الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية التكنولوجيا

قسم: الإعلام الآلي

Mémoire de Master

Spécialité : 2ème année master SIC

Thème

Intrusion Detection System based on Sppoted Hyenas Optimizer (SHO)

Présenté par :

LAHCENE REDOUANE

FAIZA CHAIMAA ATTAB

Dirigé par :

Dr. Mme. S.KOUIDRI

Promotion 2021 - 2022

سبحانك لا علم لنا إلا ما علمتنا

إنك أنت العليم الحكيم

سورة البقرة: الآية: 31

Dédicace

A la mémoire de mon père,
A ma tendre grande mère,
A ma mère a qui je dois tout mon bonheur
pour son soutien sans faille et sa confiance,
A mes chers frères et soeurs,
A tous ceux qui me sont chèrs.

Dédicace

Je dédie ce travail à :

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chers frères pour leur appui et leur encouragement,

A mes chères soeurs pour leurs encouragements permanents, et leur soutien moral,

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

J'attire un grand remercie à Mme **KOUIDRI Siham** de leur sacrifice avec nous durant la réalisation de cette thèse. Ainsi que mon binôme lahcene redouane
Merci d'être toujours là pour moi.

Résumé

Dans ce mémoire nous avons proposé un algorithme Spotted Hyena Optimizer (SHO)bio inspiré comme méthode de détection d'intrusion en réseau à partir de la base d'apprentissage NSL KDD. Un bref aperçu du système de détection d'intrusion, des algorithmes bio-inspirés associé est présenté. Les paramètres et le processus d'exécution de SHO sont discutés en détail.
Contrairement aux autres implémentations du même problème, cette implémentation donne de meilleurs résultats en terme de mesures d'évaluation. Les performances de SHO sont comparées à GA ,PSO , GWO ainsi que les algorithmes de machines learning.
Le résultat expérimental montre que notre approche est meilleure en termes de : (Accuracy, Sensitivity ,Specificity ,Precision, F1-Score et MCC).

Mots clés : métaheuristiques , algorithme de machine learning , SHO ,NSL-KDD , Sécurité Informatique.

Abstract

In this dissertation we have proposed a Spotted Hyena Optimizer (SHO)bio inspired algorithm as a network intrusion detection method from of the NSL KDD learning base. A brief overview of the intrusion detection system, associated bio-inspired algorithms is presented. The settings and running process of SHO are discussed in detail.
Unlike other implementations of the same problem, this implementation gives better results in terms of evaluation measures. SHO performance is compared to GA, PSO, GWO and machine learning algorithms.
The experimental result shows that our approach is better in terms of : (Accuracy, Sensitivity, Specificity, Precision, F1-Score and MCC).

Keywords : SHO ,NSL-KDD, machine learning algorithms.

ملخص

في هذه الأطروحة اقترحنا خوارزمية مُحسّن الضبع المرقط (SHO)
مستوحى من الحيوية كطريقة للكشف عن اختراق الشبكة من قاعدة تعلم NSL KDD.
يتم تقديم لمحة موجزة عن نظام كشف التسلل الخوارزميات الحيوية المرتبطة. تمت
مناقشة الإعدادات وعملية تشغيل SHO بالتفصيل.
على عكس التطبيقات الأخرى لنفس المشكلة ، فإن هذا التنفيذ يعطي نتائج أفضل من
حيث تدابير التقييم.تتم مقارنة أداء SHO بخوارزميات GA و PSO و GWO والتعلم
الآلي.تُظهر النتيجة التجريبية أن نهجنا أفضل من حيث:
(الدقة ، الحساسية ، الخصوصية ، الدقة ، F1-Score و MCC).

كلمة مفتاحية: التعلم الآلي,مُحسّن الضبع المرقط (SHO),قاعدة تعلم NSL KDD

Introduction générale

Au cours des dernières années, de nombreuses approches ont été proposées pour la protection des systèmes informatiques contre toute utilisation non autorisée.
Tel les approches peuvent impliquer un cryptage symétrique et asymétrique, inclure des systèmes supplémentaires tels que des pare-feu ainsi que protocoles de sécurité sophistiqués et complexes.

Comme la sécurité les mécanismes ont tendance à évoluer avec le temps, les méthodes aussi adoptée par les hackers. Parallèlement, de nouveaux types de réseaux ont fait leur apparition comme le cellulaire réseaux, Mobile Ad-Hoc Networks (MANET) [69] et les réseaux de capteurs sans fil (WSN).
Quoi est plus, les futures implémentations de réseaux mobiles 4G [70] devraient fournir des services à un large nombre de technologies d'accès sans fil hétérogènes.
Néanmoins, chacun de ces réseaux s'est avéré porteur ses propres inefficacités et vulnérabilités en matière de sécurité. Comme les approches traditionnelles ne parviennent pas à contre-attaquer complètement l'intrusion tente la nécessité d'un mécanisme supplémentaire comme la dernière ligne de défense est devenue une nécessité.
Ainsi, les systèmes de détection d'intrusion (IDS) se sont rapidement imposés comme l'un des composants les plus élémentaires de toute infrastructure de sécurité.
Un IDS est un système de sécurité capable d'identifier le comportement malveillant (déjà terminé ou en cours) contre un réseau ou un ordinateur protégé.
Sans doute, la construction d'un modèle efficace de détection d'intrusion est une tâche difficile.
C'est parce qu'un IDS doit avoir un haut taux de détection d'attaque (DR), avec un faible taux de fausses alarmes (FAR) à au même temps.
Ce qui pourrait être encore plus difficile, c'est qu'un IDS ne doit pas être exigeant en ressources de calcul et être assez intelligent pour identifier des inconnus attaques. Depuis l'apparition du premier IDS [72], une pléthore de techniques a été proposée afin de stimuler leurs performances et leur efficacité. Ce n'est que jusqu'à récemment cependant, que les chercheurs se sont inspirés de la biologie et systèmes naturels [71]. Spotted Hyena Optimizer (SHO) comme une

des nombreuses familles de techniques bio-inspirées existantes, étudie et émule le comportement d'essaims d'animaux pour résoudre des problèmes complexes. Des tâches telles que l'organisation du nid, chercher des chemins vers des sources de nourriture, ou se déplacer d'un endroit à un autre en tant qu'unité organisée ont été analysés et modélisé.
Les IDS ont appliqué ces modèles pour l'exécution de certaines procédures critiques telles que la distinction entre comportement normal et anormal, retraçant la source d'une attaque et pour l'optimisation des performances. La motivation derrière c'est assez évident : ces systèmes naturels possèdent un ensemble de caractéristiques souhaitables qui peuvent être immédiatement héritées à l'IDS résultant. Par exemple, un essaim d'insectes est capable d'accomplir des tâches complexes bien qu'il soit basé sur un certain nombre d'entités simples avec des capacités très limitées. Aussi, il est capable de remplir des engagements difficiles même si son environnement change de manière drastique et fonctionnent efficacement même si un petit nombre de sa population s'éteint.
De même, les IDS basés sur les essaims sont généralement des systèmes légers mais simples à mettre en œuvre, auto configurables, hautement adaptatifs et extrêmement robustes.
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.
Objectif du travail :
l'objectif de notre travail est de proposer une adaptation du modèle bio-inspiré SHO sur la sécurité informatique dans le but de détecter les intrusions.
Dans ce travail nous avons utilisé un modèle bio-inspiré qui est inspiré de l'hyènes tachetées ,ce modèle est un nouveau modèle que nous avons proposé au cours de nos études sur l'hyènes tachetées et avec des discussion avec notre 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.

Organisation du mémoire :

Ce travail est composé de quatre chapitres. Après l'introduction générale on trouve le premier chapitre qui fait une présentation générale sur la Sécurité Informatique, après dans le deuxième chapitre on va voir les Stratégies de détection d'intrusion basée sur les méthodes bio-inspiré.

Dans le troisième chapitre on parle sur les Caractéristiques et la classification des métaheuristiques et aprés on parle sur la Description de notre approche proposée .

Dans le quatrième 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 propos bas sur SHO et on l'applique sur le corpus et on discutera sur les résultats expérimentaux.

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

Table des matières

Chapitre 1

Sécurité Informatique

1.1. Introduction

En raison de plusieurs facteurs notamment l'ouverture des systèmes d'information sur Internet, l'évolution de la technologie et des moyens de communication ainsi que la transmission de données à travers les réseaux, des risques d'accès et de manipulation des données par des personnes non autorisées d'une façon accidentelle ou bien intentionnelle sont apparus.
Donc la mise en place d'une politique de sécurité autour de ces systèmes est devenue une nécessité incontournable.
Le système de détection d'intrusion est l'une des techniques utilisées pour garantir un contrôle permanent des attaques ainsi que la détection de toute violation de cette politique, c'est à dire toute intrusion.
dans ce premier chapitre nous introduisons les principales notions de base de la sécurité informatique y compris sa définition, ses objectifs, les problèmes et les attaques informatique et aussi les mécanismes permettant d'améliorer la sécurité.
Ensuite, nous présentons les systèmes de détection d'intrusions, leur définition, architecture, classification...etc., et nous terminons par les limites des systèmes de détection d'intrusions actuels.

1.2. Sécurité informatique

1.2.1. Définition

La sécurité informatique est définie par la protection assurée aux systèmes informatiques ainsi qu'aux données stockées, transférées ou manipulées. Cette protection doit réaliser trois principaux objectifs : l'intégrité, la disponibilité et la confidentialité des ressources du système informatique hôte.[1]

1.2.2. Objectifs de la sécurité

Dans la littérature on trouve plusieurs définitions pour les objectifs de la sécurité, mais les standards (The Federal Information Processing Standards , 2004) citent trois principaux objectifs appelés la triade CIA :[1]

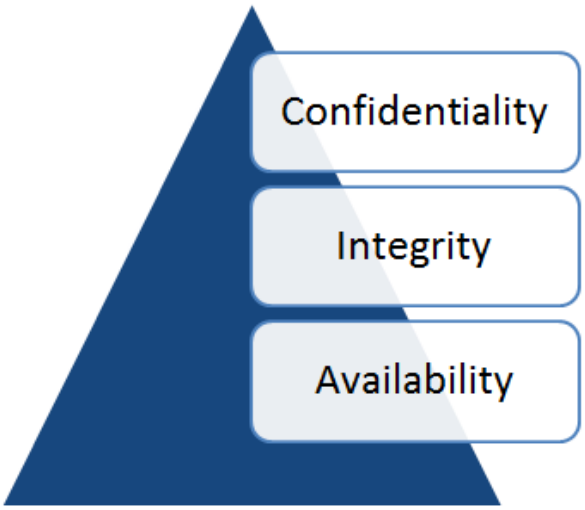

FIGURE 1.1 – *Organisation matérielle [1].*

- **La confidentialité : (confidentiality) :** Permet d'assurer que les informations sauvegardées ou transmises sur le réseau ne soient pas dévoilées à des personnes, entités ou processus non autorisés, c'est à dire seules les personnes autorisées doivent pouvoir accéder aux données ou informations ainsi protégées.
- **L'intégrité (integrity) :** Permet d'assurer que les données n'ont pas été altérées ou détruites de façon non autorisée, soit de manière accidentelle ou bien intentionnelle.
- **La disponibilité (availability) :** Cet objectif vise à assurer l'accès aux ressources du système d'information conformément aux spécifications en terme de performances. Ceci implique que le temps d'attente et le temps de service sont tous les deux relativement raisonnables.

1.2.3. SOUCIS DE LA SÉCURITÉ INFORMATIQUE

il existe trois problèmes qui affectent la sécurité informatique : les vulnérabilités, les menaces et les attaques : [2]

- **Les vulnérabilités :** Ce sont des failles ou des faiblesses dans la spécification, conception, implémentation ou bien configuration des systèmes

informatiques dont l'exploitation peut créer une intrusion.

- **Les menaces :** Une menace c'est la possibilité d'une violation d'une propriété de la sécurité en exploitant une ou plusieurs vulnérabilités d'une façon intentionnelle ou accidentelle.

- **Les attaques :** Une attaque c'est une action malveillante qui tente d'exploiter une faiblesse dans le système et de violer un ou plusieurs besoins de sécurité.

1.2.4. Classification des attaques informatiques :

Une attaque peuvent être classée selon son objectif, son point d'initiation ou la façon d'adresser la victime désirée. [3]

- **Selon l'objectif d'attaque :**
  On trouve deux types d'attaques principaux , passives et actives .
  - **Les attaques passives :**
    ce type d'attaque ne provoque pas d'altération aux ressources du système ciblé ce qui rend généralement indétectable (récupération du contenu d'un message ou bien l'observation du trafic).
  - **Les attaques actives :**
    consistent à effectuer des modifications ou bien une destruction des ressources d'un système d'une manière non autorisée. Ce type d'attaque est plus dangereux que le premier et peut causer des dégâts (usurpation de l'identité, modification,replay, déni de service...etc).

- **Selon le point d'initiation :**
  On distingue deux types d'attaques pour ce critère de classification : attaques de l'intérieur et attaques de l'extérieur.
  - **Les attaques de l'intérieur :** provenant des utilisateurs légitimes d'un système lorsqu'ils se comportent de façon non autorisée.

  - **Les attaques de l'extérieur :** venant de l'extérieur, souvent via Internet, en utilisant des techniques comme l'usurpation d'identité.

- **Selon la façon d'adresser la victime :**
  Il existe deux façons pour adresser la victime soit d'une manière directe ou bien indirecte.[4]

  - **Les attaques directes :**
    dans ce type d'attaque, l'intrus adresse ses paquets directement à la victime sans passer par un intermédiaire.

  - **Les attaques indirectes :**
    dans ce type d'attaque, l'adversaire envoie ses paquets vers une entité intermédiaire qui à son tour les retransmet vers la victime.

1.2.5. Buts des attaques :

Il existe plusieurs objectifs pour les attaques informatiques : [5]

- **Interruption :** vise la disponibilité des informations (DoS).

- **Interception :** Il existe plusieurs objectifs pour les attaques informatiques :

- **Modification :** vise l'intégrité des informations.

- **Fabrication :** vise l'authenticité des Informations.

Les quatre objectifs sont illustrés dans la figure suivante :

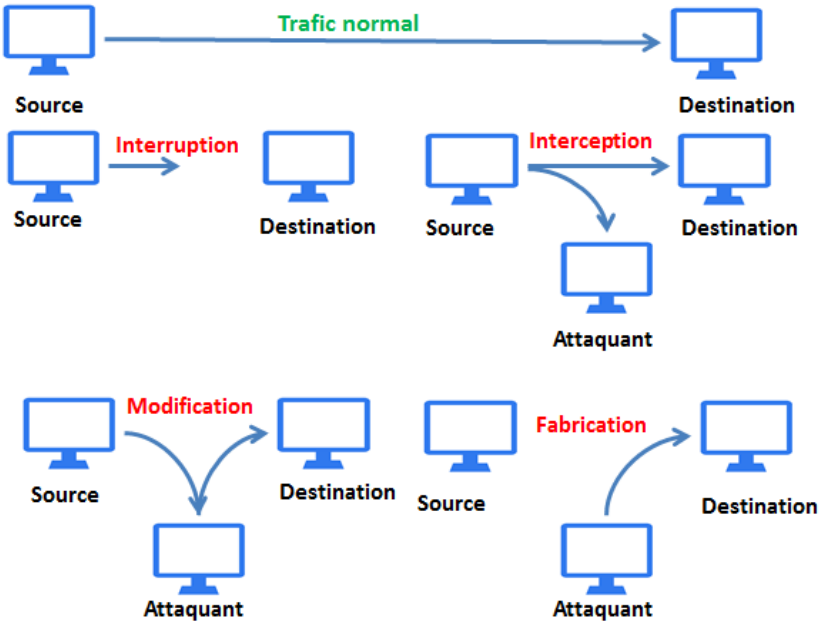

FIGURE 1.2 – *Les objectifs des attaques informatiques . [6]*

1.2.6. MOTIVATIONS DES ATTAQUES INFORMATIQUES :

Il existe plusieurs motivations d'un attaquant à vouloir exploiter une vulnérabilité et effectuer une attaque, parmi elles on peut citer les suivantes :[7]

- **motivations financières :** lorsqu'il s'agit de prendre possession des données pour rançonner l'entreprise ou le particulier.

- **motivations économique et concurrentielle :** on espionne ou on commet des actes de malveillances envers un concurrent dans le but d'acquérir un avantage commercial.

- **motivations politique ou idéologique :** comme semble l'indiquer la cyberattaque NotPetya ou le piratage de l'entreprise Ashley Madison [67].

1.2.7. Exemples d'attaques informatiques :

Il existe un nombre énorme d'attaques qui menacent les systèmes informatique à travers le monde entier, les plus connues aujourd'hui sont :

- **a) IP Spoofing :**
  Le principe de l'attaque IP Spoofing est relativement ancien (aux alentours de 1985) alors que sa première application dans une vraie attaque ne remonte qu'à 1995.
  Kevin Mitnick, un célèbre hacker, l'utilise afin de s'infiltrer dans le réseau d'un expert en sécurité informatique, Tsutomo SHimomura .
  Cette attaque consiste à usurper l'adresse IP d'une machine pour cacher la source d'attaque ou bien profiter d'une relation de confiance entre deux machines. Il existe des variantes car on peut spoofer aussi des adresses email, des serveurs DNS ...etc.[8]

- **b) Le dénis de service**
  Cette attaque consiste à envoyer des milliers des messages depuis des dizaines d'ordinateurs afin de saturer le système et donc le rendre indisponible. Ce type d'attaque est très facile à mettre en place mais très difficile à empêcher. [8]
  Un attaquant peut utiliser les DOS pour les raisons suivants :

  – obtenir le contrôle sur une machine cible ou sur un réseau. C'est le cas par exemple d'une attaque de type (SYN Flooding) qui est souvent utilisée de pair avec une tentative de spoofing.

  – masquer les traces en détruisant les stations qui auraient pu contenir des traces d'un attaquant.

  – se venger contre une personne, un administrateur ou bien encore une entreprise...etc.

  Il existe plusieurs types d'attaques DOS comme il est montré dans la figure suivante :

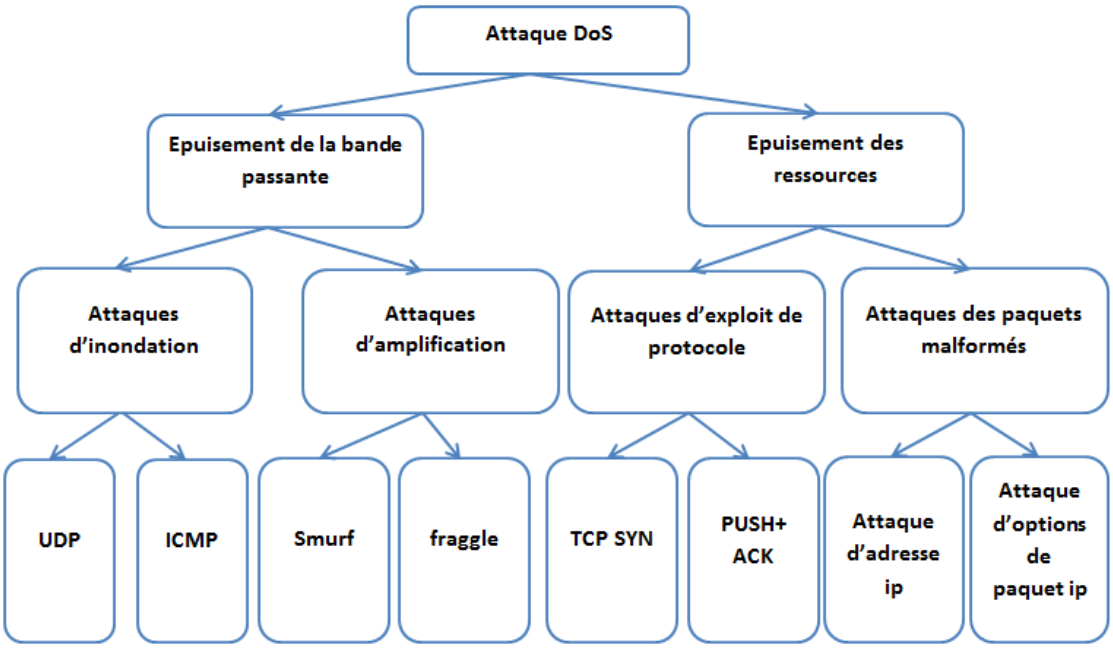

FIGURE 1.3 – *Taxonomie des attaques dos. [9]*

- **c) Probing (Sondage) :**

  Le sondage est une attaque dans laquelle le pirate analyse une machine ou un réseau pour déterminer les faiblesses ou les vulnérabilités qui pourraient être exploitées plus tard afin de compromettre le système. Cette technique est couramment utilisée dans l'exploration de données, par exemple saint, portsweep, mscan, nmap...etc.
  Cette classe d'attaque est la plus étendue et qu'elle requit une expertise technique minime,donc il est très important de protéger le système de telles intrusions car elles sont à la base d'autres attaques comme R2L (RemotetoLocal), U2R (User to Root)...etc. [10]

- **d) User to Root :**
  Ces attaques sont des exploitations dans lesquelles le pirate démarre sur le système avec un compte d'utilisateur normal et tente d'abuser des vulnérabilités du système afin d'obtenir des privilèges de super utilisateur.[11]
  En d'autre terme, l'objectif de cette attaque est d'obtenir les privilèges de l'administrateur système (Root) en allant d'un simple compte utilisateur (User) et cela en exploitant des failles dans le système comme le débordement de tampon, les erreurs de programmation..etc.

Il existe plusieurs attaques de ce type comme Eject,Ffbconfig, Fdformat, Load module, Perl,Xterm...etc.

- **e) Remote to Local :** C'est une attaque dans laquelle l'attaquant exploite les vulnérabilités d'une machine distante comme les bugs des applications, les mauvaises configuration des systèmes d'exploitation et d'autres afin d'obtenir un accès illégal à celle-ci en exploitant les privilèges d'un utilisateur local.
  Plusieurs attaques se trouvent dans cette catégorie parmi elles on cite : xlock, guest, xnsnoop, phf, Dict, warezmaster, spy, warezclient...etc. [6]

1.2.8. Techniques et mécanismes de sécurisation :

Pour réaliser les objectifs de sécurité cités dans la section 1.2.2, on doit prévoir un ensemble de mécanismes permettant de détecter toute attaque possible sur le système et même dans certains cas de prévenir ces attaques si cela est possible pour garantir un niveau élevé de protection du réseau et du système d'information.
Ces mécanismes peuvent être implémentés à différents niveaux de l'architecture réseau en couche.

a) Le chiffrement :

C'est un algorithme généralement basé sur des clefs pour transformer les données. Sa sécurité est dépendante du niveau de sécurité des clefs.
Autrement dit, le chiffrement consiste à utiliser des algorithmes permettant de coder les données en une forme non intelligible afin de les protéger contre toute divulgation non autorisée. Cette technique permet d'assurer la confidentialité des données. [6]

b) La signature numérique :

Cette technique consiste à calculer une valeur à l'aide d'un algorithme de chiffrement, cette valeur sera ajoutée à une donnée d'une façon que tout récepteur de cette donnée puisse vérifier son origine. La signature remplit deux fonctions juridiques principales :

- L'identification de l'auteur et la manifestation de son consentement. La signature numérique est le pendant électronique à la signature manuscrite,

mais la signature digitale est liée au document signé, elle n'est pas comparée à une signature témoin mais elle est vérifiée algorithmiquement alors elle est universellement vérifiable.

- Une signature numérique apporte la non répudiation à l'origine, c'est à dire l'auteur d'une action ne peut dénier l'avoir effectué. [12]

c) Le bourrage :

Données ajoutées pour assurer la confidentialité, notamment au niveau du volume du trafic. Les mécanismes de bourrage servent à modifier les caractéristiques du trafic pour assurer différents niveaux de protection contre l'analyse du celui-ci. [6]

d) Le contrôle d'accès :

Ce mécanisme consiste à vérifier les droits d'accès aux données en laissant passer que les personnes autorisées et cela pour empêcher toute exploitation de vulnérabilités venant de l'extérieur.

e) La notarisation :

Le mécanisme de notarisation consiste à reposer sur un tiers de confiance (notaire) qui détient les informations nécessaires pour assurer certains services de sécurité comme la non répudiation. [13]

f) Le pare-feu :

Un pare-feu, ou coupe-feu ou encore firewall. est un équipement ou des systèmes qui contrôlent le flux de trafic entre les différentes zones d'un réseau Donc, il assure un périmètre de protection entre le réseau interne à l'entreprise et le monde extérieur.
Voici une figure qui montre l'emplacement du pare-feu à la seine d'une entreprise à fin de protéger le réseau local et les serveurs sensibles de l'entreprise (DMZ).

Un parefeu peut assurer les tâches suivantes :

- bloquer l'accès à des services non autorisés.
- protéger en temps réel contre les menaces embarquées dans les applications.
- protéger contre les attaques de type DoS (Deni de service).

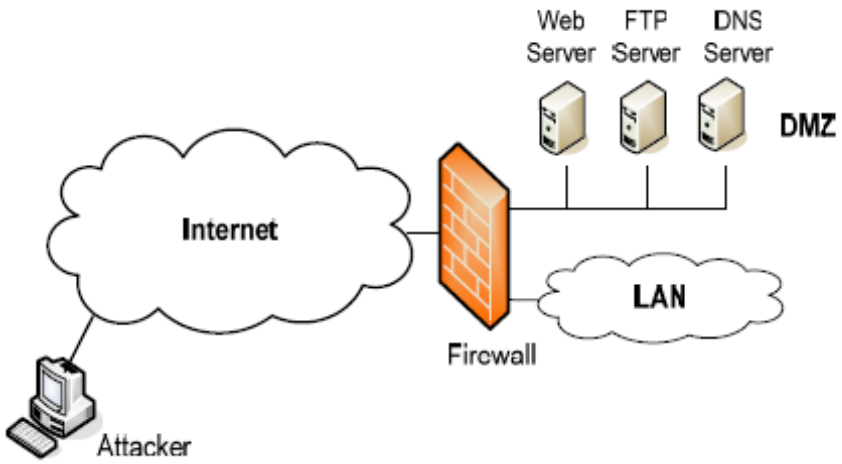

FIGURE 1.4 – *Déploiement tri hébergé d'un pare-feu de réseau d'entreprise[14].*

- intégrer des techniques de détection d'intrusions et envoyer des alertes afin de prévenir les équipes de surveillance technique.
- gérer les connexions sortantes à partir du réseau local.
- protéger le réseau interne des intrusions venant de l'extérieur.
- identifier et contrôler les applications partageant une même connexion.
- ...etc.

g) L'antivirus :

C'est un logiciel permettant de préserver le système de tout type de maliciels (virus, vers,chevaux de troie...etc). Son principe de fonctionnement peut suivre l'une des trois approches :

- comparer la signature virale du virus aux codes vérifiés.
- utiliser les métaheuristiques pour détecter les codes malveillants.
- utiliser le filtrage basé sur les règles.

h) La détection d'intrusions :

La détection des intrusions est un mécanisme de cybersécurité courant dont la tâche est de détecter les activités malveillantes dans des environnements hôte et / ou réseau.
La détection des activités malveillantes permettent de réagir en temps opportun, par exemple pour arrêter une attaque en cours. Vu l'importance de détection des intrusions, les milieux de la recherche et de l'industrie ont conçu et développé une variété de systèmes de détection d'intrusion (IDS).[15]

1.3. Systèmes de détection d'intrusions

1.3.1. Définitions

- **a) Intrusion :**

  C'est toute utilisation d'un système informatique à des fins autres que celles prévues. Autrement dit, c'est toute action malveillante qui vise l'un des objectifs de sécurité : La confidentialité, l'intégrité ou la disponibilité. [16]

- **b) Détection d'intrusions :**

  Consiste à analyser les informations collectées par les mécanismes d'audit de sécurité, à la recherche d'éventuelles attaques. [17]

- **c) Audit de sécurité :**

  C'est un examen méthodique d'une organisation ou d'un site visant à identifier ses risques, ses vulnérabilités et les faiblesses de ses protections existantes ainsi qu'à statuer sur son niveau de sécurité et à recommander des solutions aux problèmes identifiés. [17]

- **d) Système de détection d'intrusions**

  Le système de détection d'intrusions inclure tous les systèmes logiciels et matériels permettant d'automatiser les processus de surveillance et d'analyse des évènements au sein d'un système informatique afin de détecter toute activité pouvant conduire à une défaillance de sécurité.

Il peut être déployé sur une hôte, on parle alors de HostBased Intrusion Detection System (HIDS), ou bien sur un réseau, on parle alors de NetworkBased Intrusion Detection System(NIDS).

1.3.2. Architecture d'un IDS

Plusieurs architectures ont été proposées pour décrire les différents éléments constituant un système de détection d'intrusions. L'architecture la plus simple est composée de trois modules : le capteur, l'analyseur et le manager. Cette architecture est montrée dans la figure suivante :

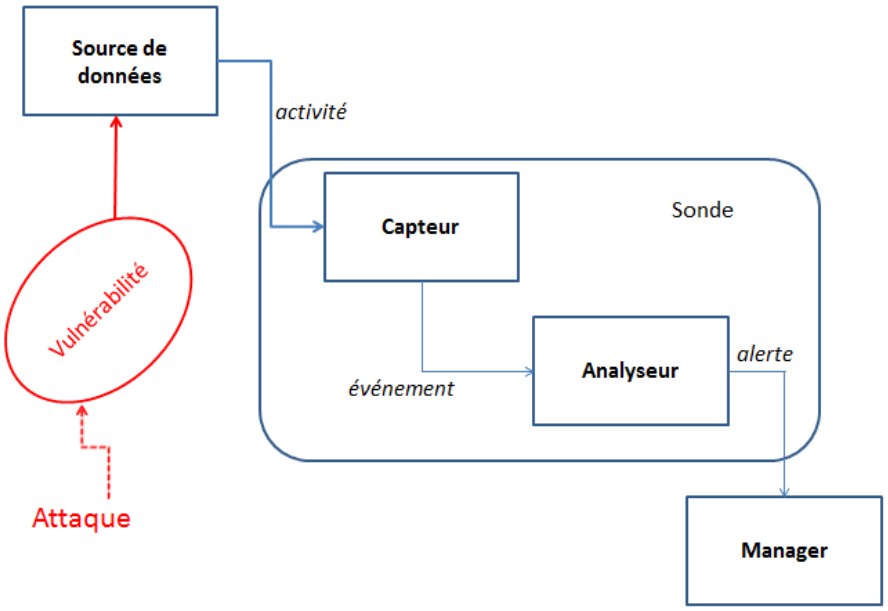

Figure 1.5 – *Déploiement tri hébergé d'un pare-feu de réseau d'entreprise[18] .*

a) Le capteur :
Chargé de collecter, filtrer et formater les informations brutes envoyées par la source de données concernant l'évolution de l'état du système. Le résultat de traitement est un message formaté appelé événement.
b) L'analyseur :
Permet d'analyser les événements générés par le capteur en détectant toute activité malveillante qui peut se produire à partir d'un sous ensemble de ces événements, et donc envoyer une alerte qui sera stockée dans les journaux du système ou bien utilisée pour lutter

contre les attaques selon le type d'IDS.

c) Le manager :

Permet de collecter et notifier les alertes envoyées par l'analyseur. éventuellement, le manager est chargé de la réaction à adopter qui peut être :

- Isolement de l'attaque pour réduire les dégâts.
- Suppression d'attaque.
- Restauration du système dans un état sain.
- Identification du problème qui a engendré cette attaque.

1.3.3. Principe de fonctionnement d'un IDS

Le fonctionnement d'un IDS et le processus de détection d'intrusions sont illustrés dans la figure suivante : [19]

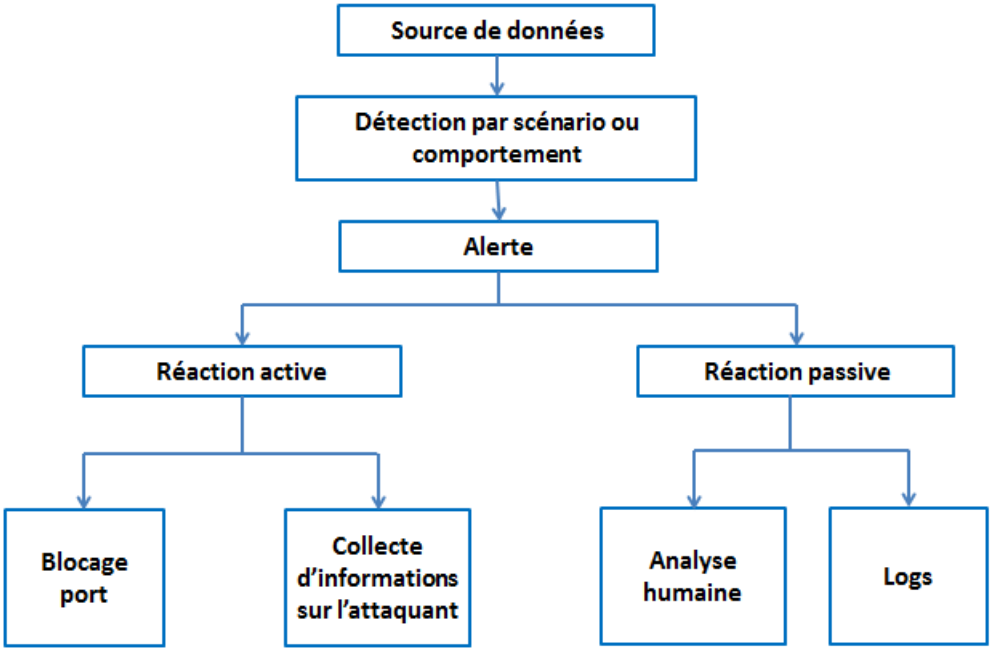

Figure 1.6 – *Fonctionnement d'un IDS [19] .*

1.3.4. Emplacement des IDS

Il existe plusieurs endroits stratégiques oú il convient de placer un IDS pour atteindre le niveau de protection attendu selon la politique de sécurité choisie. [20] Le schéma suivant illustre un réseau local ainsi que les trois positions que peut y prendre un IDS :

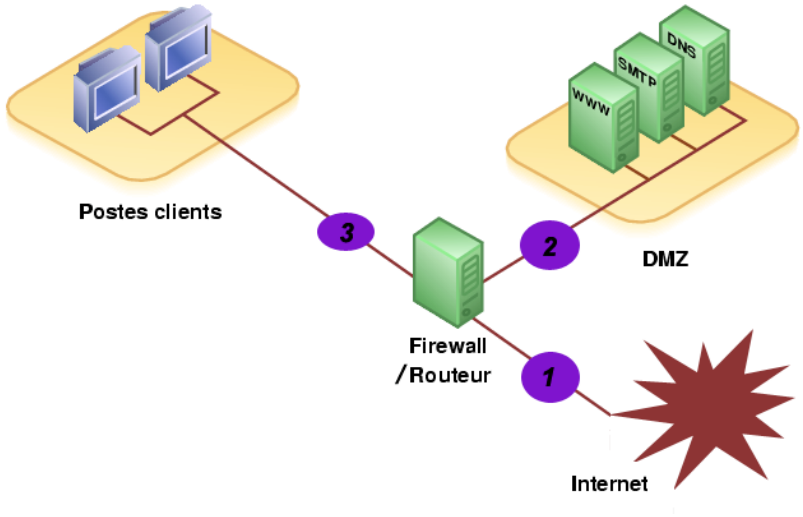

Figure 1.7 – *Emplacement d'un IDS [20].*

- **Position 1 :**
  Lorsque l'IDS prend cette position, son rôle sera de détecter l'ensemble des attaques frontales, provenant de l'extérieur, vers le parefeu. Donc, plusieurs alertes seront remontées ce qui rendra les logs difficilement consultables.

- **Position 2 :**
  Si l'IDS est placé sur la DMZ, il détectera les attaques qui n'ont pas été filtrées par le parefeu et qui relèvent d'un certain niveau de compétence. Les logs seront ici plus clairs à consulter puisque les attaques bénines ne seront pas recensées. [20]

- **Position 3 :**
  L'IDS dans cette position a pour objectif de rendre compte des attaques internes, provenant du réseau local de l'entreprise. Il peut être judicieux d'en placer un à cet endroit étant donné le fait que 80% des attaques

proviennent de l'intérieur. De plus, si des trojans ont contaminé le parc informatique (navigation peu méfiante sur internet) ils pourront être ici facilement identifiés pour être ensuite éradiqués. [20]

1.3.5. Approches pour la détection d'intrusion

Il existe deux approches pour la détection d'intrusions. La première consiste à rechercher des signatures connues d'attaques tandis que la seconde consiste à définir un comportement normal du système et à rechercher ce qui ne rentre pas dans ce comportement.
Un système de détection d'intrusions par recherche de signatures connaît ce qui est mal, alors qu'un système de détection d'intrusions par analyse de comportement connaît ce qui est bien. On parle de détection de malveillances et de détection d'anomalies.[16]

- **a) Détection de malveillances :**

  Cette approche est plutôt ancienne, remontant aux années 1990,et s'avère très efficace pour trouver des menaces connues. Elle consiste à rechercher des activités abusives par comparaison avec des descriptions abstraites de ce qui est considéré comme malveillant.
  Cette approche tente de mettre en forme des règles qui décrivent les usages non désirés, en s'appuyant sur des intrusions passées ou des faiblesses théoriques connues.
  En cas de détection d'une menace, une alerte est émise et le processus de remédiation est enclenché. [20] L'implémentation de cette approche peut être réalisée en utilisant plusieurs méthodes :

  - **Systèmes experts :**
    Ils peuvent être utilisés pour coder les signatures de malveillance avec des règles d'implication si . . .alors. Les signatures décrivent un aspect d'une attaque ou d'une classe d'attaque. Il est possible d'ajouter des nouvelles règles pour les nouvelles attaques.
  - **Analyse des transitions d'états :**
    On crée un modèle tel que le système au début ne soit pas compromis. L'intrus accède au système. Il exécute une série d'actions qui provoquent des transitions sur les états du modèle, qui peuvent être des états oú on considère que le système soit compromis. Cette approche

de haut niveau peut reconna^des variations d'attaques qui passeraient inaperçues avec des approches de plus bas niveau.

- **Réseaux de neurones :**
  La flexibilité apportée par les réseaux neuronaux permet d'analyser des données même si elles sont incomplètes ou déformées. Ils peuvent de plus permettre une analyse non linéaire de ces données. Leur rapidité permet l'analyse d'importants flux d'audit en temps réel.
  On peut utiliser les réseaux neuronaux pour filtrer et sélectionner les informations suspectes pour permettre une analyse détaillée par un système expert. On peut aussi les utiliser directement pour la détection de malveillances.
  Mais leur apprentissage est extrêmement délicat, et il est difficile de savoir quand un réseau est prêt pour l'utilisation. On peut également lui reprocher son côté boite noire (on ne peut pas interpréter les coefficients).

- **Raisonnement sur des modèles :**
  On essaye de modéliser les malveillances à un niveau élevé et intuitif d'abstraction en termes de séquences d'événements qui définissent l'intrusion.
  Cette technique peut être utile pour l'identification d'intrusions qui sont proches mais différentes. Elle permet aussi de cibler les données sur lesquelles une analyse approfondie doit être faite.

- **Algorithmes génétiques :**
  On définit chaque scénario d'attaque comme un ensemble pas forcément ordonné d'événements. Lorsqu'on veut tenir compte de tous les entremê lements possibles entre ces ensembles, l'explosion combinatoire qui en résulte interdit l'usage d'algorithmes de recherche traditionnels, et les algorithmes génétiques sont d'un grand secours.
  On peut rapprocher les méthodes utilisées à cette approche à ceux qu'on peut les rencontrer au domaine des antivirus ou encore dans le domaine de la génomique oú l'on recherche une séquence d'ADN dans un brin La détection de malveillance a deux inconvénients principales :
  * La difficulté de construction des bases de signatures.
  * La non détection des attaques non connues.

- **b) Détection d'anomalies :**
  Cette approche se base sur l'hypothèse que l'exploitation d'une faille du système nécessite une utilisation anormale de ce système, et donc un comportement inhabituel de l'utilisateur.
  Elle cherche donc à répondre à la question «le comportement actuel de l'utilisateur ou du système estil cohérent avec son comportement passé ? ».
  Il existe plusieurs méthodes pour la mise en œuvre de cette approche, parmi elles on peut citer :

  - **Observation de seuils :**
    On fixe le comportement normal d'un utilisateur à certaine valeur (seuil), par exemple le nombre maximum de mots de passe erronés, mais Il est très difficile de caractériser un comportement intrusif en termes de seuils.
    En effet, on peut avoir beaucoup de fausses alertes ou d'intrusions non détectées dans une population d'usagers non uniforme par exemple.
  - **Profilage d'utilisateurs :**
    On crée et on maintiens des profils individuels du travail des utilisateurs, auxquels ils sont censés adhérer ensuite. Au fur et à mesure que l'usager change ses activités, son profil de travail attendu se met a jour.
    Certains systèmes tentent de concilier l'utilisation de profils à court terme et de profils à long terme. Il reste cependant difficile de profiler un utilisateur irrégulier ou très dynamique. De plus, un utilisateur peut arriver à habituer lentement le système à un comportement intrusif.
  - **Profilage de programmes exécutables :**
    On observe l'utilisation des ressources du système par les programmes exécutables. Les virus, chevaux de Troie, vers, bombes logiques et autres programmes du même goût se voient démasqués en profilant la façon dont les objets du système comme les fichiers ou les imprimantes sont utilisés. Le profilage peut se faire par type d'exécutable.
  - **Profilage adaptatif à base de règles :**
    Contrairement à la détection de malveillances à base des règles, là on n'a pas besoin des connaissances d'un expert car ces règles sont générées automatiquement lors de la phase d'apprentissage. Donc, l'efficacité de cette méthode nécessite la génération de beaucoup de règles ce qui engendre des problèmes de performance.
  - **Réseaux de neurones :**
    Les réseaux neuronaux offrent une alternative à la maintenance d'un modèle de comportement normal d'un utilisateur.

Ils peuvent offrir un modèle plus efficace et moins complexe que les moyennes et les déviations standards. Cette approche a aussi beaucoup d'inconvénients comme :

* La difficulté à 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.
* Pas de prise en compte des tentatives de collusion entre utilisateurs.
* Choix délicat des différents paramètres du modèle statistique...etc.

- **c) Systèmes hybrides :**
  Pour compenser les lacunes des méthodes précédentes, certains systèmes utilisent une combinaison de la détection d'anomalies et la détection de malveillances.
  Par exemple, un administrateur peut avoir un profil qui lui permet d'accéder à certains fichiers sensibles, mais on doit vérifier que les attaques connues ne soient pas utilisées contre ces fichiers. à l'inverse,l'utilisation des fichiers comportant le mot «nucléaire »ne caractérise aucune signature d'attaque, mais cela est possible si ce n'était pas dans les habitudes de l'utilisateur.

1.3.6. Efficacité des IDS

L'efficacité d'un système de détection d'intrusions est déterminée par les mesures suivantes : [21][22][23]

- **Exactitude :**
  Un système de détection d'intrusions n'est pas exact s'il déclare comme malicieux une activité légale. Ce critère correspond au faux positif.

- **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.

- **Perfection :**
  Un système de détection d'intrusions est imparfait s'il n'arrive pas à détecter une attaque.

- **Tolérance aux pannes :**
  Le système de détection d'intrusions doit luimême résister aux attaques, en particulier dans le cas des attaques de déni de service. Ceci est important car plusieurs systèmes de détection d'intrusions s'exécutent sur des matériels ou logiciels connus vulnérables aux attaques.

- **Opportunité :**
  Un système de détection d'intrusions doit exécuter et propager son analyse d'une manière prompte pour permettre une réaction rapide dans le cas d'existence d'une attaque.

1.3.7. Limites des IDS

La plupart des systèmes de détection d'intrusions existants souffrent d'au moins deux des problèmes suivants : [24]

- Tout d'abord, les informations utilisées par le système de détection d'intrusions sont obtenues à partir d'un audit des chemins ou des paquets sur un réseau. Les données doivent parcourir un long chemin à partir de leur origine à l'IDS, donc elles peuvent potentiellement être détruites ou modifiées par un attaquant. En outre, le système de détection d'intrusions doit déduire le comportement du système à partir des données collectées, ce qui peut entraîner des interprétations erronées ou des événements manqués. Cela est appelé problème de fidélité.

- Deuxièmement, le système de détection d'intrusions utilise en permanence des ressources système supplémentaires, il surveille même lorsqu'il n'y a pas d'intrusions, car les composants du système de détection d'intrusions doivent fonctionner en permanence. C'est le problème de consommation des ressources.

- Troisièmement, parce que les composants du système de détection d'intrusions sont mis en œuvre comme programmes distincts, ils sont susceptibles d'être altérés. Un intrus peut potentiellement désactiver ou modifier les programmes exécutés sur un système, rendant la détection d'intrusions inutile ou peu fiable. C'est le problème de fiabilité.

1.4. Conclusion

Dans un monde où le progrès technologique avance à grande vitesse, où les gens, les entreprises, les organismes, les pays et même les objets sont de plus en plus connectés, les attaques informatiques sont de plus en plus fréquentes.
La question de la cybersécurité se pose à tous les niveaux et tend à devenir un enjeu essentiel ces prochaines années.
Dans ce chapitre, nous avons abordé différentes notions concernant la sécurité informatique, où nous avons présenté les différents types d'attaques, leur classification, leurs objectifs et motivations et les techniques utilisées pour protéger le système contre ces attaques. Parmi ces mécanismes, nous avons détaillé les systèmes de détection d'intrusions, vu que c'est notre objectif dans ce mémoire, qui continuent d'évoluer pour répondre aux exigences et offre un éventail de fonctionnalités capables de satisfaire les besoins de tous les types d'utilisateurs.
Donc, nous avons détaillé l'architecture des systèmes de détection d'intrusions, leur principe de fonctionnement et les différentes approches pour la détection d'intrusions où on a divisé les IDS en deux grandes catégories , les IDS comportementaux et les IDS à base de signatures.
Ces derniers consistent à détecter les attaques en se basant sur leurs signatures ce qui demande une mise à jour périodique de la base des signatures et rend la détection des nouvelles attaques impossible.
C'est pour cela qu'on a basé dans ce travail sur l'approche comportementale qui offre la possibilité de détecter les attaques inconnues en s'appuyant sur les métaheuristiques qui seront détaillés dans les prochaines chapitres.

Chapitre 2

Stratégies de détection d'intrusion basée sur des méthodes bio-inspiré

2.1. Introduction

De nos jours, les chercheurs emploient deux méthodes différentes pour résoudre les problèmes d'optimisation : les méthodes exactes et les méthodes approchées. L'utilisation des méthodes exactes offre la garantie de trouver un optimum global pour un problème donné. Cependant, ils ne sont pas directement applicables à la plupart des problèmes réels parce que le temps d'exécution requis croît de façon exponentielle avec l'ampleur du problème.
En revanche, les méthodes approchées sont généralement assez rapides, mais en général, ils offrent des solutions non optimales.
L'observation de la nature a conduit les chercheurs à emprunter les principes observés dans les phénomènes naturels pour créer des algorithmes efficaces pour la résolution de certains problèmes difficiles.
Une nouvelle ère est ouverte avec les algorithmes inspirés de la nature (bio-inspiré) qui sont des méta-heuristiques imitant la nature pour résoudre des problèmes d'optimisation.
Afin d'améliorer les résultats de détection d'intrusion, des méthodes inspirées de la nature ont été utilisées. Depuis quelques années, les chercheurs dans le domaine de sécurité des réseaux ont trouvé dans le monde naturel, une source d'inspiration inépuisable pour la conception des systèmes de détection d'intrusion.
[25]

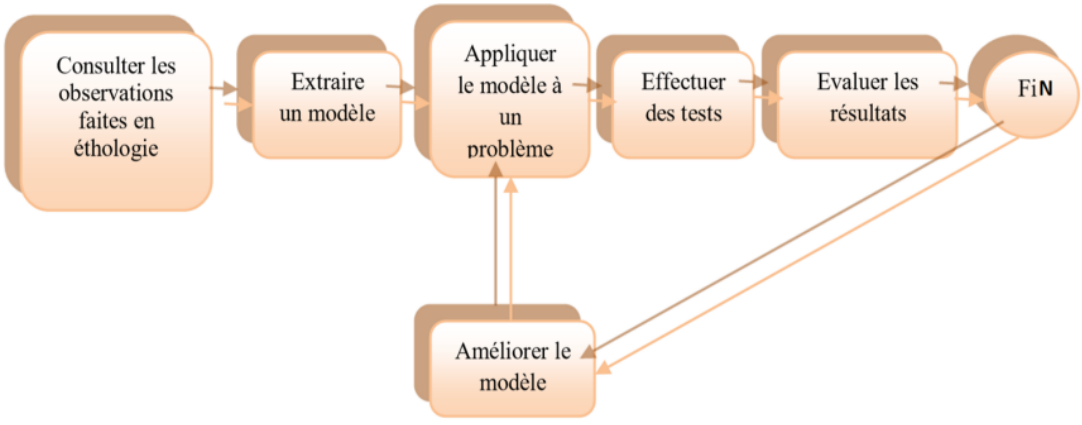

Figure 2.1 – *Le principe de développement d'une méthode biomimétique [25].*

2.2. Méthodes de résolution de problèmes

Les méthodes de résolution de problèmes peuvent être réparties en deux grandes classes :

- Les méthodes exactes.
- Les méthodes approchées.

2.2.1. Méthodes exactes :

Ce sont des méthodes qui garantissent l'obtention de la solution optimale du problème traité, Ils consistant à effectuer une énumération explicite de toutes les solutions pour assurer l'obtention de toutes les solutions ayant le potentiel d'être meilleures que la solution optimale trouvée au cours de la recherche, mais au prix de temps de calcul prohibitif et/ou d'espace mémoire souvent très grand. [26]

2.2.2. Méthodes approchées :

Dans certaines situations, il est nécessaire de disposer d'une solution de bonne qualité en un temps raisonnable, les méthodes approchées offrants cette possibilité. Ces méthodes peuvent être réparties en deux classes : [5]

2.2.2.1. Heuristiques :

Une heuristique est une méthode de résolution spécialisée à un problème particulier. Qui a pour but de trouver une solution réalisable en un temps raisonnable, mais pas nécessairement optimale.
L'usage d'une heuristique est pertinent pour calculer une solution approchée d'un problème difficile et ainsi accélérer le processus de résolution exacte. [27]

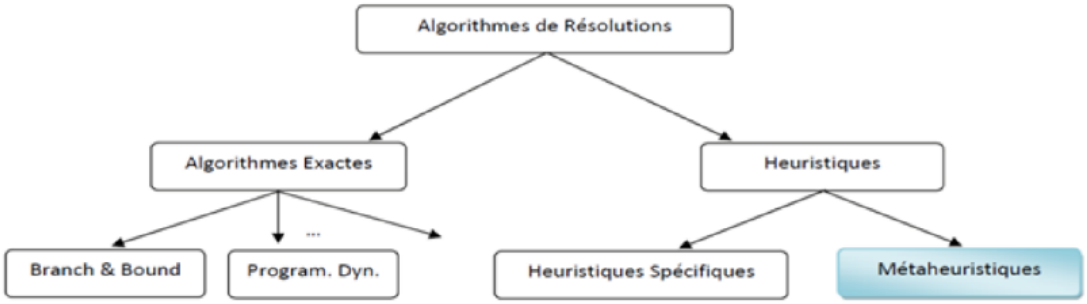

FIGURE 2.2 – *Classes des méthodes de résolutions [27].*

2.2.2.2. MÉTA-HEURISTIQUES :

Une méta-heuristique est une heuristique générique qu'il faut adapter à chaque problème. Plusieurs définitions ont été proposées pour expliquer clairement ce qu'est une méta-heuristique. Aucune de ces définitions n'est universellement reconnue. Une méta-heuristique est un processus itératif qui subordonne et guide une heuristique, en combinant intelligemment plusieurs concepts pour explorer et exploiter tout l'espace de recherche.
Des stratégies d'apprentissage sont utilisées pour structurer l'information afin de trouver efficacement des solutions optimales, ou presque-optimales . [28]
Soit S un ensemble et soit f une fonction qui associe une valeur f(s) à chaque élément S $^{*} \in S$.

L'objectif d'un algorithme d'optimisation est de déterminer un élément dans S qui minimise la fonction f.
En d'autres termes, il s'agit de déterminer S $^{*} \in Stelque$:
$f(s)* = minf(s)$

L'ensemble S contient typiquement toutes les solutions d'un problème d'optimisation, et la fonction f correspond alors à l'objectif qu'on tente d'optimiser. On peut cependant également définir S comme un ensemble de solutions ne satisfaisant pas nécessairement toutes les contraintes du problème considéré. Il existe plusieurs méta-heuristiques qui peuvent être classées comme suit :
[29]

- Les méta-heuristiques à solution unique.
- Les méta-heuristiques à base de population.

2.3. Méthodes bio-inspirées

Afin de créer des systèmes autonomes, robustes et adaptatifs, des ingénieurs et scientifiques trouvent leurs inspirations dans la nature, puisqu'elle montre des phénomènes extrêmement divers, dynamiques, robustes, complexes et fascinants. Elle trouve toujours la solution optimale pour résoudre son problème, et maintien l'équilibre parfait entre ses composantes.

2.3.1. Informatique bio-inspirée

Les méthodes bio-inspirées ont récemment gagné une importance dans l'informatique en raison de la nécessité d'une flexibilité, des moyens adaptables de résoudre les problèmes d'ingénierie. Les algorithmes bio-inspirés sont basés sur la structure et fonctionnement des systèmes naturels complexes et ont tendance à résoudre les problèmes de façon adaptable et distribué [30]

2.3.2. Motivation de l'utilisation du bio-inspiré

La nature de ses phénomènes extraordinaires nous fournit des solutions grâce à des caractéristiques telles que :

- **Emergence :** les éléments simples qui interagissent vont accomplir des tâches extraordinaires. 48 Classification de algorithmes bio-inspirés.
- **La simplicité :** de la mise en œuvre.
- **L'auto-organisation :** l'organisation interne du système se structure automatiquement sans être dirigée par une source extérieure.
- **La modularité :** le système est composé d'éléments simples qui coopèrent ensemble pour atteindre l'objectif global. Le système est donc évolutif.
- **La décentralisation :** ceci garantit un système robuste, capable de continuer à fonctionner en cas de défaillance d'un de ses composants.
- **La réactivité :** les éléments du système coopèrent et communiquent entre eux via des interactions locales. Ils sont capables de réagir instantanément aux changements d'environnement.
- **L'auto-adaptation :** l'aptitude d'un système à modifier ses paramètres de manière que son fonctionnement demeure satisfaisant en dépit des variations de son environnement.

2.3.3. Processus de création d'un algorithme inspiré de la nature

L'homme s'inspire de la nature pour développer une observation sur un phénomène naturel (Figure 2.3).
Il commence par sa modélisation en utilisant des simulations mathématiques. Une fois le modèle est raffiné, il sera utilisé pour extraire une méta-heuristique.[31]

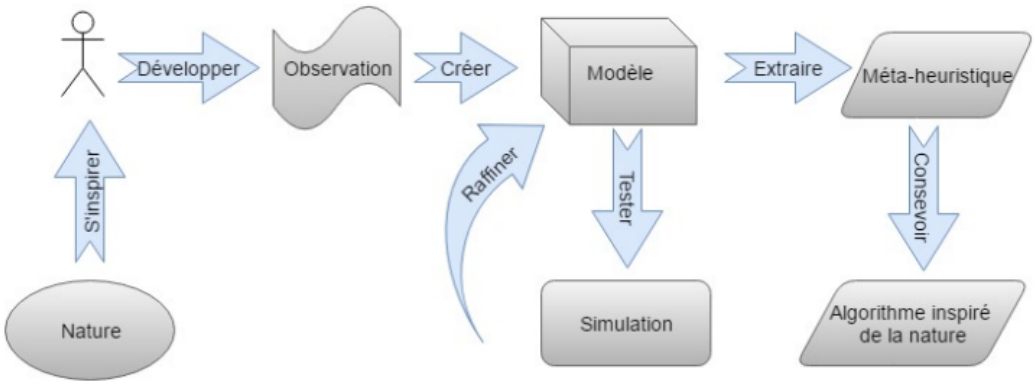

Figure 2.3 – *Passage d'un phénomène naturel à un algorithme inspiré de la nature [31].*

2.4. Classification de algorithmes bio-inspirés

Les méthodes bio-inspirés peuvent être réparties en deux grandes classes selon la source d'inspiration de la méthode bio-inspiré.

2.4.1. Algorithmes évolutionnaires :

Les algorithmes évolutionnaires sont des techniques de recherche inspirées par l'évolution biologique des espèces. Ils s'inspirent de l'évolution des êtres vivants (la théorie Darwinienne de la sélection naturelle des espèces) pour résoudre des problèmes d'optimisation. [33]
L'idée ici est que, les individus qui ont hérité des caractères bien adaptés à leur milieu ont tendance à vivre assez longtemps pour se reproduire, alors que les plus faibles ont tendance à disparaître. Ceux-ci comprennent les algorithmes génétiques, programmation génétique, stratégies d'évolution et évolution différentielle.

2.4.2. Algorithmes basés essaim :

Les algorithmes basés essaim sont des techniques d'optimisation inspirés du comportement collectif chez les espèces sociales comme les hyènes tachetée, les fourmis, les abeilles, les guêpes, les termites (fourmis blanches), les poissons et les oiseaux.
Qui sont des populations d'agents extrêmement simples, interagissant et communiquant indirectement à travers leur environnement, constituent des algorithmes distribués pour résoudre les problèmes réels difficiles. Parmi les algorithmes d'optimisation inspirés de l'intelligence en essaim les plus réussis, sont les colonies de fourmis et l'optimisation par essaim de particules, optimisation par la colonie d'abeille et récemment la recherche coucou. [27]

2.5. Caractéristiques et classification des métaheuristiques :

Les métaheuristiques ont en commun les caractéristiques suivantes :[32]

- La plupart des métaheuristiques utilisent des processus aléatoires comme moyens de récolter de l'information et de faire face à des problèmes comme l'explosion combinatoire.
- Les métaheuristiques du fait de leur capacité à être utilisées sur un grand nombre de problèmes différents, se prêtent facilement à des extensions.
- Souvent d'origine discrète à l'exception des essaims de particules et l'électromagnétisme.
- Elles sont inspirées par analogie avec la réalité : avec la physique (le recuit simulé), avec la biologie (les algorithme génétiques) ou avec l'éthologie (les colonies de fourmis) ...
- Les concepts de base des métaheuristiques peuvent être décrit de manière abstraite, sans faire appel à un problème spécifique.
- Les métaheuristiques peuvent contenir des mécanismes qui permettent d'éviter d'être bloqué dans des régions de l'espace de recherche.

2.6. TABLEAU COMPARATIVE CLASSIFICATEURS D'APPRENTISSAGE AUTOMATIQUE DANS L'ENSEMBLE D'APPRENTISSAGE :

2.6.1. L'ALGORITHME GÉNÉTIQUE (GA) :[17]

L'algorithme génétique (GA) est une méthode adaptative de recherche d'optimisation globale et simule le comportement du processus d'évolution dans la nature. Il cartographie l'espace de recherche dans un espace génétique.
Autrement dit, chaque clé possible est codée dans un vecteur appelé chromosome. Un élément du vecteur représente un gène. Tous les chromosomes constituent une population et sont estimés selon la fonction de fitness [34],[35],[36].
Une valeur de fitness sera utilisée pour mesurer la "fitness" d'un chromosome. Les populations initiales du processus génétique sont créées de manière aléatoire. GA utilise trois opérateurs pour produire une génération suivante à partir de la génération actuelle : reproduction, croisement et mutation. GA élimine les chromosomes de faible fitness et conserve ceux de haut fitness [37],[38].
Tout ce processus est répété et davantage de chromosomes de haute condition physique passent à la génération suivante, jusqu'à ce qu'un bon chromosome (individuel) soit trouvé.
L'algorithme génétique (GA) est proposé comme un outil capable d'identifier les types de connexions nuisibles dans un réseau informatique. Différentes caractéristiques des données de connexion telles que la durée et les types de connexion dans le réseau ont été analysées pour générer un ensemble de règles de classification.
Pour cette méthode métaheuristique, un ensemble de données de référence standard connu sous le nom de NSL-KDD a été étudié et utilisé pour étudier l'efficacité de la méthode proposée sur ce domaine problématique.
Les règles comprennent huit variables qui ont été simulées pendant le processus de formation pour détecter toute connexion malveillante pouvant conduire à une intrusion sur le réseau. Avec de bonnes performances dans la détection de mauvaises connexions, cette méthode peut être appliquée dans un système de détection d'intrusion pour identifier une attaque, améliorant ainsi les fonctionnalités de sécurité d'un réseau informatique. [33]

2.6.2. L'ALGORITHME PARTICLE SWARM OPTIMIZER (PSO) : [18]

Selon les auteurs dans [18] chaque particule est une connexion réseau qui représente une règle. Leur algorithme crée récursivement une population de particules à partir d'un ensemble de données d'entraînement.
Ensuite, pour chaque particule, il calcule sa fitness et met à jour les Pbest et

Gbest, c'est-à-dire la vitesse et la position valeurs de cette particule. Lorsque certains critères sont remplis, le Gbest particule (la règle la plus adaptée) est insérée dans les ensembles de règles dans le même temps, les données d'entraînement couvertes par cette règle sont supprimées.
les auteurs ont remarqué que PSO ne peut pas être directement appliqué au réseau ensembles de données d'intrusion, car dans ce cas, les attributs prennent valeurs distinctes. Pour surmonter cette limitation, ils ont également proposé un nouveau schéma de codage qui associe également des valeurs d'attribut distinctes à des valeurs entières non négatives. les résultats dans [39] atteignent de meilleurs taux de détection en incorporant un plus fonction de remise en forme précise au système.
En résumé, les IDS orientés PSO utilisent cette technique comme une étape supplémentaire d'un mécanisme de classification classique. [40]

2.6.3. L'algorithme Grey Wolf Optimizer (GWO) : [19]

l'algorithme Grey Wolf Optimizer (GWO) est une solution pour résoudre le problème de la détection d'intrusion avec les loups gris. Les loups gris sont une nouvelle approche développée par le chercheur Seyedali Mirjalili en 2014 pour résoudre les problèmes NP -complet, les loups gris basés sur le concept de groupe est également inspiré par la société des loups.
cette approche est basée sur les algorithmes de gestion des données afin que les loups gris contiennent une hiérarchie spéciale dans chaque niveau contient une catégorie de loups (alpha, betta, delta, omega).
chaque niveau, nous avons placé un Algorithme de gestion des données, donc le niveau oméga avec lequel choisi l'algorithme KNN distance Euclidienne qui permet d'effectuer un classement soit l'utilisateur soit un intrus ou non intrus ou suspect, et aussi pour le niveau delta ils ont utilisés l'algorithme KNN avec une distance manhaten est réalisé une classification, aussi pour betta avec l'algorithme KNN et la distance cheby chev, l'objectif de cette hiérarchie est de protéger le niveau alpha .[41]

2.6.4. L'algorithme Réseaux Bayésiens naïfs : [20]

cette approche est une méthode de corrélation d'alertes basées sur les RB naïfss. il utilise l'historique des observation pour construire un Rb naïfs pour chaque objectif d'intrusion.
Pendant l'étape de détection, chaque action observée se traduit par une évidence qui mit à jour chaque RB naïfs.
Selon le degré d'influence de cette action, la probabilité de chaque objectif d'intrusion change positivement ou négativement.

l'approche a pour avantage de rendre la prédiction des plans d'attaque plus facile grâce à la simplicité et l'efficacité des RB naïfss. Elle tire profit des données disponibles, et n'implique qu'une légère contribution des connaissance d'experts pour déterminer les objectifs d'intrusion.
En plus, les actions impliquées dans les plans d'attaque peuvent être identifiées et les fausses alarmes sont implicitement filtrées en se concentrant sur les actions pertinentes. Contrairement aux approches existantes, les scénarios d'attaque ne sont pas explicitement fournis par des experts, mais ils sont calculés automatiquement à partir des données d'observations.[42]

2.6.5. L'algorithmes machine learning :

2.6.5.1. L'algorithme Support Vector Machine : [21]

Le modèle SVM est formé et conçu à l'aide d'un ensemble de données NSL-KDD réduit à l'aide de l'algorithme de sélection du meilleur ensemble de fonctionnalités. Avant d'utiliser la classification SVM, une mise à l'échelle doit être effectuée.
Cette action consiste à augmenter la précision, à réduire le chevauchement et à réduire la complexité.
SVM utilise un espace de grande dimension pour trouver un hyper-plan pour effectuer une classification binaire, où le taux d'erreur est minimal.
SVM est formé à l'aide d'un ensemble de données NSL-KDD réduit, trouvant plusieurs vecteurs de support qui représentent les données de formation. Ces vecteurs supports seront formés d'un modèle par le SVM, représentant une catégorie. [43]

2.6.5.2. L'algorithme Table Descision : [22]

le Calcul de traitement maximal et le temps des tâches consommatrice a toujours été une limite dans le traitement d'énormes données d'intrusion sur le réseau.
Ce problème peut être minimisé par sélection de fonctionnalités pour condenser la taille des données du réseau impliqué. cette approche à prétraitons d'abord l'ensemble de données NSL-KDD. Puis ils étudions et analysons deux algorithmes d'arbre de décision (C4.5 et standard ID3) de data mining pour la tâche de détection intrusions et comparer leurs performances relatives. on peut conclure que l'arbre de décision C4.5 est le plus adapté à un taux de vrais positifs (TPR) élevé et à un faible taux de faux positifs (FTR) et un faible temps de calcul avec une grande précision.[44]

2.6.5.3. L'algorithme k Nearest Neighbor (K-NN) : [24],[25],[26]

Il s'agit d'une technique d'apprentissage automatique non paramétrique et la plus transparente pour la classification et la régression d'échantillons. [45] [46]
Il effectue le calcul de la séparation approximative entre plusieurs points sur le vecteur d'entrée et la position non étiquetée désignée à son K-NN.
Le paramètre k représente le nombre d'observations les plus adjacentes à l'observation donnée dans l'ensemble de données de test ou de validation.
Dans ce classificateur, un nouveau point est pris et classé selon la majorité des votes obtenus pour le point le plus proche dans les données d'apprentissage. Pour mesurer la similarité entre deux points, la distance euclidienne est utilisée comme métrique de distance.

Réf	Approche	Accuracy	Sensitivity	Specificity	Precision	F1 score	MCC
[17]	GA	0.982	0.979	0.985	0.983	0.981	0.964
[18]	PSO	0.983	0.978	0.981	0.984	0.980	0.963
[19]	GWO	0.977	0.973	0.981	0.978	0.975	0.953
[20]	Naive Bayes	0.967	0.652	0.964	0.967	0.789	0.774
[21]	SVM	0.998	0.999	0.998	0.998	0.985	0.998
[22]	Table de Descision	0.999	0.994	0.996	0.995	0.983	0.995
[23]	One R	0.966	0.938	0.967	0.996	0.946	0.959
[24]	KNN n=3	0.968	0.973	0.972	0.969	0.972	0.948
[25]	KNN n=5	0.965	0.972	0.969	0.964	0.975	0.944
[26]	KNN n=7	0.963	0.962	0.967	0.968	0.977	0.944

2.7. Conclusion :

Les méta-heuristiques sont des algorithmes génériques, souvent inspirés de la nature, conçue pour résoudre des problèmes d'optimisation complexes. Ils présentent certains avantages par rapport aux méthodes classiques d'optimisation Numérique. Ce chapitre offre un voyage sur la théorie et les applications des algorithmes méta-heuristiques bio-inspirés où nous avons dressé une revue de littérature non exhaustive sur une gamme D'algorithmes et des quelques algorithmes de machine learning.

Chapitre 3

Description de l'approche proposée

3.1. Introduction

Les méta-heuristiques forment un ensemble de méthodes utilisées en recherche opérationnelle et en intelligence artificielle pour résoudre des problèmes NP complet. Ce chapitre présente la description d'une solution proposée dont le but de classification dans un système de détection d'intrusion basé sur les l'hyènes tachettes d'optimisation (Spotted Hyena Optimization).

3.2. Inspiration

Les relations sociales sont de nature dynamique. Ceux-ci sont affectés par l'évolution des relations entre la composition du réseau et les individus quittant ou rejoignant la population. Le social l'analyse en réseau du comportement animal a été classée en trois catégories :[47]

- La première catégorie comprend les facteurs environnementaux, tels que la disponibilité des ressources et la concurrence avec d'autres espèces animales.
- La deuxième catégorie se concentre sur les préférences sociales basées sur le comportement ou la qualité individuelle.
- La troisième catégorie a moins d'attention de la part des scientifiques qui comprend les relations sociales des espèces elles-mêmes.

La relation sociale entre les animaux est l'inspiration de notre travail et corrèle ce comportement à l'hyène tachetée qui est scientifiquement nommée Crocuta.

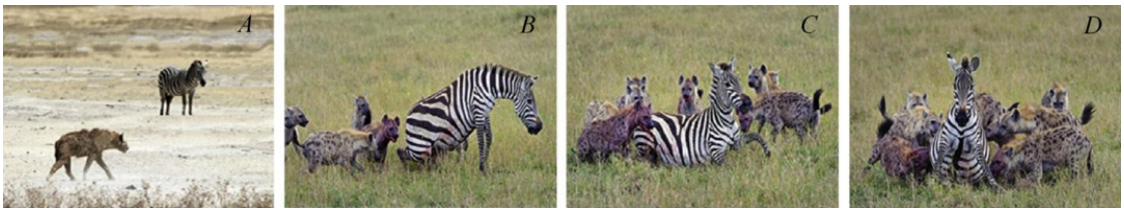

Figure 3.1 – *Comportement de chasse des hyènes tachetées : (A) recherche et suivi des proies (B) chasse (C) gênant et encerclement (D) situation immobile et attaque des proies [Advances in Engineering Software]. [48].*

Les hyènes sont de grands carnivores ressemblant à des chiens. Ils vivent dans les savanes, prairies, sous-déserts et forêts d'Afrique et d'Asie.
Ils vivent 10 à 12 ans dans la nature et jusqu'à 25 ans en prison, il existe quatre espèces d'hyènes connues, à savoir l'hyène tachetée Hyène rayée, hyène brune et protèle qui diffèrent par la taille, le comportement et le type de régime alimentaire.
Toutes ces espèces ont une attitude d'ours car les pattes avant sont plus longues que les pattes arrière. Les hyènes tachetées sont des chasseurs habiles et la plus grande des trois autres espèces d'hyènes (c'est-à-dire rayées, brunes et loup-garou). [48]
Le tacheté Hyène est aussi appelée Hyène qui rit parce que sont beaucoup semblables à un rire humain.
On les appelle ainsi parce qu'il y a taches sur leur fourrure qui est de couleur brun rougeâtre avec des taches noires. Les hyènes tachetées sont des animaux compliqués, intelligents et très sociaux avec une réputation vraiment épouvantable. Ils ont la capacité de se battre sans fin sur le territoire et la nourriture.
Dans la famille des hyènes tachetées, les membres féminins sont dominants et vivent dans leur clan.
Cependant, les membres masculins quittent leur clan quand ils sont adultes pour rechercher et rejoindre un nouveau clan. Dans cette nouvelle famille, ce sont les membres les moins bien classés pour obtenir leur part du repas.
Un membre masculin qui a rejoint le clan reste toujours avec le même membres (amis) depuis longtemps. Alors qu'une femelle, est toujours aussi sûre d'une place stable. Un fait intéressant sur les hyènes tachetées est qu'ils produisent une alerte sonore très similaire à celle de l'homme rire pour communiquer entre eux lorsqu'une nouvelle source de nourriture est trouvée.
Selon Ilany et al. [47] les hyènes tachetées vivent généralement et chasser en groupe, s'appuyer sur un réseau d'amis de confiance avoir plus de 100 membres. Et pour augmenter leur réseau, ils nouent généralement avec une autre hyène tachetée qui est l'amie d'un ami ou liée en quelque sorte par parenté plutôt que par n'importe quel inconnu repéré hyène.
Les hyènes tachetées sont des animaux sociaux qui peuvent communiquer les uns avec les autres par des appels spécialisés tels que les postures et signaux.[47]
Ils utilisent de multiples procédures sensorielles pour reconnaître leur parents et autres personnes.
Ils peuvent également reconnaître les parents de tiers et classer les relations entre leurs compagnons de clan et utiliser ceci connaissances lors de la prise de décision sociale. La piste de l'hyène tachetée proie par la vue, l'audience et l'odorat.
La figure 3.1 montre le mécanisme de suivi, de chasse, d'encerclement et d'attaque des hyènes tachetées. Les clusters cohésifs sont utiles pour une coopération

efficace entre les hyènes et aussi maximiser la forme physique.
La chasse, la technique et la relation sociale des hyènes tachetées sont modélisées mathématiquement pour concevoir SHO et effectuer une optimisation.[48]

3.3. Modèle mathématique et algorithme d'optimisation :

3.3.1. Proie encerclant :

Les hyènes tachetées peuvent être familières avec l'emplacement des proies et les encercler.
Modéliser mathématiquement la hiérarchie sociale des hyènes tachetées, nous considérons que la meilleure solution candidate actuelle est la proie cible ou l'objectif qui est proche de l'optimum car d'espace de recherche non connu a priori.
Les autres agents de recherche essayer de mettre à jour leurs positions, après que la meilleure solution candidate de recherche soit définie, sur la meilleure solution candidate optimale.
[48]

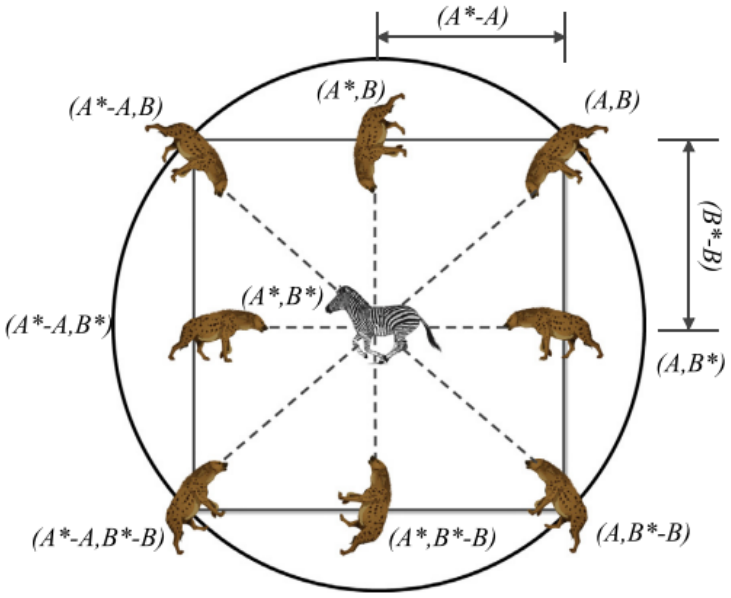

Figure 3.2 – *Vecteurs de position 2D de l'hyène tachetée. [48].*

Le modèle mathématique de ce comportement est représenté par les équations suivantes :

$$\overrightarrow{D}_h = |\overrightarrow{B}.\overrightarrow{P}_p(x) - \overrightarrow{P}(x)| \tag{3.1}$$

$$\overrightarrow{P}(x+1) = \overrightarrow{P}_p(x) - \overrightarrow{E}.\overrightarrow{D}_h \tag{3.2}$$

Où $\overrightarrow{D}_h$ définit la distance entre la proie et l'hyène tachetée,
x Indique l'itération courante.
$\overrightarrow{B}$ et $\overrightarrow{E}$ sont des vecteurs de coefficients.
$\overrightarrow{P}_p$ Indique le vecteur de position de la proie.
$\overrightarrow{P}$ est le vecteur de position d'hyène tachetée.

Cependant, (|| et .) est respectivement la valeur absolue et la multiplication par des vecteurs.
Les vecteurs $\overrightarrow{B}$ et $\overrightarrow{E}$ sont calculés comme suit :

$$\overrightarrow{B} = 2.\overrightarrow{rd}_1 \tag{3.3}$$

$$\overrightarrow{E} = \overrightarrow{2h}.\overrightarrow{rd}_2 - \overrightarrow{h} \tag{3.4}$$

$$\overrightarrow{h} = 5 - (Iteration * (5/Max_{Iteration}) \tag{3.5}$$

Ou, Itération =1,2, 3,..., $\text{Max}_{Iteration}$

Pour un bon équilibre entre l'exploration et l'exploitation. $\overrightarrow{h}$ est initialement diminué de 5 à 0 au cours du nombre maximum d'itérations $Max_{Iteration}$.
De plus, ce mécanisme favorise plus d'exploitation à mesure que la valeur d'itération augmente.
Cependant,$\overrightarrow{rd}_1$, $\overrightarrow{rd}_2$ Sont des vecteurs aléatoires dans [0, 1].
La figure 3.2 montre les effets des équations. (1) et (2) dans un environnement bidimensionnel. Dans cette figure, l'hyène tachetée (A, B) peut mettre à jour sa position vers la position de proie $A\star, B\star$).
En ajustant la valeur des vecteurs $\overrightarrow{B}$ et $\overrightarrow{E}$ ils y sont un nombre différent d'endroits qui peuvent être atteints environ le poste actuel.
Les positions probablement mises à jour d'une hyène tachetée dans l'environnement 3D sont illustrées à la Fig. 3.3.
En utilisant éq. (1) et (2), une hyène tachetée peut mettre à jour sa position au hasard autour de la proie. Par conséquent, le même concept peut encore s'étendre avec un espace de recherche à n dimensions.[48]

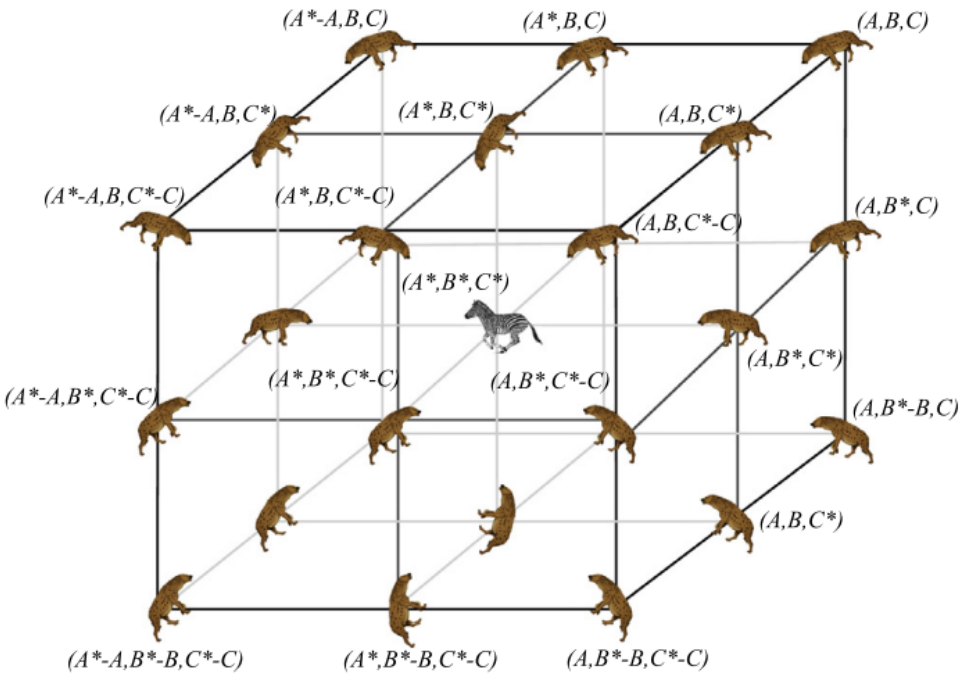

FIGURE 3.3 – *Vecteurs de position 3D et emplacements suivants possibles de l'hyène tachetée. [48].*

3.3.2. CHASSE

Les hyènes tachetées vivent et chassent généralement en groupes et dépendent d'un réseau d'amis de confiance et la capacité de reconnaître l'emplacement de proie. Pour définir mathématiquement le comportement des hyènes tachetées, nous supposons que le meilleur agent de recherche, quel que soit l'optimum, à connaître l'emplacement des proies. Les autres agents de recherche font un cluster, groupe d'amis de confiance, vers le meilleur agent de recherche et enregistré les meilleures solutions obtenues jusqu'à présent pour mettre à jour leurs positions.[48]
Les équations suivantes sont proposées dans ce mécanisme :

$$\vec{D}_h = |\vec{B}.\vec{P}_h - \vec{P}_k| \tag{3.6}$$

$$\vec{p}_k = \vec{p}_h - \vec{E}.\vec{D}_n \tag{3.7}$$

$$\vec{C}_h = \vec{P}_k + \vec{P}_{k+1} + ... + \vec{P}_{k+n} \tag{3.8}$$

Où $\vec{p}_h$ définit la position de la première meilleure hyène tachetée.

$\vec{p}_k$ Désigne la position des autres hyènes tachetées.
N indique le nombre d'hyènes tachetées qui se calcule comme suit :

$$N = count_k(\vec{p}_h, \vec{p}_{h+1}, \vec{p}_{h+2}, ..., (\vec{p}_h + \vec{M})) \tag{3.9}$$

Où $\vec{M}$ est un vecteur aléatoire dans [0.5, 1], nos définit le nombre de solutions et compter toutes les solutions candidates, après addition avec $\vec{M}$, qui sont très similaires à la meilleure solution optimale dans un espace de recherche.
$\vec{C}_h$ est un groupe ou un groupe de N nombre d'optimaux solutions.[48]

3.3.3. Attaquer des proies (exploitation) :

Afin de modéliser mathématiquement l'attaque de la proie, nous diminuons la valeur du vecteur $\vec{h}$. La variation du vecteur $\vec{E}$ est aussi diminuée pour changer la valeur du vecteur h qui peut décroître de 5 à 0 au fil des itérations.
La figure 3.3. Montre que |E| < 1 forces le groupe d'hyènes tachetées à l'assaut vers la proie. La formulation mathématique pour attaquer la proie est la suivante :

$$\vec{p}_(X + 1) = \frac{\vec{C}_h}{N} \tag{3.10}$$

Où $\vec{p}_(X + 1)$ enregistre la meilleure solution et met à jour les positions des autres agents de recherche selon la position de la meilleure recherche agent.
L'algorithme SHO permet à ses agents de recherche de mettre à jour leur position et attaque vers la proie.[48]

Figure 3.4 – *Recherche de proies (|E| > 1). [48].*

FIGURE 3.5 – *Proie attaquante (|E| < 1). [48].*

3.3.4. RECHERCHE DE PROIES (EXPLORATION)

Les hyènes tachetées recherchent principalement la proie, selon la position du groupe ou du groupe d'hyènes tachetées qui résident dans le vecteur $\overrightarrow{C}_h$. Ils s'éloignent les uns des autres pour chercher et attaquer proie. Par conséquent, nous utilisons $\overrightarrow{E}$ avec des valeurs aléatoires supérieur à 1 ou inférieur à -1 pour forcer les agents de recherche à s'éloigner de la proie.[48]
Ce mécanisme permet à l'algorithme SHO de rechercher globalement. Pour trouver une proie appropriée, la Fig. 3.4 montre que |E| > 1 facilite l'éloignement des hyènes tachetées de la proie.
Une autre constituant de l'algorithme SHO qui rend possible l'exploration est $\overrightarrow{B}$. Dans l'éq. (3), le vecteur $\overrightarrow{B}$ contient des valeurs aléatoires qui fournissent les poids aléatoires des proies. Pour montrer le comportement le plus aléatoire de l'algorithme SHO, supposons que le vecteur $\overrightarrow{B} > 1$ a priorité sur $\overrightarrow{B} < 1$ pour démontrer l'effet à distance comme on peut le voir dans l'Eq. (3).[48] Cela sera utile pour l'exploration et l'évitement des optima locaux. Selon la position d'une hyène tachetée, elle peut décider au hasard d'un poids à la proie et éventuellement la rendre rigide ou au-delà pour atteindre les hyènes tachetées. Nous avons intentionnellement besoin du vecteur $\overrightarrow{B}$ pour fournir valeurs aléatoires pour l'exploration non seulement lors des itérations initiales, mais également pour les itérations finales.
Ce mécanisme est très utile pour éviter les problèmes d'optima locaux, plus que jamais dans les itérations finales. Pour terminer, l'algorithme SHO est terminé en satisfaisant des critères de terminaison. Le pseudo-code de l'algorithme SHO montre comment SHO peut résoudre des problèmes d'optimisation, certains points peuvent être notés comme suit :[48]

- L'algorithme proposé enregistre les meilleures solutions obtenues jusqu'à

présent au cours de l'itération.

- Le mécanisme d'encerclement proposé définit un cercle en forme de voisinage autour des solutions qui peuvent être étendues à dimensions supérieures comme une hypersphère.
- Les vecteurs aléatoires $\overrightarrow{B}$ et $\overrightarrow{E}$ aident les solutions candidates à avoir hypersphères avec différentes positions aléatoires.
- La méthode de chasse proposée permet aux solutions candidates de localisé la position probable de la proie.
- La possibilité d'exploration et d'exploitation par les valeurs ajustées des vecteurs $\overrightarrow{E}$ et $\overrightarrow{h}$ et permet à SHO de facilement passage de l'exploration à l'exploitation.
- Avec le vecteur $\overrightarrow{E}$, la moitié des itérations sont dédiées à la recherche.
- (Exploration) ($|E| \geq 1$) et l'autre moitié est consacrée à la chasse (exploitation) ($|E| \leq 1$).

3.4. ÉTAPES ET ORGANIGRAMME DE SHO

Les étapes de SHO sont résumées comme suit :

- **Etape 1 :** Initialiser la population de hyènes tachetées $\overrightarrow{P}_i$ où i = 1, 2, . . ., n.
- **Etape 2 :** Choisissez les paramètres initiaux de SHO : h, B, E et N et définir le nombre maximum d'itérations.
- **Etape 3 :** Calculez la valeur de fitness de chaque agent de recherche.
- **Etape 4 :** Le meilleur agent de recherche est exploré dans la recherche donnée espace.
- **Etape 5 :** Définissez le groupe de solutions optimales, c'est-à-dire le cluster à l'aide de éq. (8) et (9) jusqu'à ce que le résultat satisfaisant soit trouvé.
- **Etape 6 :** Mettez à jour les positions des agents de recherche à l'aide de l' éq. (10).
- **Etape 7 :** Vérifiez si un agent de recherche va au-delà du frontière dans un espace de recherche donné et l'ajuster.

- **Etape 8 :** Calculez la valeur de fitness de l'agent de recherche de mise à jour et mettez à jour le vecteur Ph s'il existe une meilleure solution que la précédente solution optimale.
- **Etape 9 :** Mettre à jour le groupe de hyènes tachetées Ch à mis à jour valeur de fitness de l'agent de recherche.
- **Etape 10 :** Si le critère d'arrêt est satisfait, l'algorithme être arrêté. Sinon, retournez à l'étape 5.
- **Etape 11 :** Renvoyer la meilleure solution optimale, après arrêt des critères est satisfaite, ce qui est obtenu jusqu'ici.

Algorithme 1 : Optimiseur d'hyène tachetée.

```
Entrée : la population de hyènes tachetées Pi (i = 1, 2, . . ., n)
Résultat : le meilleur agent de recherche
procédure SHO :
Initialiser les paramètres h, B, E et N
Calculer le fitness de chaque agent de recherche
P⃗_h= le meilleur agent de recherche
C⃗_h = le groupe ou cluster de toutes les solutions optimales de loin
tant que (x < Nombre max d'itérations) faire
    pour chaque chaque agent de recherche faire
        Mettre à jour la position du courant agent par l'éq. (10)
    Mettre à jour h, B, E et N
    Vérifiez si un agent de recherche va au-delà de l'espace de recherche
     donné, puis ajustez-le
    Calculer le fitness de chaque agent de recherche
    Mettre à jour Ph s'il existe une meilleure solution que solution
     optimale précédente
    Mettre à jour le groupe Ch w.r.t Ph
    x=x + 1
retour Ph
```

3.5. La Stratégie Proposée :

3.5.1. Modèle Proposé :

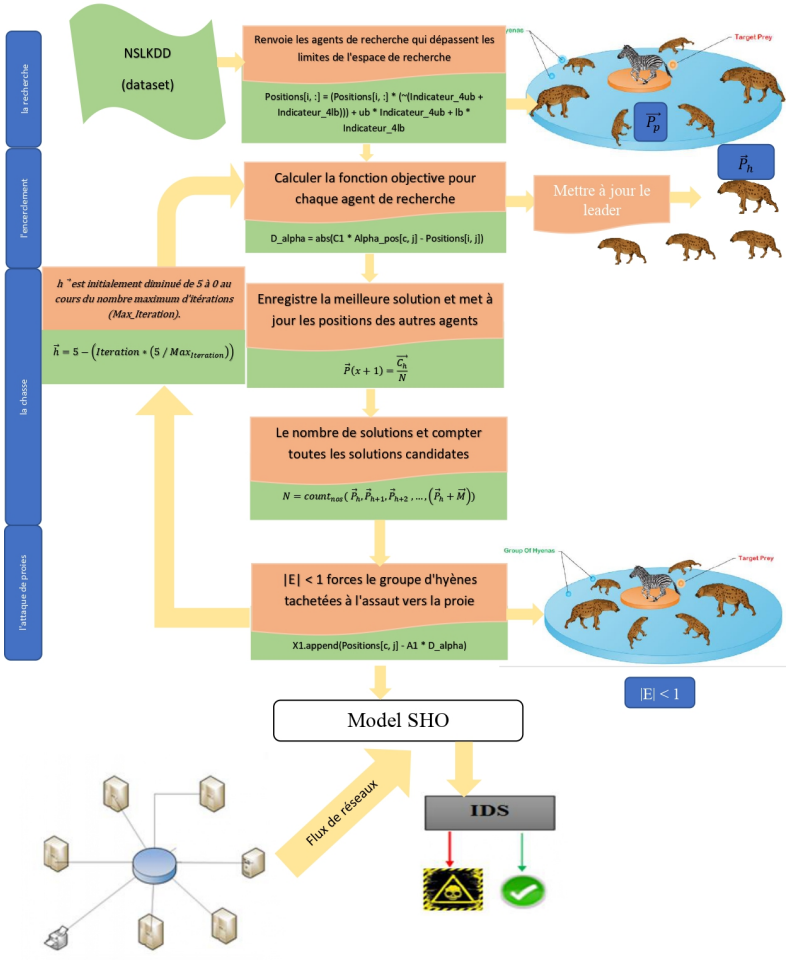

Figure 3.6 – *Représentation de l'architecture de notre proposition.*

Le processus de recherche de la solution optimale par l'algorithme SHO est le suivant :

- Tout d'abord, Générer la population initiale d'hyènes tachetées. Un certain nombre d'hyènes tachetées sont distribuées au hasard en tant que membres de la population d'hyènes dans l'espace problématique. Chaque membre de la population est une solution au problème.
- Configurez les paramètres initiaux de l'algorithme SHO, y compris le nombre maximum d'itérations, le nombre d'hyènes, la position initiale des hyènes tachetées.
- Ensuite, une solution aléatoire définie comme la population de hyènes tachetées est créée.
- Les meilleures et les pires hyènes tachetées de la population sont déterminées et un poids de forme physique de 1 et 0 leur est attribué.
- Définir un groupe (cluster) de solutions optimales dans l'espace de recherche pour atteindre résultats
- Mettre à jour la position des hyènes en recherche. Les hyènes tachetées ou les solutions au problème évoluent vers la solution optimale au fil du temps et réduisent leur rayon de recherche.
- Les poids de fitness sont également calculés pour le reste des solutions.
- Les solutions candidates autour de chacune des hyènes tachetées sont recherchées pour trouver une nouvelle solution optimale.
- Maintenant, une recherche locale est effectuée autour de l'hyène tachetée optimale avec un mécanisme d'attaque ou d'encerclement.
- L'algorithme SHO met à jour les vecteurs de caractéristiques et définit le vecteur de caractéristiques optimal à chaque itération.
- L'algorithme est répété jusqu'à ce que tous les membres de la population soient vérifiés.
- Dans la dernière itération, l'hyène tachetée optimale est considérée comme la solution optimale et utilisée comme réponse finale de l'algorithme.
- Enfin, les solutions optimales sont introduites comme résultat en tant que fonctionnalités sélectionnées.
- Différents paramètres d'évaluation ont été utilisés afin d'évaluer et de comparer les performances de l'algorithme proposé avec d'autres algorithmes.

3.6. Description :

3.6.1. Data Set utilisé (NSL-KDD) :

NSL-KDD est un ensemble de données suggéré pour résoudre certains des problèmes incohérents de l'ensemble de données KDD 99.
Bien que, encore, cette nouvelle version de l'ensemble de données KDD source de certains des problèmes discutés par McHugh [49] 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 seront cohérents 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.
Bien qu'il soit assez vieux et non une représentation parfaite des réseaux réels existants, il est en continu un indice qui est utilisé pour comparer les systèmes de détection d'intrusions dans les recherches communes. Dans la littérature la plus récente [50],[51],[52] tous les chercheurs utilisent le NSL-KDD comme ensemble de données de référence.[53]
La base de données NSL-KDD contient des enregistrements de connexion TCP/IP (125973 pour l'apprentissage et 22544 pour le test), dont chaque enregistrement est constitué de 41 attributs caractérisant la connexion, et un attribut indiquant la nature de connexion s'il s'agit d'une attaque ou non.
Les quatre catégories d'attaques existantes dans cette base sont :

- Dénis de Services.
- Probing.
- User to Root.
- Remote to User.

Le tableau suivant montre l'ensemble des attaques peuvent être inclut dans chaque type.

La classe	*Types d'attaques*
DoS	Apache2, Back, Land, Mailbomb, Neptune, Pod, Processtable, Smurf, Teardrop, Udpstrom
Probe	Mscan, Ipsweep, Nmap, Portsweep, Saint, Satan
R2L	Ftp_write, Guess_passwd, Imap, Multihop, Named, Phf, Dict, Snmpguess, Spy, Sqlattack, Warezclient, Warezmaster, Xlock, Xsnoop, Guest
U2R	Buffer_overflow, Httptunnel, Loadmodule, Xterm, Perl, Ps, Rootkit

FIGURE 3.7 – *Regroupement des attaques dans la base NSL-KDD. [48].*

La distribution des connexions de la base NSL-KDD est illustrée dans le tableau ci-dessous :

Type	*Base d'apprentissage*		*Base de test*	
	Nbr connexions	*Pourcentage*	*Nbr connexions*	*Pourcentage*
Normal	67343	53.46%	9711	43.07%
DoS	45927	36.46%	7591	33.67%
Probe	11656	09.25%	2421	10.74%
R2L	995	00.79%	2754	12.22%
U2R	52	00.04%	67	00.30%
Total	125973	100.0%	22544	100.0%

FIGURE 3.8 – *Distribution des données dans la base NSL-KDD [53].*

3.6.2. Contenu de la base de données NSL-KDD :

Cette base contient quatre groupes de données qui sont : [53][54]

- **KDDTrain+ :** qui contient toutes les données d'apprentissage du dataset NSL-KDD.
- **KDDTrain+ 20Percent :** qui représente seulement 20% des données d'apprentissage.
- **KDDTest+ :** ce groupe de données représente les données du test de la base NSL-KDD.
- **KDDTest 21 :** c'est un sous-ensemble du KDDT est+ qui n'inclut pas les enregistrements ayant un niveau de difficulté de 21 sur 21.

Type de données	$KDDTrain^{+}$	$KDDTrain^{+}$_20Percent	$KDDTest^{+}$	$KDDTest^{-21}$
Nbr d' enregistrements	125,973	25192	22,554	11850

Figure 3.9 – *Contenu de la base NSL-KDD.*

3.6.3. Les Avantages du Data Set NSL-KDD :

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 les meilleurs taux de détection sur les enregistrements fréquents. [55]
- 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.[55]

3.6.4. Numérisation des données :

Afin d'adapter le dataset NSL-KDD avec les modèles qui n'acceptent que des attributs numériques, et comme cette base contient 3 attributs alphabétiques qui sont :
protocol_type, service et flag parmi ses 41 attributs, il est nécessaire de transformer toutes ces données catégoriques en données numériques via un encodage précis. Dans notre cas, chaque valeur alphabétique est remplacée par son entier équivalent, c'est à dire, s'il existe N valeurs possibles pour l'attribut X, ces valeurs sont remplacées par des valeurs entières compris entre 0 et N1. Si on prend par exemple le cas de l'attribut protocol-type qui peut prendre trois valeurs : tcp, udp ou bien icmp, le résultat de numérisation de cet attribut sera comme suit :

Avant numérisation	*Après numérisation*
TCP	0
UDP	1
ICMP	2

Figure 3.10 – *Exemple de numérisation.*

3.6.5. Normalisation des données :

Pour garantir l'efficacité et améliorer les performances du modèle généré, il est très important d'ajuster les valeurs numériques obtenues après la phase de numérisation, puisqu'elles sont très variées et constituent un grand intervalle. Par exemple, certains attributs comme $src-bytes$ et $dst-types$ prennent des grandes valeurs tandis que d'autres comme $serror-rate$ et $same-srvrate$ ne prennent que des petites valeurs. Donc, pour éviter ce genre de problème, on est obligé d'effectuer une opération de transformation sur les données de la base NSL-KDD en utilisant une fonction bien choisie. Dans notre cas, la fonction utilisée est la fonction Min-Max décrite comme suit :

$$val_{nouv} = \frac{val_{anc} - Min_{anc}}{Max_{anc} - Min_{anc}} \tag{3.11}$$

Où :

val_{anc} : est la valeur à normaliser. Notre modèle de détection d'intrusions 52

val_{nouv} : est la valeur après la normalisation.

Min_{anc} : est la limite inférieure de l'intervalle à que valanc appartient.

Max_{anc} : est la limite supérieure de l'intervalle à que val_{anc} appartient.
En appliquant cette formule sur les données de la base NSL-KDD, on obtiendra une base normalisée dont toutes ses valeurs sont comprises entre 0 et 1.

3.6.6. les étapes :

- **Etape1 :** Renvoie les agents de recherche qui dépassent les limites de l'espace de recherche Vérifiez si un agent de recherche va au-delà du frontière dans un espace de recherche donné et l'ajuster.

- **Etape2 :** Calculer la fonction objective pour chaque agent de recherche
  $\overrightarrow{p}_k = \overrightarrow{p}_h - \overrightarrow{E}.\overrightarrow{D}_n$
  $\overrightarrow{p}_h$ Définit la position de la première meilleure hyène tachetée $\overrightarrow{p}_k$ Désigne la position des autres hyènes tachetées. Donc cette étape consiste à Calculer la valeur de fitness de chaque agent de recherche.

- **Etape3 :** $|E| < 1$ forces le groupe d'hyènes tachetées à l'assaut vers la proie Les hyènes tachetées recherchent principalement la proie, selon la position du groupe ou du groupe d'hyènes tachetées qui résident dans le vecteur ($\overrightarrow{C}_h$) .

- **Etape4** :Enregistre la meilleure solution et met à jour les positions des autres agents Le meilleur agent de recherche est exploré dans la recherche donnée
- **Etape5** : $\overrightarrow{h}$ est initialement diminué de 5 à 0 au cours du nombre maximum d'itérations $Max_{Iteration}$. De plus, ce mécanisme favorise plus d'exploitation à mesure que la valeur d'itération augmente.
- **Etape6** : Le nombre de solutions et compter toutes les solutions candidates Renvoyer la meilleure solution optimale, après arrêt des critères est satisfaite, ce qui est obtenu jusqu'ici.

3.6.7. Algorithme proposé :

Algorithme 2 : notre modele Optimiseur d'hyène tachetée.

```
import numpy as np
import random as rn
import time
from function_Eval import feval
def SHO(Positions,fobj,Lb,Ub,Max_iter,val_in, val_tar, test_dat, test_tar) :
fitness = 0
N, dim = Positions.shape[0], Positions.shape[1]
ub = Ub[1, :]
lb = Lb[1, :]
Alpha_pos = np.zeros(dim, 1)
Alpha_score = float('inf')
Alpha_pos = np.zeros(shp)
Alpha_score = float('inf')
Convergence_curve = np.zeros((Max_iter + 1, 1))
l = 0
ct = time.time()
new_fitness = []
tant que (l <= Max_iter) faire
    pour chaque i in range(N) faire
        Flag4ub = Positions[i, :] > ub
        Flag4lb = Positions[i, :] < lb
        Positions[i, :] = (Positions[i, :
        ] * ( (Flag4ub + Flag4lb))) + ub * Flag4ub + lb * Flag4lb
        new_fitness.append(feval(fobj, Positions[i, :
        ], val_in, val_tar, test_dat, test_tar))
        if new_fitness[i] < Alpha_score then
            Alpha_score = new_fitness[i]
            Alpha_pos = Positions[i, :]
    a = 5 - l * ((5) / Max_iter)
    if l == 0 then
        indn = np.zeros(dim, 1)
    else
        indn = new_fitness<fitness
```

```
pour chaque i in range(N) faire
    if indn == 1 then
        h = rn.randint(1, N-5)
    else
        h = 1
    pour chaque  j in range(dim) faire
        X1 = [] pour chaque  c in range(h) faire
            r1 = rn.random() r2 = rn.random() A1 = 2 * a * r1 - a C1 = 2 *
             r2 D_alpha = abs(C1 * Alpha_pos[j] - Positions[i, j])
             X1.append(Positions[c, j] - A1 * D_alpha)  Positions[i, j] =
             sum(X1) / dim
    Convergence_curve[l] = Alpha_score l = l + 1 fitness = new_fitness
est_fit = Convergence_curve[Max_iter − 1]
ct = time.time() − ct
returnbest_fit, Convergence_curve, Alpha_pos, ct
```

3.7. Conclusion :

Nous avons présenté dans ce chapitre une nouvelle solution pour une détection d'intrusions intelligente sur la base des hyènes tachetée. Nous avons proposé un nouveau modèle basé des hyènes tachetées pour le filtrage des connexions à des connexions normales ou anormale.

Chapitre 4

Discussion des résultats expérimentaux

4.1. Introduction

Ce chapitre est consacré à la phase d'implémentation de notre stratégie de détection d'intrusion à base d'une technique d'optimisation nommé l'hyènes tachetéees. Il permettra d'évaluer et de valider notre stratégie proposée. Pour cela, nous avons réalisé plusieurs expérimentation en utilisons plusieurs métrique que nous expliquerons dans ce chapitre.

4.2. Langage et Environnement de Travail

Les outils et environnement de programmation utilisés pour la réalisation de l'étude proposée :

4.2.1. Python :

langage de programmation polyvalent, qui existe déjà depuis assez longtemps, Quido van Rossum , son créateur, ayant débuté son développement en 1990. Stable et mature, il s'agit d'un langage de très haut niveau, dynamique et orienté objet[57]. Il soutient de multiples paradigmes de programmation au-delà de la programmation orientée objet, comme la programmation procédurale et fonctionnelle [58]. Un fichier python porte l'extension (.py).

4.2.2. Anaconda :

gestionnaire de paquets, un gestionnaire d'environnement, une distribution de données scientifiques Python/R et une collection de plus de 7 500 paquets open-source. Anaconda est gratuit et facile à installer, et il offre un soutien communautaire gratuit [59].

4.2.3. Spyder :

un environnement scientifique puissant écrit en Python, pour Python. Il est conçu par et pour les scientifiques, les ingénieurs et les analystes de données. Il présente une combinaison unique de fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec les capacités d'exploration de données, d'exécution interactive, d'inspection approfondie et de visualisation d'un ensemble scientifique [60].

4.2.4. Jupyter Notebook :

application Web à source ouverte qui vous permet de créer et de partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif.
Les utilisations incluent : nettoyage et transformation de données, simulation numérique, modélisation statistique, visualisation de données, apprentissage automatique, etc .
Le projet Jupyter est un projet open source à but non lucratif, né du projet IPython en 2014.
Il a été évolué pour soutenir la science des données interactives et le calcul scientifique dans tous les langages de programmation. Jupyter sera toujours un logiciel 100 % open-source, libre d'utilisation pour tous [61].

4.3. Bibliothèques essentielles pour l'apprentissage automatique en Python

bibliothèque de programmes ou bien librairie logicielle est un ensemble de fonctions utilitaires, regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. Les fonctions sont regroupées de par leur appartenance à un même domaine conceptuel (mathématique, graphique, tris, etc) [62].
La bibliothèque standard de Python est très grande, elle offre un large éventail d'outils [58].
Dans ce qui suit, nous allons définir les bibliothèques utilisées dans notre implémentation :

4.3.1. Pandas :

Bibliothèque open source, sous licence BSD (Berkeley Software Distribution), qui fournit des structures de données et des outils d'analyse de données performants et faciles à utiliser pour le langage de programmation Python [63].

4.3.2. Matplotlib :

Bibliothèque complète pour la création de visualisations statiques, animées et interactives en Python [64].

4.3.3. NumPy :

Paquet fondamental pour le calcul scientifique en Python.
C'est une bibliothèque Python qui fournit un objet de tableau multidimensionnel, divers objets dérivés (tels que des tableaux et des matrices masqués), et un assortiment de routines pour des opérations rapides sur des tableaux, y compris des opérations mathématiques, logiques, de manipulation de formes, de tri, de sélection, d'entrées/sorties, de transformées de fourier discrètes, d'algèbre linéaire de base, d'opérations statistiques de base, de simulation aléatoire et bien plus encore [65].

4.3.4. SKLEARN :

Un module Python intégrant des algorithmes classiques d'apprentissage machine dans le monde étroitement lié des paquets scientifiques Python (numpy, scipy, matplotlib) [66].

4.3.5. Pickle :

Le module pickle met en œuvre des protocoles binaires pour sérialiser et désérialiser une structure d'objet Python.
Le "pickling" est le processus par lequel une hiérarchie d'objets Python estconvertie en un flux d'octets, et le "unpickling" est l'opération inverse [58].

4.4. Métrique utilisée et résultat expérimentaux :

Notre approche présente un cadre complet pour sélectionner un meilleur modèle du métaheuristique pour des fonctionnalités d'ensemble de données NSL-KDD qui caractérisent efficacement le trafic normal et le distinguent du trafic anormal le but de cette expérience est de comparer notre solution avec différentes approches le résultat montre que notre stratégie est meilleure en termes F-Measure, Precision, Recall,Accuracy.. et que La méthode proposée fonctionne bien dans tous les aspects, et les résultats tirés de notre évaluation indiquent que l'approche proposée peut être utilisée pour une utilisation future :

4.4.1. Accuracy :

L'un des paramètres importants pour déterminer la précision des problèmes de classification explique la régularité avec laquelle le modèle prédit les sorties correctes et peut être mesuré comme le rapport du nombre de prédictions correctes faites par le classificateur sur le nombre total de prédictions faites par les classificateurs. . En termes de matrice de confusion.

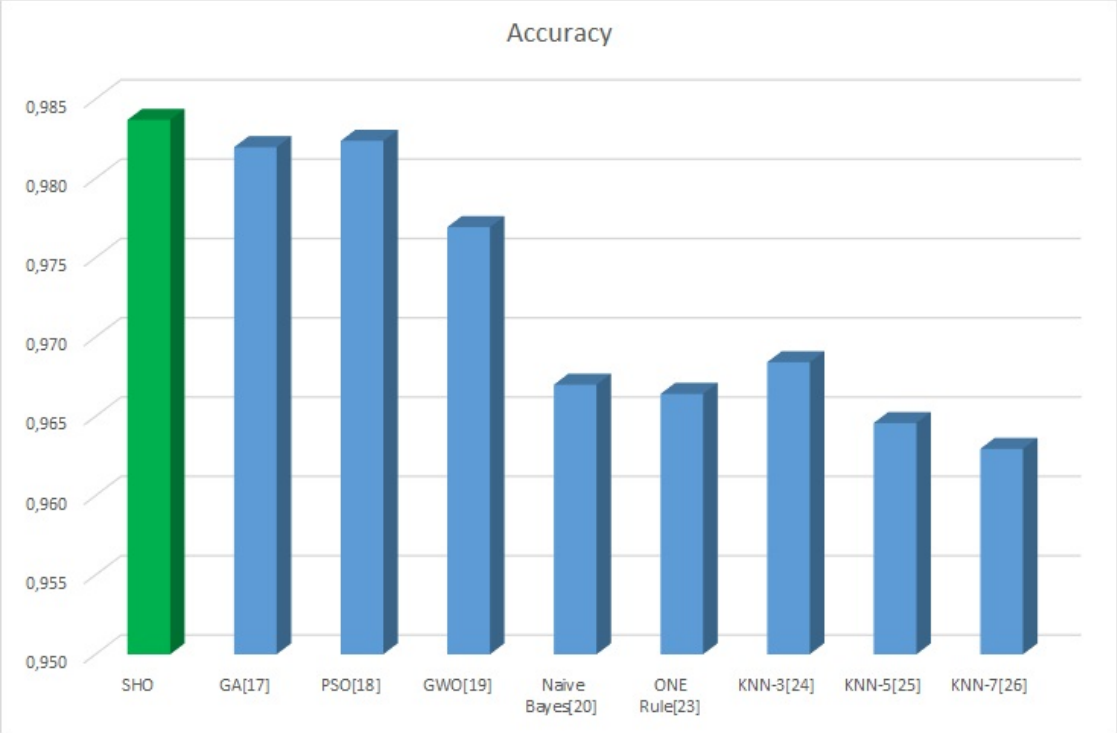

Figure 4.1 – *Comparaison des résultats Accuracy (histogramme).*

4.4.2. SENSITIVITY(LE RAPPEL) :

étant la proportion de documents correctement classés dans le système par rapport à tous les documents de la classe C_i.

Rappel = La sensibilité fait référence à la fraction d'éléments pertinents qu'une recherche IA renvoie sur le nombre total d'éléments pertinents dans la population d'origine.

S'il y a 18 documents pertinents dans l'ensemble de la population et que la recherche renvoie 9 éléments pertinents, le rappel est de 50 %.

$$Rappel(c_i) = \frac{\text{Nombre de documents bien classés dans } c_i}{\text{Nombre de documents de la classe} c_i} \tag{4.1}$$

$$R_i = \frac{V_{pi}}{V_{pi} + F_{ni}} \tag{4.2}$$

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 sufisant pour évaluer les performances d'un système. Pour cela, on définit la précision.

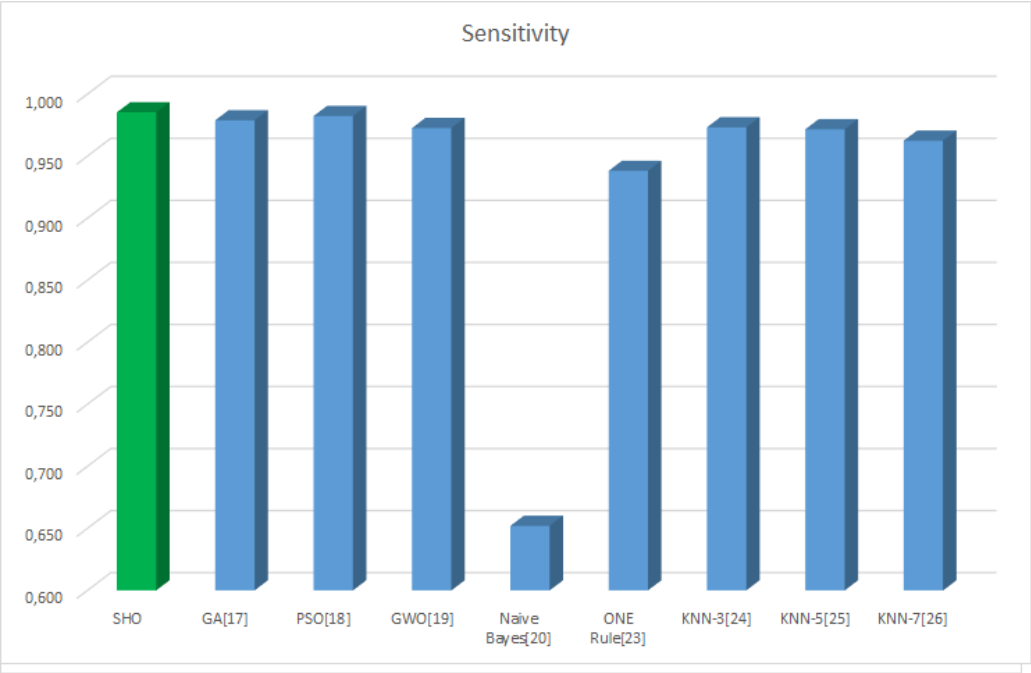

FIGURE 4.2 – *Comparaison des résultats Sensitivity (histogramme).*

4.4.3. Specificity :

La spécificité est la métrique qui évalue la capacité d'un modèle à prédire les vrais négatifs de chaque catégorie disponible.
Ces métriques s'appliquent à n'importe quel modèle catégoriel.

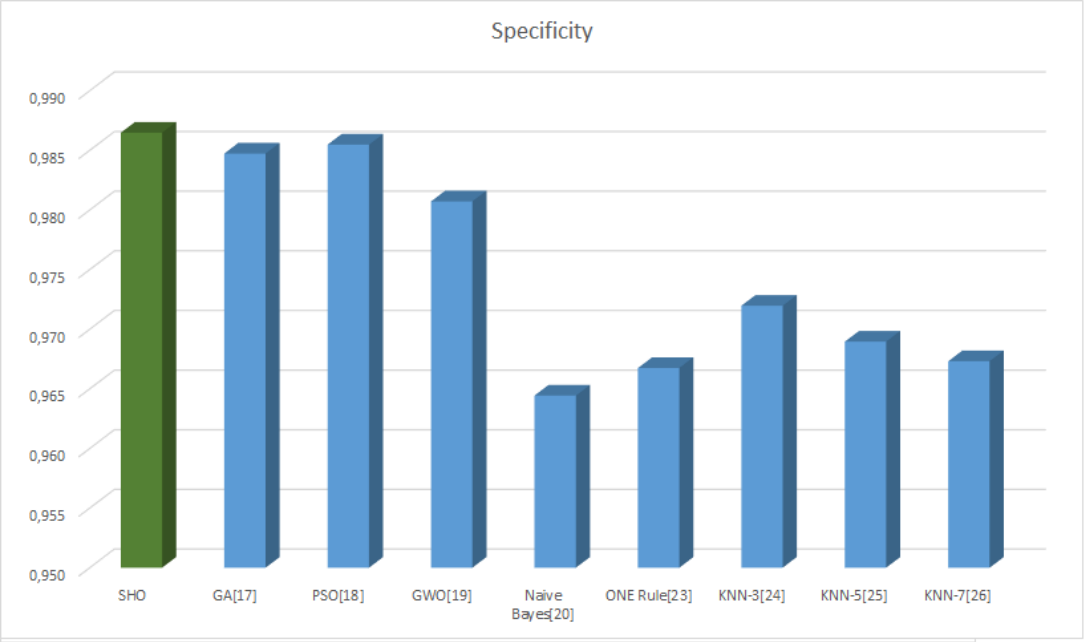

Figure 4.3 – *Comparaison des résultats Specificity (histogramme).*

4.4.4. Precision :

La précision fait référence au pourcentage d'éléments pertinents par rapport aux éléments non pertinents qu'une recherche renvoie.
Si une recherche renvoie 12 éléments de la population totale, dont 9 éléments sont pertinents et 3 non pertinents, la précision est de 60 %.

La précision vous indique dans quelle mesure une recherche évite les faux positifs.
La précision et le rappel sont tous deux importants pour le succès d'une recherche.
La précision est la proportion de documents correctement classés parmi ceux classés par le système dans C_i [56]

$$précision(c_i) = \frac{\text{Nombre de documents bien classés dans } c_i}{\text{Nombre de documents de la classe} c_i} \tag{4.3}$$

$$R_i = \frac{V_{pi}}{V_{pi} + F_{pi}} \tag{4.4}$$

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.

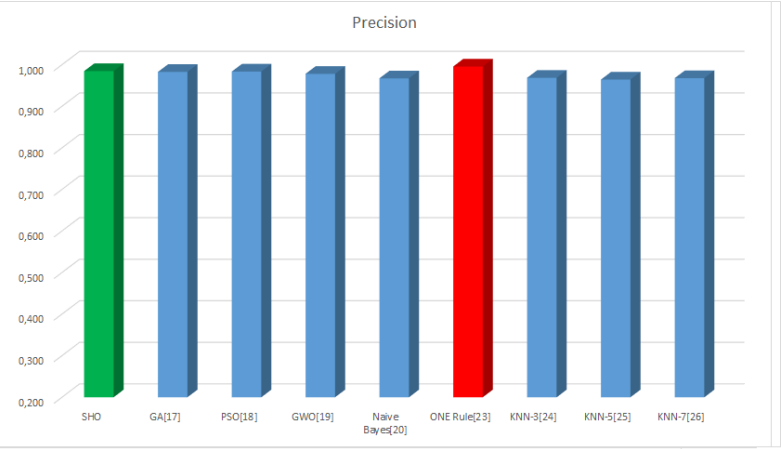

Figure 4.4 – *Comparaison des résultats precision (histogramme).*

4.4.5. F1 Score :

Le score F, également appelé score F1, est une mesure de la précision d'un modèle sur un ensemble de données. Il est utilisé pour évaluer les systèmes de classification binaires, qui classent les exemples en «positifs»ou «négatifs ».
Le score F est un moyen de combiner la précision et le rappel du modèle, et il est défini comme la moyenne harmonique de la précision et du rappel du modèle. Le score F est couramment utilisé pour évaluer les systèmes de recherche d'informations tels que les moteurs de recherche, ainsi que pour de nombreux types de modèles d'apprentissage automatique, en particulier dans le traitement du langage naturel.
Il est possible d'ajuster le F-score pour donner plus d'importance à la précision qu'au rappel, ou vice-versa. Les scores F ajustés courants sont le score F0.5 et le score F2, ainsi que le score F1 standard.

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 $F_{î^2}$ rappel et précision. On définit la mesure $F_{î^2}$ comme la moyenne harmonique entre le rappel et la précision :[56]

$$F_B = \frac{(B^2 + 1) * précision * rappel}{B^2 * précision + rappel} \tag{4.5}$$

Pour utiliser cette mesure, il est donc nécessaire de fixer préalablement un seuil de décision pour le classement, et de calculer la valeur de $F_{î^2}$ pour ce seuil. Le paramètre $F_{î^2}$ 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 F_B est fixée à 1 et la mesure est ainsi notée :[56]

$$F = \frac{2 * précision * rappel}{précision + rappel} \tag{4.6}$$

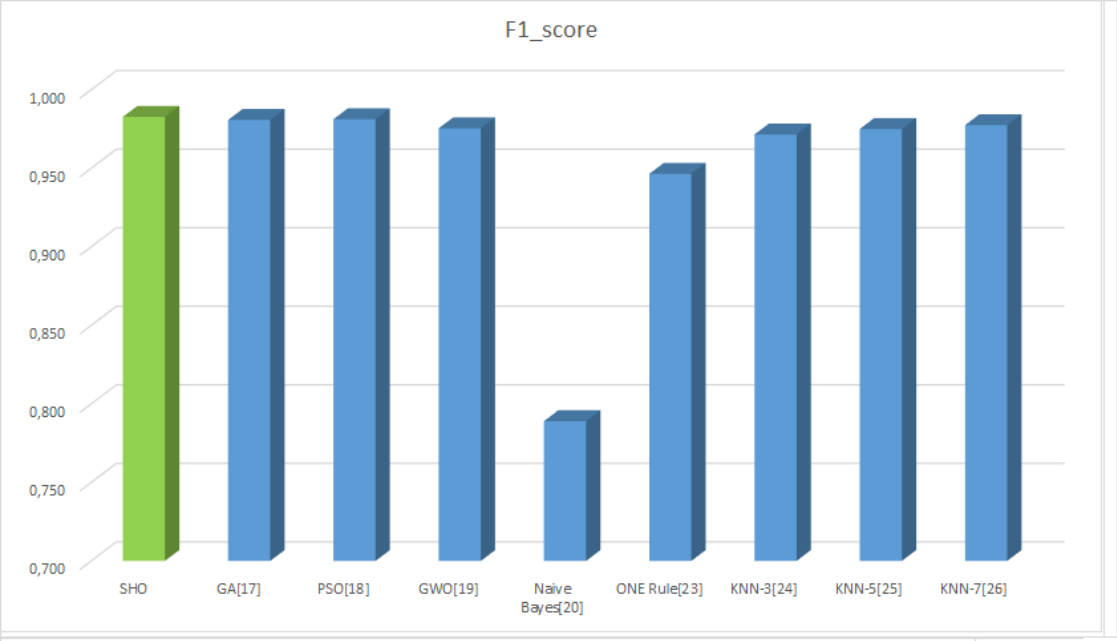

FIGURE 4.5 – *Comparaison des résultats F1_score (histogramme).*

4.4.6. MCC (THE MATTHEWS CORRELATION COEFFICIENT) :

MCC est une meilleure métrique de classification à valeur unique qui aide à résumer la matrice de confusion ou une matrice d'erreur. Une matrice de confusion a quatre entités :

- True positives (TP)
- True negatives (TN)
- False positives (FP)
- False negatives (FN)

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{4.7}$$

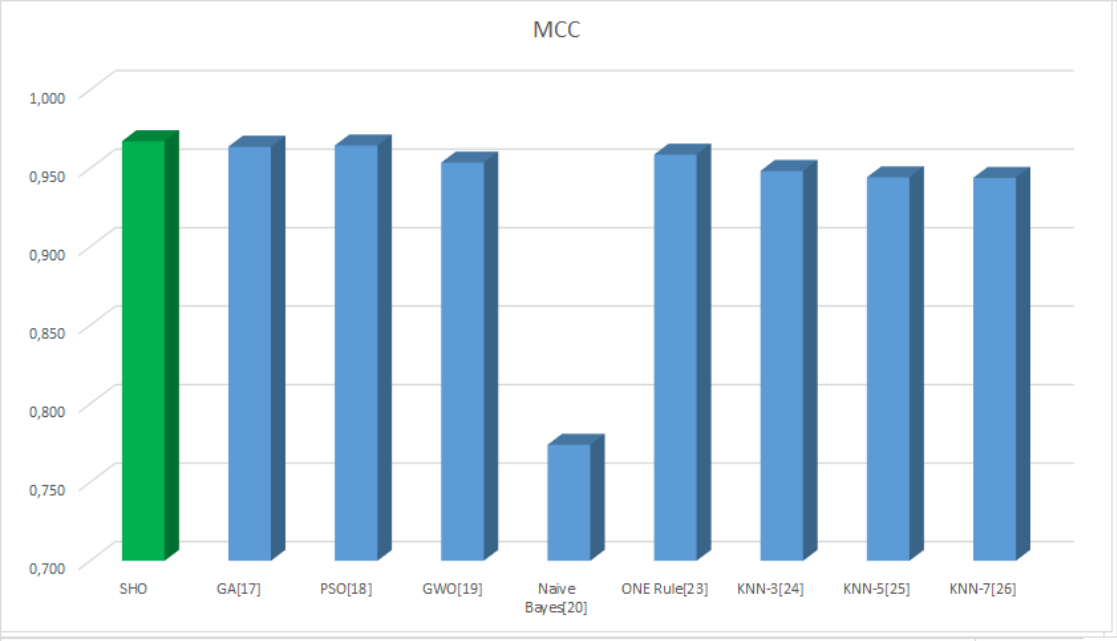

FIGURE 4.6 – *Comparaison des résultats MCC (histogramme).*

Réf	Approche	Accuracy	Sensitivity	Specificity	Precision	F1 score	MCC
	SHO	0.984	0.985	0.986	0.985	0.983	0.967
[17]	GA	0.982	0.979	0.985	0.983	0.981	0.964
[18]	PSO	0.983	0.978	0.981	0.984	0.980	0.963
[19]	GWO	0.977	0.973	0.981	0.978	0.975	0.953
[20]	Naive Bayes	0.967	0.652	0.964	0.967	0.789	0.774
[21]	SVM	0.998	0.999	0.998	0.998	0.985	0.998
[22]	Table Descision	0.999	0.994	0.996	0.995	0.983	0.995
[23]	One R	0.966	0.938	0.967	0.996	0.946	0.959
[24]	KNN n=3	0.968	0.973	0.972	0.969	0.972	0.948
[25]	KNN n=5	0.965	0.972	0.969	0.964	0.975	0.944
[26]	KNN n=7	0.963	0.962	0.967	0.968	0.977	0.944

4.5. Conception expérimentale et description des ensembles de données :

L'environnement expérimental est présenté en premier. Après cela, la conception de l'expérience et la description de l'ensemble de données sont fournies. Enfin, l'évaluation des performances est détaillée et discutée. L'algorithme SHO proposé est implémenté dans l'environnement PYTHON et sa précision est comparée avec les algorithmes GA[17], PSO[18], GWO[19] et d'autres algorithme de machine Learning. Ensuite, les algorithmes métaheuristique sont utilisés pour la sélection des caractéristiques et leurs résultats sont comparés. Aux fins de comparaison et d'analyse, différentes fonctions d'évaluation, la taille de la population et le nombre d'itérations sont examinés. Chaque algorithme est exécuté 30 fois avec une taille de population de 10 et 100 . Dans ces relations, TP, FP, TN et FN sont respectivement de vrais positifs, de faux positifs, de vrais négatifs et de faux négatifs pour identifier les fausses pages des pages légales

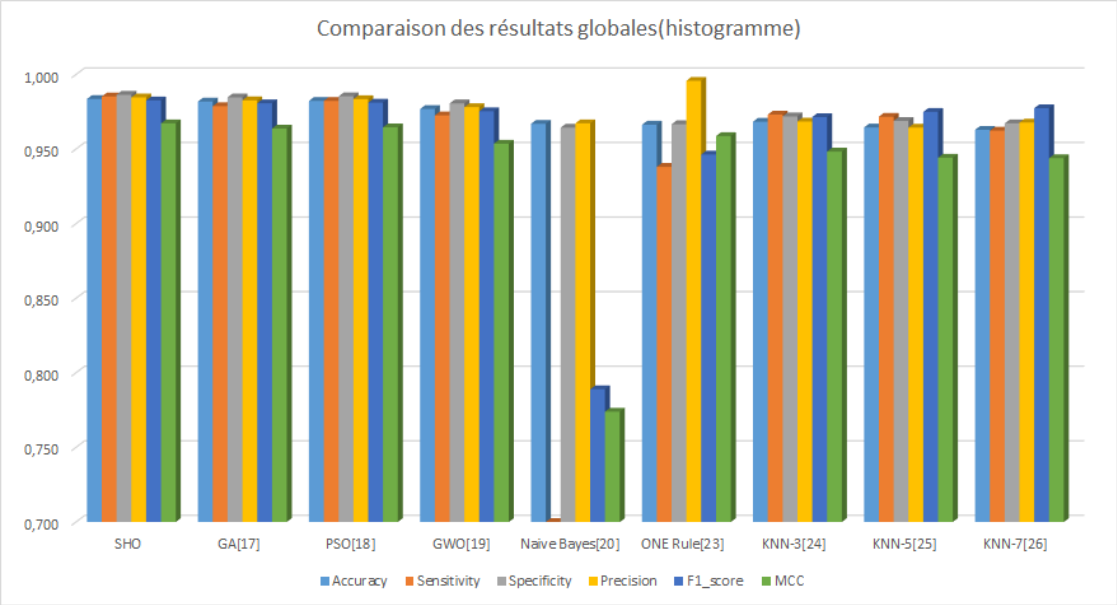

FIGURE 4.7 – *Comparaison des résultats globale (histogramme).*

4.6. CONCLUSION

Dans ce dernier chapitre, nous avons présenté les résultats obtenus à partir de plusieurs expérimentations que nous avons effectué pour arriver au meilleur taux de réussite.

Les comparaisons entre le SHO avec les autres métaheuristiques et les algorithmes de machine Learning ont montré que le SHO peut gérer différents types de contraintes et offrir de meilleures solutions que les autres optimiseurs. Ainsi, ces avantages sont les principales raisons d'utiliser le SHO pour résoudre le problème de classification dans cette étude.

Conclusion générale

De nos jours, la sécurité informatique est quasi-indispensable pour le bon fonctionnement d'un réseau. La sécurité informatique est d'actualité avec la progression du domaine informatique, la sécurité est un enjeu majeur au vue de l'évolution constante de ce domaine mais aussi à l'évolution des attaques contre les systèmes informatiques.

La sécurité absolue n'existe malheureusement pas. Mais certaines précautions telque les systèmes de détection d'intrusion peuvent faire diminuer drastiquement le risque d'avoir un système compromis. Ces systèmes font pour la plupart de l'analyse de trafic (réseau, requêtes) envoyé à un système d'information, et recherchent dans leurs bases de connaissances des éléments identifiant ce trafic comme dangereux.

Nous avons tout au long de notre travail présenté une approche métaheuristique bio-inspirée pour sélectionner les meilleures fonctionnalités de détection des intrusions. Nous avons également analysé les performances d'un classificateur SHO sur la base d'apprentissage NSL KDD. La stratégie proposé SHO a utilisé la méthode inspiré d'une façon aleatoire. Les résultats montrent que notre approche basée sur SHO atteint une meilleur précision de classification, un taux de détection le plus élevé et le taux de fausses alarmes le plus bas des systèmes de détection d'intrusion.

Pour terminer, nous tenons à souligner que nous n'avons nullement pas la prétention d'avoir présenté un travail parfait, car aucun travail scientifique ne peut l'être, ainsi nous laissons le soin à tous ceux qui nous lirons et qui sont du domaine de nous parvenir leurs remarques et suggestions pour l'enrichir et l'améliorer. Pour cela,

Comme perspective nous envisageant dans le futur d'intégrer un service qui prend en compte les informations temporelles et spatiales des connexions réseau en temps réel ; donc, il devrait être plus utile pour l'identification des comportements anormaux en réseau.

perspectives :

Mais En raison du mécanisme exécuté par l'algorithme SHO dans notre mémoire, on peut conclure que cet algorithme présente les lacunes importantes suivantes qui peuvent être surmontées pour atteindre un algorithme SHO plus précis :

- L'algorithme SHO ne recherche qu'autour d'une solution ou d'une hyène tachetée. En d'autres termes, l'algorithme recherche autour de la solution optimale récente une nouvelle hyène tachetée optimale. Cependant, la dernière solution optimale pourrait être une solution optimale locale et pourrait également faire chuter d'autres membres dans les optima locaux.
- Une recherche concentrée autour de la solution optimale actuelle entraîne la recherche d'une partie de l'espace du problème, et les autres parties qui peuvent avoir de meilleures solutions ne sont pas recherchées.

Dans le futur on peut proposer un algorithme suppose que chaque hyène tachetée recherche intelligemment son environnement et prend une meilleure position. En d'autres termes, dans l'algorithme qu'on va proposer prochainement, chaque hyène tachetée peut prendre un certain nombre de positions autour d'elle et se déplacer vers l'une de ces positions qui est la plus optimale, puis passer à la solution optimale globale au fil du temps. De plus, un mécanisme de recherche basé sur une fonction de coût est utilisé pour améliorer l'algorithme SHO.

Table des figures

Bibliographie

[1] Andress, J. (2014). The basics of information security : Understanding the fundamentals of InfoSec in theory and practice. Syngress.

[2] NETWORK SECURITY Network security is any activity designed to protect the usability and integrity(unity) of network and data. It includes both hardware and software technologies.

[3] J.K. Hao, P. Galinier, and M. Habib, âMétaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes,â Revue d'intelligence artificielle, vol. 13, no. 2, pp. 283â 324, 1999.

[4] E.G. Talbi, Metaheuristics : from design to implementation, vol. 74. John Wiley Sons, 2009.

[5] I. Boussaid, Perfectionnement de métaheuristiques pour l'optimisation continue. PhD thesis, Paris Est, 2013.

[6] W. Stallings, Network security essentials : applications and standards. Pearson, 2016.

[7] O. Lopez and F. Picard, Cyberassurance : nouveaux modèles pour quantifier l'impact économique des risques numériques. No. 3, Association d'économie financière, 2019.

[8] J.O. Gerphagnon, M. P. de Albuquerque, and M. P. de Albuquerque, âAttaques informatique,â CBPFNT-007/00,Centre brésilien de recherche physique, Rio de Janeiro â RJ â Brazil, 2004.

[9] S. Specht and R. Lee, âTaxonomies of distributed denial of service networks, attacks, tools and countermeasures,â CEL200303, Princeton University, Princeton, NJ, USA, 2003.

[10] M. S. Hoque, M. Mukit, M. Bikas, A. Naser, et al., âAn implementation of intrusion detection system using genetic algorithm,â arXiv preprint arXiv :1204.1336, 2012.

[11] S. Paliwal and R. Gupta, âDenialofservice, probing remote to user (r2l) attack detection using genetic algorithm,â International Journal of Computer Applications, vol. 60, no. 19, pp. 57â62, 2012.

[12] R. Akimana, Introduction à la sécurité informatique. African Virtual University, 2018

[13] R. Deal, Cisco router firewall security. Cisco Press, 2004

[14] K. Salah, K. Sattar, Z. Baig, M. Sqalli, and P. Calyam, âResiliency of open-source firewalls against remote discovery of lastmatching rules,â in Proceedings of the 2nd International Conference on Security of Information and Networks, SIN '09, (New York, NY, USA), p. 186â192, Association for Computing Machinery, 2009.

[15] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne, âEvaluating computer intrusion detection systems : A survey of common practices,â ACM Comput. Surv., vol. 48, Sept. 2015.

[16] P. Biondi, âArchitecture expérimentale pour la détection d'intrusions dans un système informatique,â Article de recherche,(AvrilSptembre 2001), 2001.

[17] S. Mignault, L'audit de sécurité et la protection des organisations. Université de Montréal, 2009.

[18] G. Hiet, Détection d'intrusions paramétrée par la politique de sécurité grÃœce au contrôle collaboratif des flux d'informations au sein du système d'exploitation et des applications : mise en Åuvre sous Linux pour les programmes Java. Theses, Université Rennes 1, Dec. 2008.

[19] N. Dagorn, Détection et prévention d'intrusion : présentation et limites. Laboratoire Lorrain de Recherche en Informatique et ses Applications(LORIA), 2006. https ://hal.inria.fr/inria00084202.

[20] C. Frédéric, âIds : Intrusion detection systems,â http ://wwwigm.univmlv.fr/dr/XPOSE2004/IDS/IDSSnort.html, 2005.

[21] K. TABIA, âDéveloppement de mécanismes de coopération entre algorithmes d'apprentissage automatique/classification dans un environnement incertain,â Mémoir de magistère, Université Mouloud Mammeri de Tizi ouzou, 2005.

[22] H. Debar, M. Dacier, and A. Wespi, âA revised taxonomy for intrusiondetection systems,â in Annales des télécommunications, vol. 55, pp. 361â378, Springer, 2000.

[23] A. Phillip, âPorras and alfonso valdes âlive traffic analysis of tcp/ip gatewaysâ,â in Proceeding ISOC Symposium on Network and Distributed System Security, San Diego, CA, March1998.

[24] R. Graham, âFaq : Network intrusion detection systems,â http ://www. robertgraham. com/pubs/network-intrusiondetection. html, 2000.

[25] C.HOUASSINE. "Segmentation d'images par une approche biomimétique hybride". Université M'HAMED BOUGARA BOUMERDES (2012).

[26] T. Stéphane, Data mining et statistique décisionnelle : l'intelligence des données. Editions Technip, 2012.

[27] Abbas El Dor. Perfectionnement des algorithmes d'optimisation par essaim particulaire : applications en segmentation d'images et en electronique. Other. Universite Paris-Est, 2012

[28] A ;KHORSI, "La Swarm Intelligence Dans La detection D'intrusion : Application au Loups Sauvages"(2018).

[29] https ://www.gerad.ca/alainh/Hait.pd

[30] [Mohammed, 2014] Mohammed, S. (2014). Traitement d'images par des approches bio-inspirées application à la segmentation d'images.

[31] [Labed, 2013] Labed, S. (2013). Méthodes bio-inpirées hybrides pour la résolution de problèmes complexes. PhD thesis, Thèse Doctorat en Sciences en Informatique, Université Constantine 2, Algérie.

[32] S.Mehdi, "Métaheuristiques pour la manipulation de routages alternatifs en temps réel dans un Job Shop"

[33] D. J. Day, Z. X. Zhao. Protecting against address space layout randomisation (ASLR) compromises and return-to-libc attacks using network intrusion detection systems. International Journal of Automation and Computing, vol. 8, no. 4, pp. 472â483, 2011.

[34] Feature Selection for Support Vector Machines by Means of Genetic Algorithms

[35] Balanced Accuracy for Feature Subset Selection with Genetic Algorithms

[36] A Genetic Algorithm for Feature Selection in a Neuro-Fuzzy OCR System

[37] A Two-step Feature Selection Algorithm Adapting to Intrusion Detection. In : 2009 International Joint Conference on Artificial Intelligence (2009)

[38] Feature Selection for a Fast Speaker Detection System with Neural Networks and Genetic Algorithms

[39] Wang, J.J., Jing, Y.Y., Zhang, C.F. and Zhao, J.H. (2009) Review on Multi-Criteria Decision Analysis Aid in Sustainable Energy Decision-Making.

[40] Guolong C, Qingliang C, Wenzhong G. A PSO-based approach to rule learning in network intrusion detection. Fuzzy Information and Engineering ; 2007 :666e73

[41] Intrusion Detection System with Grey Wolf Optimizer (GWO) Chaima KOUIDRI, Mebarka YAHLALI, Mohammed Amine BOUDIA, Abdelmalek AMINE, Reda Mohamed HAMOU, and Siham KOUIDRI

[42] Salem Benferhat, Tayeb Kenaza, AÃ¯cha Mokhtari. Réseaux Bayésiens naÃ¯fs pour la détection des attaques coordonnées

[43] Mukkamala, S., Janoski, G., Sung, A. (2002). Détection d'intrusion à l'aide de réseaux de neurones et de machines à vecteurs de support. Document présenté à la Conférence conjointe internationale.

[44] Sachin P. Gavhane Department Using Decision Tree Classifiers for Efficient Intrusion Detection System of Computer Engineering V.E.S. Institute of Technology Chembur-74, Mumbai, India

[45] Fukunage K et Narendra PM 1975 Un algorithme de branche et de borne pour calculer les k-plus proches voisins. IEEE Trans.

[46] Altman NS 1992 Une introduction à la régression non paramétrique du noyau et du voisin le plus proche.

[47] Ilany A, Booms AS, Holekamp KE. Topological effects of network structure on long-term social network dynamics in a wild mammal. Ecol Lett 2015

[48] Gaurav Dhiman, Vijay Kumar Spotted hyena optimizer : A novel bio-inspired based metaheuristic technique for engineering applications

[49] 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.

[50] N. Paulauskas and J. Auskalnis, âAnalysis of data preÂprocessing influence on intrusion detection using nslÂkdd dataset,â in 2017 open conference of electrical, electronic and information sciences (eStream), pp. 1â5, IEEE, 2017.

[51] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, âA deep learning approach to network intrusion detection,â IEEE Transactions on Emerging Topics in Computational Intelligence,vol. 2, no. 1, pp. 41â50, 2018.

[52] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, âEnhanced network anomaly detection based on deep neural networks,â IEEE Access, vol. 6,pp. 48231â48246, 2018.

[53] Y. Ding and Y. Zhai, âIntrusion detection system for nslÂkdd dataset using convolutional neural networks,â in Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, CSAI '18, (New York, NY, USA), p. 81â85, Association for Computing Machinery, 2018.

[54] P. Aggarwal and S. K. Sharma, âAnalysis of kdd dataset attributesÂclass wise for intrusion detection,â Procedia Computer Science, vol. 57, pp. 842â851, 2015.

[55] http : //nsl.cs.unb.ca/NSL - KDD

[56] MATALLAH Hocine, Cllassificatiion Automatiique de Textes Approche Oriientée Agent,Thèse de Magister de l'Université de Aboubekr Belkaid Telemcen, 2011.

[57] Alex Martelli. python en concentré. edition O'Reilly, paris, 2004.

[58] https ://docs.python.org/, Consulté le 29/04/2022.

[59] https ://docs.anaconda.com/anaconda/, Consulté le 29/04/2022.

[60] https ://docs.spyder-ide.org/current/index.html, Consulté le 29/04/2022.

[61] https ://jupyter.org/about, Consulté le 29/04/2022.

[62] https ://www.techno-science.net/definition/1470.html, Consulté le 29/04/2022.

[63] https ://pandas.pydata.org/docs/, Consulté le 03/04/2022.

[64] https ://matplotlib.org/, Consulté le 03/04/2022.

[65] https ://numpy.org/doc/stable/user/whatisnumpy.html, Consulté le 03/04/2022.

[66] https ://www.kite.com/python/docs/sklearn, Consulté le 03/04/2022.

[67] MansfieldDevine, The growth and evolution of DDoS October 2015Network Security 2015(10) :13-20

[68] Information Retrieval, 2nd ed. C.J. Van Rijsbergen. London : Butterworths ; 1979

[69] Y. Yang,S. Xie,Y. Cui,W. Lei,X. Zhu,Y. Yang,Y. Yu Effect of replacement of dietary fish meal by meat and bone meal and poultry by-product meal on growth and feed utilization of gibel carp, Carassius auratus gibelio

[70] Fu Y, et al. (2004) Cloning and functional of the Rhizopus oryzae high affinity iron permease (rFTR1) gene. FEMS Microbiol Lett 235(1) :169-76

[71] Williamson, Oliver, E. 2002. "The Theory of the Firm as Governance Structure : From Choice to Contract." Journal of Economic Perspectives, 16 (3) : 171-195

[72] Denning, D.E. (1987) An Intrusion Detection Model. IEEE Transactions on Software Engineering,