الجمهورية الجزَائِرية الديمقرَاطية الشعبة

République Algérienne Démocratique et Populaire

وزَارة التعليم العَالي و البحث العلمي

Ministèrede l'Enseignement Supérieur et de la Recherche Scientifique

جَامعة عمَار ثليجي بِالأَغوَاط

UNIVERSITE AMAR TELIDJI LAGHOUAT

FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE

Mémoire de Master

Domaine : Mathématiques et Informatique

Filière : Informatique

Option : Réseaux et systèmes répartis

Présenté Par :

BENDINE Samia **HADDAD Radia**

Thème :

La sécurité des réseaux basé sur le machine learning.

Soutenu publiquement devant le jury composé de :

Dr	GUELLOUMA Younes	MC(A)	(Université de Laghouat)	Président
Dr	OUBATI Sami	MC(A)	(Université de Laghouat)	Examinateur
Dr	CHAIB Noureddine	MC(A)	(Université de Laghouat)	Encadreur

Année Universitaire : 2019/2020


بسم الله الرحمن الرحيم

Dédicaces

Je dédie ce modeste travail ...

Ã mes très chers parents : *Aucune dédicace ne saurait être assez éloquente pour exprimer l'amour et l'estime que vous méritez pour tous les sacrifices que vous n'avez cessé de me donner.*

Ã mes chers frères et sœurs : *Vous occupez une place particulière dans mon cœur, je vous dirais tout simplement « je vous aime ».*

Ã tous ma famille : *En témoignage du respect et de l'affection que je porte envers vous,je vous souhaitant une vie plein de bonheur.*

Ã tous mes amis : *En témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble, je vous souhaitant un brillant avenir.*

Ã mon binôme : *Pour ton aide et ton soutien moral durant l'élaboration du travail de fin d'études.*

Ã tous ceux que j'aime et ceux qui m'aiment.

SAMIA

Dédicaces

Je dédie ce mémoire ...

À la personne *qui m'avez donné l'inspiration, l'enthousiasme et le soutien. Sans lui je ne serais jamais devenu la personne que je suis aujourd'hui. Mon père qui était avec moi tout au long de mes études pas à pas et m'a toujours donné l'amour, le courage et la confiance pour avancer.*

À la femme *qui a souffert sans me laisser souffrir, qui n'a jamais dit non à mes exigences et qui n'a épargné aucun effort pour me rendre heureuse, mon adorable mère ce travail est le fruit de tes sacrifices.*

À mes frères : *Midou, Farouk et AbedRezak.*

À mes sœurs : *Imane, Jinan, Maria.*

À mon petit ange *Fatima et ma fleur Yasmine.*

À Hasna, *merci infiniment pour ton encouragement et ta présence avec moi tout le temps, tu resteras toujours la meilleur dans ma vie.*

À Nousra, *je te remercie d'être toujours là pour moi, je te remercie d'être l'épaule sur laquelle je peux toujours compter, pour ton esprit quand j'étais dans la folie. Merci d'être une partie de moi.*

À Massilia, *je n'oublierai jamais ton encouragement et ton aide durant cette longue année.*

À Samia, *mon binôme merci pour me partager ce travail.*

Et toute personnes que je connais et qui sont chers.

RADIA

Remerciement

Nous remercions DIEU le plus puissant qui nous a donné la patience et le courage pour réaliser ce travail.

On tient tout d'abord à remercier notre encadreur Dr CHAIB noureddine pour son grand aide durant la réalisation de ce travail, pour son soutien, et sa continuelle disponibilité. Nous voulons lui exprimer nos sentiments sincères et respectueux.

Nous tenons à remercier les membres de jury, chacun a son nom d'accepter de juger et dévaluer ce travail.
Sans oublier tous ceux qui ont contribué à la réalisation de ce travail.

Résumé

L'accréditation totale du monde sur le web dans tous les domaines introduit des menaces de tout type. C'est ce qui a mis les informaticiens de sécurité et les attaquants en compétition, tous les jours des millions des techniques de sécurité sont inventés afin de protéger les systèmes contre ces attaquants. Le but de ces attaquants est de rendre un service indisponible ou détruire la confidentialité et l'intégrité des informations, l'une des attaques la plus utilisée pour atteindre cet objectif c'est l'attaque DDOS (Distributed Deniale Of Service).

Nous proposons une implémentation d'un système pour détecter ce genre d'attaque et protéger les ressources de serveur. Ce système est implémenté par une méthode de machine learning le SVM qui était écrit en python et des scripts pour l'extraction de l'ensemble de données à traiter à partir de fichier log de serveur. On a lancé une attaque DDOS (Distributed Denial of Service) contre le serveur de la plateforme E-learning de l'université Ammar Telidji. Enfin nous avons évalué le système et nous avons trouvé qu'est capable de défendre contre ce genre d'attaque.

Mots clés : Sécurité, Détection, Attaque, DDOS, Machine learning, SVM.

Abstract

The total reliance of the world on the web in all fields introduces threats of all sorts. This is what has put security informaticians and attackers in competition, every day millions of security techniques invented to protect the systems against these attackers. The purpose of these attackers is to make a service unavailable or destroy the confidentiality and integrity of the information, one of the most used attacks to achieve this purpose is the Distributed Denial Of Service (DDOS) attack.

We propose an implementation of a system to detect this kind of attack and protect server resources. This system implemented by the SVM machine learning that is written in python and scripts for extracting the dataset to be processed from the server log file. We managed a Distributed Denial Of Service (DDOS) attack against the Ammar Telidji University E-learning platform server to evaluate this system.

Keywords : Security, Detection, Attacks, DDOS, Machine learning, SVM.

Table des matières

Table des figures

Liste des abréviations

— DDOS : Distributed Denial of Service.

— SVM : Support Vector Machine.

— PHP : Hypertext Preprocessor.

— IP : Internet Protocol.

— VPN : Virtual Private Network.

— CSS : Cascading Style Sheet

— URL : Uniform Resource Locator.

— HTML : HyperText Markup Language.

— HTTP : HyperText Transfer Protocol.

— TCP : Transmission Control Protocol.

— RNN : Recurrent Neural Network.

— CNN : Convolutional Neural Networks.

— DQN : Deep Q-Networks.

— SARSA : State Action Reward State Action.

— KNN : K-Nearest Neighbors.

— DBSCAN : Density-Based Spatial Clustering of Applications with Noise.

— PCA : Principal Component Analysis.

— LSA : Latent Semantic Analysis.

— SVD : Singular Value Decomposition.

— LDA : Linear Discriminant Analysis.

— CART : Classification And Regression Trees.

— MARS : Multivariate Adaptive Regression Splines.

— LOIC : Low Orbit Ion Cannon.

Introduction générale

Les systèmes informatiques sont devenus la base de différents domaines, ils séduisirent chaque jour de plus en plus d'internautes par les nombreux avantages et la diversité des services rendus accessibles. Les utilisateurs de ces systèmes peuvent bénéficier à moindre coût de moyens de communication rapides, partager des ressources de traitement et de stockage de grandes capacités (Cloud Computing), faciliter les échanges commerciaux et financiers (e-Commerce, e-Banking), fournir et utiliser de nombreux services en ligne (e-Administration, e-Health, e-Learning, etc.), participer à des communautés virtuelles et à des réseaux sociaux, se divertir (e-Gaming, e-Television, etc.) et, plus généralement, partager et accéder à l'information.

La croissance exponentielle d'utilisation de ces systèmes les a rendus vulnérables aux attaques. Cette vulnérabilité est considérée comme un problème majeur dans le monde informatique. Tous les jours des millions des systèmes sont exposés aux attaques, vue à ces menaces le domaine de sécurité informatique est devenu l'un des domaines les plus importants et les plus chers.

L'attaque DDOS (Distributed Denial Of Service), c'est parmi les attaques les plus connues, vise à rendre un serveur, un service ou une infrastructure indisponible. Elle peut prendre différentes formes comme une saturation de la bande passante du serveur pour le rendre injoignable ou un épuisement des ressources système de la machine, et plus. Cela a incité les utilisateurs à poser des questions sur les techniques à utiliser pour détecter une attaque de ce genre et comment ils peuvent protéger leurs systèmes ?

Dans ce travail, nous avons essayé de répondre à ces questions par la création d'un système pour protéger et sécuriser les serveurs web contre ce genre d'attaque, pour la réalisation de ce système on a utilisé l'une des méthodes de machine learning le SVM et

des scripts. Ce système est appelé BanDos. L'objectif de ce système est de protéger les ressources de serveur (les pages PHP, les bases de données, ...).

Le domaine éducatif, comme tous les domaines, a été affecté par la situation actuelle due au COVID-19. Afin que le parcours éducatif des étudiants ne soit pas affecté par cette situation, ils sont dirigés vers la plateforme E-learning.

Nous avons donc profité de cette situation et maintenu notre système sur le serveur de la plateforme E-learning au niveau de l'université Ammar Thelidji.

Ce rapport se définit sur trois chapitres,

— **Le premier chapitre** contiendra une vue générale sur la sécurité informatique, aussi les concepts de base des serveurs web et son architecture.

— **Au deuxième chapitre**, nous allons définir le machine learning et détailler ces classes, nous concentrés sur quelques méthodes d'apprentissage classique.

— **Dans le troisième chapitre**, nous présenterons une description de fonctionnement de notre système et son architecture. Après nous montrerons les phases pour la réalisation et discutons les résultats obtenus.

Nous clôturerons ce rapport par une conclusion générale.

Chapitre 1

Vue générale sur la sécurité informatique et les serveurs web

1.1 Introduction

Le web est devenue une partie intégrante de tous les domaines au cours des dernières années. Nous avons remarqué son importance et son utilité dans ces circonstances particulières causées par le COVID-19. Ce qui rendait les serveurs web et le domaine de la sécurité informatique très importants. A cet effet dans ce chapitre, nous allons présenter les concepts de base de la sécurité informatique et les attaques. Ensuite, on va détailler les serveurs web et son architecture.

1.2 La sécurité informatique

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

1.2.1 Terminologie de la sécurité informatique

La sécurité informatique utilise un vocabulaire bien défini, parmi les termes les plus importants et les plus connues sont les suivants [2] :

— **Les vulnérabilités** : elles sont les failles de sécurité dans un ou plusieurs systèmes. Tout système vu dans sa globalité présente des vulnérabilités, qui peuvent être exploitables ou non.

— **Les attaques** (exploits) : elles représentent les moyens d'exploiter une vulnérabilité. Il peut y avoir plusieurs attaques pour une même vulnérabilité.

— **Les contre-mesures** : elles sont les procédures ou techniques permettant de résoudre une vulnérabilité ou de contrer une attaque spécifique.

— **Les menaces** : elles sont des adversaires déterminés capables de monter une attaque exploitant une vulnérabilité.

— **Intrusion** : c'est l'opération qui consiste à accéder, sans autorisation, aux données d'un système informatique ou d'un réseau, en contournant ou en désamorçant les dispositifs de sécurité mis en place.

— **Risque** : c'est la combinaison de menaces et de vulnérabilités.

— **Incident** : un évènement qui ne fait pas partie des opérations standard d'un service et qui peuvent provoquer une interruption de service ou altérer sa qualité.

1.2.2 Les objectifs principaux de la sécurité informatique

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. La sécurité informatique vise généralement cinq principaux objectifs :

— **Confidentialité** : il consiste à assurer que seules les personnes autorisées ont accès aux ressources échangées [3].

— **Authentification** : il a pour but d'assurer que seules les personnes autorisées ont accès aux ressources [3].

— **Intégrité** : il consiste à garantir que les données sont bien celles que l'on croit être [3].

— **Disponibilité** : elle consiste à garantir l'accès à un service ou à des ressources [3].

— **Non-répudiation** : c'est la garantie qu'aucun des correspondants ne pourra nier la transaction [3].

1.2.3 Les mécanismes de la sécurité informatique

Les mécanismes de la sécurité sont conçus pour détecter, prévenir et lutter contre les attaques. Il existe des centaines de ces mécanismes qui sont très utilisés et aussi très connus, certains d'eux sont les suivants :

— **Cryptage** : c'est une science mathématique dans laquelle on fait les études des méthodes permettant de transmettre des données de manière confidentielle. Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible ; c'est ce qu'on appelle le chiffrement, qui, à partir d'un texte clair, donne un texte chiffré ou cryptogramme. Inversement, le déchiffrement est l'action qui permet de reconstruire le texte en clair à partir du texte chiffré. Dans la cryptographie moderne, les transformations en question sont des fonctions mathématiques, appelées algorithmes cryptographiques, qui dépendent d'un paramètre appelé clef [8].

— **Antivirus** : c'est un programme qui protège une machine contre les virus. L'antivirus s'appuie sur des fichiers de signatures, puis compare la signature génétique du virus avec les codes à analyser. Certains programmes appliquent également une méthode de guidage pour détecter le code malveillant avec son comportement [8].

— **Pare-feu** : en anglais c'est le firewall, il s'agit d'un système, logiciel ou matériel, placé entre deux réseaux ou plus, et dont le rôle est de filtrer le trafic réseau se présente à ses interfaces. Il tente d'isoler ces réseaux de façon à les protéger les uns des autres. De nombreux types de pare-feu existent mais les plus connus sont :

  — **Les pare-feu à filtrage de paquets** : ce type analyse les en-têtes des paquets et applique un certain nombre de règles de filtrage basées sur les adresses IP de la machine émettrice et de la machine réceptrice.

  — **Les pare-feu applicatifs** : ce type permet de filtrer les communications application par application. Il permettra de contrôler (filtrer) les connexions réseau tentées par tel ou tel logiciel. À chaque connexion qui tente une application, ces pare-feu applicatifs vérifient que la connexion est autorisée ou alerte l'utilisateur et lui demandent l'action à effectuer [5].

— **VPN(Virtual Private Network)** : c'est une technique de réseau pour sécuriser les lignes de communication entre les membres ou les groupes qui utilisent ensemble une infrastructure de communication publique qui fournit des services de confidentialité aux lignes de communication en utilisant des protocoles de sécurité et de tunnel. Il utilise des tunnels qui fournissent une confidentialité de sécurité au chemin de communication entre les nœuds. La technique VPN est généralement appelée tunnel VPN [6].

— **IP-table** : c'est un logiciel très utilisé par le pare-feu, il fait partie du projet Netfiltre. Il est utilisé pour définir la structure de la table des règles filtrant les paquets. Plusieurs tables différentes peuvent être définies. Chaque table contient un nombre de chaînes prédéfinies, et peut aussi contenir des chaînes définies par l'utilisateur. Chaque chaîne est une liste de règles auxquelles peuvent correspondre un ensemble de paquets. Chaque règle spécifie ce qui doit être fait avec un paquet qui correspond. Il existe trois tables principales des règles, quels sont les suivants [13] :

  — **Table filtre** : c'est la table par défaut. Elle contient les chaînes prédéfinies INPUT (pour les paquets entrants dans la machine), FORWARD (pour les paquets routés à travers la machine), et OUTPUT (pour les paquets générés en local).

  — **Table Nat** : cette table est consultée lorsqu'un paquet qui crée une nouvelle connexion est détecté. Elle est composée de trois chaînes prédéfinies PREROUTING (pour modifier les paquets dès qu'ils entrent), OUTPUT (pour modifier les paquets générés localement avant qu'ils ne soient routés), et POSTROUTING (pour modifier les paquets lorsqu'ils sont sur le point de sortir).

  — **Table mangle** : cette table est employée pour un traitement spécial des paquets. Elle a deux chaînes prédéfinies, PREROUTING (pour modifier les paquets entrants avant qu'ils ne soient routés) et OUTPUT (pour modifier les paquets générés localement avant qu'ils ne soient routés).

    Certaines actions sont appliquées à un paquet, les principales actions sont les suivantes[14] :

    — **ACCEPT** : le paquet est accepté.

    — **DROP** : le paquet est rejeté.

    — **QUEUE** : le paquet est déplacé dans les processus utilisateurs. Ceci nécessite un intermédiaire (gestionnaire de queue), qui transfère le paquet à une application.

    — **RETURN** : le paquet est renvoyé à la chaîne précédentes'il s'agit d'une chaine personnalisée par l'utilisateur.

1.3 Le serveur web

Un serveur web peut faire référence à des composants logiciels (software) ou à des composants matériels (hardware) ou à des composants logiciels et matériels qui fonctionnent ensemble. L'activité de ces deux composants est différente, la différence est [15] :

— Au niveau des composants matériels, un serveur web est un ordinateur qui stocke les fichiers qui composent un site web (par exemple les documents HTML, les images, les feuilles de style CSS, les fichiers JavaScript) et qui les envoient à l'appareil de l'utilisateur qui visite le site.

— Au niveau des composants logiciels, un serveur web contient différents fragments qui contrôlent la façon dont les utilisateurs peuvent accéder aux fichiers hébergés. On trouvera au minimum un serveur HTTP.

Dans les serveurs web on a deux types, quel sont :

— **Serveur web statique** : ce type de serveur est composé d'un ordinateur (matériel) et d'un serveur HTTP (logiciel). Il est appelé statique car le serveur envoie les fichiers hébergés tels quels vers le navigateur.

— **Serveur web dynamique** : ce type possède d'autres composants logiciels, certains qu'on retrouve fréquemment dont un serveur d'applications et une base de données. Il est appelé dynamique car le serveur d'applications met à jour les fichiers hébergés avant de les envoyer au navigateur via HTTP.

1.3.1 L'architecture client-serveur

L'architecture client-serveur désigne un mode de communication entre plusieurs composants d'un réseau. Chaque entité est considérée comme un client ou un serveur. Chaque logiciel client peut envoyer des requêtes à un serveur. Un système client-serveur fonctionne selon le schéma suivant :

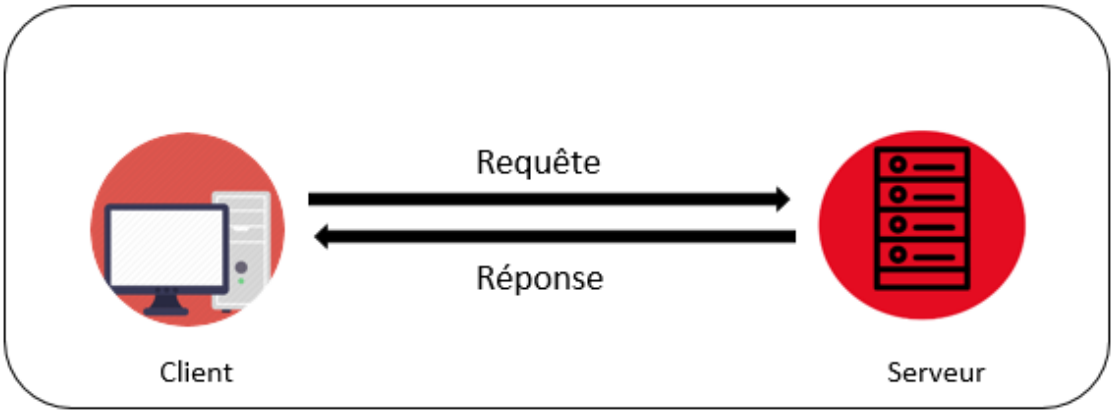

FIGURE 1.1 – Fonctionnement de système client-serveur

L'architecture client-serveur comporte un ensemble des architectures différentes [16] :

— **Architecture à 2 niveaux** : l'architecture à deux niveaux (aussi appelée architecture 2-tier, tier signifiante rangée en anglais) caractérise les systèmes clients-serveurs pour lesquels le client(niveau 1) demande une ressource et le serveur(niveau 2) la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service, comme l'indique dans le schéma suivant :

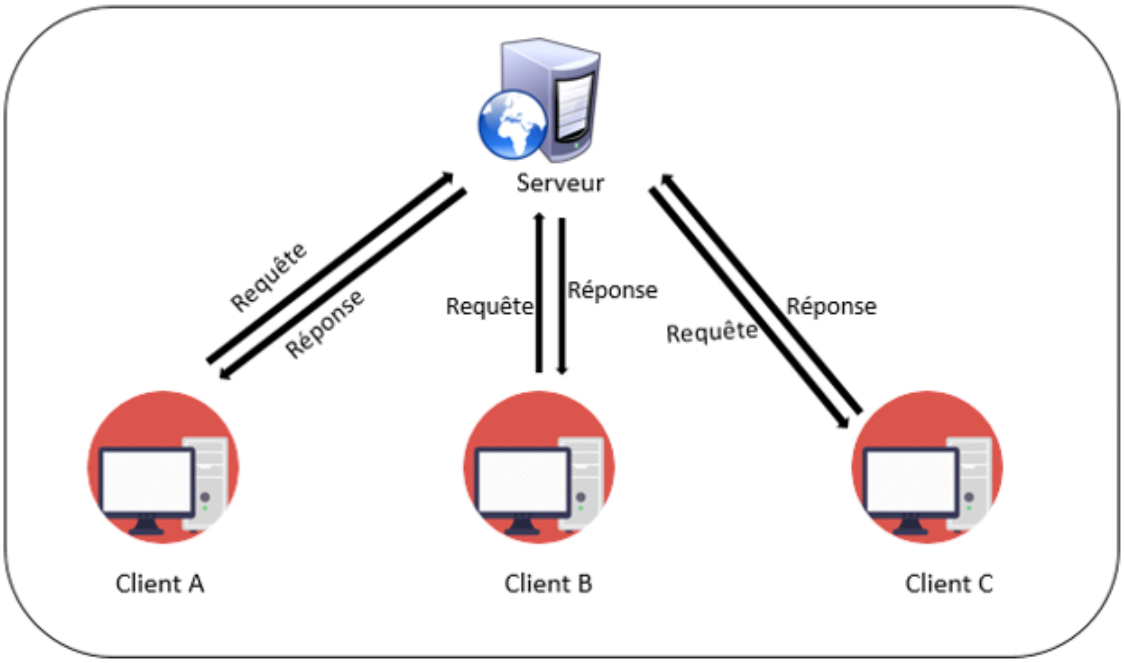

FIGURE 1.2 – Architecture client-serveur à 2 niveaux

— **Architecture à 3 niveaux** : dans l'architecture à 3 niveaux (appelés architecture 3-tier), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

  — Un client(niveau 1), c'est-à-dire l'ordinateur demandeur des ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation.

  — Le serveur d'applications(niveau 2) appelé également middleware, chargé de fournir la ressource mais faisant appel à un autre serveur.

  — Le serveur des données(niveau 3) fournissant au serveur d'applications les données dont il a besoin.

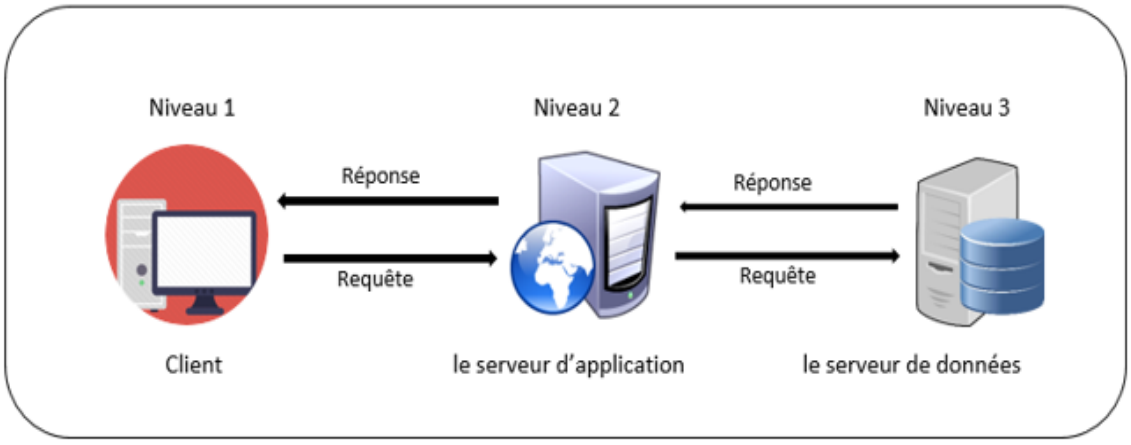

FIGURE 1.3 – Architecture à 3 niveaux

— **Architecture à N niveaux :** cette architecture permet d'atteindre un nombre de niveau plus important. La requête est exécutée par plusieurs serveurs, dans cette architecture on a un client et N serveurs.

1.3.2 Les standards de web

Le web repose sur trois standards afin d'assurer son bon fonctionnement, sont les suivants [16] :

— **URL(Uniform Resource Locator)** : indique aux navigateurs le chemin à emprunter pour accéder aux ressources contenues dans le World Wide Web.

— **HTML (HyperText Markup Language)** : c'est un langage utilisé pour décrire la mise en page et l'apparence du contenu d'un document Web.

— **HTTP (HyperText Transfer Protocol)** : c'est le protocole de transport utilisé par les navigateurs Web (Firefox, Internet Explorer...) et les serveurs Web (Apache, ...) pour communiquer entre eux. C'est lui qui est utilisé par exemple pour obtenir un fichier HTML, une image, poster un formulaire internet. Il est donc au cœur de l'internet. Techniquement, c'est un protocole texte (donc lisible en clair) s'appuyant les protocoles plus bas-niveau TCP/IP.

Le principe de fonctionnement de http est basé sur des méthodes, chaque méthode est une commande spécifiant un type de requête, c'est-à-dire qu'elle demande au serveur d'effectuer une action. En général, l'action concerne une ressource identifiée par l'URL qui suit le nom de la méthode. Les méthodes les plus utilisées sont :

— **GET** : c'est la méthode la plus courante pour demander une ressource.

— **POST** : cette méthode doit être utilisée pour ajouter une nouvelle ressource, comme un message sur un forum, un article dans un site ou encore un login et un mot de passe.

Les autres méthodes sont : **HEAD, OPTIONS, CONNECT, TRACE, PUT, DELETE**. Certains serveurs autorisent des méthodes supplémentaires permettant la gestion des ressources du serveur (par exemple WebDAV ou CALDAV).

1.3.3 Les fichiers logs

Le comportement de l'utilisateur sur un site web réside en une suite de clics de souris et de saisies sur un clavier. Ces informations déclenchent des requêtes qui ont pour résultat l'affichage de certaines pages du site. Ces requêtes sont enregistrées dans un fichier texte à mesure qu'elles sont déclenchées par les utilisateurs. Ces données sont stockées de manière standardisée de façon à ce qu'il soit possible de procéder à des analyses. Cette base de données est communément appelée fichier log. Son analyse permet en principe de savoir quelles sont les requêtes qui n'aboutissent pas (page manquante, lien erroné...) ou encore quelle est la fréquentation de chaque page [17].

On peut extraire sept informations à partir de fichiers log, sont [17] :

— Le nom du domaine ou l'adresse IP de la machine connectée.

— Le nom et le login HTTP de l'utilisateur (en cas d'accès par mot de passe).

— La date et l'heure de la requête.

— La méthode utilisée dans la requête (GET, POST, etc.) et le nom de la ressource web demandée (l'URL de la page demandée).

— Le statut de la requête i.e. le résultat de la requête (succès, échec, erreur, etc.).

— La taille de la page demandée en octets.

— Le navigateur et le système exploitation utilisé par le client.

Le format de fichier log : Le format le plus répandu de fichier log est comme la suite :

```
105.108.121.0 - - [03/May/2020:06:25:33 +0000] "GET /course/index.php?
HTTP/1.1" 200 20232 "http://elearning.lagh-univ.dz/course/request.php"
"Mozilla/5.0 (Windows NT 6.1; Win64; x64)"
```

FIGURE 1.4 – Le format du fichier log

La requête d'un utilisateur est composée par les champs suivants :

— 105.108.121.0 : l'adresse IP de l'utilisateur qui a envoyé la requête.

— 03/May/2020 :06 :25 :33 +0000 : la date est l'heure de la requête.

— GET /course/index.php ? HTTP/1.1 : la méthode de requête, la page demandée et le protocole utilisé.

— 200 : le numéro de code de réponse de serveur.

— 20232 : la taille de la page demandée en octets.

— http ://elearning.lagh-univ.dz/course/request.php : la page de référence qui à partir de laquelle la requête est lancée.

— Mozilla/5.0 (Windows NT 6.1 ; Win64 ; x64) : le navigateur et le système d'exploitation utilisés par l'utilisateur.

Les fichiers log sont très utiles pour déterminer les causes des pannes ou pour détecter les attaques contre le serveur.

1.4 Les attaques

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 systèmes et généralement préjudiciables. Les motivations des attaques peuvent être de différentes sortes :

— Obtenir un accès au système.

— Voler des informations, telles que des secrets industriels ou des propriétés intellectuelles.

— Glaner des informations personnelles sur un utilisateur.

— S'informer sur l'organisation (entreprise de l'utilisateur, etc.).

— Perturber le bon fonctionnement d'un service.

— Utiliser les ressources du système de l'utilisateur.

1.4.1 Anatomie d'une attaque

Il constitue le squelette de toutes attaques informatiques : Probe, Penetrate, Persist, Propagate, Paralyze, qui sont Fréquemment appelés « les 5 P ».
Observons le détail de chacune de ces étapes [7] :

— **Probe(analyser)** : cette étape consiste à collecter des informations sur le système par des outils afin de le pénétrer. La collecte d'informations sur le système cible peut s'effectuer de plusieurs manières, comme par exemple un scan de ports grâce au programme Nmap pour déterminer la version des logiciels utilisés, ou encore un scan de vulnérabilité à l'aide du programme Nessus.

— **Pentrate (pénétrer)** : les informations collectées dans l'étape précédente sont utilisées pour pénétrer le réseau. Des techniques comme la brute force ou les attaques par dictionnaires peuvent être utilisées pour outrepasser les protections par mot de passe.

— **Persist (persister)** : c'est la création d'un compte avec des droits de super utilisateur pour pouvoir se réinfiltrer ultérieurement. Une autre technique consiste à installer une application de contrôle à distance capable de résister à un reboot.

— **Propagate (Propager)** : cette étape consiste à observer ce qui est accessible et disponible sur le réseau local.

— **Paralyze (Paralyser)** : cette étape peut consister en plusieurs actions. Le pirate peut utiliser le serveur pour mener une attaque sur une autre machine, détruire des données ou encore endommager le système d'exploitation dans le but de planter le serveur.

1.4.2 Les types des attaques

Les attaques sont divisées à deux grandes catégories qu'elles sont :

— **Les attaques passives** : elles consistent à écouter ou copier le contenu d'un message transmis et analysé le trafic de manière illicite. Elles sont très difficiles à détecter car elles ne causent aucune altération des données [8].

— **Les attaques actives** : elles consistent à modifier les données ou les messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute [2]. Les attaques actives le plus célèbre sont :

  — **DOS (Denial of Service)** : l'attaque par déni de service vise à rendre une application informatique incapable de répondre aux requêtes de ses utilisateurs par saturation de ses ressources [9].

  — **Spoofing (usurpation d'identité)** : c'est une technique permettant à un pirate d'envoyer à une machine des paquets semblant provenir d'une adresse IP autre que celle de la machine du pirate [9].

  — **Man in the middle (homme au milieu)** : également connu sous le nom Monkey-in-the-middle attack, Session hijacking, TCP hijacking, TCP session hijacking, c'est une attaque informatique qui se réalise dans un réseau local. Son objectif est de forcer plusieurs machines d'un réseau à envoyer des données sur sa machine pour pouvoir communiquer avec leurs destinataires [10].

1.5 L'attaque DDOS (Distributed Denial of Service)

Une attaque DDOS vise à rendre un serveur, un service ou une infrastructure indisponible. L'attaque peut prendre différentes formes :

— Une saturation de la bande passante du serveur pour le rendre injoignable.

— Un épuisement des ressources système de la machine.

— L'empêchant ainsi de répondre au trafic légitime [18].

On parle de DDOS lorsque l'attaque fait intervenir un réseau de machines afin d'interrompre le ou les services visés [11].

1.5.1 Les types d'attaque DDOS

— **Les botnets** : une attaque DDOS de ce type nécessite qu'un pirate prenne le contrôle d'un réseau de machines en ligne afin de réaliser une attaque. Les ordinateurs et autres machines sont infectés par des logiciels malveillants, ce qui transforme chacun d'entre eux en bot(ou zombie). Le pirate a alors le contrôle à distance du groupe de bots, appelé botnet.

Une fois qu'un botnet a été établi, le pirate est capable de piloter les machines en envoyant des instructions actualisées à chaque Bot par une méthode de contrôle à distance. Lorsque l'adresse IP d'une victime est ciblée par le botnet, chaque Bot répond en envoyant des requêtes à la cible, ce qui peut entraîner un dépassement de capacité du serveur ou du réseau ciblé, et donc un déni de service du trafic normal. Chaque Bot était un dispositif internet légitime.

La figure suivante montrer en général l'attaque DDOS par les botnets :

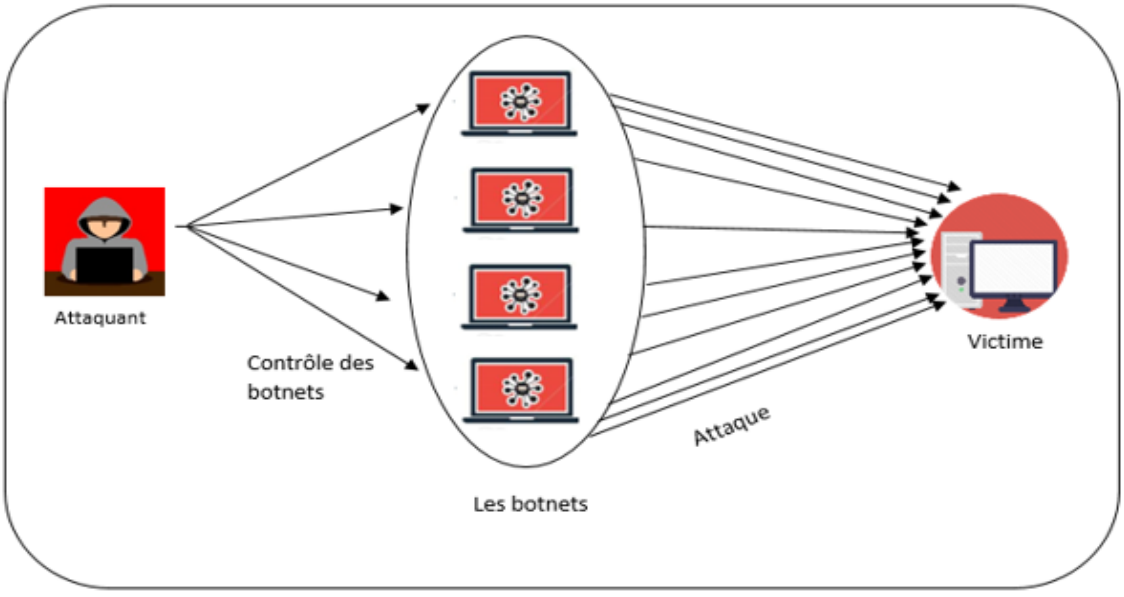

FIGURE 1.5 – Attaque DDOS par les botnets

— **Les attaques basées sur la réflexion** : ils sont connus sous le nom de smurf en anglais. Ce type d'attaque est basée sur l'utilisation de serveurs de diffusion (broadcast) pour paralyser un réseau. Dans ce type d'attaque la machine attaquante envoie une requête Ping à un ou plusieurs serveurs de diffusion en falsifiant l'adresse IP source (adresse à laquelle le serveur doit théoriquement répondre) et en fournissant l'adresse IP d'une machine cible. Ensuite, le serveur de diffusion répercute la requête sur l'ensemble du réseau, toutes les machines du réseau envoient une réponse au serveur de diffusion, le serveur broadcast redirige les réponses vers la machine cible. Ainsi, lorsque la machine attaquante adresse une requête à plusieurs serveurs de diffusion situés sur des réseaux différents, l'ensemble des réponses des ordinateurs des différents réseaux vont être routé sur la machine cible [12].

La figure suivante montrer l'attaque basée sur la réflexion :

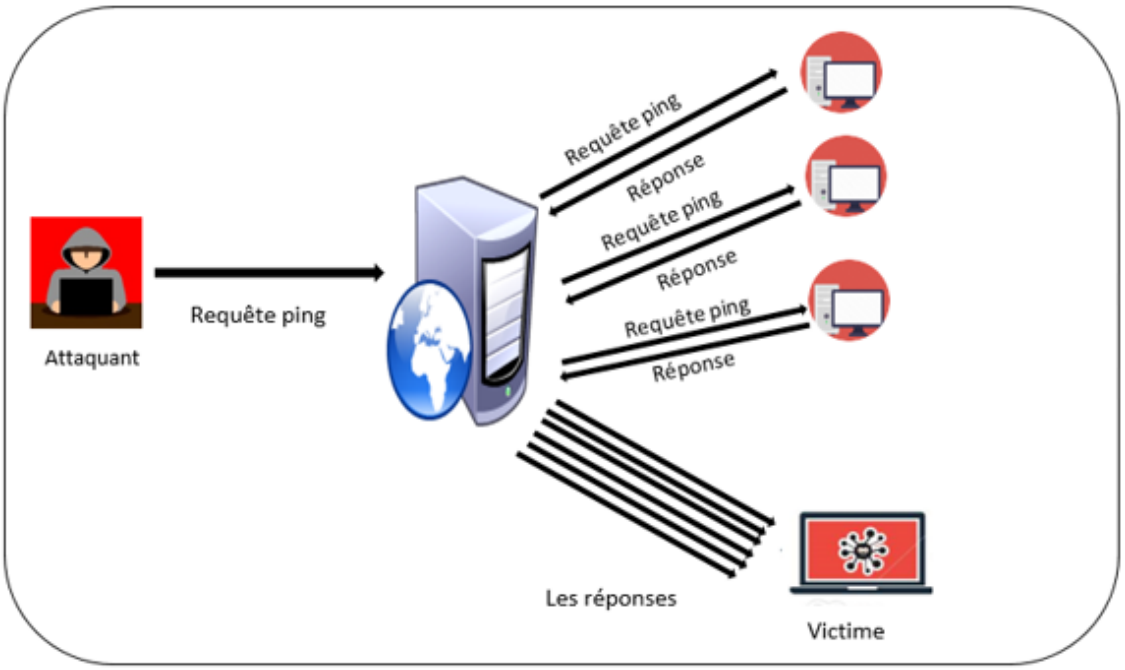

FIGURE 1.6 – Attaque basé sur la réflexion

— **Les attaques basées sur l'amplification** : ce type d'attaque a pour objectif d'épuiser la bande passante réseau disponible afin de rendre un ou plusieurs services inaccessibles. Il menait en exploitant les propriétés de certains protocoles afin de maximiser le volume de trafic généré. Par ailleurs, des attaques volumétriques visent à générer un très grand nombre de paquets par seconde afin de saturer les ressources de traitement d'une cible [11].

— **Les attaques ciblant des applications** : certaines attaques de ce type visent à épuiser les capacités de traitement d'une cible. Par exemple, un attaquant peut chercher à atteindre la limite du nombre de connexions concurrentes qu'un serveur web peut traiter. Dans ce cas, l'attaquant envoie en permanence un grand nombre de requêtes http GET ou POST au serveur ciblé. Il est également possible d'envoyer des requêtes partielles, puis de transmettre la suite de ces requêtes à intervalles réguliers, dans le but de maintenir les connexions ouvertes le plus longtemps possible et d'éviter la fermeture des connexions au-delà d'un délai fixé [11].

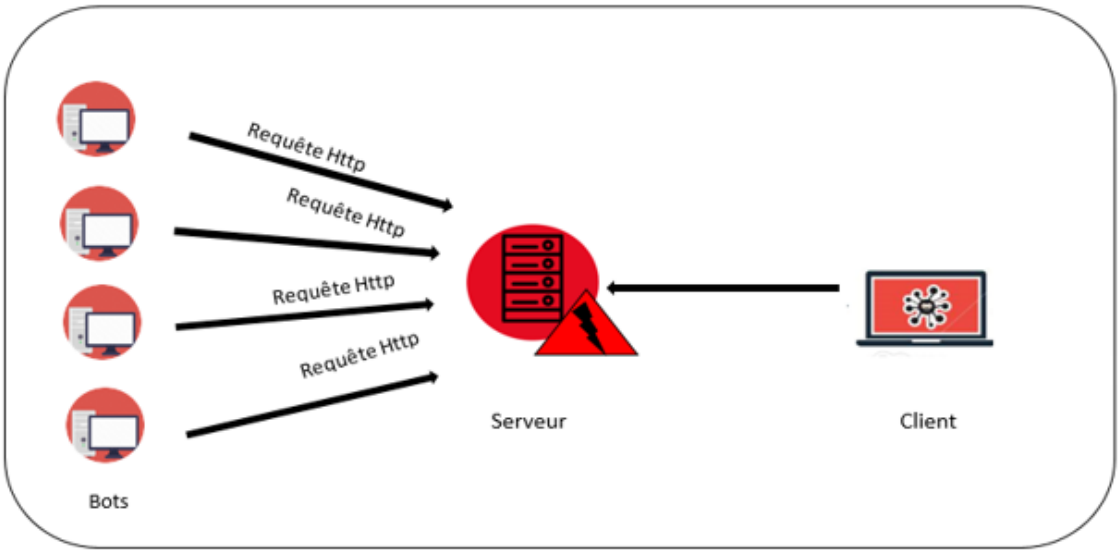

FIGURE 1.7 – Attaque ciblant application

1.6 Conclusion

Toutes machines connectée aux réseau ou tous serveurs nécessite une sécurité très élevée. Nous ne pouvons pas les protéger complétement par l'utilisation des méthodes classiques. Pour cela, de nouvelles méthodes sont implémentées qui reposent sur l'apprentissage automatique.

Chapitre 2

Le machine learning et ces techniques

2.1 Introduction

Un ordinateur n'est pas intelligent, il ne fait qu'exécuter des tâches, on lui décrit sous forme de programme quoi faire et comment le faire, c'est ce qu'on l'appelle la programmation.

Une branche de l'intelligence artificielle est utilisée pour donner la possibilité aux ordinateurs d'apprendre ou devient intelligente, cette branche est le Machine leaning.

Dans ce chapitre, nous allons définir le machine learning, ainsi les classes de machine learning et quelques méthodes d'apprentissage classique.

2.2 Machine learning

Le machine learning s'agit d'une science moderne permettant de découvrir des patterns et d'effectuer des prédictions à partir des données en se basant sur des statistiques, sur du forage de données, sur la reconnaissance de patterns et sur les analyses prédictives [19].

Le machine Learning est un ensemble de méthodes qui permettent aux ordinateurs de faire des modèles à partir des données qui leur sont soumises [21]. Les modèles de machine learning ont démontré un grand succès dans l'apprentissage de modèles complexes leur a permis de faire des prédictions sur les données non observées [20]. Il est divisible sur

quatre classes où chaque classe utilise plusieurs algorithmes, comme le montre dans la figure suivante :

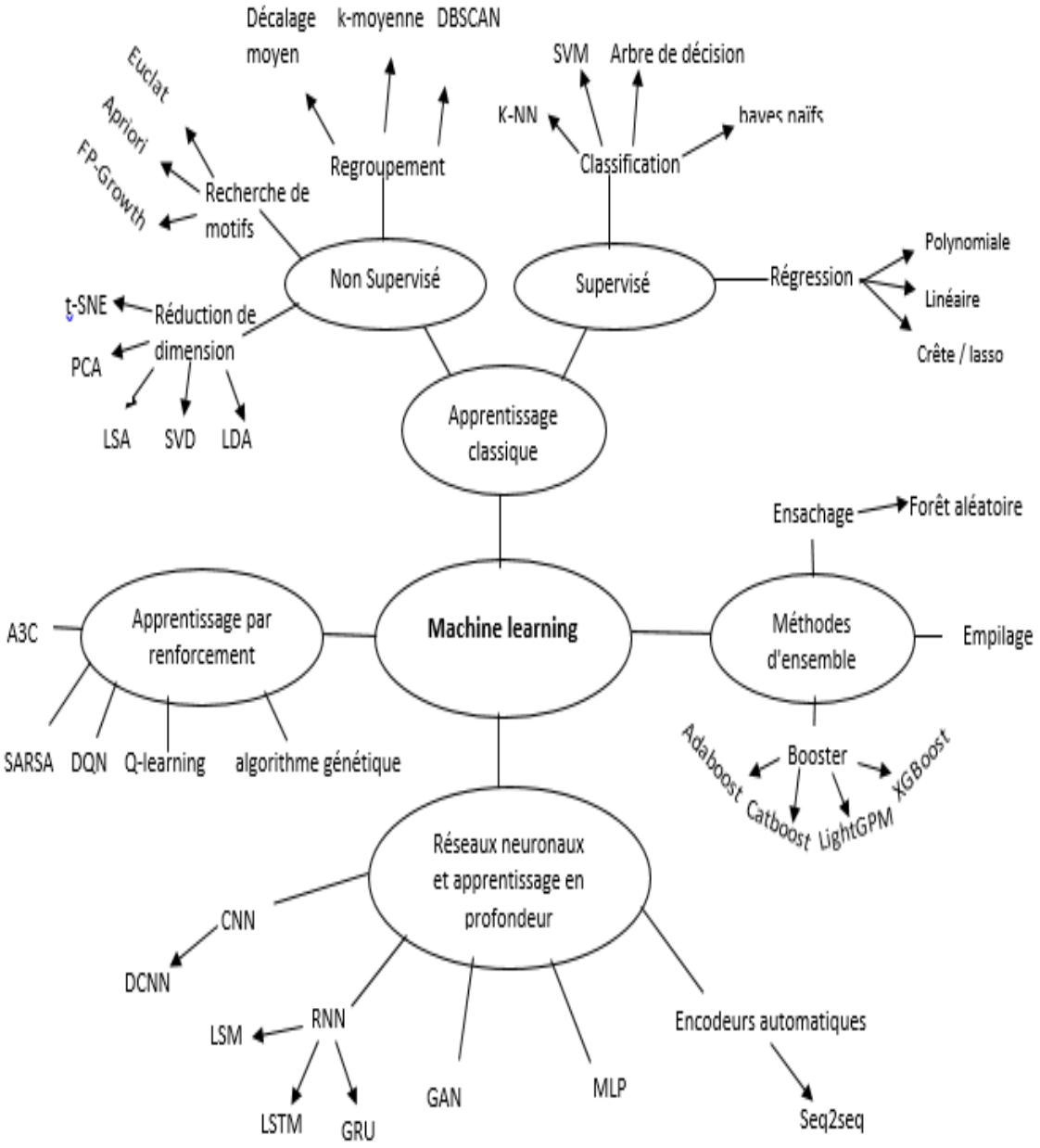

FIGURE 2.1 – Les classes de machine learning

2.3 Les classes de machine learning

Le machine learning est divisé à quatre classes principales :

2.3.1 Méthode d'ensemble

Méthode d'ensemble est l'une des techniques de machine learning les plus utilisés récemment dans différents domaines. Elles combinent plusieurs algorithmes afin d'obtenir un modèle prédictif optimal.

Méthode d'ensemble se base sur trois algorithmes qui sont ensachage (bagging), empilage(stacking) et booster. Ces trois algorithmes sont classés en deux groupes qui sont :

1. **Méthode d'ensemble séquentielle** : dans ce groupe les apprenants de base sont générés séquentiellement. La motivation fondamentale des méthodes séquentielles est d'exploiter la dépendance entre les apprenants de base (les apprenants de base sont SVM, arbre de décision, régression linéaire, régression logistique...), ces méthodes appelées Boosting.
2. **Méthode d'ensemble parallèle** : dans ce groupe les apprenants de base sont générés en parallèle. La motivation de base des méthodes parallèles est d'exploiter l'indépendance entre les apprenants de base puisque l'erreur peut être réduite de façon spectaculaire en moyenne, ces méthodes appelées Ensachage(Bagging).

Le principe de chaque algorithme est comme suite :

- Ensachage :

En anglais est appelé Bagging est une contraction de Bootstrap Aggregation. Dans cette méthode on applique sur un ensemble de formation les étapes suivantes :

— Génération de k échantillons indépendants par tirage aléatoire.

— Pour chaque échantillon, nous construisons un modèle classificateur.

— La prédiction finale pour un nouvel exemple est obtenue soit par vote majoritaire (classification) ou par moyenne des prédictions (régression).

L'intérêt majeur de cette méthode est de réduire la variance quand les prédicteurs sont instables, aussi que l'estimation d'erreur de prédiction.

- Boost :

Le boost ou boosting en anglais fait référence à un ensemble des algorithmes itératifs capable se convertit l'apprenant faible en apprenant fort. Le principe de cette méthode est formé séquentiellement les apprenants faibles, chaque itération essaye de corriger son prédécesseur afin d'améliorer les prédictions du modèle.

L'intérêt majeur de cette méthode c'est qu'on peut utiliser avec n'importe quels apprenants faibles.

- Empilage :

Empilage ou stacking en anglais est une méthode qui combine plusieurs modèles des différents types de classification ou de régression. Empilage est moins utilisé que l'ensachage et le boosting.

La procédure de l'empilage est la suivante :

— Diviser l'ensemble d'entrainement en k ensembles.

— Former plusieurs modèles de base sur la première partie et les prédictions de cette partie est utilisée comme entrées dans la deuxième partie.

— Tester les modèles de base dans la deuxième partie et les réponses correctes de cette partie sont utilisés comme sorties pour déterminer les prédictions finales. Cette méthode peut agréger des modèles très différents et d'améliorer sensiblement la qualité de la prédiction finale.

2.3.2 Réseaux neurones et apprentissage en profondeur

Les réseaux de neurones sont un ensemble des algorithmes qui ont révolutionné l'apprentissage automatique, ils s'inspirent des réseaux de neurones biologiques.

Les réseaux de neurones tentent de reconnaitre les relations sous-jacentes dans un ensemble des données. Ils peuvent s'adapter à l'évolution des entrées afin que le réseau génère le meilleur résultat possible sans avoir à repenser les critères de sortie.

Ils sont utilisés dans une variété d'applications dans les services financiers, de la prévision et de la recherche marketing à la détection des fraudes et à l'évaluation des risques, aussi la reconnaissance des formes ou le traitement du langage naturel.

La structure générale des réseaux de neurones se compose à des nœuds qui sont appelé les neurones qui sont organisés en couches. Généralement on a trois couches comme l'indique la figure suivante :

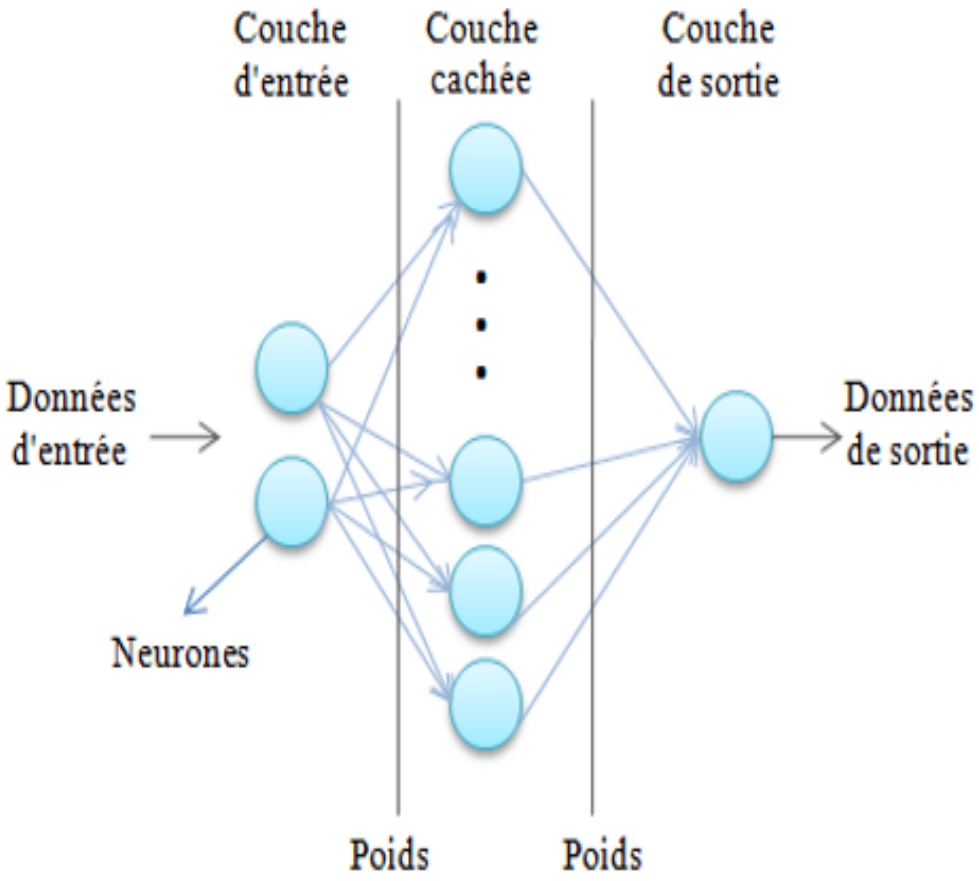

FIGURE 2.2 – La structure de réseau de neurones [28]

Neurones : ils sont un ensemble de fonction prennent une donnée d'entrée et produisent une de sortie. Tous les neurones du même groupe remplissent un type de fonction similaire.

Les couches : pour le groupement des neurones, dans la couche d'entrée les neurones reçoivent les données d'entrée, les traitent et les transmettent aux neurones dans la couche suivante. Le nombre des couches est ça dépend au problème ou domaine d'application.

Dans la couche cachée les neurones calculent de nouvelles données de sortie et les transmettent à la couche suivante. Les neurones de couche de sortie produisent les données de sortie finale.

Il est possible d'avoir un très grand nombre des couches dans le réseau de neurones complexe. Plus il y aura des couches, plus le réseau sera profond qui est connu sous le nom d'apprentissage en profondeur (Deep learning).

Poids : une valeur numérique applique à l'entrée de chacun des neurones pour calculer une donnée de sortie.

Le concept de réseau de neurones repose sur trois étapes principales :

— Pour chaque neurone dans une couche, multiplier la valeur d'entrée par le poids.

— Ensuite, pour chaque couche, additionner toutes les pondérations des neurones et ajouter un biais.

— Enfin, appliquer la fonction d'activation sur cette valeur pour calculer une nouvelle sortie.

Il existe différents types de réseau de neurones. Les deux réseaux de neurones les plus populaires sont :

— **Réseau de neurones récurrent (RNN)** : sont des réseaux de neurones spécialisés qui utilisent le contexte des entrées lors du calcul de la sortie. La sortie dépend des entrées et des sorties calculées précédemment. Les RNN conviennent aux applications où les informations historiques sont importantes.

— **Réseau de neurones de convolution (CNN)** : ces réseaux reposent sur des filtres de convolution (matrices numériques). Les filtres sont appliqués aux entrées avant que celles-ci ne soient transmises aux neurones. Ces réseaux de neurones sont utiles pour le traitement et la prévision d'images.

2.3.3 Apprentissage par renforcement

L'apprentissage par renforcement fait référence à une classe d'apprentissage automatique. Le but de cette classe est d'apprendre, à partir d'expériences successives ce qu'il convient de faire de façon à trouver la meilleure solution.

Cette classe devient très utilisée dans des divers domaines comme robotique autonome, économie, recherche opérationnelle, jeux, optimisation fondée sur la simulation, systèmes multi-agent, intelligence distribuée, statistiques et des algorithmes génétiques...

L'apprentissage par renforcement consiste, d'un agent (algorithme, robot, ...) qui interagit avec l'environnement dynamique pour trouver la solution optimale pour un problème donné. Pour atteindre la solution optimale, l'agent doit appliquer les étapes suivantes dans l'environnement :

— Il observe les effets de ses actions, cette étape est appelé l'exploration.

— Il adapte son comportement, c'est l'étape de l'exploitation.

— Il améliore ses actions futures, c'est l'étape d'apprendre.

Après ces étapes si la performance de l'agent est correcte il reçoit une récompense et si le contraire il reçoit une pénalité.

Il existe deux types d'apprentissage par renforcement :

— **Monte-Carlo** : le programme reçoit ses récompenses à la fin de l'état terminal.

— **Machine learning par différence temporelle (TD)** : les récompenses sont évaluées et accordées à chaque étape.

En plus de l'agent et de l'environnement, on peut identifier trois éléments principaux d'un système d'apprentissage par renforcement quel sont :

— **La politique** : elle définit la manière de se comporter de l'agent à un instant donné, c'est-à-dire quelle action effectuée à cet instant.

— **La fonction de récompense** : elle correspond au but du problème considéré. Elle associe chaque état de l'environnement à un nombre, la récompense. Elle définit quels sont les événements mauvais et les événements bons pour l'agent.

— **La fonction de valeur** : elle spécifie ce qui est bon à long terme. Pour parler approximativement, la valeur d'un état est la quantité totale de récompense qu'un agent peut s'attendre à accumuler dans le futur en partant de cet état.

La structure générale d'apprentissage par renforcement est comme l'indique dans la figure suivante :

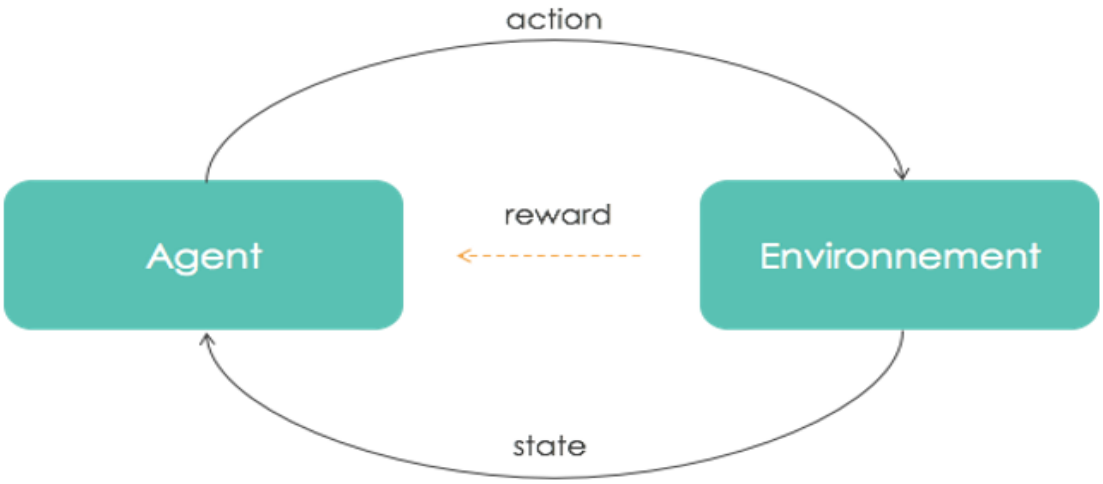

FIGURE 2.3 – La structure d'apprentissage par renforcement [31]

Les principaux algorithmes du machine learning par renforcement sont les suivants : Q-learning, Deep Q Network (DQN) et State-Action-Reward-State-Action (SARSA), algorithme génétique.

2.3.4 Apprentissage classique

L'apprentissage classique principalement est divisé à deux méthodes, sont :

— **Classification** : quand la variable à prédire prend une valeur discrète, on parle d'un problème de classification. Parmi les algorithmes de classification, on retrouve : machine à vecteur de support(SVM), arbre de décision, naïve bayes, k-plus proche voisin(KNN).
Quand l'ensemble des classes dépasse deux classes, on parle de classification multi-classes.

— **Régression** : quand la variable à prédire est une valeur spécifique, on parle d'un problème de régression, un algorithme de régression permet de trouver un modèle (une fonction mathématique) en fonction des données d'entrainement (données d'entrées). Le modèle calculé permettra de donner une estimation sur une nouvelle donnée non encore vue par l'algorithme (qui ne faisait pas partie des données d'entrainement).

Les algorithmes de régression peuvent prendre plusieurs formes en fonction du modèle qu'on souhaite construire. La régression linéaire est le modèle le plus simple, il consiste à trouver la meilleure droite qui s'approche le plus des données d'entrée. Parmi les algorithmes de régression, on retrouve : régression linéaire, régression polynomiale, régression crête...

L'apprentissage classique est regroupé en deux catégories principales, sont :

1. **Apprentissage non supervisé :**
   L'apprentissage non supervisé consiste à ne disposer que des données d'entrée et pas de variables de sortie correspondantes. L'objectif de l'apprentissage non supervisé est de modéliser la structure ou la distribution sous-jacente dans les données afin d'apprendre davantage sur les données.

   Il existe trois types d'apprentissage non supervisé :

   — **Recherche de motifs frèquent** : il est aussi connu comme reconnaissance de formes ou reconnaissance de motifs, cette technique est définie comme un processus pour identifier les régularités dans les données. Ce processus peut se faire physiquement, mathématiquement ou par l'utilisation d'algorithmes.
   La reconnaissance des formes est largement utilisée dans les domaines techniques comme la vision par ordinateur, la reconnaissance vocale, la reconnaissance faciale, etc.
   Parmi les algorithmes de recherche de motifs on a : Euclat, apriori, FP-Growth.

   — **Réduction de dimension** : cette méthode permettant de projeter des données issues d'un espace de grandes dimension dans un espace de plus petite dimension. La réduction de dimension permet de réduire la complexité d'un problème à plusieurs niveaux quel sont :
   Au niveau théorique : cela entraine automatiquement une amélioration de propriété de stabilité et de robustesse des algorithmes.
   Au niveau pratique : cela permet de limite le nombre de possibilités à tester, en réduisent l'espace des solutions, ce qui permet de traiter les données plus rapidement.
   Parmi les algorithmes de réduction de dimension on retrouve : t-SNE, PCA, LSA, SVD et LDA.

— **Regroupement (Clustering)** : La mise en groupe (cluster) consiste à séparer ou à diviser un ensemble de données en un certain nombre de groupes, fonction de regroupement base sur grouper les données en fonction de leurs similitudes. Parmi les algorithmes de regroupement, on trouve : décalage moyenne, DBSCAN, k-moyenne.

Dans cette partie on a choisi de présenter l'algorithme k-moyennes. Ce dernier c'est l'algorithme le plus facile à comprendre et à utiliser dans ce type. Il se base sur le regroupement, on lui donne un ensemble des données en entrée, et un nombre de groupes k et il va segmenter en K groupes ces données. Les individus de chaque groupe auront un degré de similarité, chaque point de données appartient à un seul groupe.

Le moyen dans les K-moyennes fait référence à la moyenne des données, c'est à-dire trouver le centre de gravité.

Le principe de k-moyennes est basé sur les points suivants :

— On définit k le nombre de groupes.

— On tire aléatoirement k individus. Ces k individus correspondent aux centres initiaux des k groupes.

— On calcule la distance entre les individus et chaque centre.

— On affecte chaque individu au centre le plus proche.

— On calcule les centres de gravité des groupes qui deviennent les nouveaux centres.

— On recommence les étapes précédentes tant que les individus sont réaffectés à des nouveaux groupes après une itération.

Parmi les avantages de cet algorithme on a :

— Simple et facile d'implémenter.

— Flexible, s'adapte aux divers changements de vos données.

— Convient à un grand nombre d'ensembles de données et est calculé beaucoup plus rapidement que le plus petit.

— Faible coût de calcul, et améliore la précision de la classification.

Ses inconvénients sont :

— Il ne peut pas être exécuté que dans des données métriques.

— Il ne permet pas de développer un ensemble optimal du groupe.

— La solution peut dépendre de l'ordre des individus.

— Il très sensible au les points éloignée.

2. **Apprentissage supervisé :**

L'apprentissage supervisé est une technique qui utilise des données d'entrée et des données attendues en sortie. Le but principal de cette technique est de trouver une fonction qui mappe les entrées et les sorties qui s'appelle un modèle de prédiction. Les données d'entrée peuvent être des valeurs numériques, alphanumériques, des images. . . , et les données de sortie (variable à prédire) peuvent être l'un des deux catégories suivantes :

— **Variable discrète** : les données de sortie peuvent prendre une valeur d'un ensemble fini de valeurs qu'on appelle des classes.

— **Variable continue** : les données de sortie peuvent prendre n'importe quelle valeur.

Parmi les algorithmes d'apprentissage classique supervisé on trouve :

— **SVM (Séparateur à Vaste Marge)**

Les SVMs sont une famille d'algorithmes d'apprentissage automatique qui permettent de résoudre des problèmes tant de classification que de régression ou de détection d'anomalie.il ont pour but de séparer les données en classes à l'aide d'une frontière aussi simple que possible appelée hyperplane, de telle façon que la distance entre les différents groupes de données et le hyperplane qui les sépare soit maximale. Cette distance est aussi appelée « marge » et les SVMs sont ainsi qualifiés de « séparateurs à vaste marge », les « vecteurs de support » étant les données les plus proches de hyperplane.

L'algorithme se base principalement sur trois astuces pour obtenir de très bonnes performances tant en qualité de prédiction qu'en complexité de calcul :

— Les points d'entrées sont le vecteur de support.

— On cherche l'hyperplan pour mieux séparer les points de la variable d'entrée par leur classe.

— Calculer la distance entre l'hyperplan et les points de données les plus proches qui est appelée la marge. L'hyperplan optimal pouvant séparer les deux classes est la ligne qui présent la marge la plus grande.

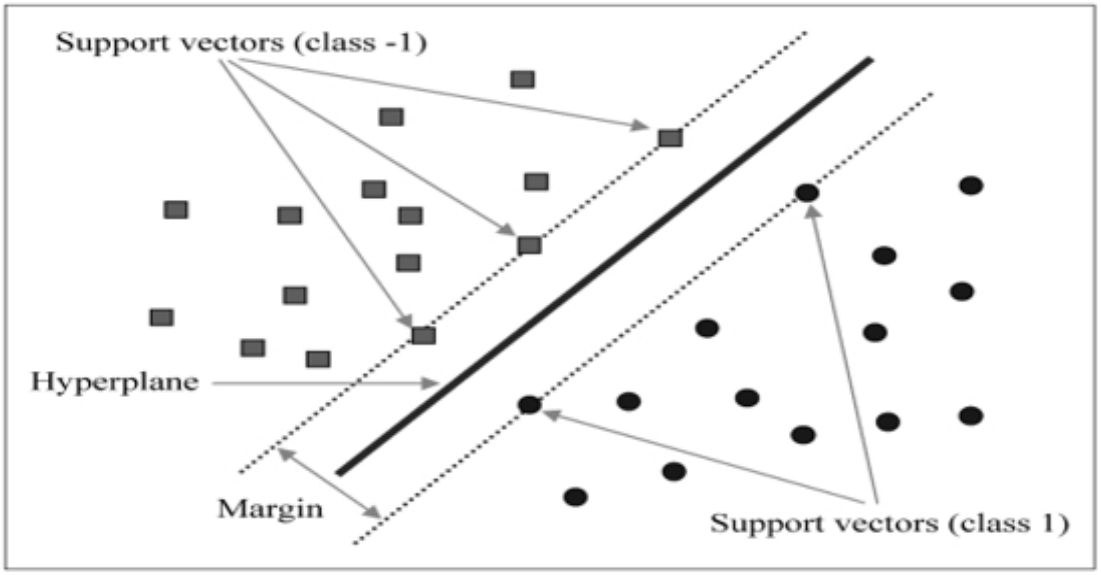

FIGURE 2.4 – La structure de SVM [37]

Les avantages de machine à vecteur de support sont :

— Sa grande précision de prédiction.

— Il fonctionne bien sûr de plus petits data sets.

— L'utilisation des sous-ensembles des points d'entrainements faites-en plus efficace.

Ses inconvénients sont :

— Il est moins efficace sur les jeux de données contenant du bruit et beaucoup d'aberrantes, sensible à les points éloignée.

— **Arbre de décision**

Les arbres de décision sont l'un des algorithmes d'apprentissage automatique supervisé, où les données sont continuellement divisées selon un certain paramètre. L'arbre peut être expliqué par deux entités, à savoir les nœuds de décision et les feuilles. Les feuilles sont les décisions ou les résultats finaux, et les nœuds de décision sont où les données sont divisées.

Ils sont capables de découvrir des interactions complexes entre des variables et de faire des prédictions précises sur de nouvelles données. Il existe deux principaux types d'arbre de décision en fouille des données :

- Les arbres de classification (oui/non type) : permettent de prédire à quelle classe la variable-cible appartient, dans ce cas la prédiction est une étiquette de classe,
- Les arbres de régression (type des données en continu) : permettent de prédire une quantité réelle (par exemple, le prix d'une maison ou la durée de séjour d'un patient dans un hôpital), dans ce cas la prédiction est une valeur numérique.

Les arbres de décision utilisent plusieurs algorithmes. La sélection de l'algorithme utilisé est basée sur le type des variables cibles, parmi les algorithmes qu'il utilise ID3, C4.5, CART et MARS.

Le principe d'arbre de décision se base sur un modèle de graphe (les arbres) pour définir la décision finale où :

- Chaque nœud interne représente un test sur un attribut (nœud de décision).
- Chaque branche représente le résultat de test.
- Chaque feuille représente une étiquette de classe (décision prise après avoir calculé tous les attributs nœud terminal).

Les chemins de racine en feuille représentent des règles de classification.

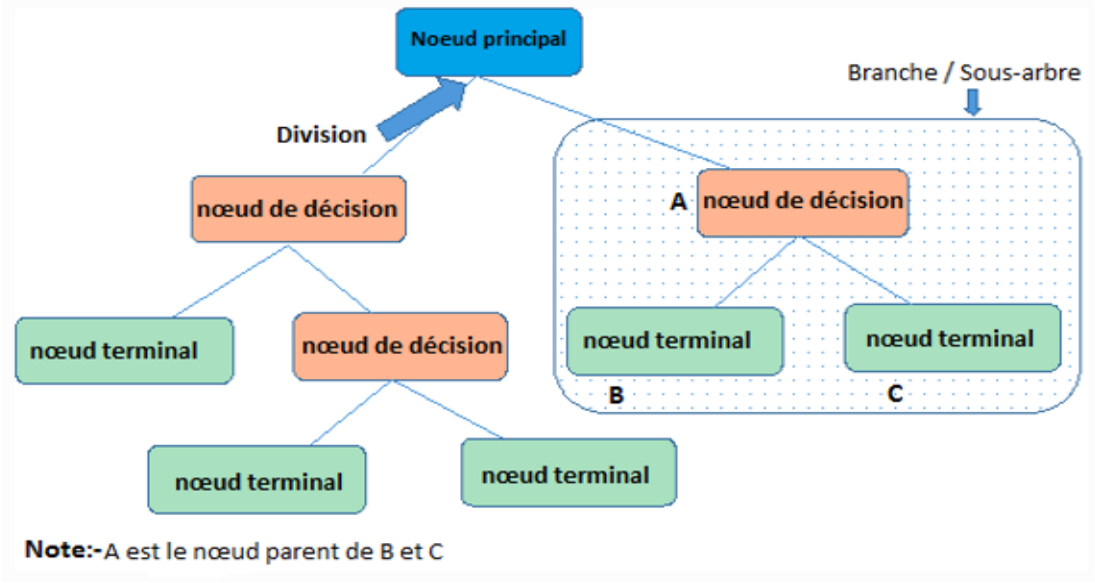

FIGURE 2.5 – La structure d'arbre de décision [40]

Les avantages de cet algorithme sont :

— Il accepte les données numériques et catégorielles et peut gérer les valeurs manquantes.

— Interprétation simple du modèle ajusté.

— Les algorithmes sont rapides et peuvent gérer de grands ensembles de données.

Ses inconvénients sont :

— Il Peut-être instable (variance élevée) donc des petits changements dans l'ensemble de données peuvent produire des modèles significativement différents.

— Impossible de modéliser facilement certains types de relations, par exemple une relation linéaire simple entre deux variables.

— Habituellement besoin d'ajustement soigneux par l'élagage.

2.4 Conclusion

Puisqu'il existe une grande variété d'algorithmes d'apprentissages automatique, tous les débutants posent une question typique, est la suivante : « quel algorithme dois-je-utiliser ? ». La réponse sur cette question est en fonction de nombreux de facteurs, notamment :

— La taille, la qualité et la nature de données.

— Le domaine.

— Le temps de calcul disponible.

— L'urgence de la tâche.

— Ce que vous voulez faire avec les données.

Sur la base de ces facteurs on a choisi la méthode SVM pour implémenter notre système.

Chapitre 3

L'implémentation du système BanDos

3.1 Introduction

Dans ce chapitre, nous y présentons d'abord les outils utilisés pour le développement de système BanDos. Nous présenterons ensuite l'architecture et le fonctionnement du système, et à la fin, nous détaillions les étapes suivies pour l'implémentation du système BanDos.

L'implémentation du système BanDos est fait sous la plateforme E-learning de l'université Ammar Telidji.

3.2 Outils de réalisation

Les principaux outils utilisés pour la mise en place de notre système sont :

— Le langage script AWK pour créer les scripts de filtrage de fichiers log et pour la mise à jour de la liste noir.

— Le langage Python 3.7 pour coder la méthode SVM sous l'application web Jupytre.

— Une logiciel LOIC (Low Orbit Ion Cannon) afin d'établir l'attaque DDOS.

3.3 Architecture du BanDos

Le système BanDos est exécuté périodiquement en parallèle avec le processus qui consiste à satisfaire les requêtes d'utilisateur. Il se compose de quatre modules. La figure suivante représente la séquence de ces modules.

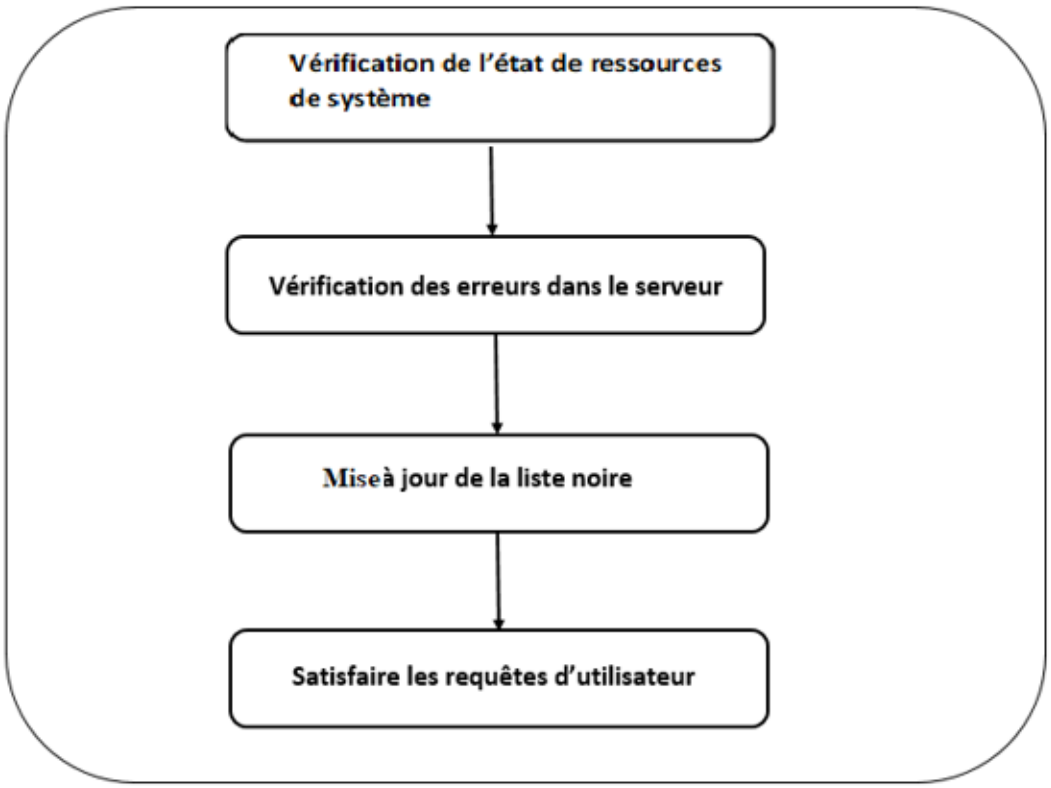

FIGURE 3.1 – Architecture de système BanDos

1. **Vérification de l'état de ressources de serveur :**
   Chaque serveur web possède un nombre fini de ressources (pages php, image, des documents), au niveau de ce module le système BanDos vérifiait la disponibilité et l'état de ces ressources.
2. **Vérification des erreurs dans le serveur :**
   Les codes d'état HTTP sont des nombres de trois chiffres qui fournissent des informations sur l'état des pages aux navigateurs Web. Ce module consiste à vérifier les erreurs au niveau du serveur, à ce stade nous nous sommes concentrés sur les erreurs de serveur web, les erreurs HTTP appartiennent à la catégorie des 400 (accès refusé) et 500 (erreur serveur).

Les 400 sont des erreurs côté client (le navigateur) et les 500 sont des erreurs côté serveur.

3. **Mise à jour de la liste noire :**
   L'utilisateur est identifié par son adresse IP, lorsqu'il se connecte au serveur nous suivrons son comportement. Le comportement d'un utilisateur est défini par le nombre de ressources demandées ainsi que chaque ressource combien de fois ait été demandée dans un intervalle du temps spécifié.

   Au niveau de ce module nous appliquons la méthode SVM pour classifier si le comportement d'un utilisateur est normal ou non, si le comportement est anormal alors l'adresse IP sera ajoutée à la liste noire.

4. **Satisfaire les requêtes d'utilisateur :**
   Ce module consiste à satisfaire les requêtes d'utilisateur après avoir confirmé que son comportement est normal.

3.4 Vue d'ensemble du BanDos

Les attaques DDOS consiste à épuiser les ressources à disposition d'un serveur, a cet effet dans la réalisation de ce système nous nous sommes concentrés sur les ressources de serveur et les adresses IP connecté, la figure suivante explique le processus qui consiste à satisfaire les requêtes d'utilisateur.

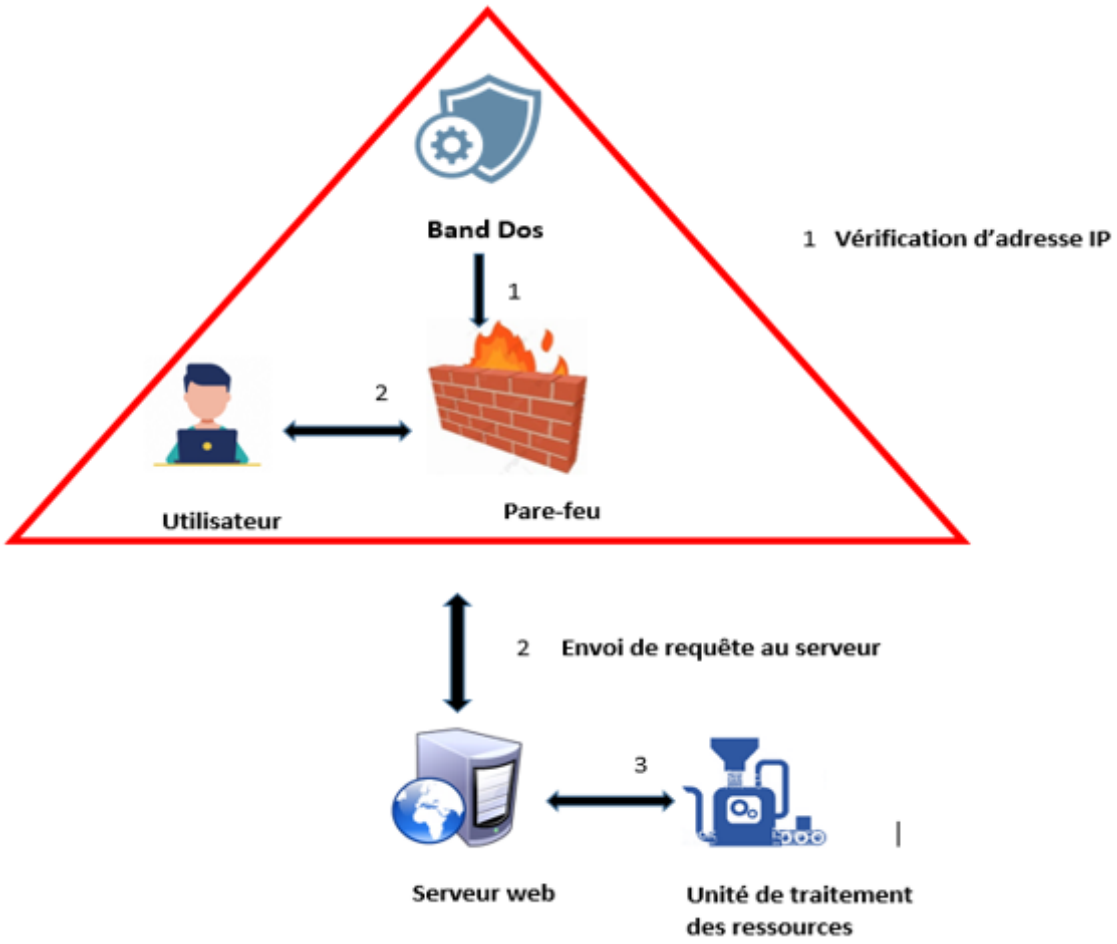

FIGURE 3.2 – Processus de traitement des requêtes

1. Vérification d'adresse IP
   - La gestion de la liste noire des adresses IP est l'une des fonctions du système BanDos. Il consiste à vérifier périodiquement les ressources demandées par un utilisateur par son adresse IP, les adresses IP suspecté sont bloquées et mettent dans la liste noire après certaines étapes. Ensuite, il fait la mise à jour de firewall.
   - Lorsqu'un utilisateur fait une connexion à la plateforme, une vérification d'adresse IP est faite au niveau du pare-feu, si l'adresse IP n'existe pas dans la liste noire alors il peut connecter, sinon la connexion rejetée.
2. Après la vérification d'adresse IP, la requête d'utilisateur est envoyée au serveur de la plateforme pour l'exécution.
3. Le serveur envoyait la requête à l'unité de traitement de ressources pour le traité et renvois la réponse.

3.5 La réalisation du BanDos

Au niveau de la réalisation nous sommes concentrés sur l'attaque Http Flood. L'attaque http flood est un type d'attaque DDOS. Cette attaque utilise des botnets pour envoyés des requêtes http dans un intervalle du temps très court afin de produire un trafic réseau très important qui rend le serveur indisponible.

Les requêtes HTTP sont généralement deux types :

— **HTTP GET** : elle consiste à récupérer du contenu comme des images, des documents, des pages PHP...

— **HTTP POST** : elle consiste à envoyer des données par l'utilisateur au serveur pour créer où faire des mises à jour de données.

Pour cela on a choisi l'adresse IP et les nombres des ressources demandées et le temps comme un vecteur.

Dans ce travail on va utiliser l'attaque HTTP GET FLOOD.

Ce travail a été réaliser en trois phases, comme suite :

3.5.1 Phase d'attaque

LOIC est un programme qui permet de réaliser des attaques DDOS. Ce qui le différencie de la plupart des outil d'attaque de ce type c'est sa facilité d'utilisation. Il propose trois types d'attaques : le Flood HTTP, le flood TCP et le flood UDP [45].

Cet outil a été utilisé sous Windows 10, le fonctionnement de l'outil est comme le montre la figure suivante :

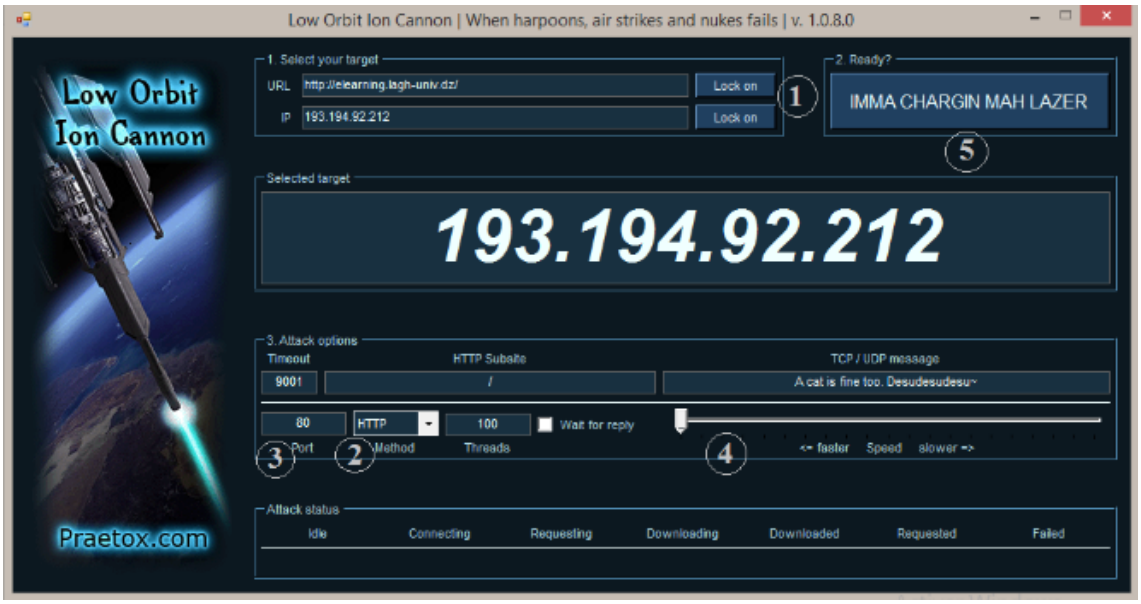

FIGURE 3.3 – Fonctionnement de LOIC

1. Adresse IP de plateforme E-Learning de l'université de Laghouat (193.194.92.212) et on clique sur Lock on.
2. Type d'attaque http.
3. Le port (80).
4. Vitesse d'attaque (maximum).
5. Lancement d'attaque.

3.5.2 Phase de classification

Au niveau de cette phase on a défini le comportement d'utilisateur par combien de fois il demande une ressource ou plusieurs dans un intervalle du temps. À partir de ça on a divisé le comportement à deux :

— **Comportement normal** : Un utilisateur normal accède aux ressources sur un serveur Web en fonction de la structure du site web qu'il visite. Par exemple, s'il accède à un site Web éducatif peut d'abord visiter la page d'accueil, puis cliquer sur la section article et enfin télécharger un article.

— **Comportement anormal** : Un utilisateur anormal accède à la page d'accueil ou télécharger le même ressource (fichier, image,...) plusieurs fois ou plusieurs ressources dans un intervalle du temps très court.

On définit le comportement comme la suite :

Comportement	**Nombre de ressources demandé**	**Temps de demande**
Normal	Moins de 40 ressources	en 2 minutes ou moins
Anormal	40 ressources ou plus	en 2 minutes ou moins

TABLE 3.1 – Table d'échelle d'intégration

Pour catégoriser le comportement des utilisateurs, nous avons basé sur le filtrage de fichier log. À scet effet, nous avons utilisé un script qui s'exécute automatiquement tous les deux minutes.

Le déroulement de script est comme la suite :

— Lire les lignes de deux dernières minutes dans le fichier log «access.log» et les déposer dans le fichier « access1.txt ».

— Prend spécifiquement les requêtes HTTP GET à partie du fichier « access1.txt ».

— Exécute un autre script pour calculer le nombre des ressources demandées par chaque adresse IP dans les deux dernières minutes.

```
#La commande pour lire les lignes de 2 dernières minutes

cat access.log | awk "/^$(date --date="-2 min" "+%b %_d %H:%M")/{p++} p" > access1.txt

#La commande pour prendre spécifiquement les requêtes HTTP GET

 cat access1.txt | grep "GET" |awk '{print $1"     "$7}'> access2.txt

#La boucle pour calculer le nombre des ressources demandées par chaque adresse IP

for ((i=0; i<=50; i++))
 do
  nbrderessource=0
  filename='access2.txt'

  while read line ;
    do
     if [[ $line == *"${ipadress[i]}"* ]]
       then

       let "nbrderessource=nbrderessource+1"

      fi
         done < $filename

     echo ${ipadress[i]} "    " $nbrderessource >> resultat.txt
done
```

FIGURE 3.4 – Script pour calculer le nombre des ressources demandées par chaque adresse IP

Le résultat final de l'exécution de ce script est sauvegardé dans le fichier « resultat.txt », ce dernier est considéré comme l'ensemble de données qu'on va traiter.

L'application de la méthode SVM se fait par les étapes suivantes :

— Définition d'un modèle de prédiction à partir de l'ensemble de données précédent, on trouve dans ce modèle les données d'entrée et les données attendues en sortie. Dans notre travail, nous avons défini l'entrée comme le nombre de fois une adresse IP demande une ressource spécifiée ou plusieurs ressources et comme sortie attendu le comportement de cette adresse. De ce modèle, le SVM apprendra à faire des décisions. Dans la figure suivante nous avons montré une partie de modèle que nous avons utilisé.

	Address-ip	Temps(s)	Nbr_de_ressources	Classe
0	31.13.103.6	1	1	1
1	31.13.103.7	1	1	1
2	41.103.139.145	15	3	1
3	31.13.103.9	20	2	1
4	41.104.79.79	20	98	0

FIGURE 3.5 – Modèle de prédiction

L'apprentissage se fait par une fonction prédéfini dans la bibliothèque Sklearn.

— Traitement d'ensemble de données que nous avons précédemment extraire de fichier log comme la suite :

Lire le fichier « resultat.csv » ligne par ligne.
Classer cette adresse selon la valeur de son comportement.
Si (comportement = 0)
Adresse est anormal
Sinon
Il est normal

3.5.3 Phase de pénalisation

Le résultat extraire précédemment sera utilisé au niveau de cette phase pour pénalisé les adresses IP suspecté, si l'adresse est suspectée il le bloqué par la command IP-table et met dans la liste noire par le script suivant :

```
filename='resultat_svm.txt'

while read line ;
  do
   if [[ $line == *"true"* ]]
     then
  iptables -A INPUT -s $1 -j DROP
  echo "$1" >> liste_noir

    fi

       done < $filename
```

FIGURE 3.6 – Script pour bloquer les adresses IP avec ip-tables

La liste noire sera vidée tous les jours à minuit par les instructions suivantes :

```
heur=$(date)

 if [[ $heur == *"00:00:00"* ]]
     then
 filename='liste_noir'
  while read line ;
   do

     iptables -D INPUT -s $1 -j DROP

   done < $filename

  echo "" > liste_noir

    fi
```

FIGURE 3.7 – Script pour vider la liste noir

3.6 Résultats et analyses

L'évaluation des performances du BanDos est faite en termes d'une classification qui a été obtenue par le SVM ainsi que le délai et le taux de détections.

— **Classification**

Cette classification est présentée sous forme d'un graphe et un tableau pour définir le comportement des adresses IP connecté :

1. La figure suivante est le graphe obtenu par le SVM et il présente le nombre de ressources demandées en fonction de temps. Nous avons considéré la classe positive (rouge) comme le comportement anormal et la classe négative (bleu) comme le comportement normal.

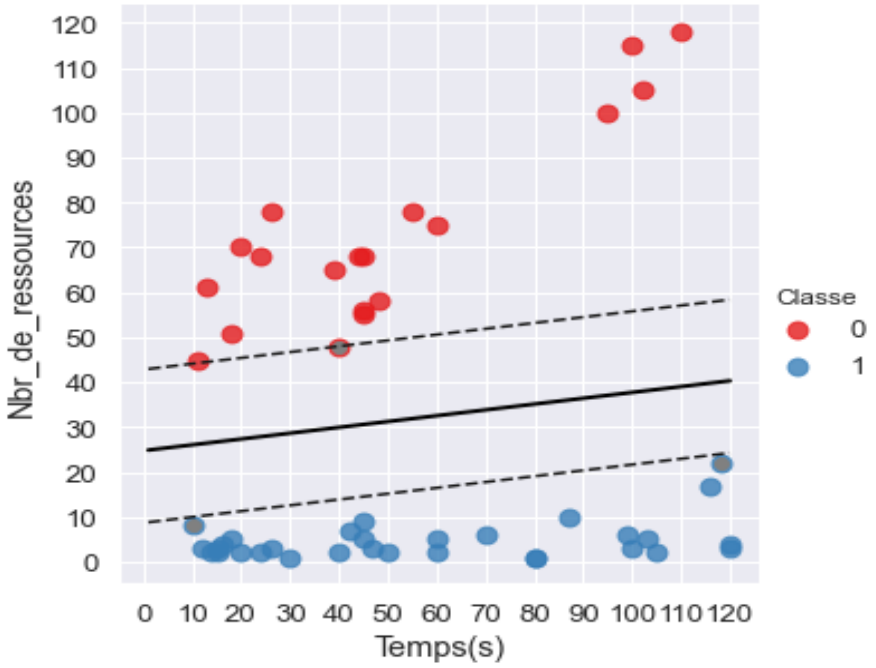

FIGURE 3.8 – Classification des adresses

2. La figure présente le tableau obtenu et il contient deux colonnes :
   — La première colonne présente les adresses IP.
   — La deuxième colonne indique si l'adresse est suspectée ou non (Si oui il affiche True sinon il affiche False).

Address-ip	Suspect
31.13.103.11	False
31.13.103.15	False
41.104.70.132	False
31.13.103.16	False
41.107.5.149	True
41.107.7.129	True
154.121.35.213	True

FIGURE 3.9 – Caractérisation des résultats

— Les adresses suspectées seront bloquées comme suivant :

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:domain
ACCEPT     udp  --  anywhere             anywhere             udp dpt:domain
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:bootps
ACCEPT     udp  --  anywhere             anywhere             udp dpt:bootps
DROP       all  --  41.107.5.149         anywhere
DROP       all  --  41.107.7.129         anywhere
DROP       all  --  154.121.35.213       anywhere
```

FIGURE 3.10 – Le blocage des adresses IP suspectées

— **Délai**

Pour étudier la métrique de délai on a appliqué le système BanDos sur les fichiers logs de serveur avec une durée de 2 minutes pendant 10 jours (2 minutes /jour). Le délai est défini par le temps de traitement de N lignes que nous avons trouvé en 2 minutes.

D'après la figure(3.13) , on remarque que le délai augmente d'une façon linéaire cela peut s'expliquer par le faite que plus le nombre de lignes augmentent plus que le délais croît.

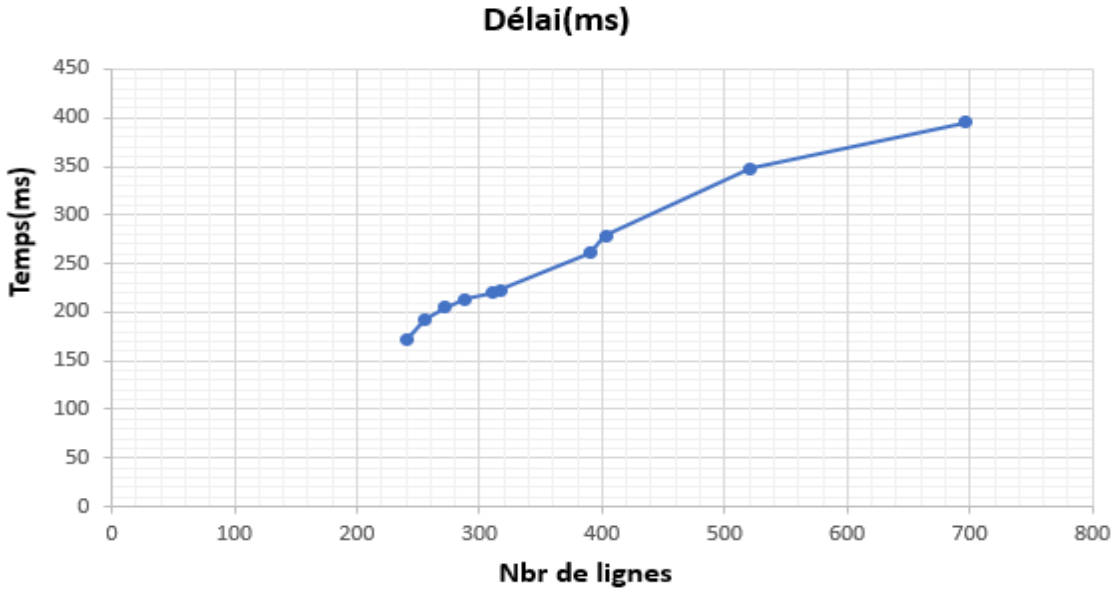

FIGURE 3.11 – Le délai d'exécution des scripts

— **Taux du détections**

On a appliqué le système BanDos sur les fichiers logs de serveur avec une durée de 1 heur pendant 10 jours (1 heur /jour).

C'est le rapport entre le nombre des entrées anormales et le nombre des entrées totales. D'après la figure(3.14) on remarque que le taux de détection se diffère en fonction du comportement des entrées.

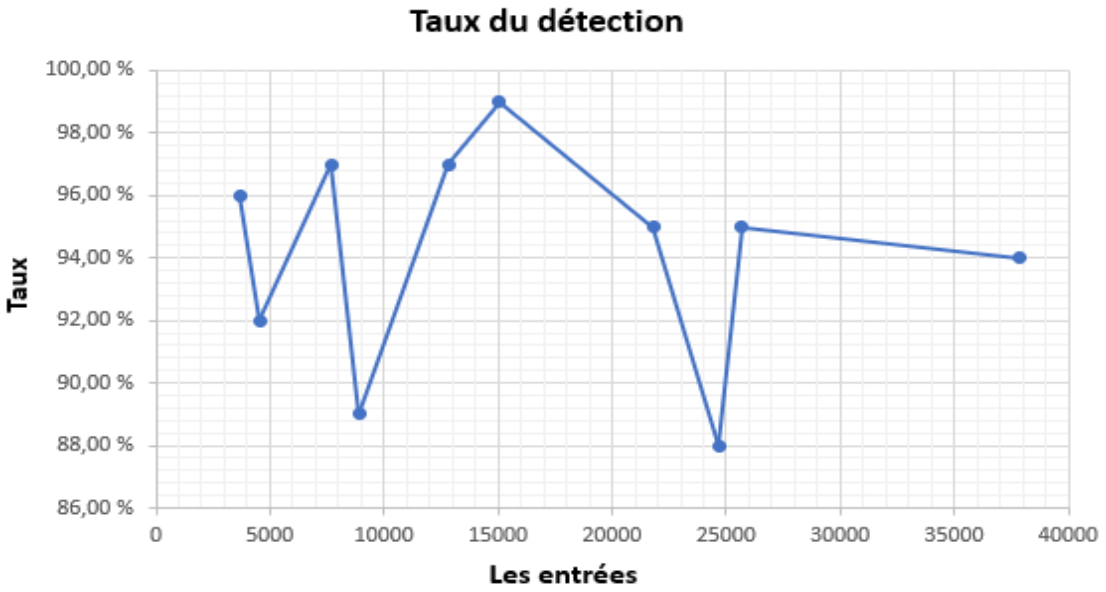

FIGURE 3.12 – Taux du détections

— **Taux du faux positif**

On a appliqué le système BanDos sur les fichiers logs de serveur avec une durée de 1 heur pendant 10 jours (1 heur /jour).

C'est le rapport entre les entrées normales et les entrées anormales détectées totales et qui est nommé par fausse détection.

La figure suivante présente le taux de faux positif en fonction de nombre d'entrées totales, on observe que le fausse détection augmente tout en augmentant le taux de détection.

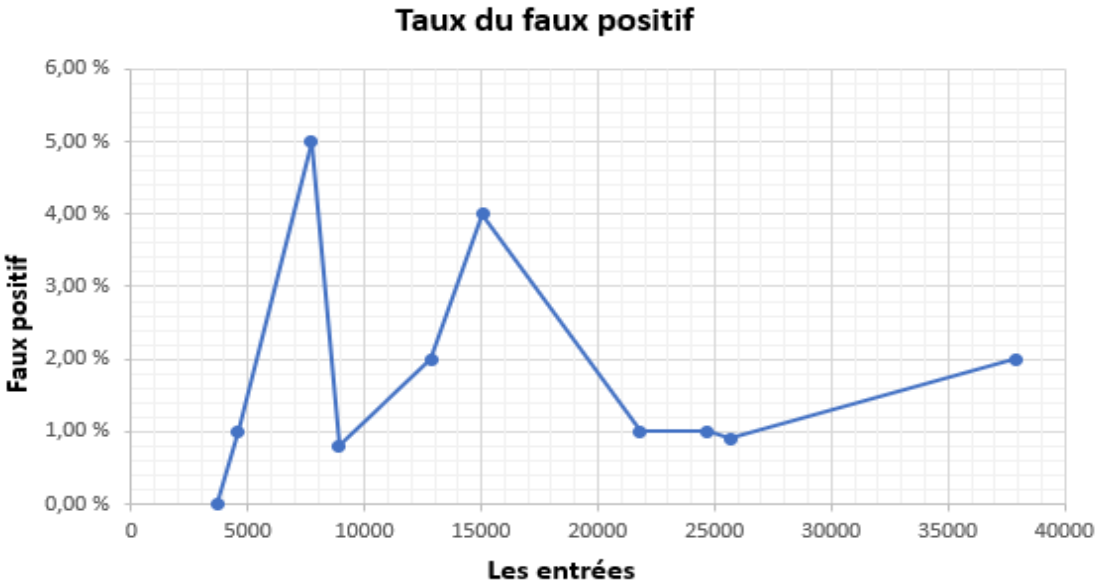

FIGURE 3.13 – Taux du faux positif

3.7 Conclusion

Dans ce chapitre, nous avons décrit brièvement les phases de réalisation de notre système Ban Dos. En effet, nous avons lancé une attaque DDOS avec l'outil LOIC, pendant l'exécution du BanDos nous avons détecté des attaques réelles contre la plateforme. À la fin de l'exécution du BanDos nous avons réussi à bloquer les adresses IP suspectées.

Conclusion générale

L'objectif de notre projet était d'implémenter un système pour protéger les ressources de serveurs web contre les attaques DDOS.

Nous avons choisi la plateforme E-learning de l'université Ammar Telidji pour réaliser notre travail.

Au début de ce projet nous avons présenté l'une des problématiques le plus poser récemment dans le monde web, pour répondre à ce problématique nous avons au départ récolté des informations générales sur la sécurité informatique ainsi que les serveurs web et son architecture. Par la suite, nous avons introduit le machine learning et ces classes. Nous nous sommes concentrée sur la classe de l'apprentissage classique et ces algorithmes.

Pour établir la partie suivante, nous avons choisi la méthode de l'apprentissage classique le SVM. Notre choix était sur la base de sa grande précision de prédiction et son bon fonctionnement sûr le plus petit ensemble de données.

La réalisation de notre système a été affecté par la situation actuelle due au COVID-19, à cet effet nous avons simulé l'attaque DDOS contre le serveur de la plateforme E-learning par utilisation d'outil LOIC.

Le système BanDos est lancé automatiquement chacun les deux minutes en parallèle avec le processus qui consiste à satisfaire les requêtes des utilisateurs, il extraira l'ensemble de données à partir de fichiers log pour classifier le comportement des utilisateurs par la méthode SVM. Ensuite, les adresses qui appartiennent à la classe suspectée seront pénalisées par l'utilisation de IP-table.

À la fin de ce travail, nous avons pu détecter les attaques DDOS lancées contre la plateforme E-learning et les bloquer.

Avantages et perspectives d'avenir

Dans le cadre de ce projet, nous avons eu l'opportunité de :

— Mettre en oeuvre nos connaissances pour détecter l'attaque DDOS.

— Familiariser avec des algorithmes de machine learning telles que SVM.

Pour cela, il est intéressant d'améliorer le système BanDos pour obtenir une nouvelle version contient les nouvelles fonctionnalités suivantes :

— L'amélioration de système BanDos pour détecter plusieur type d'attaques réseaux.

— L'intégration de système BanDos dans le système de serveur.

Bibliographie

[1] Comprendre la sécurité informatique(s.d.).redhat.com. https ://www.redhat.com/fr/topics/security (Vue le 14/08/2020).

[2] S.(2017, 20 mai). Présentation des différentes notions pour étudier la problématique de la sécurité des informations. SecuriteInfo.com.

[3] J.(2015, 26 mai). Introduction à la sécurité informatique. CommentCaMarche. https ://www.commentcamarche.net/contents/1033-introduction-a-la-securite-informatiquela-confidentialite (Vue le 14/08/2020).

[4] https ://www.itpro.fr/qu-est-ce-que-cryptage-donnees/

[5] Pirio, M. (2004). Linux Red Hat Fedora TCP/IP. ENI.

[6] E Ramadhani(Éd.).(2017, septembre). Anonymity communication VPN and Tor :a comparative study. IOP.

[7] David Burgermeister, Jonathan Krier. (2006). Les systèmes de détection d'intrusions. devellopez.com, 6.

[8] Asma CHIKH, Amina DJENNANE, « Sécurité d'une application Web à l'aide d'un système de détection d'intrusions comportementale » Mémoire de master, Université Abou Bakr Belkaid– Tlemcen ,2012.

[9] Ihsane MOUTAIB Lamia ELOFIR. (2004). Les Protocoles de sécurité dans les réseaux WiFi. Telecom lille. (Vue le 15/08/2020).

[10] BOUGHAZI Manal,LAKHAL Asma. (2017, juin). Attaque de l'homme du milieu dans les réseaux sociaux 4G. Université de 8 Mai 1945 – Guelma. Mémoire.

[11] ANSSI. (2015). Comprendre et anticiper les attaques DDoS. ANSSI, 8 12. https ://www.ssi.gouv.fr/uploads/2015/03/NP$_{G}uide_{D}DoS.pdf(Vuele$15/08/2020).

[12] J,Attaque par reflexion(Smurf). CommentCaMarche. https ://www.commentcamarche.net/contents/
attaque-par-reflexion-smurf (Vue le 16/08/2020).

[13] http ://jp.barralis.com/linux-man/man8/iptables.8.php (Vue le 16/08/2020).

[14] 11 IONOS SARL. (2017, 5 juillet). Tutoriel iptables : des règles pour les paquets de données. IONOS Digital Guide. https ://www.ionos.fr/digitalguide/serveur/outils/tutoriel-iptables-des-regles-pour-les-paquets-de-donnees/ (Vue le 15/08/2020).

[15] ZOUBIR AICHA. (2014, octobre). Conception et implémentation d'un site web dynamique pour la gestion des appels D'offres. l'Universite Abou Bekr Belkaid Tlemcen. Mémoire.

[16] ELARBI Nassim TAHAR DJEBBAR Mohamed. (2009, juin). Traitement et exploration du fichier Log du serveur web pour l'extraction des connaissances. Université Hassiba Benbouali. Mémoire.

[17] Traitement et exploration du fichier Log du serveur web pour l'extraction des connaissances par Nassim et Mohamed ELARBi etTAHAR DJEBBAR (juin 2009). Université Hassiba Benbouali Chlef (Vue le 16/08/2020).

[18] qu'est-ce qu'une attaque DDOS. (s. d.). OVHcloud. , https ://www.ovh.com/fr/anti-ddos/principe-anti-ddos.xml (Vue le 17/08/2020).

[19] Machine Learning et BigData : définition et explications - LeBigData.fr https ://www.lebigdata.fr/machine-learning-et-big-data (Vue le 03/03/2020)

[20] W Murdoch, C Singh, K Kumbier, R Abbasi-Asl, B Yu. (2019). Definitions, methods, and applications in interpretable machine learning.

[21] Anticipation des changements de notes des obligations du portefeuille d'un assureur par méthode de machine learning. (2017). Morgane LAUR, 5.

[22] Kévin Bailly (2009). Méthodes d'ensembles : Boosting, bagging et random forests.

[23] introduction to ensemble methods. (2018,16 août). Medium.com https ://medium.com/@dalmas.otieno/introduction-to-ensemble-methods-aca988f25fcb (Vue le 15/03/2020)

[24] a quick guid to boosting in ml. (2018, 21 mars).medium.com. https ://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5 (2020, 15 mars)

[25] Stacking in Machine Learning. (2019, 20 mai). GeeksforGeeks. https ://www.geeksforgeeks.org/stacking-in-machine-learning/ (Vue le 15/03/2020).

[26] R. (2020, 1 avril). Comprendre les réseaux de neurones. MonCoachData. https ://moncoachdata.com/blog/comprendre-les-reseaux-de-neurones/ (Vue le 03/04/2020).

[27] James LE.A Gentle Introduction to Neural Networks for Machine Learning. (2018).

[28] ahlem ferchichi . Propagation and reduction of uncertainties in land cover models (2018, 22janvier) Figure 5-Structure de modèle de réseau de neurones artificiels. (Vue le 03/04/2020).

[29] Apprentissage par renforcement. (2017, 6 juin). Data Analytics Post. https ://dataanalyticspost.com/Lexique/apprentissage-par-renforcement/ (Vue le 26/03/2020).

[30] Apprentissage par renforcement – de la théorie à la pratique | OCTO Talks ! (2012, 5 avril). blog.ocoto.com. https ://blog.octo.com/apprentissage-par-renforcement-de-la-theorie-a-la-pratique/ (Vue le 26/03/2020).

[31] Baptiste Saintot, Baptiste O'JEANSON. (2019, 20 février). [Idée générale de l'apprentissage par renforcement]. OCTO Talks ! https ://blog.octo.com/lapprentissage-par-renforcement-demystifie/ (Vue le 26/07/2020).

[32] Siva, C.(2019, 7 juin). Machine Learning and Pattern Recognition. dzone.com. https ://dzone.com/articles/machine-learning-and-pattern-recognition (Vue le 27/03/2020).

[33] Réduction de dimensionnalité. (2018, 13 décembre). Data Analytics Post. https ://dataanalyticspost.com/Lexique/reduction-de-dimensionnalite/ (Vue le 27/03/2020).

[34] An Introduction to Support Vector Machines and Other Kernel-based Learning Methods (20016) Nello Christianini and John Shawe-Taylor.103 115.

[35] Laurent Younes, Sebastien Gadat. (2008, 3 juillet). A Stochastic Algorithm for Feature Selection in Pattern Recognition. Journal of Machine Learning Research. https ://www.jmlr.org/papers/volume8/gadat07a/gadat07a.pdf.

[36] Nasaoui, H. Les 10 plus populaires algorithmes du machine learning. fr.slidshare.net.(2020, 7 mars). https ://fr.slideshare.net/HakimNasaoui/les-10-plus-populaires-algorithmes-du-machine-learning-135086708 (Vue le 27/03/2020).

[37] Chen, S.-T., Hsiao, Y.-H., Huang, Y.-L., Kuo, S.-J., Tseng, H.-S., Wu, H.-K., Chen, D.-R. (2009). Comparative Analysis of Logistic Regression, Support Vector Machine and

Artificial Neural Network for the Differential Diagnosis of Benign and Malignant Solid Breast Tumors by the Use of Three-Dimensional Power Doppler Imaging. Korean Journal of Radiology, 10(5), 464 (Vue le 05/03/2020).

[38] Gupta, P. (2018, 20 juin). Decision Trees in Machine Learning - Towards Data Science. Medium. https ://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 (Vue le 26/03/2020).

[39] Mayur Kulkarni. (2017, 7 septembre). Decision Trees for Classification : A Machine Learning Algorithm. XORIANT. https ://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html (Vue le 12/03/2020).

[40] India, S. (2020, 3 juillet). Machine Learning Algorithms Hands-on Training | SpringboardIN Blog. Springboard Blog. https ://in.springboard.com/blog/machine-learning-algorithms-decision-tree-random-forest/ (Vue le 25/07/2020).

[41] Al-Masri, A. (2019, 15 mai). How Does k-Means Clustering in Machine Learning Work ? Medium. https ://towardsdatascience.com/how-does-k-means-clustering-in-machine-learning-work-fdaaaf5acfa0 (Vue le 29/03/2020).

[42] Benzaki, Y. (2018, 12 avril). Tout ce que vous voulez savoir sur l'algorithme K-Means. Mr. Mint : Apprendre le Machine Learning de A à Z. https ://mrmint.fr/algorithme-k-means (Vue le 01/04/2020).

[43] Issarane, H. (2019, 5 mars). K-means : Definition Avantages / Inconvénients. Le DataScientist. https ://le-datascientist.fr/k-means (Vue le 01/04/2020).

[44] Inter blocage de processus. Y. CHALLAL, H. BETTAHAR, M. VAYSSADE.

[45] Aoudia Manel Lahouassa sahar. (2019). Mise en place d'une plateforme pour la détection des attaques DOS. Université SAAD DAHLAB de Blida .Mémoire.