusions manquent de communication et de coopération. Ceci limite

la capacité à détecter les attaques distribuées à grande échelle.
- La plupart des systèmes de détection d'intrusions sont construits dans une architecture hiérarchique, qui est une structure arborescente avec un système de contrôle au sommet, des unités d'agrégation d'informations aux noeuds internes, et des unités de sonde aux noeuds de feuilles. Dans ce type de système, une grande quantité de données transférée à travers le réseau peut résulter une congestion du réseau.
- En raison de la dépendance dans les structures hiérarchiques, beaucoup de systèmes de détection d'intrusions sont susceptibles d'être attaqués. Un attaquant peut découper une branche de contrôle du système de détection d'intrusions en attaquant un noeud interne ou même en le décapitant en entier. Typiquement, de tels composants critiques ont été durcis pour résister aux attaques directes.
Néanmoins, d'autres techniques de survivabilité tel que la redondance, la mobilité, le recouvrement dynamique etc... manquent dans les implémentations actuelles.
- Beaucoup de systèmes de détection d'intrusions ne peuvent pas combiner adéquatement l"historique des alarmes intrusives pour analyser les comportements intrusifs futurs.
Pour surmonter ces imperfections, les systèmes de détection d'intrusions à base d'agents qui sont distribués, scalables, et reconfigurables sont devenus populaires.[**21**]

3.4.6 Une vue générale de quelque systèmes de détection d'intrusions existants

Il existe plusieurs systèmes de détection d'intrusions qui ont été développés. Dans cette section, nous présenterons quelques systèmes de détection d'intrusions existants.[**24**]

3.4.6.1 IDES

IDES (Intrusion-Detection Expert System) a été développé par SRI International System Design Laboratory. Il représente le modèle de référence pour un grand nombre de systèmes de détection d'intrusions. Il a été conçu pour surveiller un seul hôte et il traite uniquement les données d'audit. Ce système de détection d'intrusions est indépendant du système surveillé, il fonctionne sur une machine dédiée, reliée au système par un réseau. Afin de détecter les violations de sécurité en temps réel, IDES s'appuie aussi bien sur une approche statistique que sur un système expert[**25**][**26**] .Ainsi, il est constitué de deux éléments importants :
-Le détecteur d'anomalie : qui est responsable de la détection des comportements atypiques, en utilisant des méthodes statistiques du modèle de Denning [**31**]
-Le système expert : qui est chargé de détecter les attaques suspectes en s'appuyant sur une base de connaissances de scénarios d'attaques connus.[**25**]

3.4.6.2 NIDES

NIDES (Next- Generation IDES) [**25**][**26**]est une version améliorée du système de détection d'intrusions IDES. Il assure la détection d'intrusions sur plusieurs hôtes (distribuées) en se basant toujours sur les données d'audit. Il n'y a aucune analyse

du trafic réseau. Il utilise les mêmes algorithmes qu'IDES.

3.4.6.3 NADIR

NADIR (Network Anomaly Detection and Intrusion Reporter)[**25**][**26**][**27**] est un système expert qui a été conçu pour le réseau ICN (Integrated Computing Network) du Laboratoire National Los Alamos. Son but est d'analyser les activités réseaux des utilisateurs et d'ICN en se basant sur les règles du système expert qui définissent la politique de sécurité et les comportements suspects. L'inconvénient majeur de ce système est qu'il ne peut être porté sur d'autres réseaux, étant donné que les protocoles réseaux d'ICN ne sont pas standards.

3.4.6.4 DIDS

DIDS (Distributed Intrusion Detection System)[**25**][**26**][**27**] est un système de détection d'intrusions basé réseau qui se base sur l'approche hiérarchique. Afin d'éviter la dégradation des performances de système, DIDS délègue certaines analyses locales aux hôtes locaux. Son architecture se compose de trois entités :
-Le « Host Monitor » : Il en existe un par hôte. Il collecte les données de l'hôte surveillé, fait une première analyse simple sur ces données puis transmet les événements pertinents au « DIDS Director ».
-Le « LAN Monitor » : Il en existe un pour chaque segment LAN. Il surveille le trafic sur le LAN, collecte les informations réseaux et reporte au « DIDS Director » les activités suspectes et non autorisées qui se sont produites sur le réseau.
-Le « DIDS Director » : Il analyse les rapports reçus du « LAN Monitor » et des « Host Monitor » afin de détecter les attaques potentielles.

3.4.6.5 GrIDS

GrIDS (Graph-Based Intrusion Detection System)[**28**] a été conçu pour détecter des attaques à grande échelle. GrIDS considère les réseaux larges comme une agrégation de sous réseaux. Les données concernant l'activité des hôtes et le trafic réseau entre ces hôtes sont rassemblées dans des graphes d'activité qui révèlent la structure causale de l'activité réseau.Les noeuds d'un graphe d'activité correspondent aux hôtes constituant le réseau alors que les arêtes représentent l'activité réseau entre les différents hôtes.
Durant la phase de détection, GrIDS analyse les caractéristiques des graphes d'activité et compare ces graphes à des formes intrusives connues. S'il y a des similitudes entre ces graphes et des attaques connues, il en informe l'officier de sécurité.

3.4.6.6 CSM

CSM (Cooperating Security Manager)[**26**][**27**] est un système de détection d'intrusions qui peut être utilisé dans un environnement de réseau distribué. Son principal objectif est de détecter les activités intrusives de façon non centralisée car utiliser un directeur central qui coordonnerait toutes les activités limiterait la taille du réseau « le problème d'incrémentabilité ». Pour cela, CSM doit s'exécuter sur chaque hôte connecté au réseau.Ainsi, au lieu de reporter les activités anormales à un directeur central, les CSM communiquent entre eux pour détecter d'une manière coopérative les intrusions réseaux. Les composants principaux de

ce système de détection d'intrusions sont :
-Un système de détection d'intrusions local (IDS) :
qui assure la détection d'intrusions pour un hôte local.
-Un gestionnaire de sécurité : qui coordonne la détection d'intrusions distribuée entre les CSM.
-Un gestionnaire d'intrus (IH : intruder handling component) : dont le rôle est d'entreprendre les actions nécessaires lorsqu'une intrusion est détectée.

3.4.6.7 AAFID

Le système AAFID (Autonomous Agent for Intrusion Detection)[**29**][**30**] est la première tentative d'utilisation des agents autonomes pour les systèmes de détection d'intrusions basés réseau où plusieurs agents indépendants opèrent de manière coopérative pour assurer la surveillance du système cible. La décision finale du système est le résultat de coopération entre ces différents processus.

3.5 Conclusion

Nous avons présenté dans ce chapitre une étude des systèmes de détection d'intrusions.
Ils nous est paru évident que ces systèmes sont à présent indispensables aux entreprises afin d'assurer leur sécurité.
La plupart des systèmes de détection d'intrusions sont construits dans une architecture hiérarchique dont une grande quantité de données transférée à travers le réseau peut résulter une congestion de réseau et sont susceptibles d'être attaqués .
Pour offrir un système de détection d'intrusions pour les réseaux actuels mais aussi pour la nouvelle génération, nous proposons une nouvelle génération de systèmes de détection d'intrusions fondés sur des architectures utilisant des méthodes heuristique distribuées et basés sur des modèles bio-iformatique pour modéliser et implémenter une détection d'intrusions distribuée et intelligente.

CHAPITRE 4 :

Méta-Heuristiques

4 Méta-Heuristiques

4.1 Introduction

Les métaheuristiques forment un ensemble de méthodes utilisées en recherche opérationnelle et en intelligence artificielle pour résoudre des problèmes d'optimisation réputés difficiles. Résoudre un problème d'optimisation combinatoire, c'est trouver l'optimum d'une fonction, parmi un nombre fini de choix, souvent très grand. Les applications concrètes sont nombreuses, que ce soit dans le domaine de la production industrielle, des transports ou de l'économie partout où se fait sentir le besoin de minimiser des fonctions numériques, dans des systèmes où interviennent simultanément un grand nombre de paramètres.
De nombreuses définitions ont été faites dans la littérature, dans ce chapitre nous retiendrons que deux définitions "A metaheurisic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem.
"A metaheuristic is an iterative master process that guides and modifies the operations of subordinate heuristics to effciently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a constructive method."[**32**]
Alors une Méta heuristique est une méthode algorithmique capable de guider et d'orienter le processus de recherche dans un espace de solution, souvent très grand à des régions riches en solutions optimales. Le fait de rendre cette méthode abstraite et plus générique conduite à un vaste utilisation pour des champs d'applications différents.
A ces applications, les Méta heuristiques permettent, de trouver des solutions, peut- être pas toujours optimales, en tout cas très proches de l'optimum et en un temps raisonnable. Elles se distinguent en cela des méthodes dites exactes, qui garantissent certes la résolution d'un problème, mais au prix de temps de calcul prohibitifs.

4.2 Optimisation Combinatoire

En mathématique, l'optimisation combinatoire recouvre toutes les méthodes qui permettent de déterminer l'optimum d'une fonction avec ou sans contraintes. Soit S un ensemble de solutions à un problème d'optimisation et f une fonction objective qui mesure la valeur f(s) avec s $\in$ S. Pour un problème de minimisation on cherche à déterminer une solution s qui minimise la fonction objective. Un minimum est une solution qui fait partie des solutions réalisable, dans le domaine d'optimisation on distingue deux types de minimums :

— **Minimum Local :** une solution s est minimum local par apport à une structure de voisina N si $\forall$ s' $\in$ N(s),f(s) $\leq$ f(s')

— **Minimum Global :** une solution s est minimum global si $\forall$ s'$\in$ S, f(s) $\leq$ f(s')
— **Voisinage :** le voisinage est une fonction notée N qui associe un sous ensemble de S a toute solution s, les voisins de s sont s'$\in$ N(s).

4.3 Intensification et diversification

Le principe d'intensification et de diversification est un point critique pour tout Méta heuristique, il consiste à trouver un compromis entre les deux tendances duales suivantes :

— Il s'agit d'une part d'intensifier l'effort de recherche vers les zones les plus prometteuses de l'espace de solutions.
— Il s'agit d'autre part de diversifier l'effort de recherche de façon à être capable de découvrir de nouvelles zones contenant de meilleur es combinaisons.

La façon d'intensifier ou de diversifier la recherche dépend d'une méta heuristique à une autre et dans la plus part des cas la modification des paramètres, dont le Méta heuristique dépond, permettent à celle-ci d'échapper à une convergence prématurée, autrement dit d'échapper d'un minimum local. Pour les approches dites perturba ives, l'intensification de la recherche se fait en favorisant l'exploration des meilleurs voisins d'une solution. La diversification d'une approche perturbatrice se fait généralement en introduisant une part d'aléatoire, par exemple autorisé avec une faible probabilité la recherche à choisir des voisins de moins bonne qualité.
Pour les approches constructives, l'intensification de la recherche se fait en favorisant, à chaque étape de la construction, le choix de composants appartenus aux meilleurs combinaisons précédemment construites. La diversification se fait en introduisant une part d'aléatoire permettant de choisir avec une faible probabilité de moins bons composants.
En général, plus on intensifie la recherche d'un algorithme en l'incitant à explorer les combinaisons proches des meilleures combinaisons trouvées, et plus il converge rapidement. Cependant, si l'on intensifie trop la recherche, l'algorithme risque d'être stagner autour d'optima locaux.
L'équilibre entre intensification et diversification dépend du temps de calcul dont on dispose pour résoudre un problème donné. Plus ce temps est petit et plus on a intérêt à favoriser l'intensification pour converger rapidement, quitte à converger vers des combinaisons de moins bonne qualité. Cet équilibre dépend également de l'instance du problème à résoudre, plus particulièrement de la topologie de répartition des solutions réalisables. Différentes approches ont été proposés pour adapter les valeurs des paramètres manipulés. Les plus répondues sont ceux qui adoptent dynamiquement les valeurs au cours de la recherche de solution. Enfin, difficile de trouver les valeurs adéquates permettant d'intensifier et de diversifier la recherche à la fois.

4.4 Classification des Méta heuristiques

Les problèmes d'optimisation combinatoire sont souvent des problèmes très difficiles dont la résolution par des méthodes exactes peut s'avérer très longue ou peu réaliste. L'utilisation de méthodes heuristiques permet d'obtenir des solutions de bonne qualité en un temps de résolution raisonnable. Les heuristiques sont aussi très utiles pour le développement de méthodes exactes fondées sur des techniques d'évaluation et de séparation (Branch and Bound).
Une heuristique est un algorithme qui a pour but de trouver une solution réalisable, sans garantie d'optimalité, contrairement aux méthodes exactes qui garantissent des solutions exactes. Comme les algorithmes de résolution exacte sont de complexité exponentielle pour les problèmes difficiles, il peut être plus judicieux de faire appel aux heuristiques pour calculer une solution approchée d'un problème ou aussi pour accélérer le processus de résolution exacte. Généralement une heuristique est conçue pour un problème particulier, mais les approches peuvent contenir des principes plus généraux. On parle de Méta heuristique.
Une manière de classifier les Méta heuristiques est de distinguer celles qui travaillent avec une population de solutions de celles qui ne manipulent qu'une seule solution à la fois. Les méthodes qui tentent itérativement d'améliorer une solution sont appelées méthodes de recherche locale ou méthodes de trajectoire.
La méthode Tabou, le Recuit Simulé et la Recherche à Voisinages Variables sont des exemples typiques de méthodes de trajectoire. Ces méthodes construisent une trajectoire dans l'espace des solutions en tentant de se diriger vers des solutions optimales. L'exemple le plus connu de méthode qui travaille avec une population de solutions est l'algorithme génétique. La figure suivante donnera un panorama des méthodes les plus utilisées.

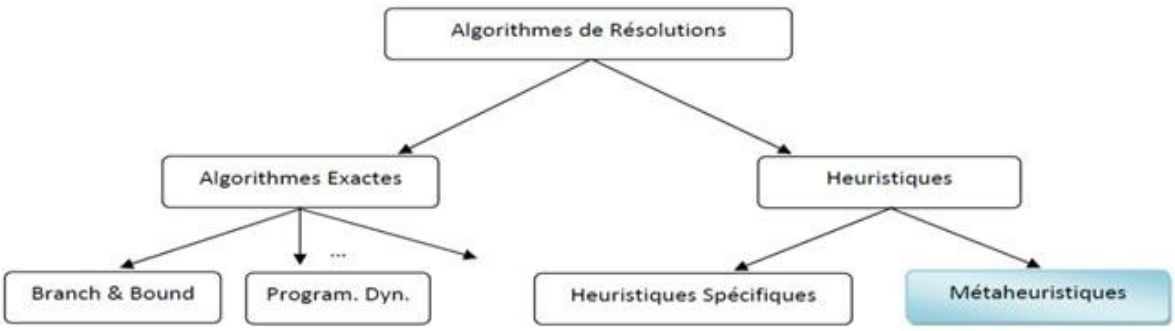


FIGURE 9: Classes des méthodes de résolutions

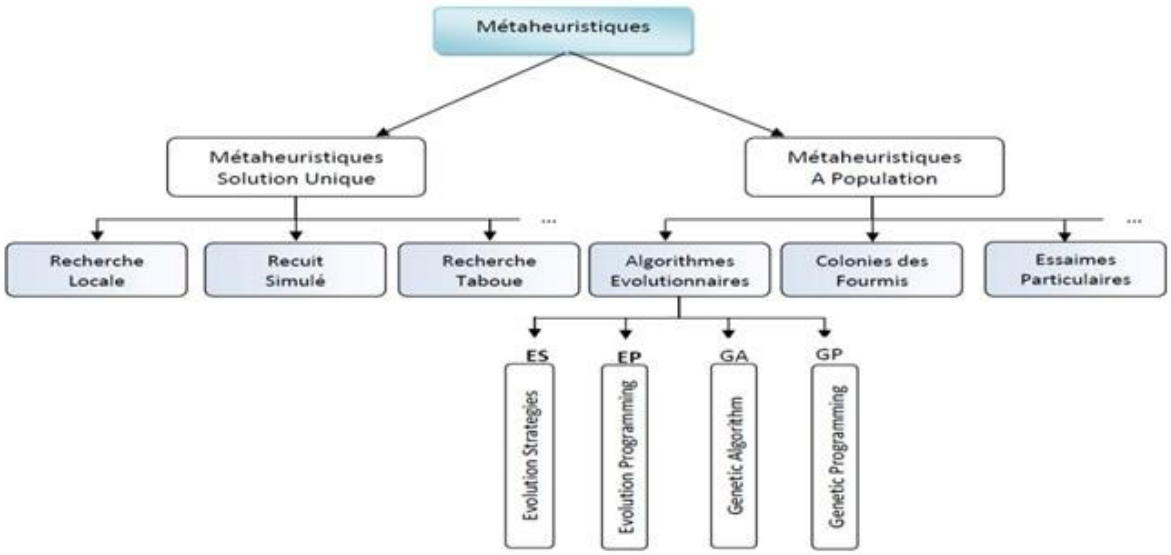


FIGURE 10: Classes des méta heuristiques

4.4.1 Heuristiques

Une heuristique est un algorithme qui fournit rapidement (en un temps polynomial) une solution approchée et réalisable, pas nécessairement optimale, pour un problème d'optimisation difficile. Cette méthode approximative est le contraire d'un algorithme exact qui donne une solution optimale pour un problème donné. Il y a une multitude d'heuristiques qui ont déjà été proposées dans la littérature. Nous pouvons citer des heuristiques très simples telles que les algorithmes gloutons **[33]** ou les approches par amélioration itérative. Le principe des méthodes gloutonnes est de faire une succession de choix optimaux localement, jusqu'à ce que l'on ne puisse plus améliorer la solution, et ce, sans retour en arrière possible. Le fonctionnement d'une heuristique gloutonne est similaire à celui d'un algorithme glouton exact. La différence réside dans le fait que nous n'imposons plus que la solution obtenue soit optimale, nous obtenons donc un algorithme d'approximation.

4.4.2 Méta heuristiques

Des heuristiques plus poussées, adaptables à un grand nombre de problèmes différents, sans changements majeurs dans l'algorithme, ont été mises au point et ont donné naissance à une nouvelle famille d'algorithmes d'optimisation stochastiques : les méta-heuristiques. Le terme méta-heuristique a été inventé par Fred Glover en 1986, lors de la conception de la recherche tabou.**[33]**
Les Méta heuristiques forment une famille d'algorithmes d'optimisation visant à résoudre des problèmes d'optimisation difficile, pour lesquels nous ne connaissons pas de méthodes classiques plus efficaces. Elles sont généralement utilisées comme des méthodes génériques pouvant optimiser une large gamme de problèmes différents, d'où le qualificatif méta. Leur capacité à optimiser un problème à partir d'un

nombre minimal d'informations est contrebalancée par le fait qu'elles n'offrent aucune garantie quant à l'optimalité de la meilleure solution trouvée. Cependant, du point de vue de la recherche opérationnelle, ce constat n'est pas forcément un désavantage, puisque l'on préfère toujours une approximation de l'optimum global trouvée rapidement à une valeur exacte trouvée dans un temps rédhibitoire.
Il existe un grand nombre de Méta heuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. La plupart des méta-heuristiques utilisent des processus aléatoires comme moyens de récolter de l'information et de faire face à des problèmes comme l'explosion combinatoire. Les Méta heuristiques peuvent être considérées comme des algorithmes stochastiques itératifs, où elles manipulent une ou plusieurs solutions à la recherche de l'optimum. Les itérations successives doivent permettre de passer d'une solution de mauvaise qualité à la solution optimale. L'algorithme s'arrête après avoir atteint un critère d'arrêt, consistant généralement en l'atteinte du temps d'exécution imparti ou en une précision demandée. Ces méthodes tirent leur intérêt de leur capacité à éviter les optima locaux, soit en acceptant des dégradations de la fonction objectif au cours du traitement, soit en utilisant une population de points comme méthode de recherche.
Les Méta heuristiques sont souvent inspirées de processus naturels qui relèvent de la physique (l'algorithme du recuit simulé), de la biologie de l'évolution (les algorithmes génétiques) ou encore de l'éthologie (les algorithmes de colonies de fourmis ou l'optimisation par essaim particulaire).
Les Méta heuristiques se caractérisant par leur capacité à résoudre des problèmes très divers, elles se prêtent naturellement à des extensions. Pour illustrer celles-ci, nous pouvons citer :

- Les Méta heuristiques pour l'optimisation multi objectif [**33**] : où il faut optimiser plusieurs objectifs contradictoires. Le but ne consiste pas ici à trouver un optimum global, mais à trouver un ensemble d'optima, qui forment une surface de compromis pour les différents objectifs du problème
- Les Méta heuristiques pour l'optimisation multimodale [**33**] : où l'on ne cherche plus l'optimum global, mais l'ensemble des meilleurs optima globaux et/ou locaux .
- Les Méta heuristiques pour l'optimisation de problèmes bruités : où il existe une incertitude sur le calcul de la fonction objectif, dont il faut tenir compte dans la recherche de l'optimum. [**34**]
- Les Méta heuristiques pour l'optimisation dynamique[**33**] : où la fonction objectif varie dans le temps, ce qui nécessite d'approcher l'optimum à chaque pas de temps .
- Les Méta heuristiques hybrides [**33**] : qui consistent à combiner différentes Méta heuristiques, afin de tirer profit des avantages respectifs.
- Les Méta heuristiques parallèles[**33**] : où l'on cherche à accélérer le calcul, en répartissant la charge de calcul sur des unités fonctionnant de concert. Le problème revient alors à adapter les Méta heuristiques pour qu'elles soient distribuées.

4.5 Méta heuristiques perturbatives

Les approches perturbaives les plus connues sont les algorithmes génétiques, décrits en 4.5.1, et la recherche locale, décrite en 4.5.2.

4.5.1 Les algorithmes génétiques

Les algorithmes génétiques s'inspirent de la théorie de l'évolution et des règles de la génétique qui expliquent la capacité des espèces vivantes à s'adapter à leur environnement par la combinaison des mécanismes suivants :

- la sélection naturelle fait que les individus les mieux adaptés à l'environnement tendent à survivre plus longtemps et ont donc une plus grande probabilité de se reproduire
- la reproduction par croisement fait qu'un individu hérite ses caractéristiques de ses parents, de sorte que le croisement de deux individus bien adaptés à leur environnement aura tendance à créer un nouvel individu bien adapté à l'environnement
- la mutation fait que certaines caractéristiques peuvent apparaître ou disparaître de façon aléatoire, permettant ainsi d'introduire de nouvelles capacités d'adaptation à l'environnement, capacités qui pourront se propager grâce aux mécanismes de sélection et de croisement.

Les algorithmes génétiques reprennent ces mécanismes pour définir une méta-heuristique. L'idée est de faire évoluer une population de combinaisons, par sélection, croisement et mutation, la capacité d'adaptation d'une combinaison étant ici évaluée par la fonction objectif à optimiser.

4.5.2 Recherche locale

Une recherche locale explore l'espace des combinaisons de proche en proche, en partant d'une combinaison initiale et en sélectionnant à chaque itération une combinaison voisine de la combinaison courante, obtenue en lui appliquant une transformation élémentaire.

4.6 Méta-heuristiques constructive

Les approches constructives construisent une ou plusieurs combinaisons de façon incrémentale, c'est-à-dire, en partant d'une combinaison vide, et en ajoutant des composants de combinaison jusqu'à obtenir une combinaison complète. Ces approches sont dites "basées sur les modèles" dans[**38**], dans le sens où elles utilisent un modèle, généralement stochastique, pour choisir à chaque itération le prochain composant de combinaison à ajouter a la combinaison en cours de construction.
Il existe différentes stratégies pour choisir les composants à ajouter à chaque itération, les plus connues étant les stratégies gloutonnes et gloutonnes aléatoires, décrites en 4.6.1, les algorithmes par estimation de distribution, décrits en 4.6.2 et la méta-heuristique d'optimisation par colonies de fourmis, introduite en 4.6.3

4.6.1 Algorithmes gloutons et gloutons aléatoires

Les algorithmes gloutons (greedy) construisent une combinaison en partant d'une combinaison vide et en choisissant à chaque itération un composant de combinaison pour lequel une heuristique donnée est maximale.

4.6.2 Algorithmes par estimation de distributions

Les algorithmes par estimation de distribution (Estimation of Distribution Algorithms ; EDA)[**39**] sont des algorithmes gloutons aléatoires itératifs : à chaque itération un ensemble de combinaisons est généré selon un principe glouton aléatoire similaire à celui décrit en 4.6.1. Cependant, les EDA exploitent les meilleures combinaisons construites lors des itérations précédentes pour construire de nouvelles combinaisons.

4.6.3 Optimisation par colonies de fourmis

Il existe un parallèle assez fort entre l'optimisation par colonies de fourmis (Ant Colony Optimization ; ACO) et les algorithmes par estimation de distribution[**40**] . Ces deux approches utilisent un modèle probabiliste glouton pour générer des combinaisons, ce modèle évoluant en fonction des combinaisons précédemment construites dans un processus itératif d'apprentissage. L'originalité et la contribution essentielle d'ACO est de s'inspirer du comportement collectif des fourmis pour faire évoluer le modèle probabiliste. Ainsi, la probabilité de choisir un composant est définie proportionnellement à une quantité de phéromone représentant l'expérience passée de la colonie concernant le choix de ce composant. Cette quantité de phéromone évolue par la conjugaison de deux mécanismes : un mécanisme de renforcement des traces de phéromone associées aux composants des meilleures combinaisons, visant à augmenter la probabilité de sélection de ces composants ; et un mécanisme d'évaporation, visant à privilégier les expériences récentes par rapport aux expériences plus anciennes.

4.7 Les loups sauvages

4.7.1 Introduction

Dans la nature, plusieurs espèces sont caractérisées par le comportement social.Les bancs de poissons, les nuées d'oiseaux, et les troupeaux d'animaux et les meutes de bêtes sauvages, sont le résultat du besoin biologique qui les pousse à vivre en groupe.De ces principes là, les chercheurs se sont inspirés pour développer des méthodes basées sur les comportements de ces animaux, et ont donné naissance à ce que l'on appelle par Métaheuristique. Ce mot concerne toutes les méthodes qui modélisent l'interaction des agents (animaux) qui sont en mesure de s'auto-organiser. Elles représentent des méthodes de résolution de problème combinatoires qui consistent à réitérer certains processus jusqu'à obtenir la solution optimale. L'un des animaux les plus organisés et les plus rigoureux dans leur travail est le loup. Les loups possèdent une très grande capacité de communication. Et grâce à leur intelligence, une méthode appelée méthode des loups a été

développer. Dans cette méthode, les loups artificielles représentent des agents qui en collaborant les un avec les autres, résolvent des problèmes complexes d'optimisation combinatoire.

4.7.2 Les loups dans la nature

— Dans le domaine animalier, le terme « meute » désigne un groupe de canidés, entre 2 et 15 individus, généralement proches parents.[**41**]

FIGURE 11: Une meute de loups

— Les loups vivent en meutes organisées selon une hiérarchie stricte dirigée par un couple de loups.
— Une véritable hiérarchie règne au sein de la meute : chaque membre a un rôle précis.[**41**]

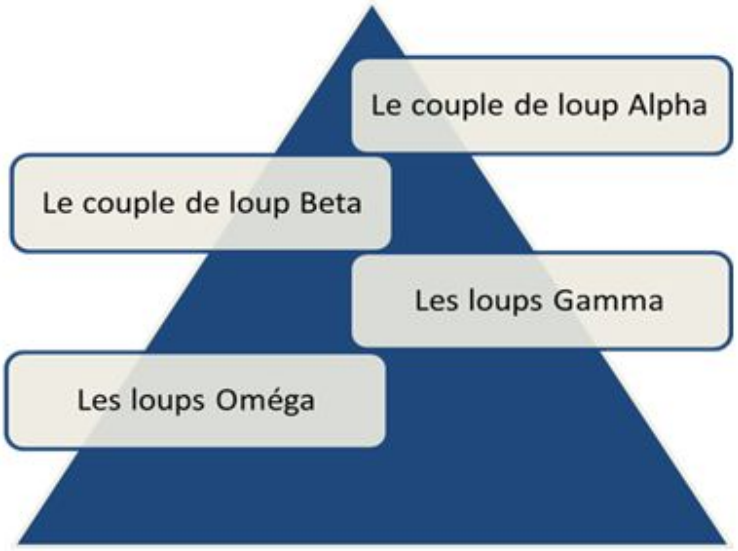


FIGURE 12: Hiérarchie de la meute

— En haut de la pyramide, nous avons les alpha.Ce sont les leaders. Ils fonctionnent par deux , un couple de loups dominant, mâle et femelle. Ils sont reconnaissables à leur queue levée quand les autres l'ont basse. [**41**]
— En dessous , nous avons les bêta.Ce sont les bras droits du couple dominant appelés à les remplacer en cas de décès ou si le couple alpha perd sa place de leader. Les bêta exécutent les ordres du couple leader et jouent également les gardes du corps pour eux.[**41**]
— Puis viennent les gamma le gros de la meute, ce sont les dominés. Ils suivent les instructions décidées par le couple alpha mais transmises par les bêta dont ils dépendent. Ils font vivre la meute au quotidien. [**41**]
— Enfin, la meute accueille aussi un oméga. La meute comprend des membres spécialistes, prédateurs, nourrices et, les plus malheureux d'entre tous, souffre-douleurs et boucs émissaires de la bande, Ils désamorcent la tension, et c'est sur eux que converge toute l'agressivité sociale de la famille... l'oméga se révèle indispensable pour apaiser le stress de ses congénères et éviter les risques de blessure et rétablir l'équilibre au sein de la meute.[**41**]

4.7.3 comment les loups chassent

1. La faim fait sortir le loup de sa tanière, ce qui peut être effectivement vrai. L'acte de chasse est inné chez le loup, comme chez tous les prédateurs carnivores.
2. Les loups ont la faculté de mémoriser les endroits où ils ont déjà chassé. Ils repèrent leur proie à l'odeur et aussi à la vue. Ils peuvent ainsi parcourir de longues distances pour chasser. [**42**]

FIGURE 13: Comment les loups chassent(1,2)

3. Les loups localisent leur proie et ne la quitte pas des yeux.
4. approchant le plus près possible sans que cette dernière ne les sente.
5. Il arrive aux loups de marquer de longs arrêt sans pour autant quitter leur victime des yeux.[**43**]

FIGURE 14: Comment les loups chassent(3,4,5)

6. La meute attendra le moment idéal pour se jeter sur sa proie. Durant l'attaque les loups harcèlent leur proie, jusqu'à épuisement de cette dernière.[**42**]

FIGURE 15: Comment les loups chassent(6)

4.7.4 Le modèle informatique

On a utilisé le principe de la chasse pour la modélisation , L'explication dans le tableau suivant.

Modèle Biologique	**Modèle Informatique**
La faim fait sortir le loup de sa tanière	Le déclanchement du processus de classification
Les loups ont la faculté de mémoriser les endroits où ils ont déjà chassés.	Les instances observées sont sauvegardé
les loups visent la proie (la peur de la proie (forte ou faible))	calculer la probabilité des deux classe intruse et non intruse
Ils peuvent ainsi parcourir de longues distances pour chasser ,les loups localisent leur proie et ne la quitte pas des yeux, approchant le plus près possible sans que cette dernière ne les sente	Les loups sont les classifieurs à utiliser (3 classifieurs differents : distance euclidiene,distance manhatan , disatnce minkowski)
les loups attaquent ensemble	Les classes intruses sont les proies attaqués, et les classe non intruse sont les proies non attaqués
La chasse est terminé	Les instances traitées sont sauvegardées.

TABLE 2: passage du modèle biologique vers le modèle informatique

4.7.5 Domaine d'application

Nombreux sont les domaines d'application des algorithmes des loups sauvages, citons quelques unes :

— Classification de données.
— L'optimisation de fonction.
— Clustering de données.
— L'ordonnancement de tâches.

4.8 Conclusion

Donc la recherche informatique a donné naissance des méthodes heuristique qui donnent des résultats approchés dans un temps raisonnable. Il existe plusieurs types d'heuristiques et la recherche en mouvement vers les heuristiques inspirées de la biologie qui ont donné des meilleurs résultats dans le domaine de l'intelligence artificielle.
Nous avons présenté une approche inspirée des loups sauvages pour résoudre un problème de securité d'informatique en l'occurrence la détection d'intrusion .Cette méthode est inspirée de la nature.

CHAPITRE 5 :

Implémentation et résultats

5 Implémentation et résultats

5.1 Knowledge Disovry and Data Mining (KDD Cup 1999 Data)

KDD Cup 1999 Data :

• Il s'agit de l'ensemble de données utilisé pour le troisième International Knowledge Discovery and Data Mining Tools compétition, qui s'est tenue en conjonction avec KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining.

• La tâche de la concurrence a été de construire un détecteur d'intrusion de réseau, un modèle prédictif permettant de distinguer entre les connexions " mauvaises ", appelé les intrusions (ou les attaques) et les connexions normales " bonnes ".

• Cette base de données contient un ensemble standard de données a vérifiés, qui comprend une grande variété d'intrusions simulées dans un environnement de réseau militaire.[**44**]

5.1.1 Le Contenue de KDD Cup 1999 Data

- Logiciel pour détecter les intrusions réseau protège un réseau informatique des utilisateurs non autorisés, y compris peut-être des initiés. La tâche d'apprentissage Détecteur intrusion est de construire un modèle prédictif (c.-à-d. un classificateur) capable de faire la distinction entre les connexions " mauvaises ", appelé les intrusions ou les attaques et les connexions normales " bonnes

- Le programme d'évaluation détection Intrusion 1998 DARPA a été préparé et géré par MIT Lincoln Labos. L'objectif était d'étudier et d'évaluer les recherches en détection d'intrusion. Un ensemble standard de données à vérifiés, qui comprennent une grande variété d'intrusions simulées dans un environnement de réseau militaire, a été fourni. Le concours de détection intrusion 1999 KDD utilise une version de ce jeu de données.[**44**]

- Lincoln Labos mettre en place un environnement d'acquérir neuf semaines de données vidage TCP brut pour un réseau local (LAN), simulant un LAN de Force aérienne américaine typique. Ils opéraient le LAN comme s'il s'agissait d'un véritable environnement de Force aérienne, mais il parsemé de multiples attaques.

- Les données brutes de formation a été environ quatre giga octets de données binaires compressées de vidage TCP de sept semaines de trafic réseau. Cela a été transformé en enregistrements de connexion environ 5 millions. De même, les deux semaines de données de test a donné environ 2 millions entrées de connexion.

- Une connexion est comme une séquence de paquets TCP commençant et se terminant à certains puits de temp , entre lesquels les données transitent vers et à partir d'une adresse IP de source vers une adresse cible selon un protocole bien défini. Chaque connexion est étiquetée comme soit normal, soit comme une attaque, avec exactement un type d'attaque spécifique. Chaque enregistrement de connexion se compose d'environ 100 octets.[**44**]

- Les attaques se répartissent en quatre grandes catégories :[**44**]

1- **DOS :** déni de service, par exemple : syn flood .

2- **R2L :** accès non autorisé depuis une machine distante, deviner par exemple mot de passe .
3- **U2R :** accès non autorisé à des privilèges locaux super-utilisateur (root), par exemple : « débordement de tampon " les attaques diverses .
4- **Sondage :** surveillance et autres sonder, par exemple : port de numérisation.
- Il est important de noter que les données de test ne sont pas de la même distribution de probabilité que les données d'apprentissage, et il inclut des types spécifiques d'attaque pas dans les données d'apprentissage. Cela rend la tâche plus réaliste. Certains experts de l'intrusion estiment que plus de nouvelles attaques sont des variantes des attaques connues et la « signature » d'attaques connues peut être suffisante pour prendre des nouvelles variantes. Les ensembles de données contiennent un total de 24 types d'attaque formation, avec un 14 types supplémentaires dans les données de test uniquement.[**44**]
- Stolfo et coll. défini des fonctionnalités qui aident à distinguer les connexions normales des attaques. Il y a plusieurs catégories de fonctions dérivées.
- La " même hôte " caractéristiques examine seulement les connexions dans le passé deux secondes qui ont la même destination héberger sous la connexion en cours et calcule les statistiques relatives au comportement du protocole, maintenance, etc..
- Une liste complète de l'ensemble des fonctionnalités définies pour les enregistrements de connexion est donnée dans les trois tableaux ci-dessous. Le schéma de données de l'objet dataset du concours est disponible sous forme lisible par une machine.Et d'autre fonctionnalités voire [**44**]

nom de la fonction	**description**	**type**
duration	longueur (nombre de secondes) de la connexion	continu
protocol_type	type de protocole, par exemple :tcp, udp, etc..	discret
service	service réseau sur la destination par exemple :http, telnet, etc..	discret
src_bytes	nombre d'octets de données de source à destination	continu
dst_bytes	nombre d'octets de données de destination à la source	continu
flag	état normal ou erreur de la connexion	discret
land	1 si la connexion est depuis/vers le même hôte/port ; 0 sinon	discret
wrong_fragment	nombre de " mauvais " fragments	continu
urgent	nombre de paquets urgents	continu

TABLE 3: Caractéristiques de base des connexions TCP individuelles.

nom de la fonction	**description**	**type**
hot	nombre d'indicateurs " chauds "	continu
num_failed_logins	nombre de tentatives de connexion qui ont échoué	continu
logged_in	1 si correctement connecté ; 0 sinon	discret
num_compromised	number of "compromised" conditions	continu
root_shell	1 si le shell root est obtenue ; 0 sinon	discret
su_attempted	1 si la commande " su root" tenté ; 0 sinon	discret
num_root	nombre d'accès " route "	continu
num_file_creations	nombre d'opérations de création de fichier	continu
num_shells	nombre d'invites de shells	continu
num_access_files	nombre d'opérations sur les fichiers de contrôle d'accès	continu
num_outbound_cmds	nombre de commandes sortants dans une session ftp	continu
is_hot_login	1 si la connexion appartient à la liste " chaude " ; 0 sinon	discret
is_guest_login	1 si la connexion est une connexion " guest" ; 0 sinon	discret

TABLE 4: Fonctionnalités de contenu au sein d'une connexion suggérée par la connaissance du domaine.

nom de la fonction	**description**	**type**
count	nombre de connexions vers le même hôte que la connexion en cours dans les deux dernières secondes Remarque : Les fonctionnalités suivantes se refer aux connexions des méme hotes	continu
serror_rate	% de connexions qui contiennent des erreurs ” SYN”	continu
rerror_rate	% de connexions qui contiennent des erreurs ” REJ”	continu
same_srv_rate	% des connexions au même service	continu
diff_srv_rate	% de connexions à différents services	continu
srv_count	nombre de connexions pour le même service que la connexion en cours dans les deux dernières secondes Remarque : Les fonctionnalités suivantes se réfère aux connections du même service.	continu
srv_serror_rate	% de connexions qui contiennent des erreurs ” SYN”	continu
srv_rerror_rate	% de connexions qui contiennent des erreurs ” REJ”	continu
srv_diff_host_rate	% de connexions sur différents hôtes	continu

TABLE 5: Caractéristiques de circulation calculées à l'aide d'une fenêtre de temps de deux secondes.

- Donc la KDD 99 est un corpus benchmark pour la detection d'intrusion , caractérisé de 42 attributs de différents services pour la detection d'intrusion, et l'attribut numéro 42 est la classe de la connexion .
-les chercheurs ont fait plusieurs recherche pour optimiser le corpus KDD 99 et ont arriver a un corpus benchmark appeler NSL-KDD.

5.1.2 NSL-KDD

- NSL-KDD est un ensemble de données suggéré pour résoudre certains des problèmes inhérents de l'ensemble de données KDD 99 qui sont mentionnées dans . Bien que, encore, cette nouvelle version de l'ensemble de données KDD souffre de certains des problèmes discutés par McHugh [**45**]et peut ne pas être un parfait représentant des réseaux réels existants, en raison de l'absence de public ensemble de données sur le réseau IDS, nous pensons qu'il peut toujours être appliqué comme un ensemble de données de référence efficace pour aider les chercheurs à comparer les méthodes de détection d'intrusion différents. En outre, le nombre d'enregistrements dans le NSL-KDD former et tester les ensembles sont raisonnables. Cet avantage rend abordable pour exécuter les expériences sur l'ensemble complet sans avoir à choisir au hasard une petite partie. Par conséquent, les résultats de l'évaluation des travaux de recherche différents sera cohérentes et comparables.

-La tâche de la concurrence a été de construire un détecteur d'intrusion de réseau, un modèle prédictif permettant de distinguer entre les connexions " mauvaises ", appelé les intrusions (ou les attaques) et les connexions normales " bonnes ". Cette base de données contient un ensemble standard de données devant être vérifiés, qui comprend une grande variété d'intrusions simulées dans un environnement de réseau militaire.

5.1.2.1 Les avantages de la NSL-KDD par rapport à KDD 99

-L'ensemble de données NSL-KDD présente les avantages suivants sur l'ensemble de données KDD original :
-Il n'inclut pas les documents redondants dans la rame, classifieurs ne seront pas biaisés vers comptes rendus plus fréquents.
-Il n'y a aucun enregistrement en double dans les ensembles de test proposée ; par conséquent, la performance des apprenants ne sont pas faussées par les méthodes qui ont le meilleurs taux de détection sur les enregistrements fréquents.[**46**]
-Le nombre d'enregistrements sélectionnés dans chaque groupe de niveau de difficulté est inversement proportionnel au pourcentage d'enregistrements dans le jeu de données KDD original. Ainsi, les taux de classification des méthodes d'apprentissage machine distincte varient dans une gamme plus large, ce qui le rend plus efficace d'avoir une évaluation précise des techniques d'apprentissage différents.
-Le nombre d'enregistrements dans l'apprentissage et test est raisonnable, ce qui le rend abordable pour exécuter les expériences sur l'ensemble complet sans avoir choisir au hasard une petite partie. Par conséquent, les résultats de l'évaluation des travaux de recherche différents seront cohérents et comparables.[**46**]

5.1.2.2 Statistiques De La NSL-KDD

Une des plus importantes lacunes dans apprentissage ou corpus des données KDD de l'apprentissage est le grand nombre de disques redondants, ce qui provoque les algorithmes d'apprentissage prend beaucoup de temps. En outre, l'existence de ces enregistrements répétés dans le corpus de test entraînera les résultats de l'évaluation d'être biaisés par les méthodes qui ont le meilleurs taux de détection sur les enregistrements fréquents.[**46**]

	Enregistrements original	**Enregistrements distincts**	**Taux de réduction**
Attaques	3,925,650	262,178	93.32%
Normal	972,781	812,814	16.44%
Total	4,898,431	1,074,992	78.05%

TABLE 6: La valeur statistique d'enregistrements redondants dans le l'apprentissage "KDD".

	Enregistrements original	**Enregistrements distincts**	**Taux de réduction**
Attaques	250,436	29,378	88.26%
Normal	60,591	47,911	20.92%
Total	311,027	77,289	75.15%

TABLE 7: La valeur statistique d'enregistrements redondants dans le test "KDD".

En outre, ils ont analysé le niveau de difficulté des enregistrements de l'ensemble de données KDD. Étonnamment, environ 98 % des dossiers dans le corpus de l'apprentissage et 86 % des enregistrements dans le corpus test étaient correctement classés tous les 21 apprenants.

5.2 Le langage de programmation Java

Java est un langage de programmation et une plateforme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts [**47**]

Java est-il disponible gratuitement en téléchargement ?

Oui, vous pouvez télécharger Java gratuitement. Pour obtenir la dernière version, rendez-vous sur java.com.

Si vous construisez un dispositif imbriqué ou grand public et que vous souhaitez y intégrer Java, contactez Oracle pour plus d'informations sur la façon d'inclure Java dans votre dispositif.[**47**]

Pourquoi dois-je procéder à la mise à niveau vers la dernière version de Java ?

La dernière version de Java comprend d'importantes améliorations en matière de performances, de stabilité et de sécurité pour les applications Java exécutées sur votre ordinateur. L'installation de cette mise à jour gratuite garantit que les applications Java sont toujours exécutées de manière sécurisée et efficace.[**47**]

Lorsque vous téléchargez le logiciel Java, vous avez accès à l'environnement JRE (Java Runtime Environment). Cet environnement se compose de la Java Virtual Machine (JVM), des classes standard de la plate-forme Java et des bibliothèques

Java de prise en charge. L'environnement JRE correspond à la partie exécution du logiciel Java et permet d'exécuter ce dernier dans votre navigateur Web.[**47**]

5.2.1 L'editeur Netbeans de langage java

NetBeans est un projet open source ayant un succès et une base d'utilisateur très large, une communauté en croisance constante, et près 100 partenaires mondiaux et des centaines de milliers d'utilisateur à travers le monde. Sun Microsystems a fondé le projet open source NetBeans en Juin 2000 et continue d'être le sponsor principal du projet.[**48**]

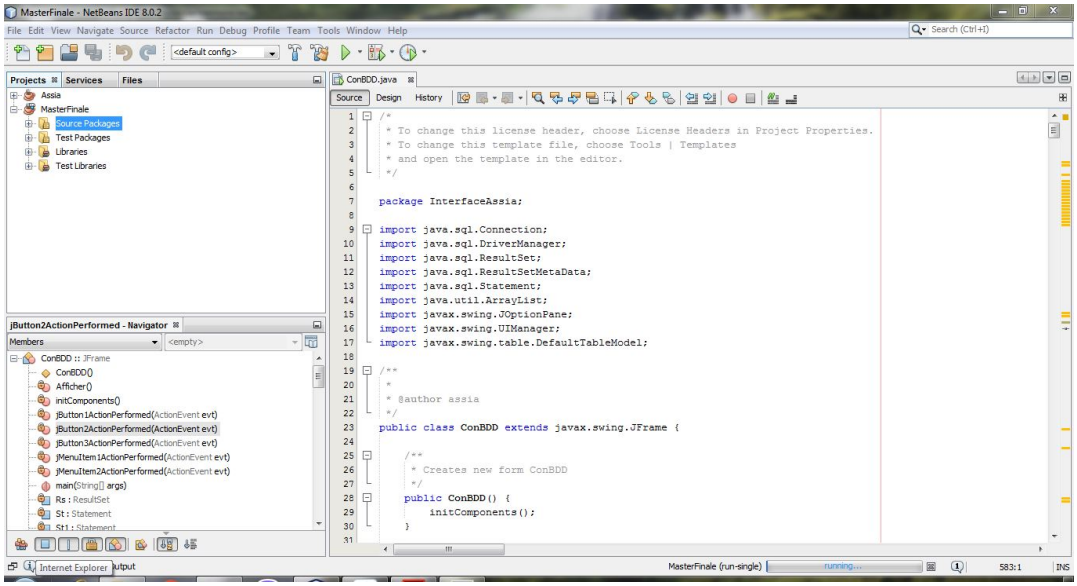


FIGURE 16: Interface netbeans.

Aujourd'hui, deux projets existent : L'EDI NetBeans et la Plateforme NetBeans.
L'EDI NetBeans est un environnement de développement - un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java - mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans. L'EDI NetBeans est un produit gratuit, sans aucune restriction quant à son usage.[**48**]
Également disponible, La Plateforme NetBeans ; une fondation modulable et extensible utilisée comme brique logicielle pour la création d'applications bureautiques. Les partenaires privilégiés fournissent des modules à valeurs rajoutées qui s'intègrent facilement à la Plateforme et peuvent être utilisés pour développer ses propres outils et solutions.[**48**]
Les deux produits sont open source et gratuits pour un usage commercial et non-commercial. Le code source est disponible pour réutilisation sous la Common Development and Distribution License (CDDL).[**48**]

5.3 Quelques distances Utilisées Dans notre Modèle

- Dans notre modèle on a utilisé quelques distances pour le calcul , parmi les distances utilisées on trouve :[**49**]

5.3.1 Distance euclidienne :

Dans un hyperespace, espace à n dimensions, la distance euclidienne entre les points X (x_1 ; $x_2 \ldots$; x_n) et Y (y_1 ; $y_2 \ldots$; y_n) est :

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

5.3.2 Distance manhattan

Dans un hyperespace, espace à n dimensions, la distance manhattan entre les points X (x_1 ; $x_2 \ldots$; x_n) et Y (y_1 ; $y_2 \ldots$; y_n) est :

$$\sum_{i=1}^{n} |x_i - y_i|$$

5.3.3 Distance minkowski

Dans un hyperespace, espace à n dimensions, la distance minkowski entre les points X (x_1 ; $x_2 \ldots$; x_n) et Y (y_1 ; $y_2 \ldots$; y_n) est :

$$\sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

le **p** utiliser dans notre cas est égale à 3 (***p=3***).

5.4 les mesures de performance de classifieurs

Nous considérons ici un problème simple de classification pour lequel nous nous intéressons à une classe unique C et nous voulons évaluer un système qui nous indique si une instance peut être associé ou non à cette classe C.
- Ce problème est un problème de classification à deux classes (C et non C noté ¬ C). Si on peut maitriser ce problème simple(bi-classe) , on peut aussi maitriser les mesures de validation de plusieurs classe ou bien multi-classe.

5.4.1 Matrice de contingence(matrice de confusion)

Pour évaluer un système de classification , nous utilisons un corpus étiqueté (corpus d'apprentissage et même pour le test) pour lequel on connaît la vraie catégorie de chaque connexion ou instance, et le résultat obtenu par le classifieur. Pour ce corpus, nous pouvons construire la matrice de contingence pour chaque classe (Voir table 8), qui fournit 4 informations essentielles :[**50**]
- **Vrai Positif (VP) :** Le nombre de connexions attribuées à une catégorie convenablement(documents attribués a leur vraie catégorie).

- dans notre cas connexions attribuées **normal** par le modèle et leurs vraies catégories **normal** dans le corpus.
- **Faux Positif (FP) :** Le nombre de documents attribués à une catégorie inconvenablement,(Documents attribués à des mauvaises catégories).
- dans notre cas connexions attribuées **anomaly** par le modèle et leurs vraies catégories **normal** dans le corpus.
- **Faux Négatif (FN) :** Le nombre de documents inconvenablement non attribués,(Qui auraient dû être attribués à une catégorie mais qui ne l'ont pas été).
- dans notre cas connexions attribuées **normal** par le modèle et leurs vraies catégories **anomaly** dans le corpus.
-**Vrai Négatif(VN) :** Le nombre de documents non attribués à une catégorie convenablement,(Qui n'ont pas à être attribués à une catégorie, et ne l'ont pas été).
- dans notre cas connexions attribuées **anomaly** par le modèle et leurs vraies catégories **anomaly** dans le corpus.

Catégorie C_i		**Jugement d'expert**	
		Normal	**Anomaly**
Jugement du classifieur	**Normal**	VP_i	FP_i
	Anomaly	FN_i	VN_i

TABLE 8: Matrice de confusion

5.4.2 Précision et Rappel

- Certains principes d'évaluation sont utilisés de manière courante dans les différents domaines . Les performances en terme de classification sont généralement mesurées à partir de deux indicateurs traditionnellement utilisés c'est les mesures de rappel et précision. Initialement elles ont été conçues pour les systèmes de recherche d'information, mais par la suite la communauté de classification de textes les a adoptées.
- Formellement, pour chaque classe C_i, on calcule deux probabilités qui peuvent être estimées à partir de la matrice de contingence correspondante, ainsi ces deux mesures peuvent être définies de la manière suivante :[**50**]
- **Le rappel** étant la proportion de documents correctement classés dans par le système par rapport à tous les documents de la classe C_i.

$$Rappel(C_i) = \frac{Nombre\_de\_documents\_bien\_classés\_dans\_C_i}{Nombre\_de\_documents\_de\_la\_classe\_C_i}$$

$$R_i = \frac{VP_i}{VP_i + FN_i}$$

- Le rappel mesure la capacité d'un système de classification à détecter les documents correctement classés. Cependant, un système de classification qui considérerait tous les documents comme pertinents obtiendrait un rappel de 100%. Un

rappel fort ou faible n'est pas suffisant pour évaluer les performances d'un système. Pour cela, on définit la **précision**.
- **La précision** est la proportion de documents correctement classés parmi ceux classés par le système dans C_i .[**50**]

$$Précision(C_i) = \frac{Nombre\_de\_documents\_bien\_classés\_dans\_C_i}{Nombre\_de\_documents\_classé\_dans\_C_i}$$

$$P_i = \frac{VP_i}{VP_i + FP_i}$$

- La précision mesure la capacité d'un système de classification à ne pas classer un document dans une classe, un document qui ne l'est pas. Comme elle peut aussi être interprétée par la probabilité conditionnelle qu'un document choisi aléatoirement dans la classe soit bien classé par le classifieur.
Ces deux indicateurs pris l'un indépendamment de l'autre ne permettent d'évaluer qu'une facette du système de classification : la qualité ou la quantité.

5.4.3 Bruit et silence

- On peut également définir les notions de Bruit (B) et de Silence (S) qui sont respectivement les notions complémentaires de la précision et du rappel.[**50**]
- On utilise aussi la notion de bruit qui présente le problème selon le point de vue opposé de la précision. Le bruit est le pourcentage de textes incorrectement associés à une classe par le système :[**50**]

$$Bruit(B) = 1 - Précision(P) = \frac{FP_i}{VP_i + FP_i}$$

- La notion de silence est le point de vue opposé du rappel. Le silence est le pourcentage de connexion à associer à une classe incorrectement non classés par le système :[**50**]

$$Silence(S) = 1 - Rappel(R) = \frac{FN_i}{VP_i + FN_i}$$

5.4.4 TP_rate et FP_rate

Les deux mesures TP-rate et FP-rate [**51**]

$$TP\_rate = \frac{VP_i}{VP_i + FP_i}$$
$$FP\_rate = \frac{FP_i}{VP_i + FP_i}$$

5.4.5 F-measure et entropie

- Observés conjointement, les indicateurs les plus célèbres à savoir le rappel et la précision, sont une estimation courante de la performance d'un système de classification.
- Cependant plusieurs mesures ont été développées afin de synthétiser cette double information. Nous ne retiendrons ici la mesure F_β décrite dans (Van Rijsbergen, 1979) . La F-mesure est la mesure de synthèse communément adoptée depuis les années 80 pour évaluer les algorithmes de classification de données textuelles à partir de la précision et du rappel.
- Elle est employée indifféremment pour la classification (Non supervisé) ou la catégorisation (Supervisé), pour la problématique de recherche d'information ou de classification. Elle permet donc, de combiner, selon un paramètre !, rappel et précision.
- On définit la mesure F_β comme la moyenne harmonique entre le rappel et la précision :[**50**]

$$F_\beta = \frac{(\beta^2 + 1) * précision * rappel}{\beta^2 * précision + rappel}$$

Pour utiliser cette mesure, il est donc nécessaire de fixer préalablement un seuil de décision pour le classement, et de calculer la valeur de F_β pour ce seuil. Le paramètre ! permet de choisir l'importance relative que l'on souhaite donner à chaque quantité. On choisit en général de donner la même importance aux deux critères, donc habituellement, la valeur de ! est fixée à 1 et la mesure est ainsi notée :[**50**]

$$F = \frac{2 * précision * rappel}{précision + rappel}$$

-**Entropie** : c'est la perte d'information , elle se calcule par la formule :

E= - précision*log(précision)

5.5 Implementation de l'algorithme

5.5.1 L'algorithme inspiré des loups sauvages

Comme on a vu dans le chapitre précédent les loups ont un fonctionnement biologique.Suite aux recherches faites dans ce domaine et lors des discussions avec mon encadreur de thèse , Nous avons atteint la decision de developper un nouveau modèle basé sur les loups sauvages qui est un algorithme de boosting avec trois classifieurs .
Les proies sont représentées par la base de test , l'idée du modèle est qu'il existe deux cas, proies non attaquées pour les connexions "normale"(non intrus) et l'autre proies attaquées pour les connexions "anomaly"(intrus), la peur de la proie forte ou faible (probabilité des deux classes intruse ou non intruse) et les distances sont représentées par les trois classifieurs(trois loups), la classification se fait par

le passage par les trois classifieurs. à l'avènement d'une nouvelle connexion du test ,d'une part chaque classifieur calcule la probabilité normale et la distance moyenne avec les connexions normales(non intruses) de la base d'apprentissage ,d'autre part même pour les connexions anomaly les classifieurs calculent la probailté anomaly et la distance moyenne de la base d'apprentissage si la sommation des (distances*probabilité) des classifieurs avec normale est inférieure de la sommation des (distances*probabilité) des classifieurs avec anomaly donc proies non attaquées "normale"(non intruse) classe "normale" et l'algorithme étiquette la connexion avec la classe "normale"(non intruse) si le cas contraire proies à attaquées "anomaly" et l'algorithme étiquette la connexion avec la classe "anomaly"(intruse).
-Pour le calcul des distances on a trois classifieurs le premier utilise la distance euclidienne dans le calcul , le calcul de la distance moyenne se fait par la somme de la distance de la connexion test avec toutes les connexions normale dans la base d'apprentissage divisée par le nombre des connexions normales dans la base d'apprentissage pour calculer **D1**, et méme le classifieur calcule la somme de la distance de la connexion test avec toutes les connexions anomaly dans la base d'apprentissage divisée par le nombre des connexions anomaly dans la base d'apprentissage pour calculer **D1'**.
- le deuxiéme classifieur fait le méme calcul sauf que la distance utilisée est la distance manhattan , on calcule **D2** pour la classe normale et **D2'** pour la classe anomaly .
- le troisième classifieur fait le méme calcul sauf que la distance utilisée est la distance minkowski , on calcule **D3** pour la classe normale et **D3'** pour la classe anomaly.

5.5.2 L'algorithme de boosting

5.5.2.1 Les données de l'algorithme :

— **La distance moyenne normale :**(c'est la somme de la distance de la connexion(instance) de test avec tous les connexion normale de la base d'apprentissage deviser par le nombre de connexion normale de la base d'apprentissage).

1. **D1** : distance moyenne normale et la distance utiliser dans ce classifier c'est la distance euclidienne.
2. **D2** : distance moyenne normale et la distance utiliser dans ce classifier c'est la distance Manhattan.
3. **D3** : distance moyenne normale et la distance utiliser dans ce classifier c'est la distance Mikowski.

— **La distance moyenne anomaly :**(c'est la somme de la distance de la connexion(instance) de test avec tous les connexion anomaly de la base d'apprentissage divisée par le nombre de connexion anomaly de la base d'apprentissage).

1. **D1'** : distance moyenne anomaly et la distance utilisée dans ce classifier c'est la distance euclidienne.

2. **D2'** : distance moyenne anomaly et la distance utilisée dans ce classifier c'est la distance Manhattan.
3. **D3'** : distance moyenne anomaly et la distance utilisée dans ce classifier c'est la distance Mikowski.

— **la probabilité normale :**(c'est le nombre des instances normale divisée par la somme de nombre des instances normale avec le nombre des instances anomaly.

$$P1 = \frac{N1}{N1 + N2}$$

— **la probabilité anomaly :**(c'est le nombre des instances anomaly divisée par la somme de nombre des instances normale avec le nombre des instances anomaly.

$$P2 = \frac{N2}{N1 + N2}$$

5.5.2.2 l'algorithme proposé :

algorithme loups ;
Debut
pour i=1 à n faire
debut
Si (Loup1+ Loup2+Loup3)<(Loup1'+ Loup2'+Loup3')
Alors $Class_i$***= "normal"***
Si non
$Class_i$***= "anomaly" ;***
Fin ;
Fin.

Avec :
— Loup1 =D_i1 * P1
— Loup2 =D_i2 * P1
— Loup3 =D_i3 * P1
— Loup1' =D_i1' * P2
— Loup2' =D_i2' * P2
— Loup3' =D_i3' * P2

Et :
— n : le nombre de connexion de base de test.
— Classi : la classe attribué a la connexion **"i"**

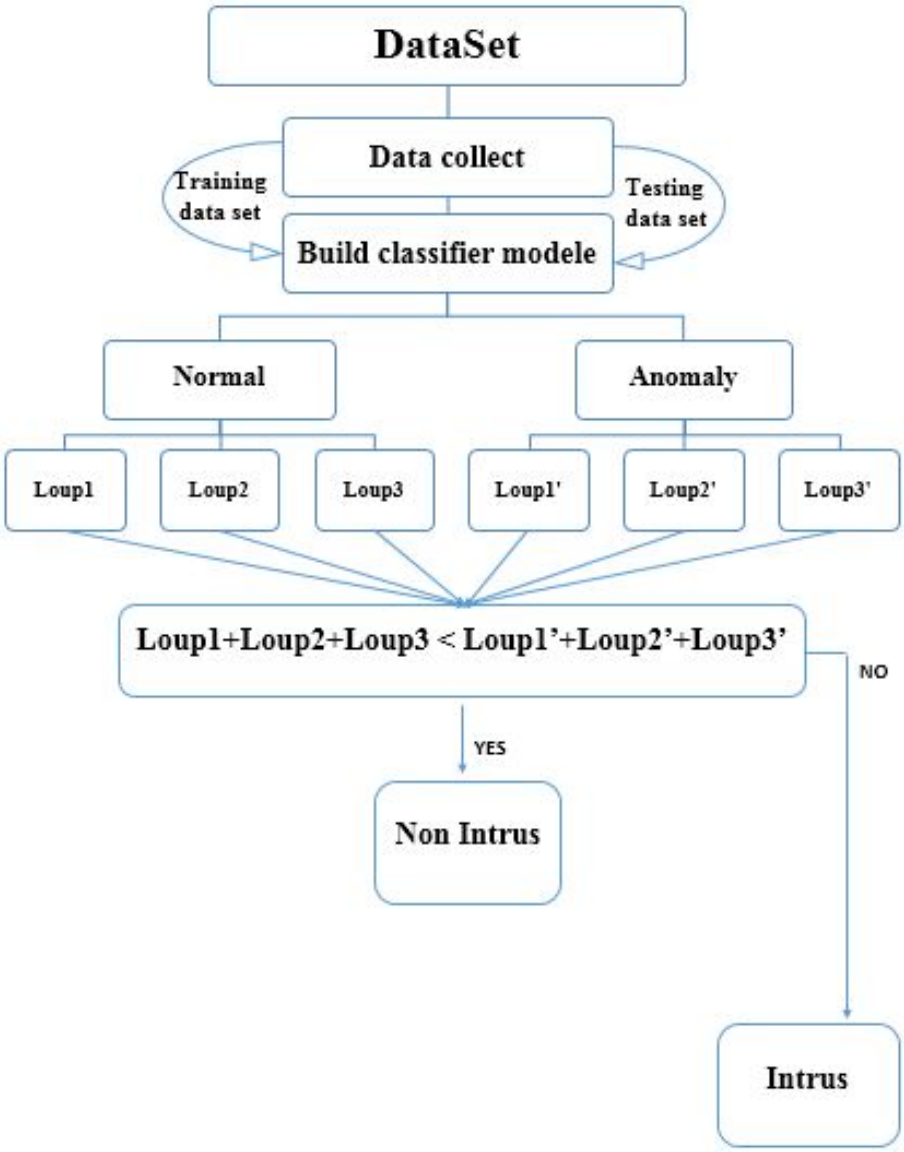


FIGURE 17: Illustration de notre systeme

5.5.3 Presentation de l'application

L'interface contient deux onglets (Traitement , Evaluation) :

— **Traitement :**

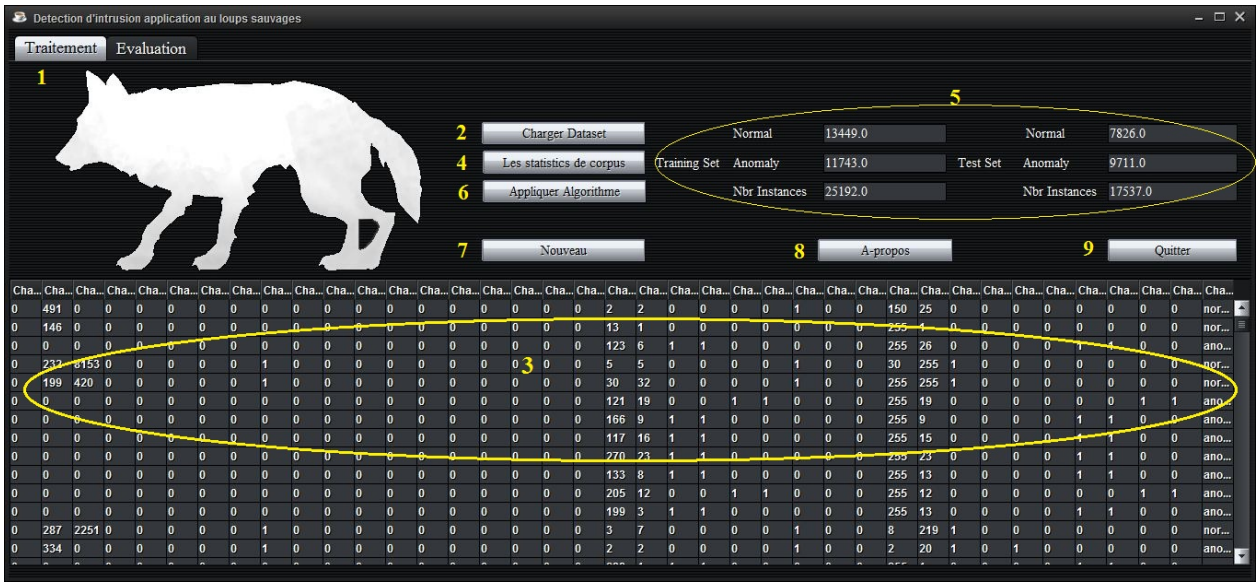


FIGURE 18: Capture d'ecran de l'interface (Onglet :Traitement)

1. Choix d'onglet **("Traitement")**.
2. Bouton**"Charger dataset"** pour charger le corpus.
3. L'affichage du corpus dans un tableau.
4. Bouton**"Les statistiques du Corpus"** pour afficher les statistiques de corpus.
5. Zone contient les statistiques du corpus.**(Normal,Anomaly et Nombres d'instances de (Training set et Test set))**
6. Bouton**"Appliquer algorithme"** pour lancer l'algorithme.
7. Bouton**"Nouveau"** pour un nouveau test.
8. Bouton**"A-propos"**pour afficher les information sur cette application.
9. Bouton**"Quitter"** pour quitter l'application.

— **Evaluation :**

FIGURE 19: Capture d'ecran de l'interface (Onglet :Evaluation)

1. Choix d'onglet **("Evaluation")**.
2. Zone pour afficher la matrice de confusion.
3. Zone pour afficher les resultats de l'algorithme.

5.5.4 Tests et resultats

- Dans notre travail on a pris la base d'apprentissage NSL KDD avec un pourcentage de 20% (25192 instances) , et on a pris la base de test NSL KDD complete (17537instances). Le tableau ci-dessous contient les résultats de notre modèle avec différents nombres d'attributs de la NSL-KDD .

1. **Les résultas de notre algorithme :**

attributs / mesures	**5**	**10**	**15**	**20**	**25**	**30**	**35**
Rappel	0.5646	0.5646	0.5598	0.5621	0.5603	0.6536	**0.6537**
Precision	0.9969	0.9969	0.9608	**0.9971**	0.9608	0.9955	0.9955
F-mesure	0.7209	0.7209	0.7074	0.7189	0.7078	0.7891	**0.7892**
Entropie	0.0030	0.0030	0.0399	**0.0028**	0.0399	0.0044	0.0044
Bruit	0.0030	0.0030	0.039	0.0028	0.0391	0.0044	0.0044
Silence	0.2790	0.2790	0.4401	0.4378	0.4396	0.3463	0.3462
TP_rate	0.5646	0.5646	0.5598	0.5621	0.5603	0.6536	0.6537
FP_rate	0.0765	0.0765	0.4367	0.0064	0.4298	0.0156	0.0156
Accuracy	0.5726	0.5726	0.5600	0.5683	0.5608	0.7054	0.7055
Matrice C	9681 30	9681 30	9331 380	9683 28	9331 380	9668 43	9668 43
	7464 362	7464 362	7336 490	7542 284	7322 504	5123 2703	5121 2705
T.d'exécution	46 m	92 m	138 m	184 m	230 m	276 m	322 m

TABLE 9: les résultats de la détection d'intrusion avec notre modèle

2. **Visualisation des résultats :**

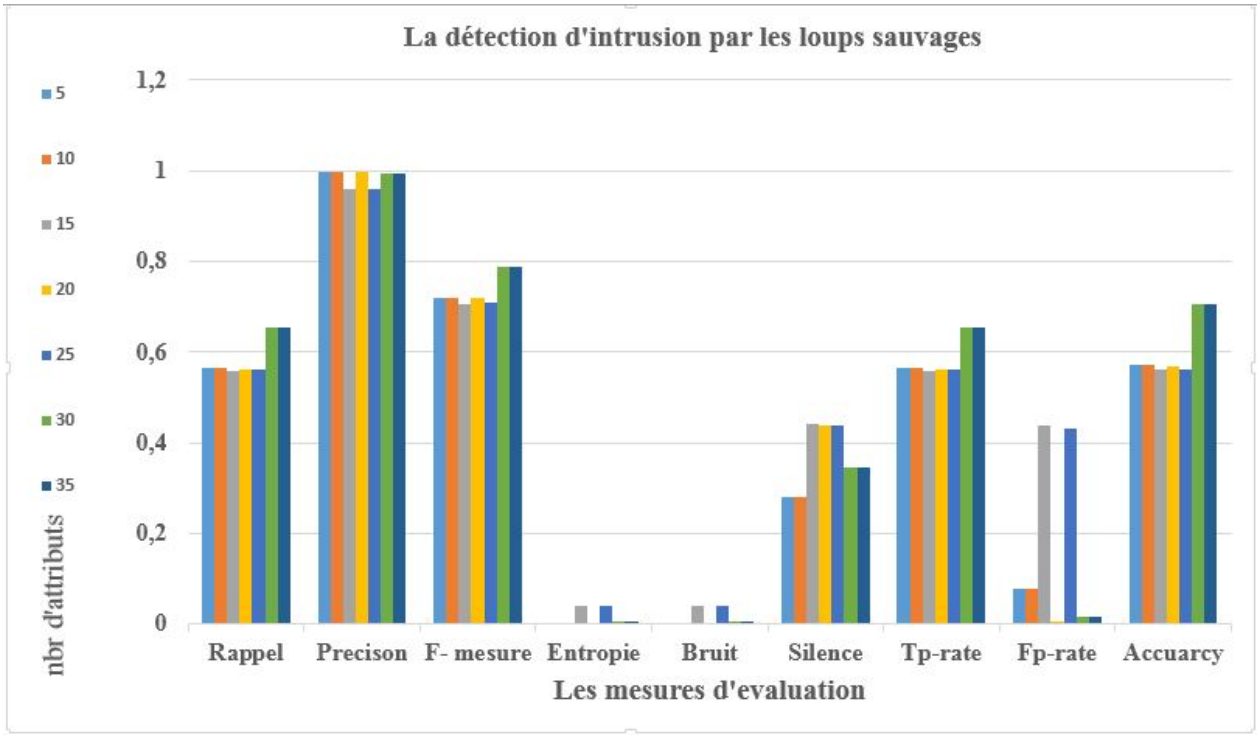


FIGURE 20: visualisation des résultats (histogramme)

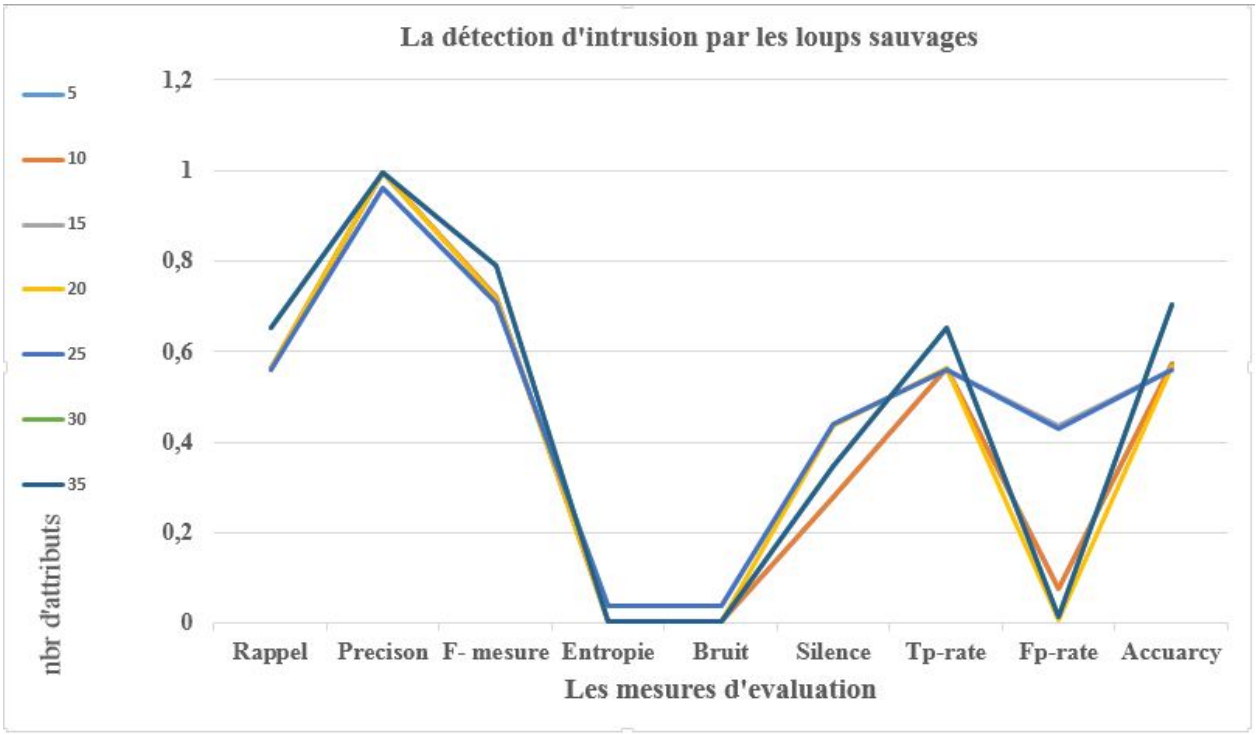


FIGURE 21: visualisation des resultats(courbe)

3. **La discussion des résultats :** a partir des figures (20,21) on conclue que :
   - Pour le rappel, de 5 à 25 attributs de la NSL KDD la valeur est stable a 0.56 à peu pret, et à 30 et 35 attributs la valeur augmente à 0.65.
   - Pour la précision avec 5 et 10 attributs la valeur est stable à 0.9969 ainsi qu'avec 30 et 35 attributs la valeur et de 0.9955 tandis que avec 15 et 25 attributs la valeur diminue à 0.9608 alors qu'avec 20 attributs elle augmente à 0.9971
   - pour la f-mesure de 5 à 25 attributs la valeur varie entre 0.70 et 0.72 on remarque qu'avec 30 et 35 attributs la valeur augmente à 0.7892 et donne de bons resultats.
   - pour l'entropie "perte d'information" avec 5 et 10 attributs la valeur est à 0.003 avec 15 et 25 attributs on a obtenu de mauvais resultats la valeur diminue avec 30 et 35 attributs par contre on obtient un bon resultat avec 20 attributs dans la valeur est de 0.0028.

4. **Comparaison avec d'autres algorithmes :** On a ajouter trois classifieur (SVM , Naive Bayes , Les Abeilles Sociales) pour une etude compratraive avec notre modele.

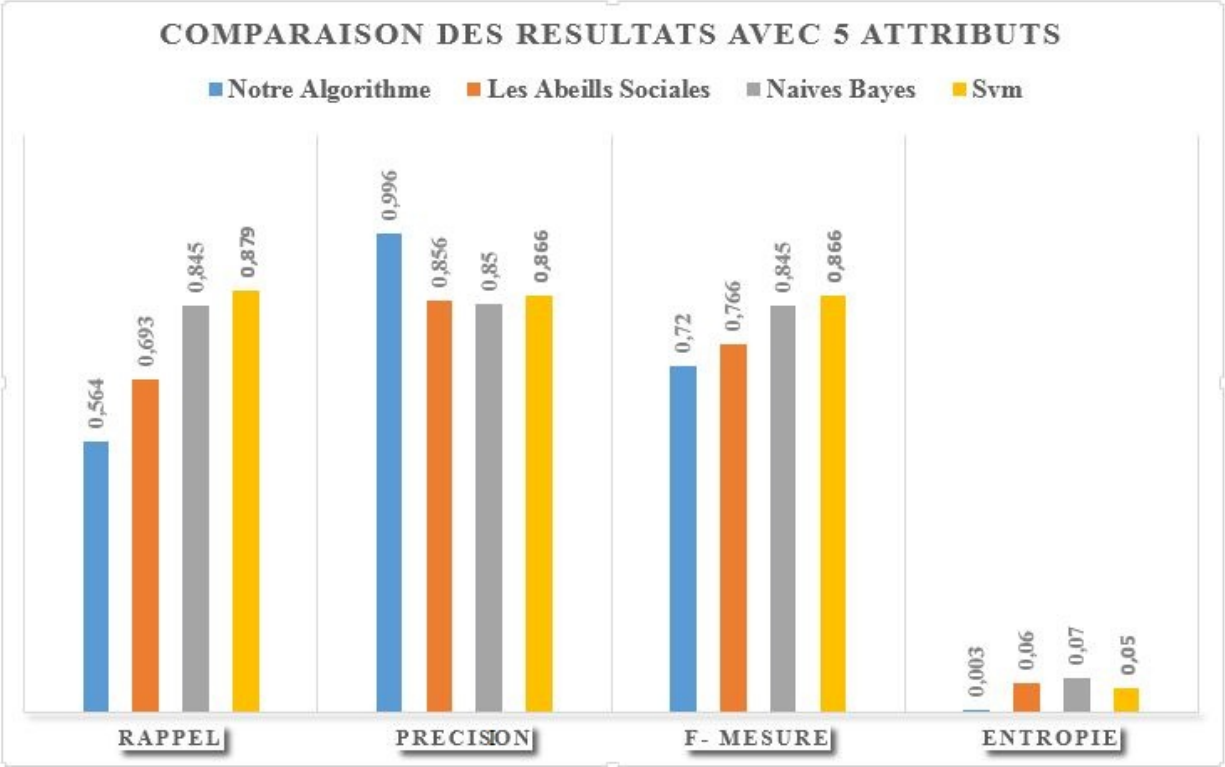


FIGURE 22: Comparaison des résultats(nbr attributs=5)

Comme on a vu dans la figure précédente(figure 22) , notre algorithme a donné dans le cas de 5 attributs de la Kdd une f-mesure inférieure aux trois autres algorithmes , mais elle est acceptable , aussi nous a donné une precision et une entropie meilleures , et pour le rappel notre algorithme a donné un résultat acceptable .

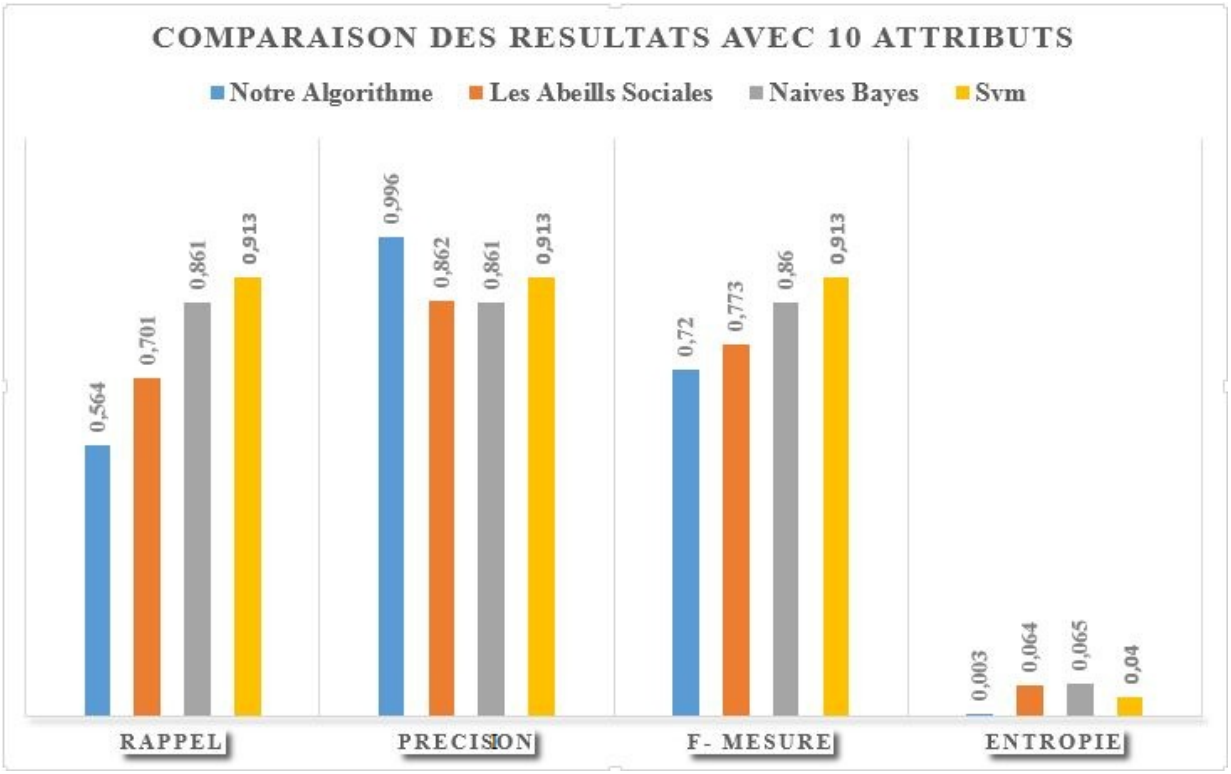


FIGURE 23: Comparaison des résultats(nbr attributs=10)

Comme on a vu dans la figure précédente(figure 23) , notre algorithme a donné dans le cas de 10 attributs de la Kdd une f-mesure inférieure aux trois autres algorithmes , mais elle est acceptable , aussi nous a donné une precision et une entropie meilleures , et pour le rappel notre algorithme a donné un résultat acceptable

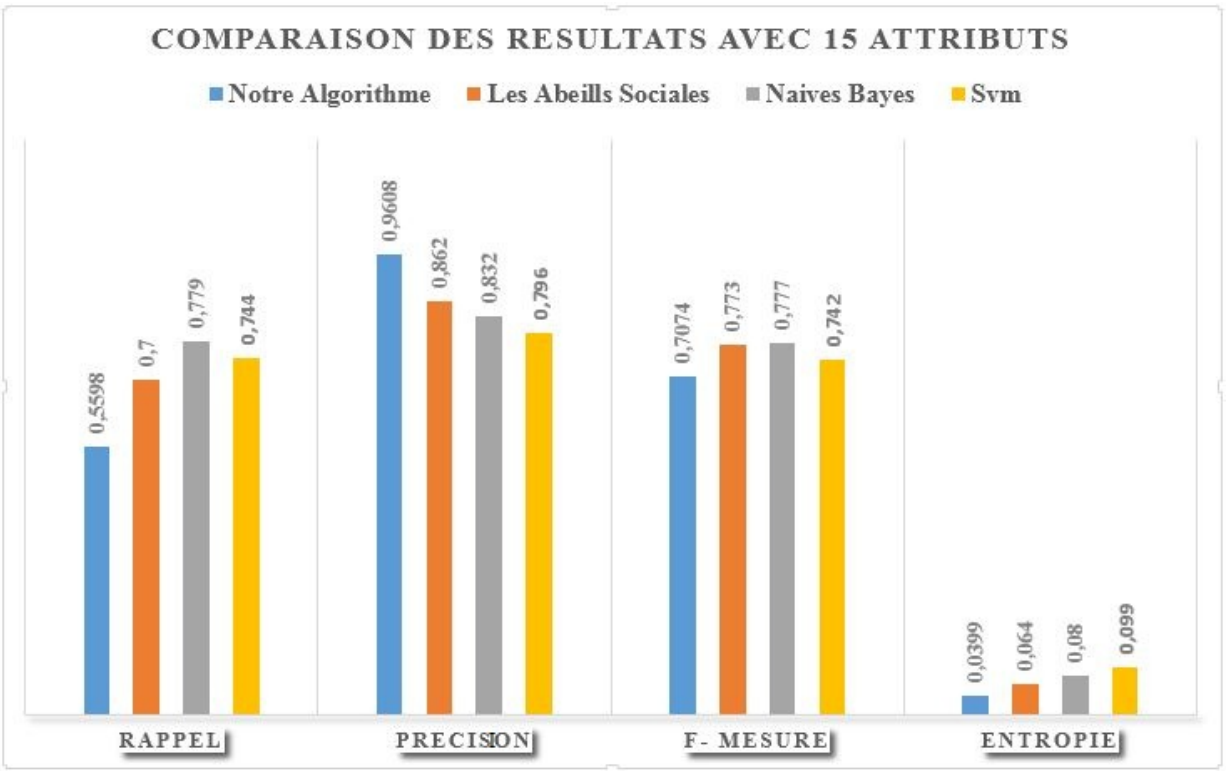


FIGURE 24: Comparaison des résultats(nbr attributs=15)

Comme on a vu dans la figure précédente(figure 24) , notre algorithme a donné dans le cas de 15 attributs de la Kdd une f-mesure inférieure aux trois autres algorithmes , mais elle est acceptable ,aussi nous a donné une precision et une entropie meilleures , et pour le rappel notre algorithme a donné un résultat acceptable .

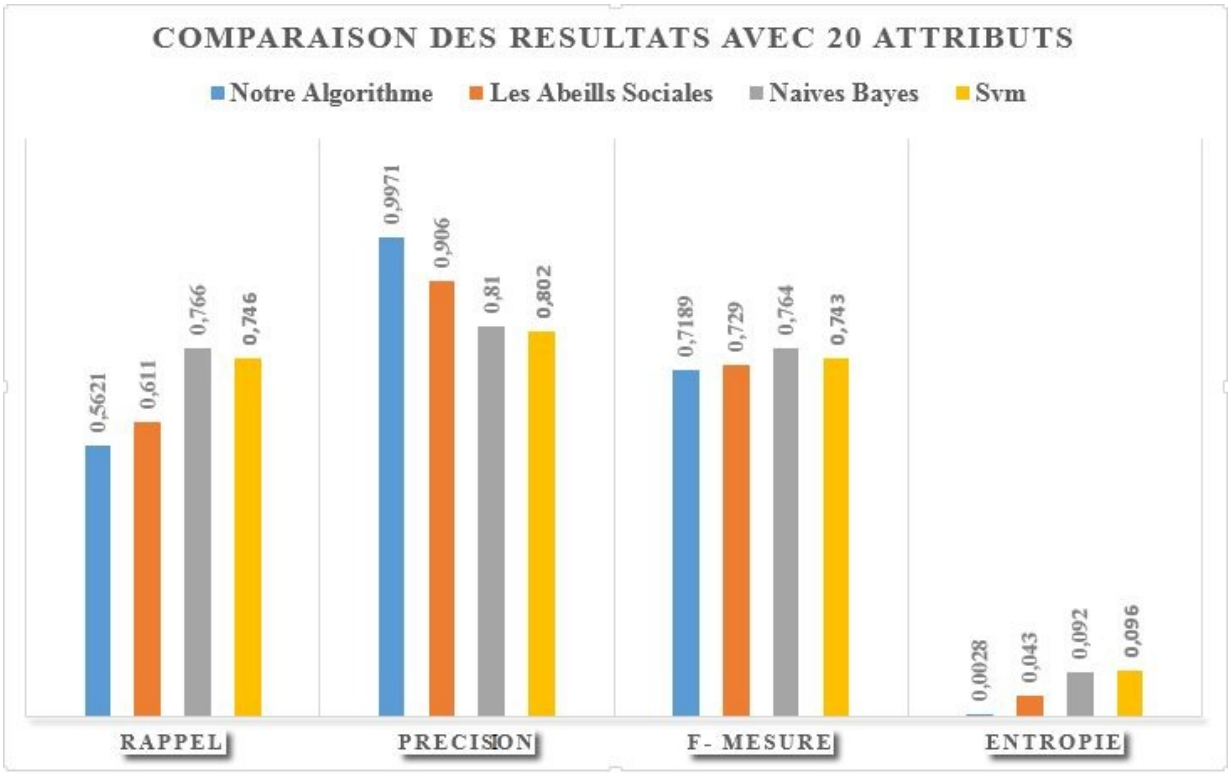


FIGURE 25: Comparaison des résultats(nbr attributs=20)

Comme on a vu dans la figure précédente(figure 25) , notre algorithme a donné dans le cas de 20 attributs de la Kdd une f-mesure inférieure aux trois autres algorithmes, mais elle est acceptable approximative de l'algorithme 'les abeilles sociales' , aussi nous a donné une precision et une entropie meilleures , et pour le rappel notre algorithme a donné un résultat acceptable .

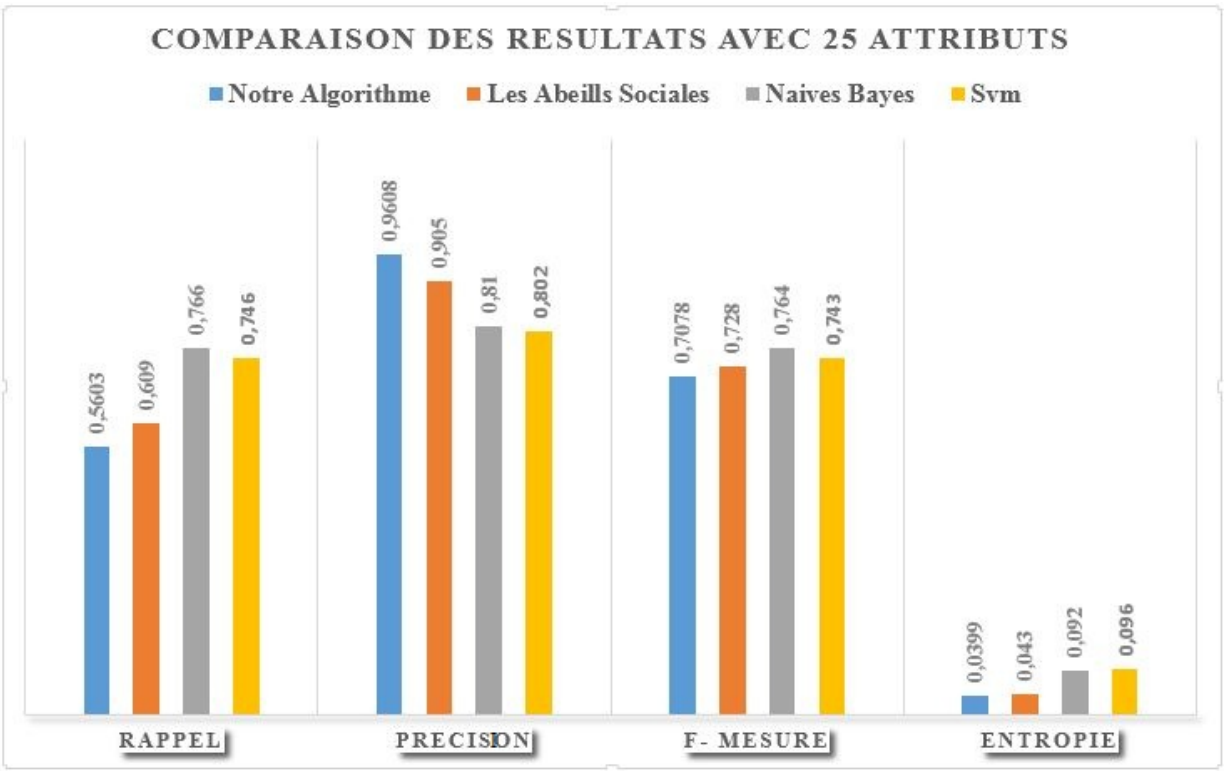


FIGURE 26: Comparaison des résultats(nbr attributs=25)

Comme on a vu dans la figure précédente(figure 26) , notre algorithme a donné dans le cas de 25 attributs de la Kdd une f-mesure inférieure aux trois autres algorithmes , mais elle est acceptable , aussi nous a donné une precision et une entropie meilleures , et pour le rappel notre algorithme a donné un résultat acceptable

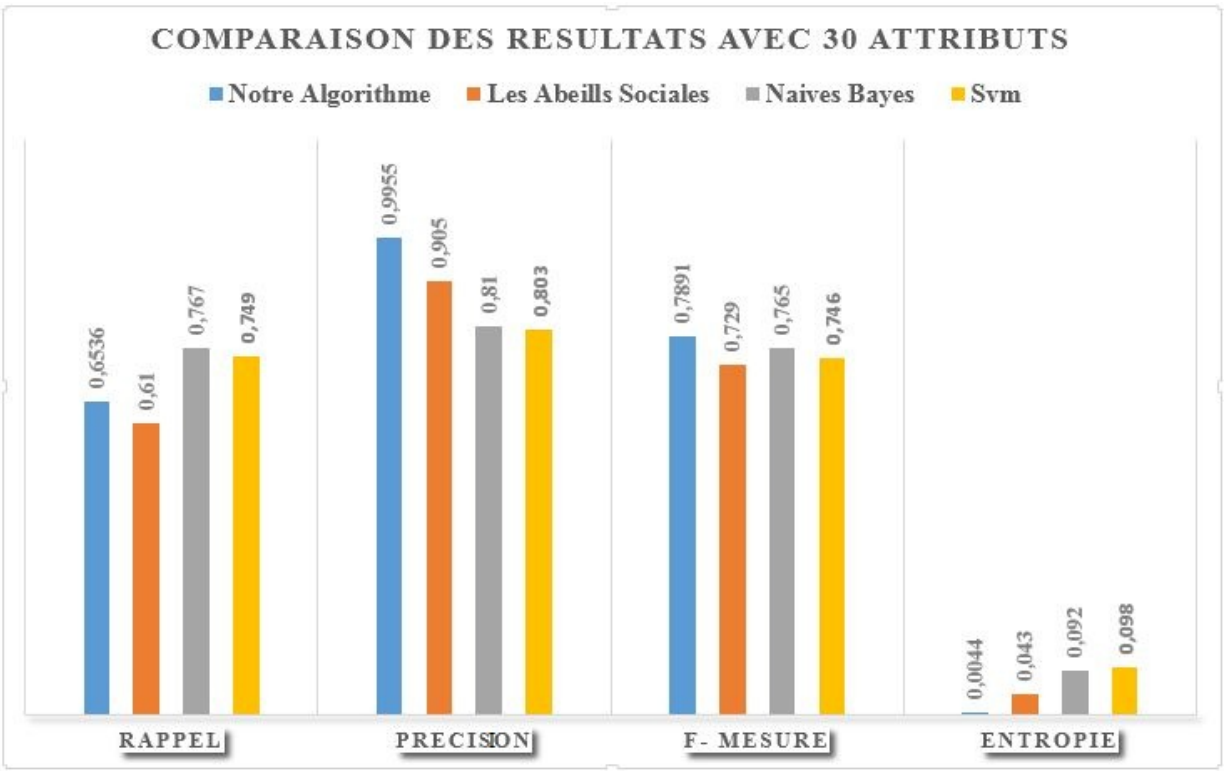


FIGURE 27: Comparaison des résultats(nbr attributs=30)

Comme on a vu dans la figure précédente(figure 27) , notre algorithme a donné dans le cas de 30 attributs de la Kdd une f-mesure, precision et entropie meilleures que les trois autres algorithmes , et pour le rappel notre algorithme a donné un résultat acceptable et bon.

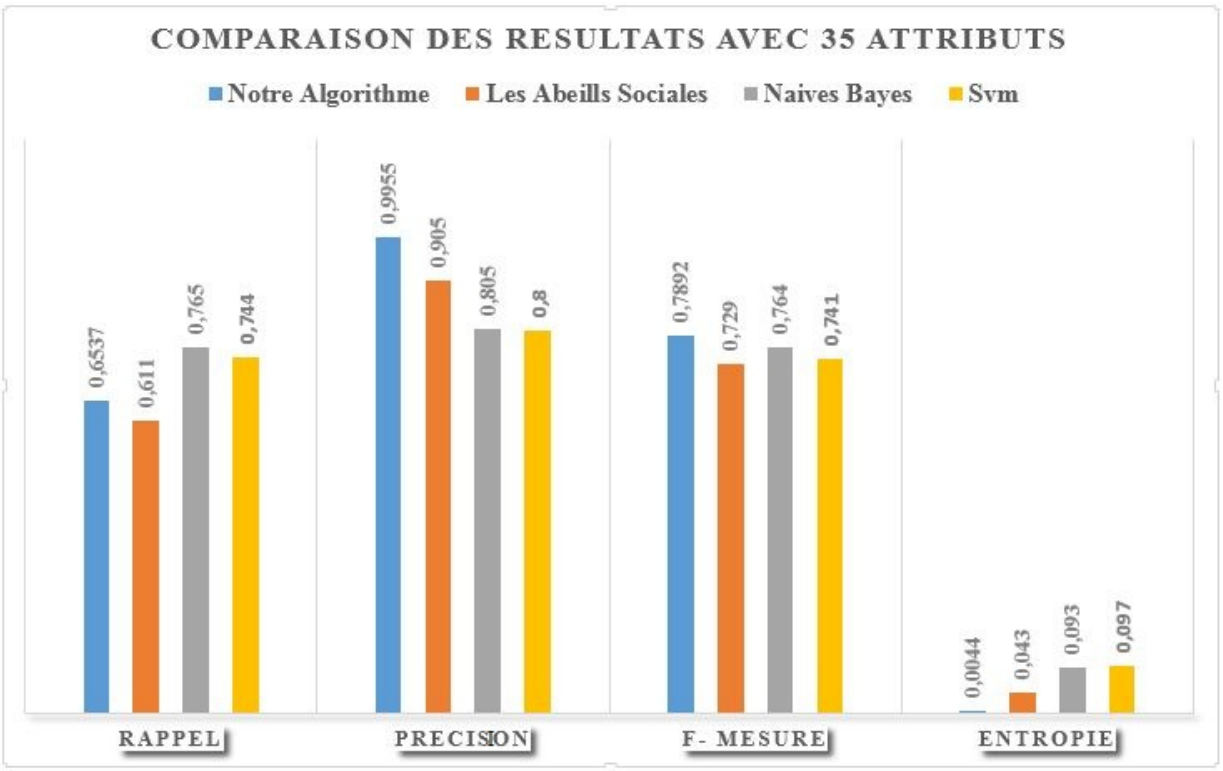


FIGURE 28: Comparaison des résultats(nbr attributs=35)

Comme on a vu dans la figure précédente(figure 28) , notre algorithme a donné dans le cas de 35 attributs de la Kdd une f-mesure, precision et entropie meilleures que les trois autres algorithmes , et pour le rappel notre algorithme a donné un résultat acceptable et bon.

5.6 Conclusion

Nous avons présenté dans ce chapitre une nouvelle solution pour une détection d'intrusions distribuée et intelligente sur la base des loups sauvages.
Nous avons proposé un nouveau modèle basé sur trois classifieurs inspirées des loups sauvages pour le filtrage des connexions à des connexions normales ou anomaly.
Nous avons analysé les resultats de cette approche sur le corpus (NSL KDD), l'addition des colonnes de la kdd donnent de bons résultats ,et les résultats s'optimisent surtout pour la precision et la perte d'information et un petit peu pour la f-mesure et le rappel et on conclue que le nombre d'attributs 30 et 35 sont l'idéal pour ce corpus .Ces valeurs peuvent changer pour d'autres corpus.

Conclusion Générale

6 Conclusion Générale

Dans ce travail ,nous nous sommes intéressés aux techniques bio-inspirées après des recherches dans le domaine et précisément on parle des loups sauvages ,on a prouvé que l'algorithme des loups sauvages peut détecter les intrusions , et filtrer les connexions "anomaly" des connexions "normale" avec une bonne façon.

L'objectif du travail est d'aborder un modèle ou un algorithme pour la détection d'intrusion ,et qui donne de bons résultats dans la sécurité informatique, et qui concurrence les autres algorithmes , suite aux des longues recherches et des bonnes orientations de mon encadreur, notre but pour cette thèse est presque réalisé avec un oeil futuriste dans le domaine bio-inspiré qui est riche en méthodes et techniques .

Notre projection est d'adapter notre modèle à d'autre domaines tel que les réseaux sociaux , le domaine médicale et même dans les domaines linguistiques tel que le traitement automatique de la langue par l'application de notre méthode a des données textuelles et voire pour le texte mining.

En perspectives d'autres recherches dans le domaine bio-inspiré et l'application d'autres méthodes et meta-heuristique seraient introduites dans la sécurité informatique ainsi que d'autres domaines , parce que la nature est vaste et les secrets de ces methodes bio-inspirées dans différents domaines ont montré leur efficacité .

Bibliographique

Bibliographique

— [**1**] http ://www.univ-bouira.dz/fr/images/uamob/fichiers/Cours/CoursHerzallah
— [**2**] http ://www.webreview.dz/IMG/pdf indexation_automatique_de_la_reherche.
— [**3**] https ://perso.limsi.fr/jacquemi/IRI-TR/dess-iri.
— [**4**] http ://dspace.univ-tlemcen.dz/bitstream/112/5832/1/Les-mesures-de-similarite-dans-un-systeme-de-recherche-Dinformation.
— [**5**] https ://fr.wikipedia.org/wiki/Modele_vectoriel.
— [**6**] http ://benhur.teluq.ca/SPIP/inf6104/spip.php ?article98
— [**7**] Philipe Biondi , « Architecture expérimentale pour la détection d'intrusions dans un système informatique » ,Avril-Septembre 2001.
— [**8**] Salima Hassas , « Systèmes Complexes à base de Multi-Agents Situés » ,Université Claude Bernard-Lyon 1,2003.
— [**9**] Nicolas Nobelis, Un modèle de Case-Based Reasoning pour la détection d'intrusion, Université nice SOPHIA ANTIPOLIS,Rapport de stage DEA RSD/ESSI3 SAR,2002.
— [**10**] Liran LERMAN, Les systèmes de détection d'intrusion basés sur du machine learning, Université LIBRE de BRUXELLES.
— [**11**] Ghenima BOURKACHE, Un IDS réparti basé sur une société d'agents intelligents, Université M'hamed BOUGARA de BOUMERDES,thèse magistère,2006.
— [**12**] Nicolas Baudoin et Marion Karle , « NT Réseaux : IDS et IPS » ,Ingénieurs 2000. 2003-2004.
— [**13**] H.Debar, M.Dacier et A.Wespi , « A revised taxonomy for intrusion detection systems » ,Annales des télécommunications. July-August 2000.
— [**14**] A.Phillip, Porras et Alfonso Valdes , « Live traffic analysis of tcp/ip getwayes » ,Proc. ISOC Symposium on Network and Distributed System Security (NDSS98). (San Diego, CA, March, 98), Internet Society
— [**15**]Ludovic Mé et Cédric Michel, « La détection d'intrusion : bref aperçu et derniers développements » ,Mars 1999.
— [**16**] J. Anderson, « Computer security threat monitoring and surveillance » , 1980.
— [**17**] Dorothy E. Denning, « An intrusion detection model » , IEEE Transactions on software engeneering, SE-13 :222–232, 1987.
— [**18**] Ludovic Mé et Véronique Alanou, « Détection d'intrusion dans un système informatique : Méthodes et outils ».
— [**19**] Jacob Zimmermann et Ludovic Mé, « Les systèmes de détection d'intrusions : principes algorithmiques ».
— [**20**] Klauss Muller, « IDS : Système de détection d'intrusion, partie I »,LinuxFocus article number 292.
— [**21**] Chunsheng Li, Qingfeng Song, et Chengqi Zhang, « MA-IDS Architecture for Distributed Intrusion Detection using Mobile Agents »,Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004).

— [**22**] Noria Foukia, « IDReAM- Intrusion Detection and Response executed with Agent Mobility- A distributed and Self- Organizing Approach », Thèse de doctorat de l'Université de Genève. 2003.
— [**23**] Farah Abdel Majid Barika, « Vers un IDS Intelligent à base d'Agents Mobiles », Mémoire de DEA de Université de Tunis, Institut Supérieur de Gestion. 2003.
— [**24**] LABED Ines, Proposition d'un système immunitaire artificiel pour la détection d'intrusions, Université MENTOURI de CONSTANTINE,thèse magistère,2006.
— [**25**] Mykerjee. B & Heberlein. L.T & Levitt .K.N, « Network Intrusion Detection »,IEEE Network, Vol 8, No 3, pp .26-41, 1994.
— [**26**] K. Price, « Intrusion Detection Pages »,Purdue University, 1998.
— [**27**] B. White & E. A. Fisch, & U. W. Pooch, « Cooperating Security Managers : A Peer- Based Intrusion Detection System »,IEEE Network Journal, pp. 20-23, January/February 1996.
— [**28**] S. Staniford-Chen & S. Cheung & R. Crawford & M. Dilger & J. Frank & J.Hoagland & S. Templeton & K. Levitt & S. Walnum & C. Wee, & R. Yip , « GrIDS-A Graph-Based Intrusion Detection System for Large Network »,s. Proc of the 19th National Information Systems Security Conference, 1996.
— [**29**] J. S. Balasubramaniyan & J. O. Garcia-Fernadez & D. Isacoff & E. H. Spafford & D. Zamboni, « An Architecture for Intrusion Detection using Autonomous Agents », Technical Report Coast-TR-98-05, Computer Sciences Department, Purdue University, 1998.
— [**30**] M. Crosbie & E. H. Spafford, « Active Defense of a Computer System using Autonomous Agents », Technical Report CSD-TR-95-008, Purdue University, 1995.
— **31** D. Denning, «An intrusion detection models », IEEE, transaction on software engineering ,13(2) : 222-232, 1987.
— [**32**] S. Voß, S. Martello, I.H. Osman and C. Roucairol (Eds), "Meta- Heuristics - Advances and Trends in Local Search Paradigms for Optimization". Kluwer Academic Publishers, Dordrecht, The Netherlands, (1999)
— [**33**] Abbas El Dor. Perfectionnement des algorithmes d'optimisation par essaim particulaire : applications en segmentation d'images et en electronique. Other. Universite Paris-Est, 2012.
— [**34**] Johann Dreo. Adaptation de la metaheuristique des colonies de fourmis pour l'optimisation dicile en variables continues. Application en genie biologique et medical.. Other. Universite Paris XII Val de Marne, 2003. French
— [**35**] John H. Holland. Adaptation and artificial systems. University of Michigan Press, 1975.
— [**36**] David. E. Goldberg. Genetic algorithms in search, optimization and machine learning. Addison- Wesley, 1989.
— [**37**] Agoston E. Eiben and Jim E. Smith. Introduction to Evolutionary Computing. Springer, 2003.
— [**38**] Charles Fleurent and Jacques A. Ferland. Object-oriented implementa-

tion of heuristic search methods for graph coloring, maximum clique, and satisfiability. DIMACS Series in Discrete Mathema- tics and Theoretical Computer Science, 26 :619–652, 1996.

— [**39**] Pedro Larranaga and Jose A. Lozano. Estimation of Distribution Algorithms. A new tool for Evolutionary Computation. Kluwer Academic Publishers, 2001.

— [**40**] Mark Zlochin, Mauro Birattari, Nicolas Meuleau, and M. Dorigo. Modelbased search for combinatorial optimization : A critical survey. Annals of Operations Research, 131 :373–395, 2004.

— [**41**] https ://pictapic.com/media/1726337734521789876_1021724760

— [**42**] http ://fandeloup.centerblog.net/rub-sa-majeste-LE-LOUP-35.html

— [**43**] http ://www.franceloups.fr/chasse.htm

— [**44**]http : //kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

— [**45**]J. McHugh, “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, vol. 3, no. 4, pp. 262–294, 2000.

— [**46**]http : //nsl.cs.unb.ca/NSL − KDD/

— [**47**]https : //www.java.com/fr/download/faq/whatisjava.xml

— [**48**]https : //netbeans.org/indexf r.html

— [**49**]" Measuring distances ",Applied multivariate statistics – Spring,Swiss Federal Institue of Technology Zurich,2012.

— [**50**]MATALLAH Hociine, Cllassiifiicatiion Automatiique de Textes Approche Oriientée Agent,Thèse de Magister de l’Université de Aboubekr Belkaid Telemcen, 2011.

— [**51**]Damien François, Binary classification performances measure,v1.0 - 2009.