République Tunisienne
Ministère de l'Enseignement Supérieur et de la recherche scientifique

Université de Gafsa
Institut Supérieur des Sciences Appliquées et de la Technologie de Gafsa

Cycle de Formation en Mastère Professionnelle dans la Discipline Expert en cyber sécurité

Mémoire de *MASTERE*
Expert Cyber Sécurité

N° d'ordre: 02−ECS

MEMOIRE

Présenté à

L'Institut Supérieur des Sciences Appliquées et de Technologie de Gafsa

(Département Informatique et télécommunication)

En vue de l'obtention Diplôme en

MASTERE

Dans la discipline Expert Cyber Sécurité

Par

TLILI Ramzi

REALISATION D'UN MODELE DE DETECTION D'ATTAQUE DDOS EN SDN A L'AIDE DE L'APPRENTISSAGE AUTOMATIQUE

	Soutenu devant les membres de jury composé de :	
M.	**Wajdi SAADAOUI**	*Président*
Mme	**Rabaa IBRAHMI**	*Rapporteur*
M.	**Ali KOTTI**	*Encadreur*
M.	**Mounir TELLI**	*Co-Encadreur*

A.U : 2022 – 2023

Dédicace

Je dédie ce mémoire à mon Dieu, ainsi qu'à des personnes spéciales qui ont joué un rôle essentiel dans mon parcours :

À mon Dieu,

Tu as été ma lumière, ma force et ma boussole tout au long de cette aventure. Tu m'as guidé, béni et soutenu à chaque étape de ce chemin. Je te suis reconnaissant pour ta grâce infinie et pour avoir été ma source d'inspiration.

À mon père, Hedi,

Tu as été mon roc, ma source d'inspiration et mon plus grand soutien. Tes encouragements constants, ta sagesse et ton amour inconditionnel m'ont donné la force de persévérer dans les moments difficiles. Je suis reconnaissant pour tes précieux conseils et ta présence indéfectible à mes côtés.

À ma mère, Zohra,

Malgré les épreuves que tu as traversées, tu as toujours été là pour moi. Ta force, ta résilience et ton amour inébranlable sont une source d'admiration. Je te dédie ce mémoire en espérant que tu retrouveras rapidement la santé et que tu continueras à être une source d'inspiration pour nous tous.

À mon frère, Mohamed,

Tu es mon compagnon de vie, mon complice et mon meilleur ami. Tes encouragements, ton soutien inconditionnel et ton humour ont illuminé mon chemin. Merci d'avoir toujours cru en moi et de m'avoir poussé à donner le meilleur de moi-même.

À mes sœurs, Amel, Sonia, Soumaya, Sana, Manel, Asma,

Vous êtes mes anges gardiens. Votre présence, votre soutien et votre amour sans limites ont été une véritable bénédiction dans ma vie. Vos sourires, vos câlins et vos mots d'encouragement ont été ma source de réconfort. Je vous dédie ce mémoire avec gratitude et reconnaissance.

À mes amis,

Vous avez été mes compagnons de route, mes confidents et mes partenaires dans cette aventure. Vos encouragements, vos conseils et votre amitié précieuse ont été des piliers sur lesquels je me suis appuyé. Merci d'avoir partagé ces moments inoubliables avec moi.

Enfin, je dédie ce mémoire à tous ceux qui croient en moi, qui m'ont soutenu et encouragé tout au long de ce parcours. Votre confiance et votre soutien ont été ma motivation pour atteindre mes objectifs. Je vous suis reconnaissant du fond du cœur.

Que cette dédicace témoigne de ma gratitude éternelle envers chacune de ces personnes qui ont laissé une empreinte indélébile dans ma vie

Remerciement

Je voudrais commencer par exprimer ma profonde gratitude à Dieu pour Sa guidance et Sa bénédiction tout au long de mon parcours.

Je tiens également à remercier sincèrement mon encadreur, M. KOTTI Ali, ainsi que mon Co-encadreur, M. TELLI Mounir, pour leur précieuse assistance, leurs conseils éclairés et leur soutien inconditionnel. Leur expertise et leur disponibilité ont été d'une valeur inestimable pour la réussite de mon travail.

Je souhaite également exprimer ma reconnaissance envers les membres du jury qui ont généreusement consacré leur temps et leur expertise pour évaluer mon travail. Je suis reconnaissant de leur participation et de l'attention qu'ils ont portée à mon travail.

Enfin, je tiens à remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail. Leur soutien moral, leurs encouragements et leurs conseils ont été d'une grande importance pour moi. Leur présence à mes côtés a été une source de motivation et de réconfort tout au long de ce parcours.

À tous ceux qui ont joué un rôle dans la réalisation de ce travail, je vous exprime ma plus profonde gratitude. Votre soutien et votre contribution ont été essentiels à ma réussite.

Liste des abréviations

IDS : Intrusion Détection System (Système de détection d'intrusion)

SDN : Software-Defined Networking (Réseau défini par logiciel)

DoS : Denial of Service (Déni de service)

DDoS : Distributed Denial of Service (Déni de service distribué)

DNS : Domain Name System (Système de noms de domaine)

IP : Internet Protocol (Protocole Internet)

MAC : Media Access Control (Contrôle d'accès au support)

SQL : Structured Query Language (Langage de requête structuré)

ARP : Address Resolution Protocol (Protocole de résolution d'adresse)

QoS : Quality of Service (Qualité de service)

IPAM : IP Address Management (Gestion des adresses IP)

IA : Intelligence Artificielle

LR : Logistic Regression (Régression Logistique)

K-NN : K-nearest Neighbors (K-plus proches voisins)

SVM : Support Vector Machine (Machine à vecteurs de support)

NB : Naive Bayes

DT : Decision Tree (Arbre de décision)

RF : Random Forest

Liste des figures

Tables des matières

Introduction générale

La sécurité informatique est essentielle dans notre monde numérique en constante évolution. Les entreprises et les organisations dépendent de leurs systèmes informatiques pour stocker des données, gérer leurs opérations et communiquer avec leurs clients. La sécurité informatique vise à protéger ces systèmes contre les menaces et les attaques malveillantes.

Une violation de la sécurité informatique peut entraîner des conséquences graves, telles que des pertes financières, des dommages à la réputation et des atteintes à la vie privée. C'est pourquoi il est crucial pour les entreprises de mettre en place des mesures de sécurité solides pour protéger leurs systèmes, leurs réseaux et leurs données.

En garantissant la sécurité informatique, les entreprises peuvent prévenir les interruptions de service, les vols de données et les autres conséquences néfastes pour leurs activités. Cela leur permet également de maintenir la confiance de leurs clients et de protéger la confidentialité des informations sensibles.

La sécurité informatique est d'une importance primordiale dans notre monde numérique. Les entreprises doivent prendre des mesures pour protéger leurs systèmes et leurs données contre les menaces et les attaques. En investissant dans la sécurité informatique, elles peuvent prévenir les pertes financières, protéger leur réputation et assurer la continuité de leurs activités.

Chapitre 1 : Etat de l'art

Introduction

La sécurité informatique est un domaine essentiel dans le monde numérique d'aujourd'hui. Avec la dépendance croissante aux technologies de l'information, il est crucial de protéger les systèmes, les réseaux et les données contre les menaces et les attaques malveillantes. La sécurité informatique vise à prévenir les violations de la confidentialité, les pertes de données, les interruptions de service et d'autres conséquences néfastes pour les entreprises et les organisations. En mettant en place des mesures de sécurité adéquates, les entreprises peuvent protéger leurs actifs numériques et maintenir leurs opérations en toute confiance.

Ce chapitre se concentre sur la sécurité informatique et les attaques auxquelles les systèmes et les réseaux sont confrontés.

I. Sécurité informatique

La sécurité informatique est essentielle dans notre monde numérique en constante évolution. Avec la prolifération des technologies de l'information et des communications, ainsi que l'accroissement des menaces.

1) Définition

La sécurité informatique protège l'intégrité des technologies de l'information telles que les systèmes informatiques, les réseaux et les données contre les attaques, les dommages ou les accès non autorisés.[1]

2) Les outils de sécurités informatiques

- ✓ Pare-feu (Firewall)

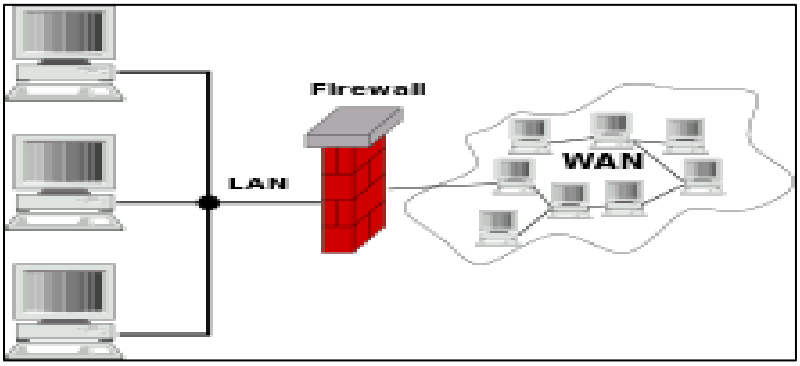

Figure 1 : Pare-feu (Firewall)

Un pare-feu est un dispositif matériel ou logiciel qui surveille et contrôle le trafic réseau en fonction de règles prédéfinies, permettant de bloquer les connexions non autorisées et de protéger le réseau contre les attaques.

Les pare-feux sont des éléments essentiels de la sécurité informatique. Ils agissent comme une barrière de protection entre un réseau interne et des réseaux externes non fiables, tels qu'Internet.[2]

- ✓ **Antivirus et antimalware**

Figure 2: Antivirus et antimalware

Ces logiciels sont conçus pour détecter, prévenir et supprimer les logiciels malveillants, tels que les virus, les vers, les chevaux de Troie et les logiciels espions, afin de protéger les systèmes contre les infections.

- **Outils de chiffrement :**

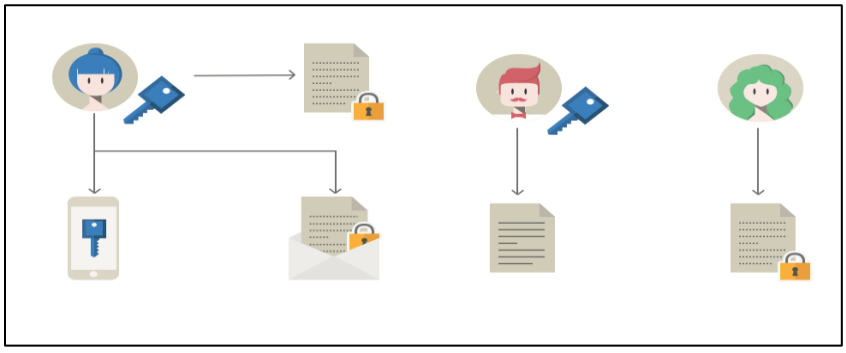

Figure 3 : Les outils de chiffrement

Ils permettent de protéger les données en les rendant illisibles pour toute personne non autorisée. Le chiffrement peut être utilisé pour sécuriser les communications, les fichiers ou les périphériques de stockage.

- **Outils de sauvegarde et de récupération :**

Figure 4 : Sauvegarde et récupération

Les outils de sauvegarde et de récupération sont essentiels pour assurer la protection et la disponibilité des données. Ils permettent de créer des copies de sauvegarde des informations et des systèmes informatiques afin de pouvoir les restaurer en cas de perte ou de dommage. Ces outils sont utilisés pour sauvegarder les fichiers, les bases de données et les systèmes complets, et ils offrent des fonctionnalités telles que la planification des sauvegardes et la vérification de l'intégrité des données. En cas d'incident, tels qu'une défaillance matérielle, une erreur humaine ou une cyberattaque, les données peuvent être récupérées à partir des sauvegardes, minimisant ainsi les pertes et les interruptions des activités.

En résumé, ses outils sont cruciaux pour assurer la protection et la continuité des données et des systèmes informatiques.

3) Les attaques informatiques

- **Définition**

Une attaque informatique fait référence à toute action délibérée visant à compromettre l'intégrité, la confidentialité ou la disponibilité des systèmes informatiques, des réseaux ou des données.

Une « attaque » est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables.[3]

Elle vise à exploiter les vulnérabilités et les faiblesses des systèmes informatiques dans le but de causer des dommages, d'accéder à des informations sensibles, de perturber les opérations normales ou de gagner un avantage illégitime.

- **Les phases d'attaques informatiques**

Une attaque informatique se déroule généralement en plusieurs phases qui permettent aux attaquants d'atteindre leurs objectifs

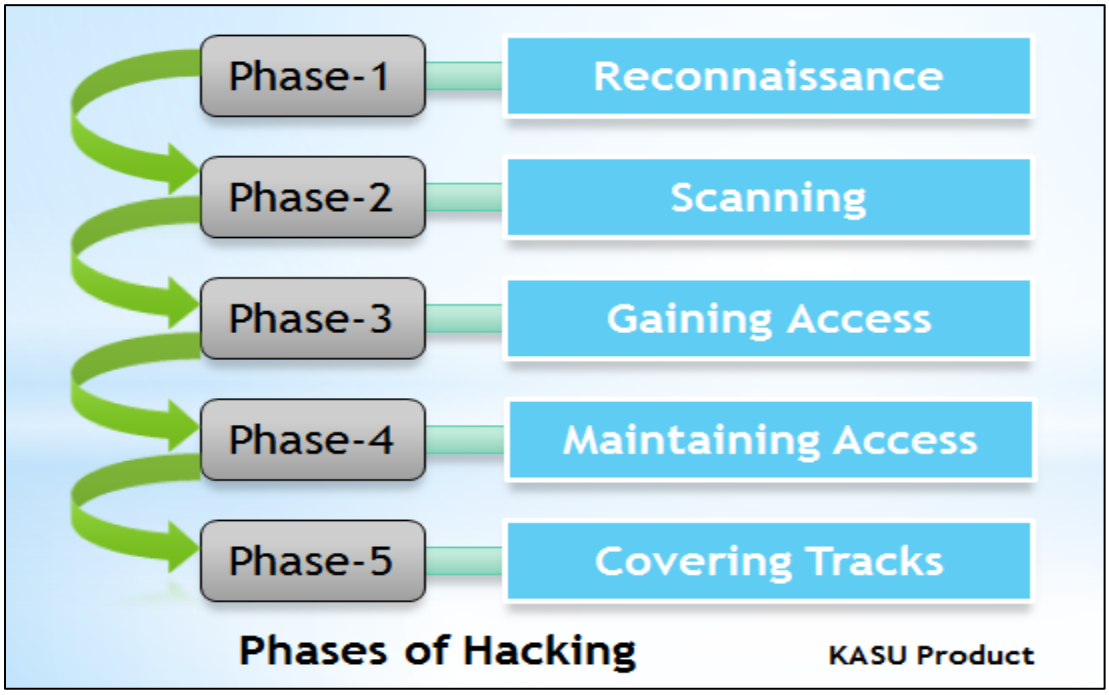

Figure 5 : Les phases d'attaques informatiques

- Reconnaissance : Collecte d'informations sur la cible, tels que les systèmes, les réseaux, les utilisateurs et les vulnérabilités potentielles.
- Scan : Recherche et identification des vulnérabilités sur les systèmes cibles, notamment en analysant les ports ouverts et les services exposés.
- Exploitation : Utilisation de techniques pour exploiter les vulnérabilités identifiées et obtenir un accès non autorisé aux systèmes cibles.
- Maintien de l'accès : Établissement de mécanismes pour maintenir l'accès aux systèmes compromis, tels que l'installation de backdoors ou la création de comptes d'utilisateurs malveillants.

- Effacement des traces : Suppression ou altération des preuves de l'attaque afin de dissimuler l'activité malveillante et de rendre plus difficile la détection et l'attribution.

✓ **Les différents types d'attaques réseaux**

Les attaques réseau sont des actions malveillantes visant à compromettre les systèmes, les protocoles ou les communications d'un réseau, mettant ainsi en danger la confidentialité, l'intégrité et la disponibilité des données. Voici quelques types d'attaques :

- **Attaque d'interception de données :**

Sniffing : L'attaquant utilise des outils d'écoute réseau pour capturer et analyser le trafic en transit, dans le but d'intercepter des données sensibles telles que des mots de passe, des informations d'identification ou des données confidentielles.

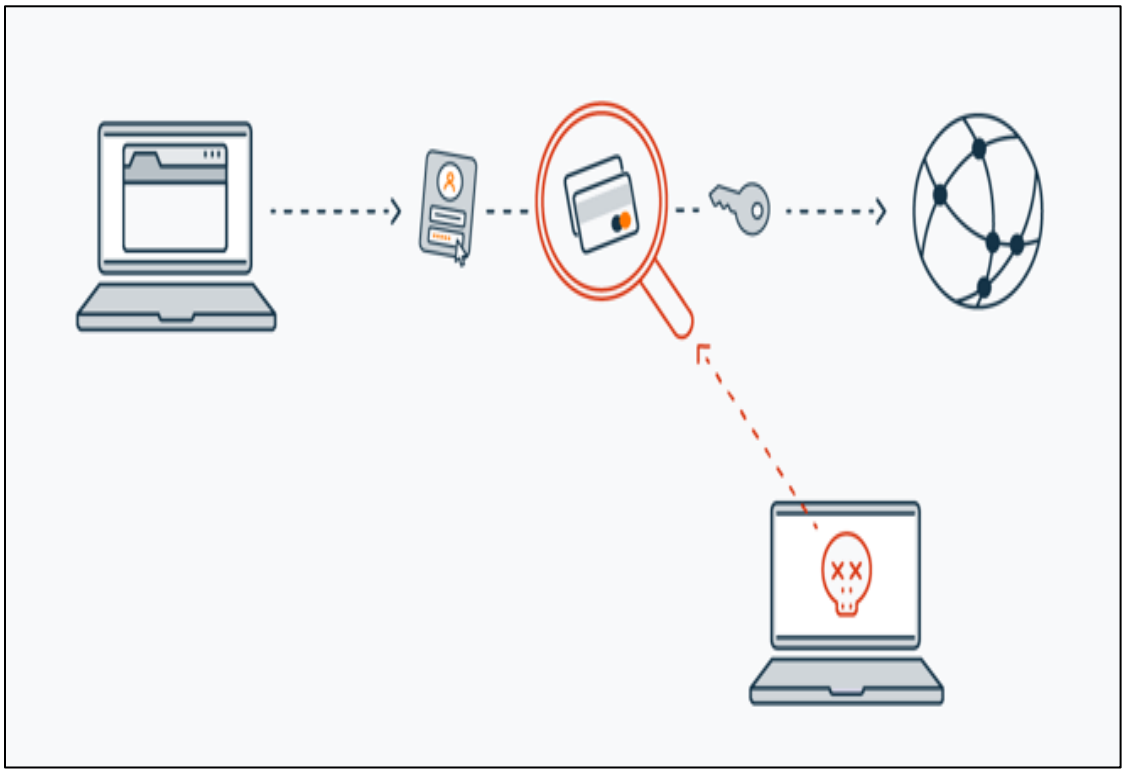

Figure 6 : Attaque d'interception de données

- **Attaques de déni de service (DoS) :**

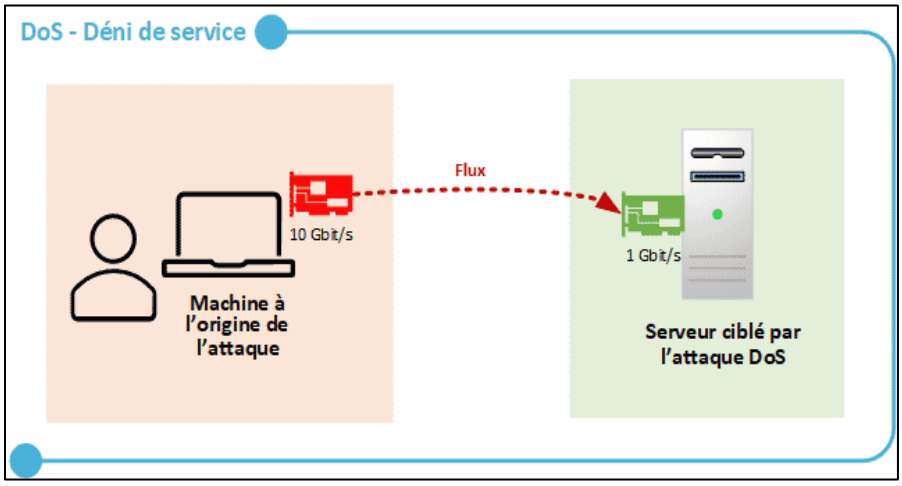

Figure 7 : Attaque DOS

Flooding : L'attaquant inonde un réseau, un serveur ou une application avec une grande quantité de trafic légitime, épuisant ainsi les ressources disponibles et rendant les services indisponibles pour les utilisateurs légitimes.

SYN Flood : L'attaquant envoie une grande quantité de demandes de connexion SYN à un serveur, mais ne finalise jamais la connexion, laissant ainsi des connexions en attente et épuisant les ressources du serveur.[4]

DNS Amplification : L'attaquant exploite les serveurs DNS mal configurés pour envoyer des requêtes DNS falsifiées avec une adresse IP de la cible, ce qui provoque une inondation de réponses DNS vers la cible et surcharge ses ressources.

- **Attaques de déni de service distribué (DDoS)**

Botnet : L'attaquant utilise un réseau de machines compromises (zombies) appelé botnet pour lancer une attaque coordonnée contre une cible. Chaque machine dans le botnet envoie une grande quantité de trafic vers la cible, rendant difficile la mitigation de l'attaque.

Reflective DDoS : L'attaquant exploite des serveurs mal configurés pour renvoyer des réponses à une cible, en utilisant l'adresse IP de la cible comme adresse source. Cela amplifie l'attaque, car la cible reçoit une quantité massive de trafic provenant de multiples sources.[5]

- **Attaques d'injection :**

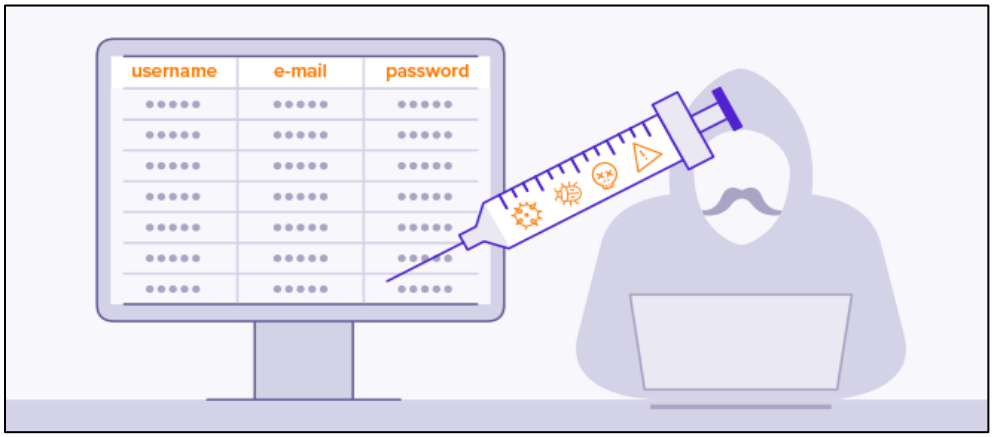

Figure 8 : Attaque d'injection

SQL Injection : L'attaquant insère des commandes SQL malveillantes dans des champs d'entrée d'une application Web pour exécuter des opérations non autorisées sur la base de données sous-jacente.

Command Injection : L'attaquant injecte des commandes système malveillantes dans des champs d'entrée pour exécuter des commandes arbitraires sur le serveur cible.

- **Attaques de spoofing :**

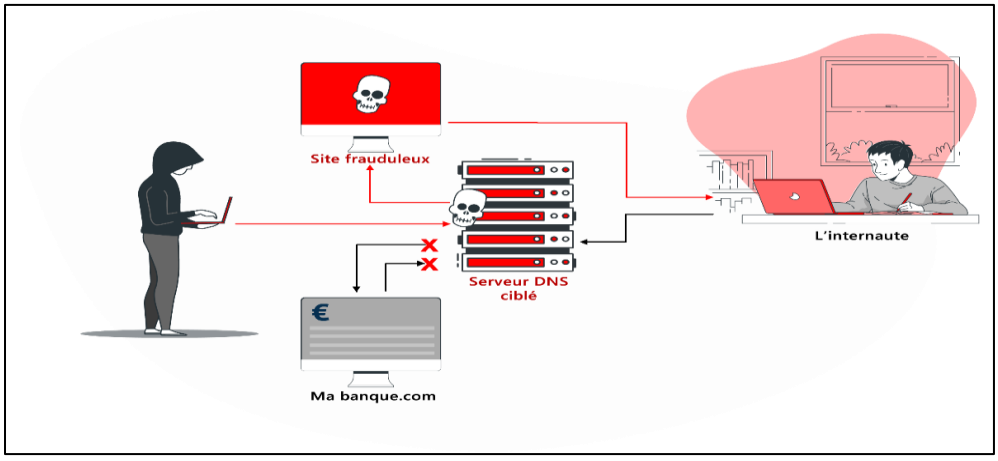

Figure 9 : Attaque spoofing

IP Spoofing : L'attaquant falsifie l'adresse IP source de ses paquets pour masquer son identité ou pour tromper les systèmes de sécurité en les faisant croire que le trafic provient d'une source légitime.

ARP Spoofing : L'attaquant falsifie les tables d'adresse ARP pour associer une adresse IP à une fausse adresse MAC, ce qui lui permet d'intercepter ou de rediriger le trafic réseau.

- **Attaques de déchiffrement :**

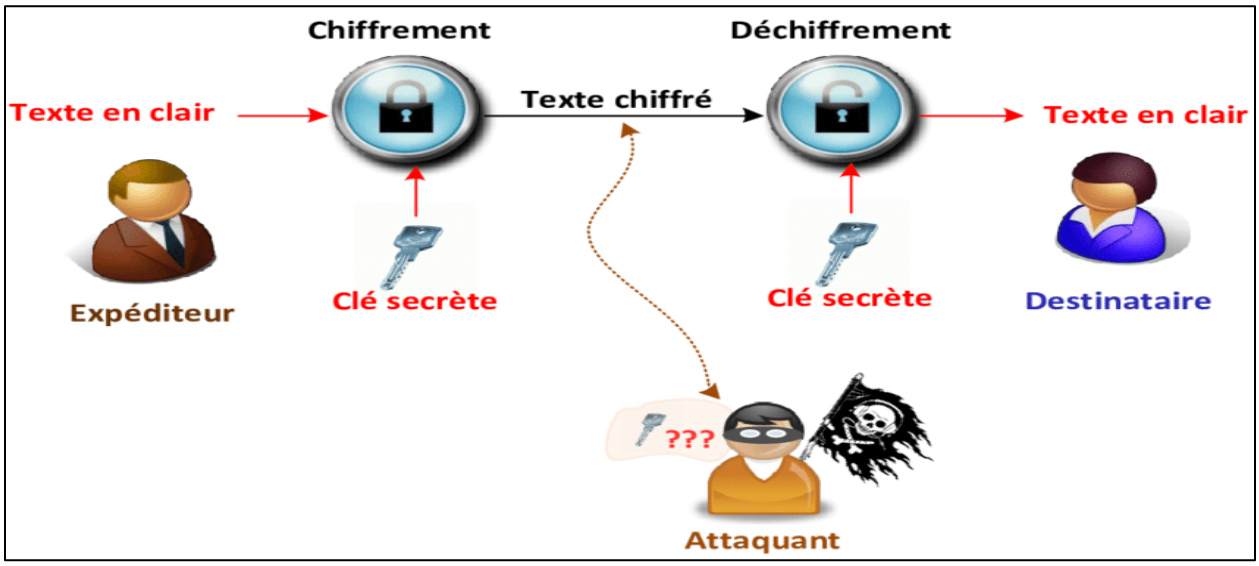

Figure 10 : Attaque de déchiffrement

Brute-Force : L'attaquant essaie toutes les combinaisons possibles de clés pour déchiffrer un message ou accéder à des données protégées.

Attaque par dictionnaire : L'attaquant utilise une liste prédéfinie de mots de passe ou de clés pour tenter de déchiffrer des données

- La sécurité informatique est un élément essentiel pour assurer la protection des systèmes, des réseaux et des données dans un environnement numérique en constante évolution. Les attaques informatiques sont de plus en plus sophistiquées et ciblées, mettant en évidence la nécessité de mettre en place des mesures de sécurité efficaces pour contrer ces menaces.

II. IDS (Intrusion Détection System)

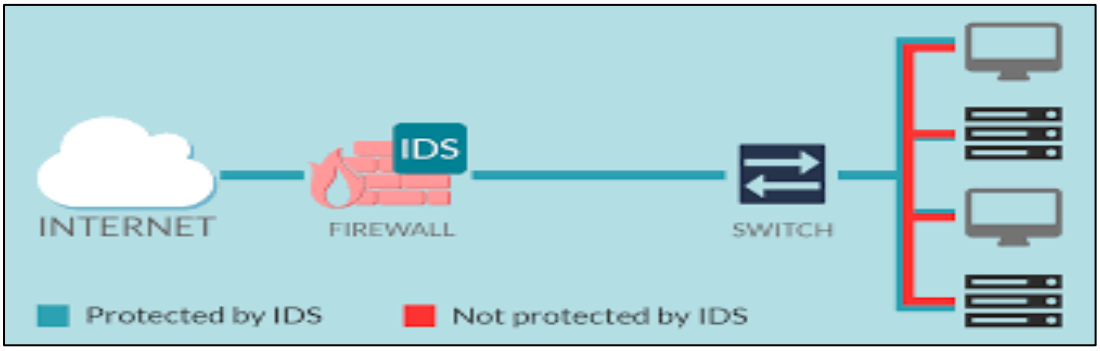

Figure 11 : Intrusion Detection System

1) Définition

Un IDS est un système de détection d'intrusion qui surveille activement les réseaux, les systèmes et les applications pour détecter les tentatives d'intrusion ou les activités malveillantes.[6]

2) Fonctionnement

Les IDS fonctionnent en analysant les données et en cherchant des signes d'activités suspectes ou malveillantes, tels que des connexions de sources inconnues, des changements dans le comportement du système ou des attaques de déni de service (DDoS). Les IDS peuvent également utiliser des signatures ou des règles prédéfinies pour détecter des activités connues.

3) Avantages et limites

✓ **Avantages** :

- Les IDS peuvent détecter les menaces rapidement, ce qui permet aux administrateurs de réponse contre elles avant que des dommages importants ne soient causés.
- Les IDS peuvent être configurés pour répondre automatiquement aux intrusions détectées.
- Les IDS peuvent aider à prévenir les pertes de données potentielles ainsi que les dommages à la réputation de l'entreprise.

✓ **Limites** :

- Les IDS ne sont pas toujours en mesure de détecter toutes les activités malveillantes, surtout si l'attaquant utilise des techniques de piratage avancées et sophistiquées.

- Les IDS ont tendance à générer un grand nombre d'alertes fausses positives, qui peuvent être difficiles à gérer manuellement.
- Les IDS sont souvent complexes à configurer et à maintenir, nécessitant souvent des experts en sécurité informatique pour une utilisation efficace.

En somme, les IDS sont une technologie importante dans la sécurité de l'information, mais doivent être utilisés avec précaution, avec une compréhension approfondie de leur fonctionnement et de leurs limites.

III. SDN

Les réseaux traditionnels sont basés sur une architecture centralisée, où tous les appareils communiquent avec un serveur central. Cependant, cette architecture peut poser des problèmes de fiabilité, de disponibilité et de sécurité. C'est dans ce contexte que l'apparition de la notion de Software-Defined Networking (SDN) a apporté une nouvelle approche dans la gestion des réseaux.[7]

1) Définition

Les réseaux définis par logiciel (SDN - Software-Defined Networking) sont une nouvelle approche des réseaux informatiques qui visent à résoudre ces problèmes. Au lieu d'avoir un contrôle centralisé, les réseaux SDN séparent le plan de contrôle du plan de données. Le plan de contrôle est centralisé et géré par un contrôleur SDN, tandis que le plan de données est distribué et géré par des appareils de réseau.

2) Architecture SDN

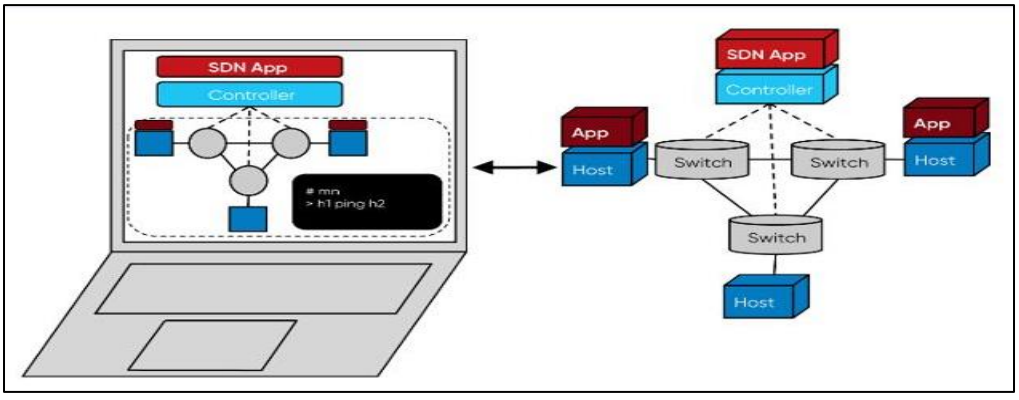

Figure 12 : Architecture SDN

✓ **Applications SDN**

Ces applications relaient les actions et demandent des ressources via le contrôleur SDN à l'aide d'API. Les applications SDN peuvent prendre diverses formes et servir diverses fonctions, telles que la gestion de réseau, la fourniture d'analyses, l'ajout sécurité ou des fonctions de réseau communes. Les exemples incluent la gestion des adresses IP (IPAM), la gestion de la qualité de service (QoS), l'équilibrage de charge ou la détection et l'atténuation d'un déni de service (DoS) cyberattaques.

✓ **Contrôleur SDN**

Les applications SDN envoient des instructions au contrôleur SDN, qui transmet ces informations aux composants de mise en réseau. Le contrôleur SDN collecte également les informations réseau à partir du matériel et les renvoie aux applications.

3) Avantage et limites SDN

✓ **Avantage**

- Gestion centralisée : le SDN permet une gestion centralisée des réseaux, ce qui facilite la configuration, la surveillance et la gestion des réseaux.
- Automatisation : le SDN permet une automatisation accrue des tâches de gestion de réseau, ce qui réduit les erreurs humaines et les coûts de main d'œuvre.
- Flexibilité : le SDN permet de reconfigurer rapidement les réseaux en fonction des besoins des applications, ce qui permet aux entreprises d'adapter rapidement leur infrastructure aux besoins changeants.
- Réduction des coûts : le SDN permet de réduire les coûts en consolidant les équipements réseau, en simplifiant la gestion des réseaux et en automatisant les tâches de gestion.
- Sécurité accrue : le SDN permet une sécurité accrue grâce à la centralisation de la gestion de la sécurité, à la surveillance continue des activités de réseau et à la capacité de limiter l'accès aux ressources réseau.

✓ **Limites**

- Dépendance à l'égard de la programmation : Le SDN nécessite une programmation et une configuration manuelles pour mettre en place et maintenir le réseau. Cela peut être coûteux et complexe, en particulier pour les réseaux de grande envergure.

- Fiabilité : La centralisation du contrôle peut constituer un point de défaillance unique, ce qui rend le réseau vulnérable aux pannes.
- Sécurité : Le SDN peut introduire de nouveaux vecteurs d'attaque, car les attaquants peuvent accéder au contrôleur SDN centralisé et perturber le réseau.
- Performance : Les réseaux SDN peuvent présenter des temps de latence plus longs que les réseaux traditionnels, car les paquets doivent être envoyés au contrôleur pour prendre une décision, puis renvoyés à l'équipement de commutation.

IV. Les attaques DDoS sur les réseaux

1) Définition d'attaque DDOS

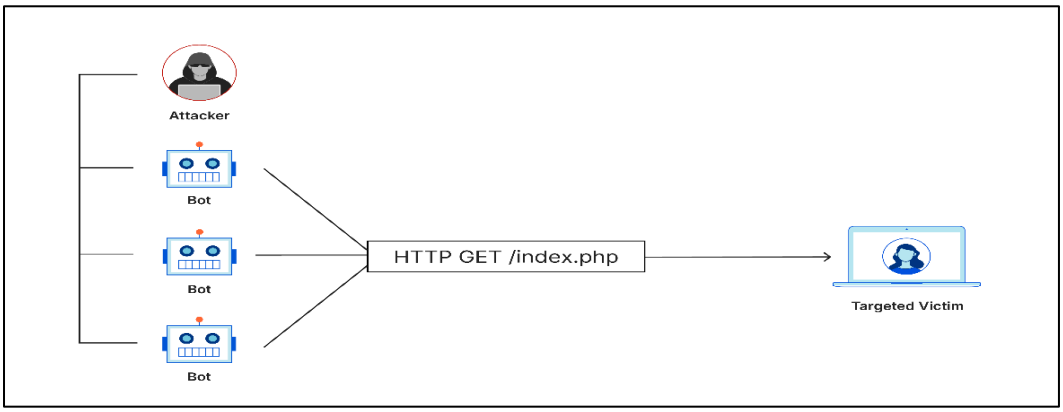

Figure 13 : Attaque DDOS

Une attaque DDoS, ou déni de service distribué, est un type de cyberattaque qui vise à rendre un site Web ou une ressource réseau indisponible en l'inondant de trafic malveillant afin de l'empêcher de fonctionner.

Dans une attaque par déni de service distribué (DDoS), un attaquant submerge sa cible de trafic Internet indésirable, ce qui empêche le trafic légitime d'atteindre sa destination.

D'un point de vue général, une attaque DDoS ou DoS peut être comparée à un embouteillage inattendu causé par des centaines de fausses demandes de covoiturage. Les demandes semblent légitimes pour les services de covoiturage, qui envoient alors des chauffeurs pour récupérer les passagers, ce qui engorge les rues de la ville. Le trafic légitime habituel est ainsi entravé, empêchant les personnes d'arriver à destination.

2) Les types d'attaques DDOS

Contrairement à d'autres attaques de cybercriminels, les attaques DDoS n'essayent pas d'infiltrer un système. Cependant, elles peuvent faire partie d'une attaque plus vaste. Par exemple, lorsque qu'un système est paralysé, les attaques peuvent être utilisées pour distraire les opérateurs de serveurs, les empêchant ainsi de détecter une attaque en cours sur un autre système. Si la réactivité d'un système est retardée en raison d'une attaque DoS ou DDoS, les hackers ont alors la possibilité de modifier les requêtes sur le système surchargé en utilisant des réponses manipulées. Il existe trois catégories de stratégies qui reposent sur ces attaques [3]:

- **La surcharge de la bande passante**
- **Attaques d'épuisement des ressources**

✓ Attaques de surcharge de bande passante (Flood)

Ces attaques visent à submerger la bande passante d'un réseau en envoyant une quantité massive de trafic illégitime. Cela peut être réalisé en utilisant des botnets ou des amplificateurs, ce qui entraîne une congestion du réseau et une réduction de la disponibilité des services.

- **Attaque Smurf** : qui exploite les vulnérabilités du protocole ICMP (Internet Control Message Protocol). L'attaquant envoie des paquets ICMP avec une adresse IP source falsifiée vers un réseau de diffusion ou de multidiffusion. Les routeurs de ce réseau, pensant que les paquets proviennent de la cible réelle, renvoient des réponses ICMP massives à cette dernière. En utilisant une adresse IP falsifiée, l'attaquant amplifie considérablement le volume de trafic, submergeant la bande passante de la cible et la rendant indisponible pour les utilisateurs légitimes. Les attaques Smurf peuvent avoir un impact significatif sur la disponibilité des services en ligne

✓ Attaques d'épuisement des ressources

Ces attaques exploitent les limites de capacité des ressources d'un système, telles que le nombre maximal de connexions simultanées, la mémoire ou la puissance de calcul.

- **HTTP Flood** : Dans cette variante d'attaque de surcharge de ressources DDoS qui est la plus simple, l'attaquant inonde le serveur Web de la cible d'un grand nombre de requêtes HTTP.

Pour cela, il doit juste accéder aux pages du projet cible jusqu'à ce que le serveur s'effondre sous la charge de requêtes.

- **Ping Flood :** Pour ce type d'attaque, les hackers envoient également des paquets ICMP (Internet Control Message Protocol) modifiés, appelés Echo Request. Ces paquets sont généralement envoyés par des botnets à grande échelle. Les conséquences sont un ralentissement du système, voire un blocage du système, car ce dernier doit répondre à ce flux massif de requêtes.
- **SYN Flood :** Le SYN Flood est une attaque visant à provoquer un déni de service et à rendre un réseau indisponible. Elle s'inscrit dans le cadre du protocole TCP (Transmission Control Protocol), et son objectif est de submerger le serveur cible de requêtes SYN (Synchronized) en masse. Normalement, une connexion TCP entre le serveur et un client nécessite un échange de messages en trois étapes : SYN, SYN-ACK et ACK. Le client qui souhaite se connecter au serveur envoie d'abord un paquet SYN. En réponse à cette requête, le serveur lui envoie un message SYN-ACK (Synchronized Acknowledgment), et le client doit finalement envoyer une réponse ACK (Acknowledgment) pour établir la connexion de manière définitive. Si la dernière étape n'est pas réalisée, le système sera paralysé, car le serveur n'a pas reçu de confirmation finale de la connexion et mettra du temps à libérer les ressources, générant un temps d'attente. Si un grand nombre de connexions semi-ouvertes sans message ACK apparaissent, il y a un risque de surcharge et les ressources du serveur peuvent être épuisées.
- **UDP Flood :** Avec ce type d'attaque, les hackers utilisent la connexion UDP (User Datagram Protocol). Contrairement à la transmission TCP, les données peuvent être transférées via UDP sans nécessiter une connexion établie au préalable. Dans le cas des attaques DoS et DDoS, des paquets UDP sont envoyés à des ports aléatoires sur le système ciblé. Le système tente alors en vain de déterminer quelles applications attendent les données transférées, ce qui entraîne l'envoi de paquets ICMP à l'expéditeur avec le message "Destination inaccessible". Si un système reçoit un grand nombre de requêtes de ce type, cela entraîne une surcharge des ressources et entraîne une indisponibilité temporaire pour les utilisateurs. [8]

Conclusion

En conclusion, la sécurité informatique est une préoccupation majeure dans le contexte actuel, où les attaques sont de plus en plus sophistiquées et fréquentes. Les entreprises doivent prendre des mesures proactives pour protéger leurs systèmes et leurs données sensibles contre les attaques malveillantes. Le déploiement de technologies émergentes telles que le Software-Defined Networking (SDN) offre de nouvelles possibilités pour renforcer la sécurité des réseaux d'entreprise. Cependant, il est important de rester vigilant face aux nouvelles menaces et d'adopter des approches basées sur l'intelligence artificielle pour détecter et atténuer les attaques de manière proactive.

Chapitre 2 : Les stratégies de détection d'attaque

Introduction

Les attaques informatiques, y compris les attaques DDoS (Distributed Denial of Service), représentent une menace sérieuse pour la disponibilité et l'intégrité des systèmes en ligne. Les attaques DDoS visent à submerger un système cible en inondant son réseau ou ses ressources avec un trafic excessif, ce qui entraîne une dégradation ou une interruption du service pour les utilisateurs légitimes.

Face à ces attaques, les organisations doivent mettre en place des stratégies de détection efficaces pour identifier rapidement les activités malveillantes et prendre des mesures d'atténuation appropriées. Les stratégies de détection d'attaques DDoS sont conçues pour surveiller en continu le trafic réseau, analyser les modèles et les comportements suspects, et détecter les signes caractéristiques des attaques DDoS.

I. Intelligence artificielle

L'intelligence artificielle (IA) est un vaste domaine de l'informatique qui se concentre sur la création de machines intelligentes capables d'accomplir des tâches qui nécessitent généralement l'intelligence humaine. L'IA est une discipline interdisciplinaire qui utilise diverses approches, et les avancées en matière d'apprentissage automatique et d'apprentissage en profondeur ont particulièrement suscité un changement de paradigme dans presque tous les secteurs de l'industrie technologique.

L'intelligence artificielle permet aux machines de modéliser, voire d'améliorer, les capacités de l'esprit humain. Des développements tels que les voitures autonomes et la prolifération d'assistants intelligents comme Siri et Alexa font de l'IA une partie de plus en plus intégrée de la vie quotidienne. C'est également un domaine dans lequel les entreprises de toutes les industries investissent massivement.[9]

Figure 14 : Intelligence artificielle

II. Machine Learning

1) Définition

Le Machine Learning, ou apprentissage automatique, est un domaine de l'intelligence artificielle qui se concentre sur le développement de techniques et d'algorithmes permettant aux ordinateurs d'apprendre et de s'améliorer à partir de données, sans être explicitement programmés. Il est largement utilisé dans de nombreux domaines tels que la reconnaissance d'image, la classification de texte, la prédiction, la recommandation et bien d'autres.

2) Les types d'apprentissage automatique

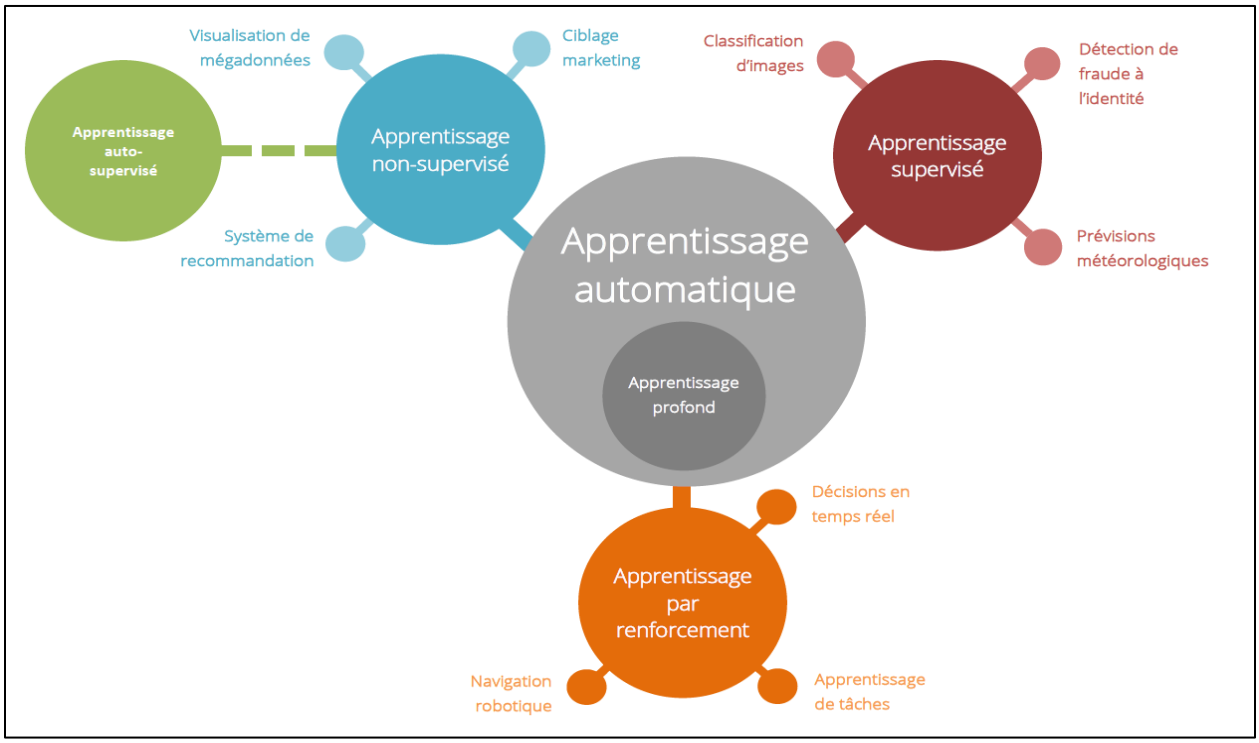

Figure 15 : Les types d'apprentissage automatique

✓ Apprentissage supervisé

Avec l'apprentissage supervisé, la machine peut apprendre à faire une certaine tâche en étudiant des exemples de cette tâche. Par exemple, elle peut apprendre à reconnaître une photo de chien après avoir été exposée à des millions de photos de chiens. Ou bien, elle peut apprendre à traduire le français en chinois après avoir vu des millions d'exemples de traduction français-chinois.

D’une manière générale, la machine peut apprendre une relation $f : x \rightarrow y$ qui relie x à y en analysant des millions d’exemples d’associations $x \rightarrow y$. [5]

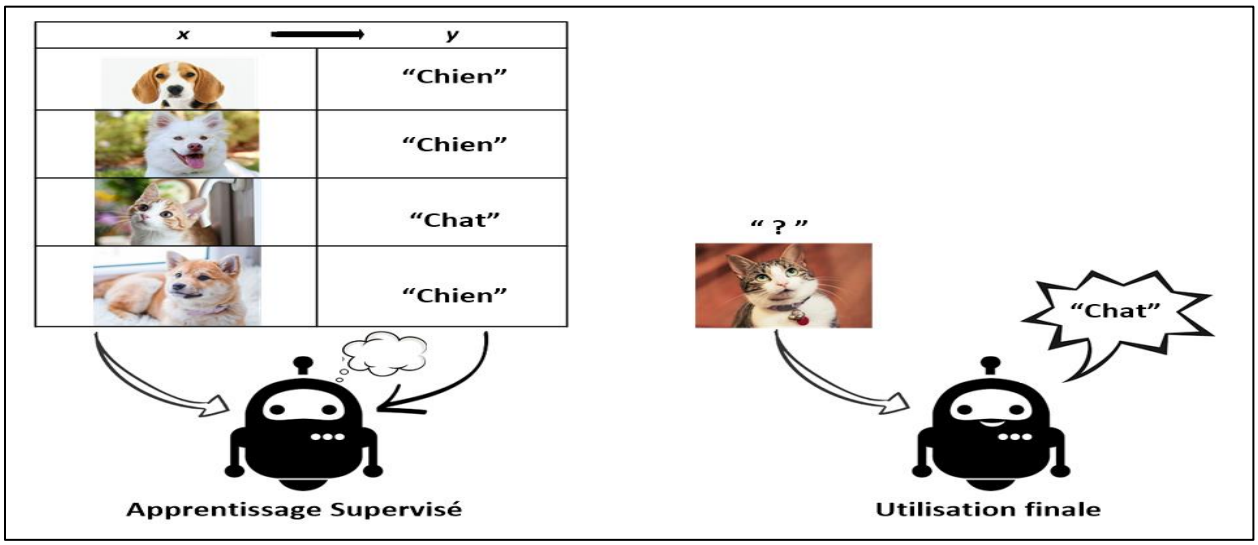

Figure 16 : Apprentissage supervisé

- ✓ **Apprentissage non supervisé**

L'apprentissage non supervisé est une technique d'apprentissage automatique où l'algorithme analyse un ensemble de données sans disposer d'étiquettes ou de réponses préalables. L'objectif de l'apprentissage non supervisé est de découvrir des structures, des patterns ou des regroupements intrinsèques dans les données.

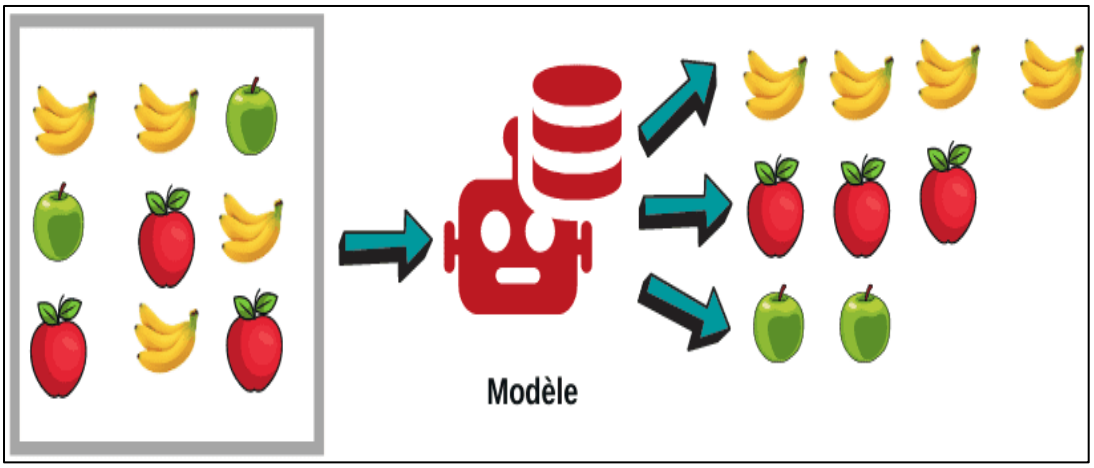

Figure 17 : Apprentissage non supervise

✓ **Apprentissage par renforcement**

L'apprentissage par renforcement est une forme d'apprentissage automatique où un modèle d'intelligence artificielle est entraîné à prendre les meilleures décisions dans un environnement donné. Le modèle apprend en interagissant avec cet environnement, en prenant des actions et en évitant les erreurs. Par exemple, l'apprentissage par renforcement peut être utilisé pour entraîner un modèle à jouer aux échecs ou au poker, où il apprend en essayant différentes actions, en évitant les erreurs et en améliorant ses performances en ajustant ses décisions. Cette méthode repose sur l'utilisation de récompenses et de sanctions pour encourager le modèle à prendre les bonnes décisions et à éviter les erreurs.

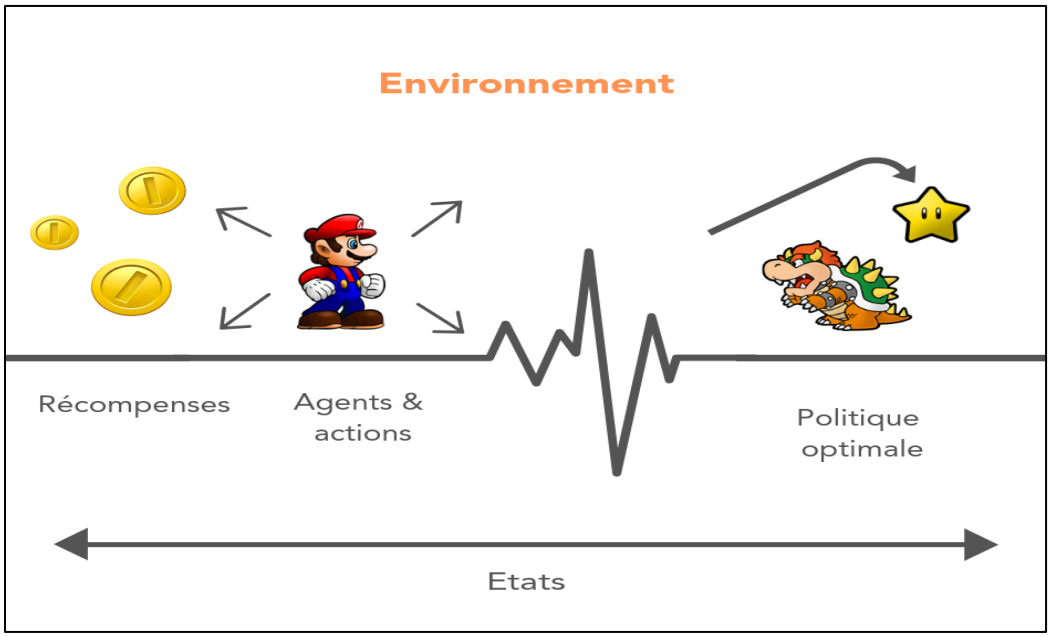

Figure 18 : Apprentissage par renforcement

3) Les algorithmes Machine Learning

Un algorithme de Machine Learning est un ensemble de procédures et de règles mathématiques appliquées aux données afin de permettre à une machine d'apprendre et d'évoluer sans être explicitement programmée. En d'autres termes, il permet à un ordinateur de tirer des conclusions à partir d'un ensemble de données d'entraînement, puis de généraliser ces conclusions pour de nouvelles données. Le Machine Learning est une branche de l'intelligence artificielle qui permet

aux ordinateurs de simuler l'apprentissage humain et de prendre des décisions de manière autonome.

- ✓ **LR (Artificial Logistic Regression)**

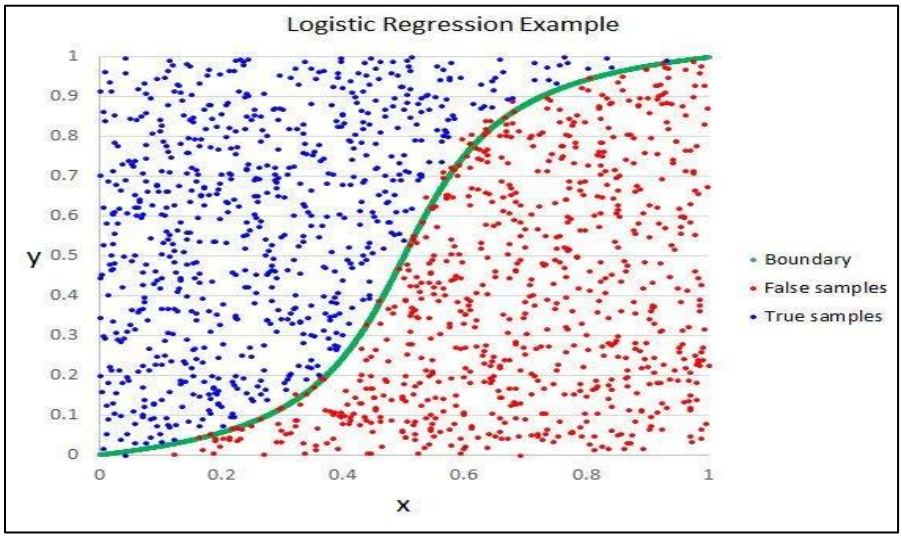

Figure 19 : algorithme Logistic Regression

L'algorithme Logistic Regression est une méthode de classification largement utilisée pour prédire une variable binaire (deux valeurs distinctes). Comme son nom l'indique, il s'agit d'une régression qui utilise une fonction logistique pour produire des résultats entre 0 et 1. Ce type d'algorithme est souvent utilisé en machine Learning pour résoudre des problèmes de classification tels que la détection de spam et la prédiction d'une réponse positive ou négative à une question. Il existe deux types de Logistic Regression :

- ✓ La Binary Logistic Regression : est utilisée pour prédire une variable binaire
- ✓ La Multinomial Logistic Regression : est utilisée pour prédire une variable catégorique ayant plus de deux expressions possibles.

Le principe de l'algorithme Logistic Regression repose sur une analyse de régression logistique à deux états qui consiste à déterminer la probabilité d'appartenance d'un individu à une classe. Pour cela, on utilise une formule mathématique appelée fonction de régression logistique qui prend en compte les caractéristiques (ou variables) de chaque individu et leur associe un poids (ou coefficient) qui indique l'influence de chaque variable sur la réponse. La régression logistique peut également être utilisée pour résoudre des problèmes de régression (où la variable à prédire est continue plutôt que binaire). Dans ce cas, l'algorithme est appelé

régression logistique multiple et utilise plusieurs variables d'entrée pour prédire une sortie continue. Pour appliquer l'algorithme Logistic Regression, il est nécessaire de diviser les données en deux ensembles : un ensemble d'entraînement et un ensemble de test. L'ensemble d'entraînement est utilisé pour apprendre le modèle, tandis que l'ensemble de test permet de vérifier si le modèle prédit correctement les résultats pour les nouvelles données. La Logistic Regression est un algorithme relativement simple à comprendre et à mettre en œuvre, mais il peut être sensible aux données manquantes ou erronées. Afin d'obtenir des résultats fiables, il est important de bien comprendre les caractéristiques des variables utilisées et de s'assurer que les données utilisées pour entraîner le modèle sont cohérentes et représentatives du problème à résoudre.

En conclusion, l'algorithme Logistic Regression est un outil efficace pour résoudre des problèmes de classification et de régression. Il est largement utilisé en machine Learning et peut être appliqué à une variété de problèmes différents. Cependant, il est important de bien comprendre les caractéristiques du problème à résoudre et de s'assurer que les données et les variables utilisées sont cohérentes et représentatives pour obtenir des résultats fiables.

- ✓ **K-NN (K-nearest Neighbors)**

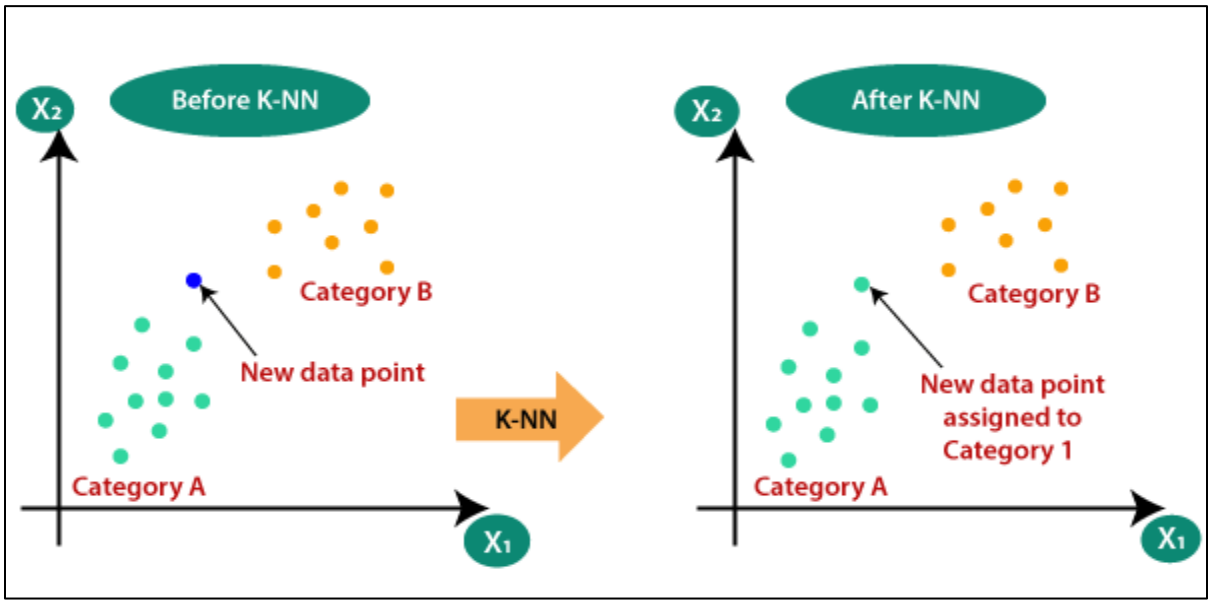

Figure 20 : **Algorithme *K-NN***

Dans l'apprentissage automatique, K-Nearest Neighbors (K-NN) est un algorithme de classification répandu qui est souvent utilisé comme référence dans la classification de modèles plus complexes. L'algorithme K-NN est une méthode populaire parmi les chercheurs et les scientifiques des données. Il est également souvent utilisé dans les systèmes de recommandation, les systèmes de reconnaissance de texte et d'image, ainsi que dans d'autres tâches de classification.

L'algorithme K-NN fonctionne en comparant l'exemple de test à l'ensemble des exemples d'entraînement les plus proches. Les exemples les plus proches sont déterminés par mesure de distance. L'algorithme calcule la distance entre l'exemple de test et chaque exemple d'entraînement, puis sélectionne les K exemples les plus proches. La valeur de K est un paramètre déterminé avant l'exécution de l'algorithme. Une valeur typique pour K est 5. Une fois que les K exemples les plus proches ont été sélectionnés, l'algorithme K-NN classe l'exemple de test en sélectionnant la classe majoritaire parmi ces exemples.

Par exemple, si les 5 exemples les plus proches d'un nouvel exemple sont étiquetés « chien » et 4 exemples sont étiquetés « chat », l'algorithme K-NN classera le nouvel exemple comme un chien.

- ✓ **Avantages**
    - Il est simple à comprendre et à utiliser.
    - Il peut être utilisé pour des données de n'importe quelle taille, y compris des ensembles de données volumineux.
    - Il peut être utilisé pour des problèmes de classification multiclasse.
- ✓ **Les limites**
    - Il peut être lent pour des ensembles de données volumineux.
    - L'exactitude de la classification dépend de la qualité de l'ensemble des exemples d'entraînement.
    - Si le nombre de classes est grand, l'algorithme peut être moins efficace.
    - Le choix d'une valeur de K optimale peut être difficile et dépend des données.

En conclusion l'algorithme K-NN est une méthode simple et populaire utilisée dans l'apprentissage automatique pour la classification et la prédiction. Il utilise une approche non-paramétrique, ce qui signifie qu'il ne suppose pas une distribution de probabilité sous-jacente pour les données d'entraînement et ne nécessite pas de phase d'apprentissage formelle. Cette méthode peut être

appliquée à des problèmes de classification multi classe et peut être utilisée pour des données de toutes tailles. Cependant, le choix d'une valeur de K optimale peut être difficile et dépend des données.

- ✓ **SVM (Support Vector Machine) :**

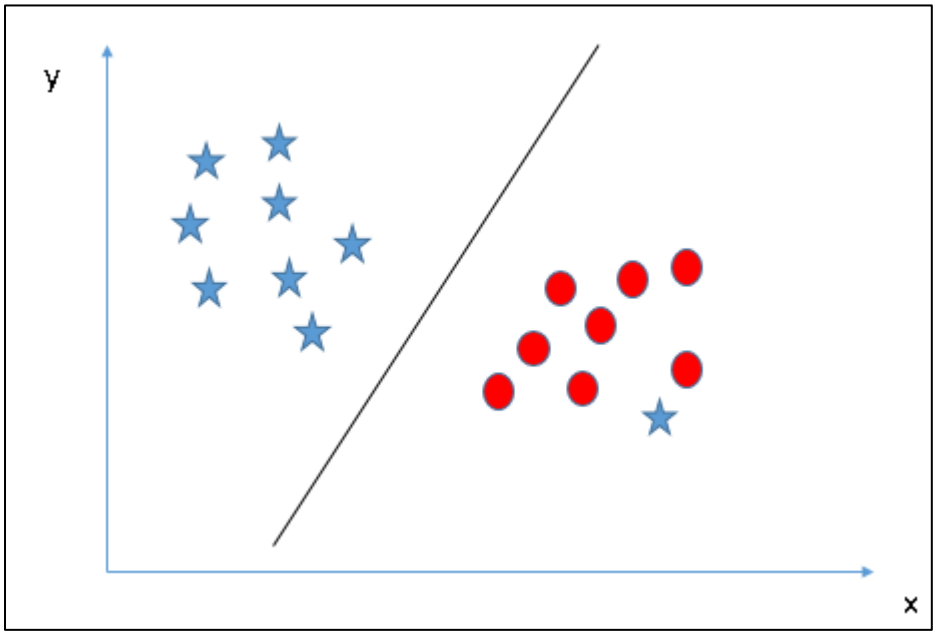

Figure 21 : Algorithme Support Vector Machine

La machine à vecteurs de support est un algorithme d'apprentissage automatique utilisé pour la classification et la régression. Il fonctionne en trouvant la meilleure ligne ou le meilleur plan qui sépare les données en deux classes, ou en trouvant la meilleure fonction qui relie les variables indépendantes aux variables dépendantes.

Un SVM commence par diviser un ensemble de données en deux classes. Il cherche ensuite la meilleure ligne ou le meilleur plan qui sépare ces deux classes de manière à maximiser la marge entre les deux. La marge est définie comme la distance entre la ligne ou le plan et les points les plus proches de chaque classe.

- ➢ **Avantages**
  - Ils sont efficaces dans les espaces de grande dimension.
  - Ils sont relativement peu sensibles au surapprentissage.
  - Ils sont capables de gérer des données à grande échelle.

- **Limites**
    - Ils sont sensibles à la pertinence des paramètres d'initialisation.
    - Ils ne sont pas très efficaces pour le traitement de données très bruyantes ou mal structurées.
    - Ils ne sont pas très efficaces pour le traitement de données très grandes ou très communes.

En conclusion, les SVM sont un algorithme d'apprentissage automatique puissant et polyvalent qui peut être utilisé pour résoudre une grande variété de problèmes de classification et de régression. Cependant, comme pour tout algorithme, il y a des avantages et des inconvénients, et les utilisateurs doivent être conscients de ces facteurs lors de la sélection de l'algorithme à utiliser pour leur travail.

- ✓ **NB (Naive Bayes) :**

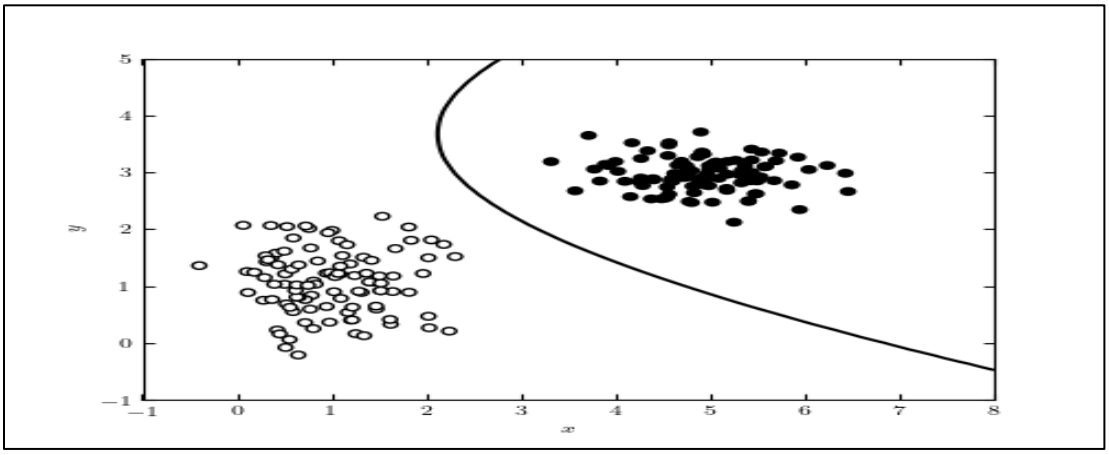

Figure 22 : Algorithme Naive Bayes

Naive Bayes est un algorithme simple mais puissant d'apprentissage automatique. Il est souvent utilisé pour la classification de texte en raison de sa grande précision et de sa rapidité d'exécution. L'algorithme de Naive Bayes est basé sur le théorème de Bayes, qui est une méthode de calcul de la probabilité conditionnelle. Il suppose également que toutes les caractéristiques sont indépendantes les unes des autres, d'où le terme "naïf".

Dans le contexte de la classification de texte, chaque document est représenté par un vecteur de caractéristiques, telles que la fréquence des mots clés, des termes spécifiques et des balises.

L'algorithme calcule ensuite les probabilités conditionnelles de chaque classe (par exemple, "spam" ou "non-spam") en fonction de ces caractéristiques.

L'algorithme est entraîné sur un ensemble de données d'entraînement avec des étiquettes de classe connues. Il utilise ces données d'entraînement pour ajuster les paramètres du modèle statistique. Pour classer un nouveau document, l'algorithme calcule les probabilités conditionnelles de chaque classe pour ce document, puis sélectionne la classe avec la probabilité la plus élevée.

L'algorithme Naive Bayes est souvent utilisé dans des domaines tels que la détection de spam, la classification de documents, l'analyse de sentiments et la reconnaissance de la parole. Il est apprécié pour sa grande précision et sa rapidité d'exécution, même sur de grands ensembles de données.

Cependant, il peut ne pas fonctionner aussi bien sur des données avec des caractéristiques fortement corrélées ou des données qui ne sont pas bien réparties dans l'espace des caractéristiques

- ✓ **RF (Random Forest)**

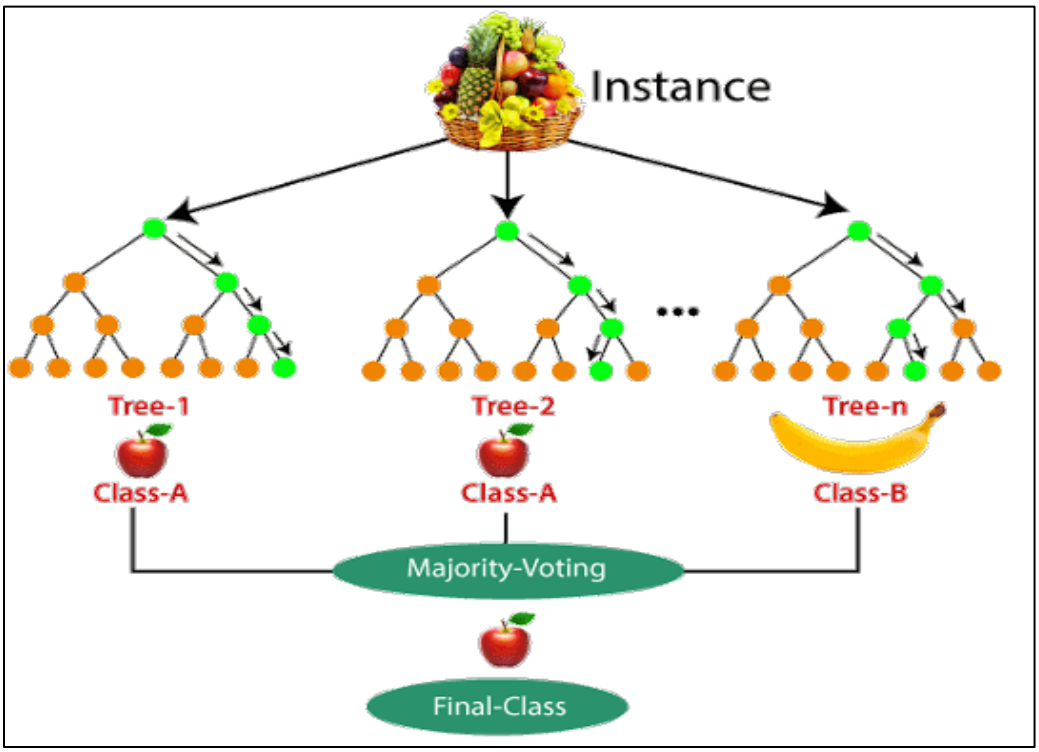

Figure 23 : Algorithme Random Forest

C’est un algorithme d'apprentissage automatique qui combine plusieurs arbres de décision pour créer un modèle plus robuste et précis.

Les arbres de décision sont des modèles qui prennent une série de décisions en fonction des caractéristiques d'une observation pour aboutir à une prédiction. Cependant, les arbres de décision peuvent être instables et sujet à sur-apprentissage si le nombre de variables est important.

Avec Random Forest, on construit plusieurs arbres de décision indépendants les uns des autres, chacun étant entraîné sur un sous-ensemble aléatoire des observations et sur un sous-ensemble aléatoire des variables. Les arbres individuels sont ensuite combinés en prenant la moyenne de leurs prédictions pour fournir une prédiction finale plus précise et robuste.

- **Avantages**
    - Capacité à gérer des données qui ont une forte corrélation entre les variables via sa méthode de sélection aléatoire de variables dans chaque arbre de décision.
    - Peut être utilisé pour la classification ou la régression, et il est particulièrement utile pour les données bruyantes.
- **Limites**
    - Contrairement aux arbres de décision traditionnels, Random Forest n'a pas besoin de prétraiter les données (supprimer des informations, modifier les données...) pour être efficace.

En conclusion, Random Forest est un algorithme d'apprentissage automatique qui combine de multiples arbres de décision pour produire un modèle plus précis et fiable. Il est utile pour le traitement des données bruyantes, des données qui ont de forte corrélation entre les variables et des données non prétraitées.

✓ **DT (Decision Tree)**

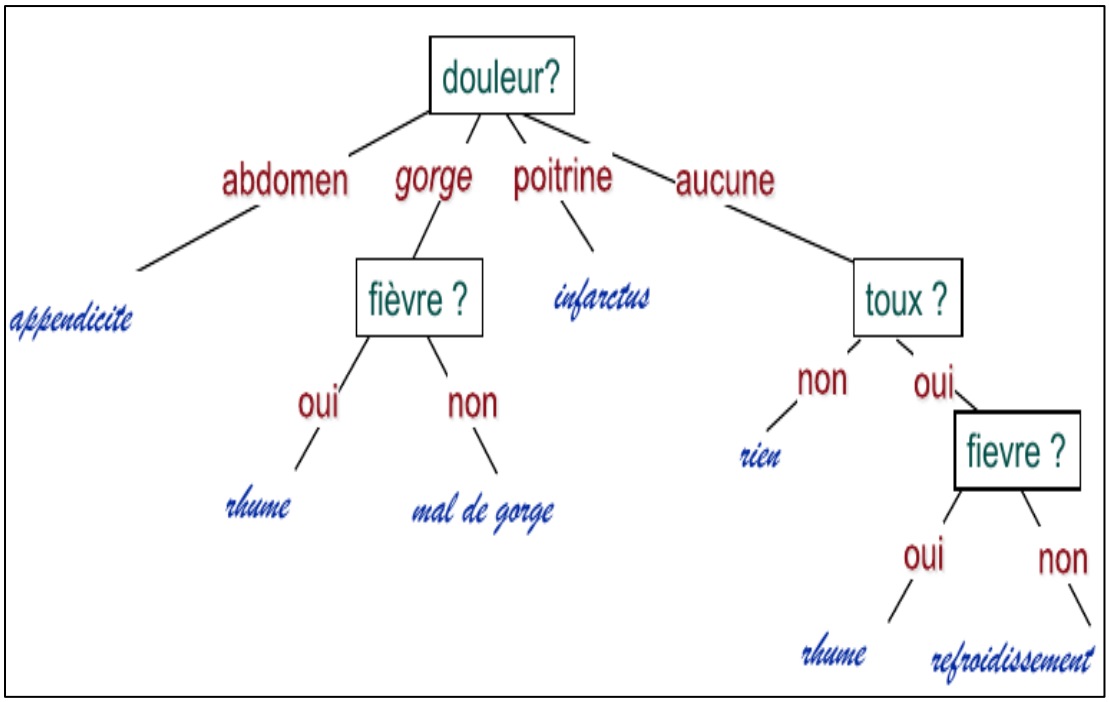

Figure 24 : algorithme Decision Tree

Un arbre de décision, ou "decision tree" en anglais, est une méthode de modélisation prédictive et d'analyse de données qui utilise un graphique structuré en forme d'arbre pour représenter un ensemble de résultats possibles auxquels on peut arriver en faisant une série de choix.

Cette méthode est couramment utilisée dans les domaines de la finance, de la gestion de la qualité, des soins de santé, de la recherche opérationnelle, de la surveillance des conditions atmosphériques, de la technologie et de l'apprentissage automatique.

Le processus de création d'un arbre de décision comprend plusieurs étapes :

Tout d'abord, les données sont collectées et analysées pour identifier les variables et les caractéristiques pertinentes pour l'analyse.

Ensuite, les données sont divisées en ensembles d'entraînement et de test, qui sont utilisés pour construire et valider le modèle. Le modèle est construit en utilisant une séquence d'opérateurs de

décomposition qui séparent les données en sous-ensembles, en fonction des réponses à des questions spécifiques. Ces sous-ensembles sont séparés en nœuds ou "nodes" qui représentent chacun une décision ou une catégorie.

Une fois que l'arbre de décision est construit, il est utilisé pour prédire les résultats futurs en classant les données d'entrée dans les nœuds appropriés en utilisant les "branches" de l'arbre. Les résultats sont ensuite présentés sous forme de diagrammes ou de tables qui montrent la probabilité des résultats.

a) Avantages

- La possibilité d'analyser des données complexes
- La capacité d'identifier des relations et des tendances cachées
- La facilité de compréhension et d'interprétation des résultats.

b) Limites

- Vulnérables à des biais dans les données d'entrée
- Surajustement ou le sur-apprentissage du modèle.

En Conclusion, l'arbre de décision est donc une méthode largement utilisée dans le domaine de l'analyse de données et de la modélisation prédictive, et il peut être utilisé dans de nombreux contextes différents pour aider à prendre des décisions informées et à prédire les résultats futurs.

III. Les réseaux de neurones

1) Définition

Un réseau de neurones artificiels ou Neural Network est un système informatique s'inspirant du fonctionnement du cerveau humain pour apprendre. [11]

Les réseaux de neurones sont des modèles d'apprentissage automatique qui utilisent des neurones artificiels interconnectés pour effectuer des calculs et apprendre à partir des données afin de résoudre des tâches complexes.

2) Utilité des réseaux de neurones

Les réseaux de neurones sont utilisés dans une variété de domaines, y compris la reconnaissance d'images, la traduction automatique, la reconnaissance vocale et la classification de données.

- ✓ **Apprentissage en profondeur pour la reconnaissance d'images :**

L'apprentissage en profondeur, notamment les réseaux de neurones convolutifs, a révolutionné la reconnaissance d'images. Les réseaux de neurones profonds peuvent extraire des caractéristiques visuelles complexes à partir d'images, permettant ainsi une classification précise. Cette technologie a des applications dans de nombreux domaines, notamment la sécurité, la médecine, la robotique et la surveillance.

- ✓ **Apprentissage par renforcement profond pour les jeux et la robotique :**

L'apprentissage par renforcement profond est utilisé pour entraîner des agents autonomes à prendre des décisions dans des environnements complexes. Cette technique est particulièrement efficace dans les jeux et la robotique, où les agents peuvent apprendre à jouer à des jeux ou à naviguer dans des environnements sans intervention humaine. Les réseaux de neurones profonds sont utilisés pour représenter la politique de l'agent, permettant ainsi des performances supérieures.

- ✓ **Réseaux de neurones convolutifs pour la reconnaissance de la parole :**

Les réseaux de neurones convolutifs sont largement utilisés pour la reconnaissance de la parole. Ces réseaux peuvent extraire des caractéristiques acoustiques à partir de signaux audios et les utiliser pour identifier des mots ou des phrases. Les réseaux de neurones convolutifs sont également utilisés pour la transcription automatique de la parole et l'assistance vocale.

IV. Dataset

1) Définition

Un data set (ou jeu de données ou ensemble de données) est un ensemble de données numérisées (statistiques, textes, images, son, vidéo…). C'est l'ensemble de données que l'on fournit aux algorithmes de machine Learning pour que ce dernier crée des « modèles », c'est-à-dire des équations statistiques qui permettent à l'algorithme de réaliser des prévisions sur les futures ventes, sur le comportement d'un consommateur, d'identifier le contenu d'une photo, d'une vidéo, de traduire automatiquement un texte ou d'anticiper une panne matérielle ou un défaut de paiement d'un débiteur, par exemple.

2) Les sources les plus populaires pour trouver des datasets

Il existe de nombreux jeux de données disponibles pour l'apprentissage automatique (machine Learning), chacun adapté à des tâches spécifiques telles que la classification, la prédiction, la segmentation...

Voici quelques-unes des sources les plus populaires pour trouver des jeux de données de Machine Learning :

- ✓ **Kaggle :**

Kaggle est une plateforme en ligne de science des données qui propose des compétitions de machine Learning et de nombreuses ressources pour l'apprentissage automatique, y compris des jeux de données.

- ✓ **UCI Machine Learning Repository :**

Cette archive publique contient de nombreux jeux de données de machine learning, avec des descriptions détaillées et des informations sur les tâches correspondantes.

- ✓ **Google Dataset Search :**

Cette plateforme de recherche de Google permet de rechercher des jeux de données de machine learning provenant de différentes sources, y compris des universités, des gouvernements et des organisations à but non lucratif.

- ✓ **OpenML** :

OpenML est une plateforme en ligne de partage de données de Machine Learning, qui permet aux utilisateurs de télécharger, partager et explorer des jeux de données.

- ✓ **AWS Open Data :**

Amazon Web Services (AWS) propose un large éventail de jeux de données gratuits et publics dans divers domaines, y compris la science, la finance, la santé, ...

V. Comparaison des solutions de détection d'attaque DDOS

Lorsqu'il s'agit de détecter les attaques DDoS (Distributed Denial of Service), il existe deux méthodes principales :

- ✓ L'ancienne méthode qui repose sur des algorithmes de détection basés sur des règles prédéfinies
- ✓ Nouvelle méthode qui utilise l'intelligence artificielle (IA) pour détecter les attaques.

L'ancienne méthode de détection d'attaques DDoS utilise des systèmes de détection d'intrusion (IDS) et des pare-feux qui recherchent des modèles spécifiques de trafic réseau pour détecter les attaques. Ces systèmes fonctionnent en examinant les paquets de données entrants et sortants, comparant les modèles de trafic à une base de données de règles préétablies pour identifier les attaques. Les IDS et les pare-feux génèrent des alertes lorsqu'une activité suspecte est détectée.

L'intelligence artificielle (IA) utilise des algorithmes d'apprentissage automatique pour analyser les données de trafic réseau en temps réel et détecter les attaques DDoS. À mesure que les algorithmes apprennent les modèles de trafic normaux, ils peuvent détecter les anomalies qui indiquent une attaque DDoS en cours. Les systèmes d'IA sont plus sophistiqués que les systèmes de détection basés sur des règles, car ils peuvent détecter des attaques complexes, telles que les attaques cachées et les attaques de couches multiples. Contrairement aux systèmes de détection basés sur des règles, les systèmes d'IA ne nécessitent pas de configuration manuelle ou de mises à jour régulières de la base de données de règles. Les systèmes d'IA peuvent également surveiller en permanence le trafic réseau et s'adapter rapidement aux nouveaux modèles de trafic.

En conclusion, la détection d'attaques DDoS à l'aide de l'IA est plus efficace et plus rapide que la méthode traditionnelle basée sur des règles. Bien que les systèmes de détection basés sur des règles soient utiles pour détecter les attaques les plus simples, ils ne suffisent pas à détecter les attaques sophistiquées. Les systèmes d'IA offrent une capacité de détection plus avancée, ce qui permet aux organisations de mieux se protéger contre les attaques DDoS.

Conclusion

L’intelligence artificielle joue un rôle crucial dans le domaine de la sécurité informatique. Grâce à ses capacités d'apprentissage automatique et d'analyse avancée, elle permet de détecter et de prévenir efficacement les attaques et les menaces en constante évolution. L'utilisation de l'intelligence artificielle permet d'améliorer la détection des comportements anormaux, d'identifier les schémas d'attaques complexes et de prendre des mesures proactives pour renforcer la sécurité. En outre, l'IA permet également d'automatiser certaines tâches de sécurité, réduisant ainsi la charge de travail pour les professionnels de la sécurité. En combinant les connaissances humaines avec les capacités de l'IA, nous pouvons renforcer la sécurité des systèmes informatiques et protéger les entreprises contre les menaces numériques croissantes.

Chapitre 3 : Expérimentations

Introduction

Ce chapitre se concentre sur la classification du trafic d'attaque DDoS en utilisant plusieurs algorithmes d'apprentissage automatique. L'objectif est de développer des modèles capables de détecter et de catégoriser les attaques DDoS à partir de l'analyse du trafic réseau. En renforçant la sécurité informatique, cette approche vise à protéger les entreprises contre les interruptions de service et les pertes financières, en identifiant rapidement les attaques et en prenant les mesures appropriées pour minimiser les conséquences néfastes.

I. Environnement de travail

1) Choix de logiciel

Figure 25 : Logo google colab

Google Colab est un logiciel de développement et d'exécution de code basé sur Jupyter Notebook, offrant un accès gratuit à des ressources de calcul puissantes dans le cloud. Il est facile à utiliser, permettant d'écrire, d'exécuter et de visualiser le code Python de manière interactive. Grâce à sa compatibilité avec des bibliothèques populaires telles que TensorFlow et PyTorch, il est idéal pour les projets d'apprentissage automatique et de science des données. En étant accessible depuis n'importe quel appareil connecté à Internet, il facilite le partage de travaux et favorise la collaboration. En résumé, Google Colab offre une plateforme pratique et efficace pour développer et exécuter du code, en mettant à disposition des ressources de calcul puissantes et en facilitant le travail collaboratif dans le domaine de l'apprentissage automatique et de l'analyse de données.

2) Choix du langage de programmation

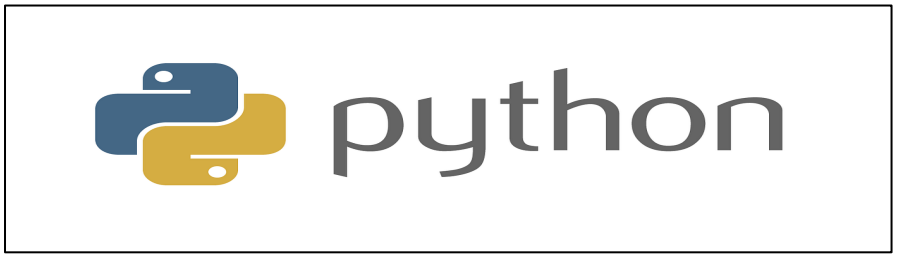

Figure 26 : Logo Python.

Python est un langage de programmation populaire et polyvalent, apprécié pour sa simplicité et sa lisibilité. Il est souvent utilisé pour développer des applications, des sites web, des scripts et des analyses de données. Python se distingue par sa syntaxe claire et intuitive, ce qui facilite l'apprentissage pour les débutants en programmation. Il offre également une grande variété de bibliothèques et de frameworks qui permettent d'étendre ses fonctionnalités et de développer des projets complexes. Que ce soit pour automatiser des tâches, créer des interfaces utilisateur ou analyser des données, Python est un choix populaire en raison de sa simplicité et de sa polyvalence.

Python joue un rôle clé dans science des données tâches et est utilisé pour effectuer des calculs statistiques complexes, visualiser des données et créer algorithmes d'apprentissage automatique.[12]

3) Choix du dataset CIC-DDoS2019

Le choix du dataset CIC-DDoS2019 pour mon projet de détection des attaques DDoS peut avoir plusieurs avantages :

- ✓ Fiabilité : Le CIC-DDoS2019 est un dataset fiable et précis car il a été collecté à partir de vraies attaques DDoS enregistrées dans un environnement contrôlé. Cela garantit que les données sont pertinentes et correspondant à la réalité.
- ✓ Taille du dataset : Le CIC-DDoS2019 contient un grand nombre d'instances qui peuvent être utilisées pour entraîner mon modèle de détection.
- ✓ Variété : Le dataset CIC-DDoS2019 présente une grande variété de types d'attaques DDoS, de protocoles de communication réseau et de flux de trafic. Cela garantit que le modèle est capable de détecter différents types d'attaques et de trafic.
- ✓ Accessibilité : Le CIC-DDoS2019 a été publié gratuitement pour une utilisation publique.

4) Outils et bibliothèques utilisés

- ✓ datetime : permet de manipuler des dates et heures en Python.
- ✓ matplotlib : bibliothèque permet de créer des graphiques et des visualisations en Python.
- ✓ numpy : cette bibliothèque est largement utilisée en tant que base pour la manipulation de tableaux multidimensionnels en Python
- ✓ pandas : bibliothèque utilisée pour manipuler et analyser des données en Python.
- ✓ train_test_split : cette fonction de la bibliothèque "sklearn.model_selection" permet de diviser les données en ensembles d'entraînement et de test. Cette fonction est utilisée pour préparer les données pour l'entraînement et l'évaluation des différents algorithmes de classification.
- ✓ LogisticRegression : classe de la bibliothèque "sklearn.linear_model" est utilisée pour implémenter les modèles de régression logistique.
- ✓ KNeighborsClassifier : classe de la bibliothèque "sklearn.neighbors" est utilisée pour implémenter les algorithmes de classification K-NN.
- ✓ SVC : cette classe de la bibliothèque "sklearn.svm" est utilisée pour implémenter les machines à vecteurs de support. Cette méthode est largement utilisée pour les problèmes de classification.
- ✓ GaussianNB : cette classe de la bibliothèque "sklearn.naive_bayes" est utilisée pour implémenter l'algorithme de classification naïve bayésienne gaussienne.
- ✓ DecisionTreeClassifier : cette classe de la bibliothèque "sklearn.tree" est utilisée pour implémenter les arbres de décision pour les problèmes de classification.
- ✓ RandomForestClassifier : classe de la bibliothèque "sklearn.ensemble" est utilisée pour implémenter les forêts d'arbres aléatoires pour les problèmes de classification.
- ✓ confusion_matrix : cette fonction de la bibliothèque "sklearn.metrics" permet de calculer la matrice de confusion pour évaluer les performances d'un algorithme de classification.
- ✓ accuracy_score : cette fonction de la bibliothèque "sklearn.metrics" permet de calculer le score d'exactitude d'un modèle de classification.

II. Préparation du code

1) Importation des bibliothèques

Figure 27 : Importation des bibliothèques

Cette étape du code importe les bibliothèques nécessaires et les méthodes pour l'analyse des données, la construction des modèles et l'évaluation des performances.

2) Préparations des données

```
class MachineLearning():

    def __init__(self):

        print("Loading dataset ...")

        self.counter = 0

        self.flow_dataset = pd.read_csv('/content/drive/MyDrive/memoire/cicddos2019_dataset.csv')
        self.flow_dataset.iloc[:, 2] = self.flow_dataset.iloc[:, 2].astype(str)
        self.flow_dataset.iloc[:, 2] = self.flow_dataset.iloc[:, 2].str.replace(',', '')
        self.flow_dataset.iloc[:, 3] = self.flow_dataset.iloc[:, 3].astype(str)
        self.flow_dataset.iloc[:, 3] = self.flow_dataset.iloc[:, 3].str.replace(',', '')
        self.flow_dataset.iloc[:, 5] = self.flow_dataset.iloc[:, 5].astype(str)
        self.flow_dataset.iloc[:, 5] = self.flow_dataset.iloc[:, 5].str.replace(',', '')


        self.X_flow = self.flow_dataset.select_dtypes(exclude=['object']).iloc[:, :-1].values
        self.X_flow = self.X_flow.astype('float64')

        self.y_flow = self.flow_dataset.iloc[:, -1].values

        self.X_flow_train, self.X_flow_test, self.y_flow_train, self.y_flow_test = train_test_split(self.X_flow, self.y_flow, test_size=0.25, random_state=0)
```

Figure 28 : Préparation des données et division en ensembles d'entraînement et de test

Cette partie du code effectue le chargement du jeu de données, réalise des opérations de prétraitement des données telles que la conversion de types et la manipulation des colonnes, puis divise les données en ensembles d'entraînement et de test. Cette préparation des données est essentielle pour l'entraînement et l'évaluation des modèles d'apprentissage automatique ultérieurs.

3) Performances des différents algorithmes de classification sur un jeu de données

```
def LR(self):

    print("------------------------------------------------------------------------------")
    print("Logistic Regression ...")

    self.classifier = LogisticRegression(solver='liblinear', random_state=0)
    self.Confusion_matrix()

def KNN(self):

    print("------------------------------------------------------------------------------")
    print("K-NEAREST NEIGHBORS ...")

    self.classifier = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
    self.Confusion_matrix()

def SVM(self):

    print("------------------------------------------------------------------------------")
    print("SUPPORT-VECTOR MACHINE ...")

    self.classifier = SVC(kernel='rbf', random_state=0)
    self.Confusion_matrix()

def NB(self):

    print("------------------------------------------------------------------------------")
    print("NAIVE-BAYES ...")

    self.classifier = GaussianNB()
    self.Confusion_matrix()


def DT(self):

    print("------------------------------------------------------------------------------")
    print("DECISION TREE ...")

    self.classifier = DecisionTreeClassifier(criterion='entropy', random_state=0)
    self.Confusion_matrix()

def RF(self):

    print("------------------------------------------------------------------------------")
    print("RANDOM FOREST ...")

    self.classifier = RandomForestClassifier(n_estimators=10, criterion="entropy", random_state=0)
    self.Confusion_matrix()
```

Figure 29 : Performances des différents algorithmes

Le code présente une analyse des performances de différents algorithmes de classification, tels que la régression logistique (LR), les k-plus proches voisins (KNN), les machines à vecteurs de support (SVM), le naïve Bayes (NB), l'arbre de décision (DT) et la forêt aléatoire (RF). Chaque algorithme

est exécuté et évalué en utilisant une matrice de confusion pour mesurer la précision de la classification. Les résultats obtenus permettent de comparer les performances de ces algorithmes et d'identifier celui qui convient le mieux pour résoudre le problème de classification donné.

4) Calcul de la matrice de confusion et évaluation des performances du modèle

```
def Confusion_matrix(self):
    self.counter += 1

    self.flow_model = self.classifier.fit(self.X_flow_train, self.y_flow_train)

    self.y_flow_pred = self.flow_model.predict(self.X_flow_test)

    print("------------------------------------------------------------------------------")

    print("confusion matrix")
    cm = confusion_matrix(self.y_flow_test, self.y_flow_pred)
    print(cm)

    acc = accuracy_score(self.y_flow_test, self.y_flow_pred)

    print("succes accuracy = {0:.2f} %".format(acc*100))
    fail = 1.0 - acc
    print("fail accuracy = {0:.2f} %".format(fail*100))
    print("------------------------------------------------------------------------------")
```

Figure 30 : Calcule matrice de confusion et évaluation des performances

Ses lignes du code calculent la matrice de confusion en utilisant les prédictions du modèle de machine Learning. Elle affiche ensuite la matrice de confusion, le taux de succès et d'échec du modèle.

5) Visualisation des résultats des algorithmes

```
x = ['TP','FP','FN','TN']
x_indexes = np.arange(len(x))
width = 0.10
plt.xticks(ticks=x_indexes, labels=x)
plt.title("Résultats des algorithmes")
plt.xlabel('Classe predite')
plt.ylabel('Nombre de flux')
plt.tight_layout()
plt.style.use("seaborn-darkgrid")
# plt.style.use("dark_background")
# plt.style.use("ggplot")
if self.counter == 1:
    y1 = [cm[0][0],cm[0][1],cm[1][0],cm[1][1]]
    plt.bar(x_indexes-2*width,y1, width=width, color="#1b7021", label='LR')
    plt.legend()
if self.counter == 2:
    y2 = [cm[0][0],cm[0][1],cm[1][0],cm[1][1]]
    plt.bar(x_indexes-width,y2, width=width, color="#e46e6e", label='KNN')
    plt.legend()
if self.counter == 3:
    y3 = [cm[0][0],cm[0][1],cm[1][0],cm[1][1]]
    plt.bar(x_indexes,y3, width=width, color="#0000ff", label='NB')
    plt.legend()
if self.counter == 4:
    y4 = [cm[0][0],cm[0][1],cm[1][0],cm[1][1]]
    plt.bar(x_indexes+width,y4, width=width, color="#e0d692", label='DT')
    plt.legend()
if self.counter == 5:
    y5 = [cm[0][0],cm[0][1],cm[1][0],cm[1][1]]
    plt.bar(x_indexes+2*width,y5, width=width, color="#000000", label='RF')
    plt.legend()
    plt.show()
```

Figure 31 : Traçage des diagrammes

Ses lignes du code tracent un diagramme à barres représentant les résultats des algorithmes. Elle utilise les valeurs de la matrice de confusion pour chaque algorithme et les affiche sur un graphique avec des étiquettes et des légendes appropriées. Enfin, elle affiche le diagramme à barres à l'aide de la fonction plt.show().

6) Entraînement et prédiction des modèles

```
def main():

    start_script = datetime.now()
    ml = MachineLearning()
    start = datetime.now()
    ml.LR()
    end = datetime.now()
    print("LEARNING and PREDICTING Time: ", (end-start))

    start = datetime.now()
    ml.KNN()
    end = datetime.now()
    print("LEARNING and PREDICTING Time: ", (end-start))

    #start = datetime.now()
    #ml.SVM()
    #end = datetime.now()
    #print("LEARNING and PREDICTING Time: ", (end-start))

    start = datetime.now()
    ml.NB()
    end = datetime.now()
    print("LEARNING and PREDICTING Time: ", (end-start))

    start = datetime.now()
    ml.DT()
    end = datetime.now()
    print("LEARNING and PREDICTING Time: ", (end-start))

    start = datetime.now()
    ml.RF()
    end = datetime.now()
    print("LEARNING and PREDICTING Time: ", (end-start))
```

Figure 32 : Exécution des algorithmes d'apprentissage automatique

Cette partie du code exécute les algorithmes d'apprentissage automatique et mesure leur temps d'exécution. Cela permet d'obtenir des informations sur les performances des modèles et d'optimiser le choix de l'algorithme en fonction de la durée d'exécution requise.

7) Mesure du temps d'exécution du script

```
    end_script = datetime.now()
    print("Script Time: ", (end_script-start_script))

if __name__ == "__main__":
    main()
```

Figure 33 : Mesure du temps d'exécution du script

Cette partie du code est dédiée à la mesure du temps d'exécution du script. Elle utilise la fonction datetime.now() pour enregistrer le moment de début et de fin du script, puis calcule la différence pour obtenir le temps total d'exécution. Ce temps est ensuite affiché à l'écran en tant que "Script Time". Cette partie permet d'évaluer la performance globale du script et de vérifier si le temps d'exécution est raisonnable

III. Analyse des résultats

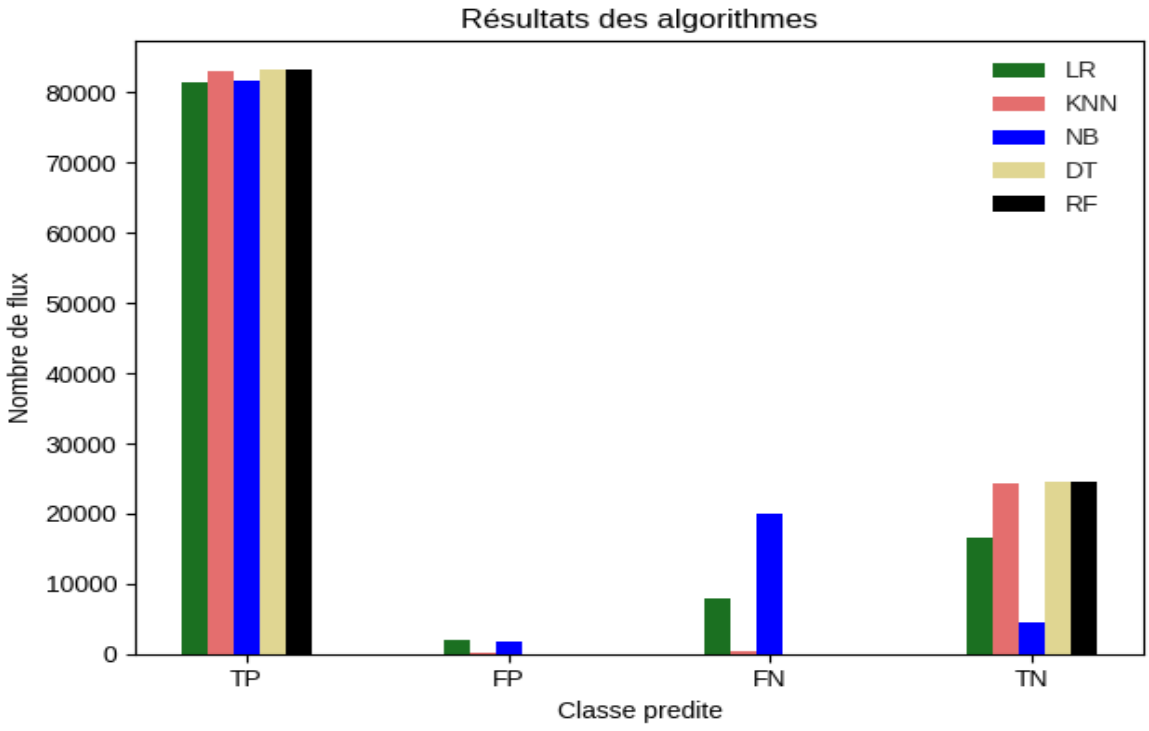

Figure 34 : Les résultats obtenu des algorithmes

1) Analyse des résultats Régression logistique (Logistic Regression)

La matrice de confusion indique un total de 81 417 vrais positifs (TP), 1 873 faux positifs (FP), 7 950 faux négatifs (FN) et 16 603 vrais négatifs (TN). L'exactitude (accuracy) du modèle est de 90,89%, ce qui signifie qu'il prédit correctement la classe dans 90,89% des cas. L'erreur (fail accuracy) est de 9,11%. Le temps d'apprentissage et de prédiction est de 1 minute et 38 secondes.

2) Analyses des résultats K-NN (K-Nearest Neighbors)

La matrice de confusion affiche 83 077 vrais positifs (TP), 213 faux positifs (FP), 317 faux négatifs (FN) et 24 236 vrais négatifs (TN). L'exactitude est de 99,51%, ce qui indique une performance élevée du modèle. L'erreur est de 0,49%. Le temps d'apprentissage et de prédiction est de 4 minutes et 54 secondes.

3) Analyse des résultats Naïve Bayes

La matrice de confusion montre 81 639 vrais positifs (TP), 1 651 faux positifs (FP), 19 954 faux négatifs (FN) et 4 599 vrais négatifs (TN). L'exactitude est de 79,97%, ce qui indique une performance relativement plus faible par rapport aux autres algorithmes.

L'erreur est de 20,03 %. Le temps d'apprentissage et de prédiction est de 2,28 secondes.

4) Analyse des résultats arbre de décision (Decision Tree)

La matrice de confusion présente 83 271 vrais positifs (TP), 19 faux positifs (FP), 13 faux négatifs (FN)et 24 540 vrais négatifs (TN). L'exactitude est de 99,97%, ce qui montre une excellente performance du modèle. L'erreur est de seulement 0,03%.

Le temps d'apprentissage et de prédiction est 8.89 secondes.

5) Analyse des résultats Random Forest

L'algorithme Random Forest a obtenu une matrice de confusion avec 83 261 vrais positifs (TP), 29 faux positifs (FP), 10 faux négatifs (FN) et 24 543 vrais négatifs (TN). L'exactitude (accuracy) est de 99,96%, avec un taux d'erreur (fail accuracy) de 0,04%.

Le temps d'apprentissage et de prédiction est 8.90 secondes.

- Ces résultats fournissent une évaluation de la performance des différents algorithmes utilisés. On observe que le random forest, la régression logistique, le K-NN et l'arbre de décision obtiennent des performances élevées avec des taux d'exactitude élevés. En revanche, le modèle de naïve Bayes présente une précision relativement plus faible.

Conclusion

Ce chapitre présente une approche basée sur l'apprentissage automatique pour la classification du trafic d'attaque DDoS. En renforçant la sécurité informatique, cette méthode permet de protéger les entreprises en détectant et en catégorisant efficacement les attaques, minimisant ainsi les perturbations et les impacts financiers négatifs.

Conclusion générale

En conclusion, ce projet a exploré l'utilisation de différents algorithmes d'apprentissage automatique pour la classification du trafic en attaque DDoS. Les résultats obtenus ont montré que certains algorithmes, tels que les k plus proches voisins et l'arbre de décision, ont obtenu des performances exceptionnelles avec des taux de précision élevés.

L'intégration de l'intelligence artificielle et de l'apprentissage automatique dans la sécurité informatique offre des avantages significatifs en termes de détection des attaques et de prévention des menaces.

Ce projet a permis d'explorer différentes approches et d'obtenir des résultats encourageants. Cependant, il convient de souligner que la sécurité informatique est un domaine en constante évolution, et les attaques évoluent rapidement. Par conséquent, il est important de maintenir une veille technologique et d'adapter continuellement les stratégies de sécurité pour faire face aux nouvelles menaces.

Une perspective importante à considérer est la nécessité de rester constamment informé des développements récents et de s'adapter en permanence aux nouvelles techniques d'attaque et aux variantes de DDoS. Les chercheurs et les professionnels de la sécurité doivent maintenir une veille technologique active pour assurer une défense efficace.

En résumé, ce projet démontre l'importance de l'intelligence artificielle dans la sécurité informatique et souligne le potentiel des algorithmes d'apprentissage automatique pour renforcer la protection contre les attaques DDoS. L'intégration de ces technologies permet d'améliorer la détection, la prévention et la réaction face aux menaces numériques, contribuant ainsi à assurer la sécurité et la continuité des systèmes informatiques des entreprises.

Annexe

Creation d'un SDN (Software Defined Networks)

➢ Les composants de SDN

Les composants de SDN sont des éléments clés dans l'architecture d'un réseau défini par logiciel (SDN). Ils jouent des rôles spécifiques pour permettre la programmabilité, la centralisation du contrôle et la virtualisation du réseau, offrant ainsi une flexibilité et une gestion améliorées du réseau. Voici quelques-uns des composants principaux dans un environnement SDN :

- Mininet : Mininet est un émulateur de réseau virtuel qui permet de créer des réseaux SDN à petite échelle sur une seule machine. Il permet de simuler des hôtes, des commutateurs et des liaisons virtuelles dans un environnement contrôlé.

- OpenDaylight : est une plateforme open source de contrôle de réseau SDN. Il agit en tant que contrôleur centralisé qui gère et contrôle les commutateurs réseau compatibles SDN.

➢ Installation et configuration SDN

- **Installation Mininet**
  - ✓ Installation d'une machine virtuelle Mininet depuis « https://github.com/mininet/mininet/releases/ »
  - ✓ Une fois l'installation terminer on ouvre la machine virtuelle avec l'un des logiciels de virtualisation (exemple : VMware Workstation)

Figure 35:Mininet

- **Installation et configuration OpenDayLight**

✓ Création d'une nouvelle machine virtuelle Ubuntu en suivant les étapes du processus de création de machine virtuelle.

✓ Téléchargement de l'archive d'OpenDaylight en utilisant cette commande :

wget <URL_de_téléchargement_de_l'archive>

✓ Décompression de l'archiver en utilisant la commande :

tar -zxvf <nom_de_l'archive.tar.gz>

✓ Ouvrir OpenDayLight en suivant les commande

« cd chemin\vers\OpenDaylight »

«.\bin\karaf.bat »

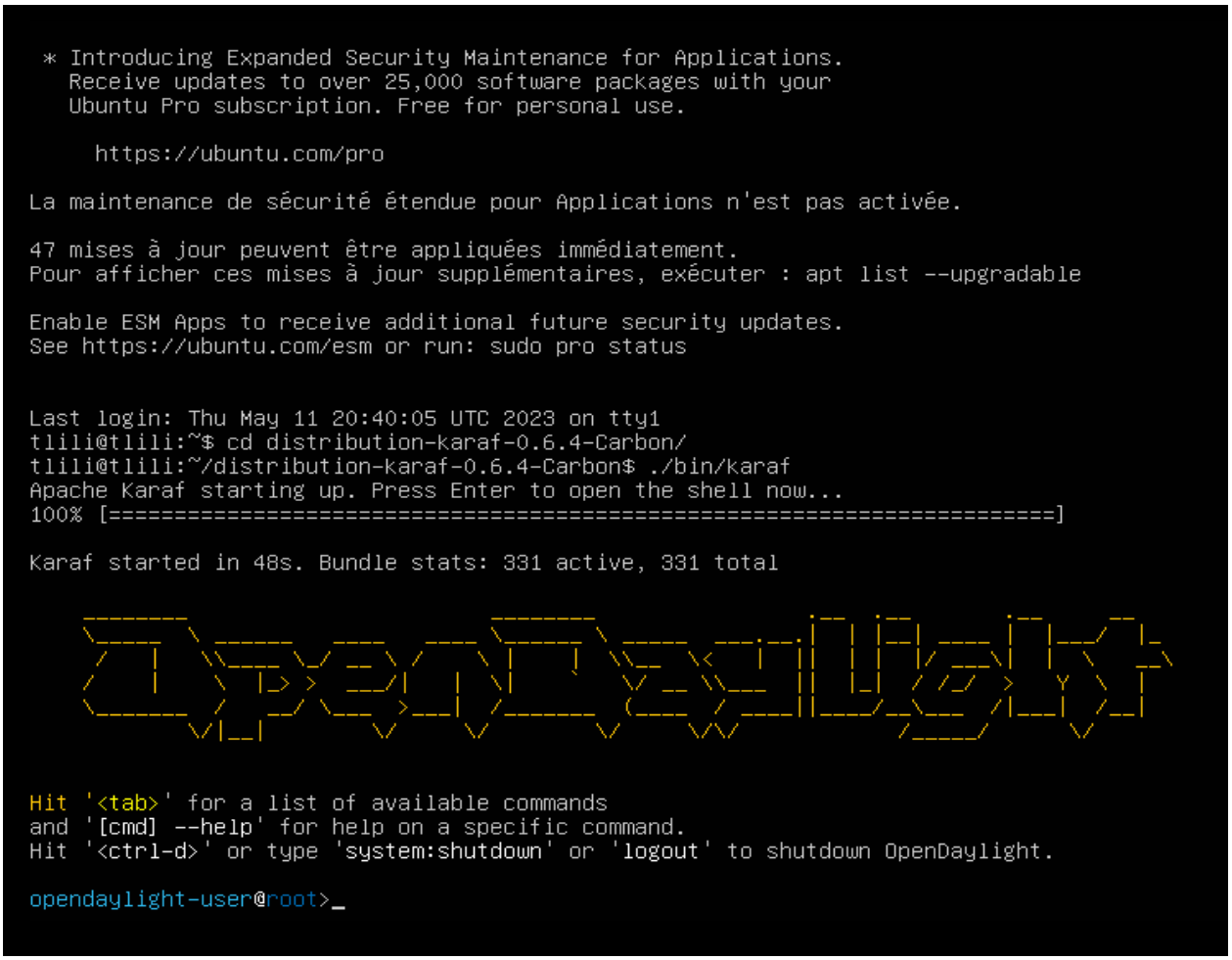

Figure 36 : OpenDayLight

Les attaques testées sur SDN

- ✓ Attaque ICMP (Ping) Flood
- ✓ Attaque UDP Flood
- ✓ Attaque TCP-SYN Flood
- ✓ Attaque Land

```
mininet@mininet-vm:~/mininet/custom$ sudo python  generate_ddos_trafic.py
Unable to contact the remote controller at 192.168.79.131:6653
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s2) (h5, s2) (h6, s2) (h7, s3) (h8, s3) (h9, s3) (h10, s4) (h11, s4)
 (h12, s4) (h13, s5) (h14, s5) (h15, s5) (h16, s6) (h17, s6) (h18, s6) (s1, s2) (s2, s3) (s3, s4) (s
4, s5) (s5, s6)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
------------------------------------------------------------------------------
Performing ICMP (Ping) Flood
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Performing UDP Flood
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Performing TCP-SYN Flood
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Performing LAND Attack
------------------------------------------------------------------------------
```

Figure 37: Attaque sur SDN

Après avoir effectué des tests, il est important de noter que les réseaux SDN peuvent être vulnérables à certaines attaques.

Bibliographie

[1]: (Hat, 2018)

[2]: RACHEDI, HABIBA. *Réalisation d'un pare-feu sécurise sous proxy sous linux*. Diss. Université Ibn Khaldoun-Tiaret-, 2012.

[3]: Djamel, Mechhat, and Houfel Abdelmalek. *Réalisation d''un crypto-système basé sur l''algorithme TEA (Tiny Encryption Algortithme) pour les systèmes embarqués cas : Smartphones*. Diss. Université Mouloud Mammeri, 2017.

[4] : Hamdi, Omessaad, Mbaye Maissa, and Francine Krief. "Génération automatique de signatures par apprentissage pour les systèmes de détection d'intrusions." *7ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d'Information*. 2012.

[5] : Hakim, Amarir, Danes Adrien, and Doffe Sidney. "La protection des réseaux contre les attaques DoS." *Université Paris Descarte* (2009): 1-26.

[6] : Barracuda Networks. (s.d.). Intrusion Detection System. Glossary. Récupéré sur https://www.barracuda.com/support/glossary/intrusion-detection-system

[7] : Sezer, Sakir, et al. "Are we ready for SDN? Implementation challenges for software-defined networks." *IEEE Communications Magazine* 51.7 (2013): 36-43.

[8]: (ionos, 2019)

[9] : Laurière, Jean-Louis. "Intelligence artificielle : résolution de problèmes par l'homme et la machine." (1987).

[10] : (Saint-Cirgue, 2019)

[11] : https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition

[12] : Corbo, A. (29 décembre 2022). "Le rôle clé de Python dans les tâches de la science des données". Récupéré le [date d'accès], à partir de https://builtin.com/software-engineering-perspectives/python.

Résumé

Le SDN surveillent aide l'administrateur a gérer et protéger le réseau informatique contre tout accès non autorisé d'utilisateurs.

Le but de ce projet est de proposer un modèle prédictif (c'est-à-dire un classificateur) capable de faire la distinction entre les "mauvaises connexions" (attaques) et les "bonnes (normales) connexions" après avoir appliqué une extraction de caractéristiques de la base d'apprentissage CICDDoS2019.

Mots clés : SDN, Sécurité, Machine Learning, DDoS, LR, K-NN, DT, NB, RF, Python.

Abstract

SDN helps administrators manage and protect the computer network against unauthorized user access.

The purpose of this project is to propose a predictive model (i.e., a classifier) capable of distinguishing between "bad connections" (attacks) and "good (normal) connections" after applying feature extraction from the CICDDoS2019 dataset.

Keywords: SDN, Security, Machine Learning, DDoS, LR, K-NN, DT, NB, Python.