RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Abderrahmane Mira - Bejaia
Faculté des Sciences Exactes
Departement d'informatique

Mémoire du Projet de Fin d'Études

En vue de l'obtention du diplôme de master Professionnel en informatique

Option : Administration et Sécurité des Reseaux

Thème

Approche par classification du problème de détection de botnets.

Réalisé par :

Messouaf Yacine

Composition du jury :

Président : Mme Tahakourt Zineb - U.A/Mira Bejaia

Examinateur : Mme El bouhissi brahami houda - U.A/Mira Bejaia

Encadreur : Mr. AMROUN Kamal - U.A/Mira Bejaia

Co-Encadreur : Mr. EL SAKAAN Nadim - U.A/Mira Bejaia

Promo 2019/2020

REMERCIEMENTS

Dieu merci pour la santé, la volonté, le courage et la détermination qui m'ont accompagné tout au long de la préparation de ce mémoire.

Je tiens à remercier mon co-encadreur Mr El-SAKAAN Nadim pour ses précieuses orientations ainsi que Mr amroune kamal d'avoir accepté d'être mon encadreur.

Je remercie également les membres de jury d'avoir consacré de leur temps pour l'évaluation de mon modeste travail.

En ce moment précis, toutes mes pensées vont vers mes honorables parents en reconnaissance à leur esprit de sacrifice et de dévouement ainsi qu'à leur soutien constant moral et matériel et ce, pour m'avoir permis de construire un avenir certain.

A la fin, je remercie tous ceux qui ont contribué à la réalisation de ce projet de près ou de loin.

DÉDICACES

À mes chers parents, ma mère et mon père, pour l'éducation qu'ils m'ont prodiguée avec tous les moyens et au prix de tous les sacrifices qu'ils ont consentis à mon égard, pour leur patience, leur amour et leurs encouragements. Que ce travail leur apporte joie et fierté.

À mes chers frères Ishek et Othman.

À l'âme du défunt ma chère grand-mère et à toute ma famille.

À mon chère enseignant de mathématique hakim.

À tous mes enseignants de département d'informatique.

À mes très chers amis : hichem, mohamed, Abdelhak, yanis, layachi, tamouh, lamri, islem, imad, fateh, walid, djamel, nassim, oussama, lamine et notamment dhirar et nassim pour leurs aides et soutiens.

Yacine

TABLE DES MATIÈRES

LISTE DES ABRÉVIATIONS

TIC : Technologies de l'information et de la communication
c&c : Command and Conquer
HTTP : Hypertext Transfer Protocol
OSI : Open Systems Interconnection
TCP : Transmission Control Protocol
IP : Internet Protocol
P2P : peer to peer
IoT : Internet of Things
W3C : World Wide Web Consortium
IETF : Internet Engineering Task Force
IEEE : Institute of Electrical and Electronics Engineers
ETSI : European Telecommunications Standards Institute
COAP : Constrained Application Protocol
UDP : User Datagram Protocol
URI : Uniform Resource Identifier
REST : Representational State Transfer
DNS : Domain Name System
mDNS : multicast Domain Name System
6LowPAN : IPv6 over Low -Power Wireless Personal Area Networks
WPAN : Wireless Personal Area Network
IPv6 : Internet Protocol version 6
WIFI : Wireless Fidelity
RF : Radio Frequency
RFID : Radio Frequency IDentification
3G : troisième génération
GSM : Global System for Mobile

UMTS : Universal Mobile Telecommunications System
MoCA : Montreal Cognitive assessMent
CTU : Centre de Télé-enseignement Universitaire
kbps : Kilobits Per Second
DoS : Denial of Service attack
VM : virtual machine
ICMP : Internet Control Message Protocol
ARP : Adresse Resolution Protocol
CHAOS : Create Havoc Around Our System
CHAOSNet : Create Havoc Around Our System Network
DHCP : Dynamic Host Configuration Protocol
SMTP : Simple Mail Transfer Protocol
POP : Post Office Protocol
IGMP : Internet Group Management Protocol
IPv4 : Internet Protocol version 6
TLS : Transport Layer Security
DES : Data Encryption Standard
RC4 : Rivest Cipher 4
MAC : Media Access Control
SHA : Secure Hash Algorithm
MD5 : Message digest 5
IRC : Internet Relay Chat
SSL : Secure Socket Layer
3DES : Triple Data Encryption Standard
DSS : Digital Signature Standard
NB : Naive Bayesian
NBA : Naive Bayesian Augmented
SVM : Support Vector Machine
GA : Genetic Algorithms
RBF : Radial Basis Function
MSVM : Multiclass Support Vector Machine
ID3 : Iterative Dichotomized
CART : Classification and Regression Trees
BFT : Best First Tree
SLIC : Supervised Learning In Quest
MDL : Minimum Description Length
SPRINT : Scalable Parallelizable Induction of Decision Tree algorithme
HTML : HyperText Markup Language

TABLE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE :

L'Internet des objets est en train de devenir l'un des principales tendances façonnant le développement des technologies dans le secteur des TIC (Technologies de l'information et de la communication) en général. Le passage d'un Internet utilisé pour interconnecter les appareils des utilisateurs finaux à Internet utilisé pour interconnexion d'objets physiques qui communiquent les uns avec les autres ou avec les humains est effectué afin d'offrir un service qui nécessite de repenser à nouveau certaines approches conventionnelles habituellement utilisées en réseau informatique, fourniture et gestion de services.

L'attaque par déni de service distribuées est l'une des plus grandes menaces pour un réseau informatique, cette menace est amplifiée par l'apparition de l'internet des objets qui offre une grande opportunité grâce au nombre d'objets connectés possédant une capacité de calcul et de communication intéressante, par conséquent ces objets présentent des vulnérabilités importantes et s'ils ne sont pas correctement configurés et sécurisés, ils peuvent être détournés et utilisés comme machines zombies.

Un botnet est un réseau de machines zombies infectés par un logiciel malveillant appeler bot qui s'exécute sur la machine de la victime sans son approbation, ce réseau de machines infecter présente un danger énorme qui réside dans sa capacité à lancer des attaques DoS distribuées à tout moment.

Afin de mieux comprendre son fonctionnement un botnet est généralement composé d'un serveur de commande et de contrôle (C&C), un client Bot (Machine Zombie) et un bot intermédiaire qui crée généralement un canal entre le serveur (C&C), et les clients en utilisant le protocole Internet Relay Chat (IRC).

Selon l'architecture, il existe deux grandes catégories de botnets : une architecture centralisée où le maître de bot donne des instructions au serveur C&C qui les transmet aux clients et une architecture distribuée dont le maître de bot ordonne directement à un client qui gère les commandes aux autres comme dans le cas des réseaux peer-to-peer.

Les botnets sont devenus sophistiqués au fil du temps, le protocole IRC a été le premier protocole

à être utilisé comme moyen de communication entre le serveur C&C et les clients. Désormais, HTTP (Hypertext Transfer Protocol) est également utilisé comme protocole de communication dans les architectures centralisées.

Avec la propagation des réseaux P2P, le concept a été rapidement détourné pour permettre une propagation plus rapide des malwares et des instructions des maîtres de bot. Au fur et à mesure de l'apparition des solutions basées sur l'analyse du trafic, les botnets ont commencé à crypter les commandes.

C'est pour cela que nous allons explorer des approches que nous considérons adaptables pour le contexte loT afin de Détecter les virus botnet et de Prévenir les attaques de ces derniers en comparant deux solution basée sur l'apprentissage automatique.

Afin de mener à bien notre travail, nous avons organisé ce rapport en quatre chapitres :

Le premier chapitre intitulé « sécurité des réseaux » couvre les modèles OSI et TCP/IP, les réseaux peer to peer ainsi qu'une définition de l'Internet des objets, ses protocoles approprier, son architecture et les dimensions de sa sécurité.

Le deuxième chapitre titré « Introduction du cas d'étude » comporte une description des capteurs de données sur lesquelles des algorithmes de classification vont être appliquer, leurs chronologies, l'état de leurs adresses IP ainsi que la définition de quelques protocoles contenues dans leurs paquets.

Le troisième chapitre nommé « Présentation des algorithmes de classification » décrit les algorithmes de classification intitulé : Naive bayes classifier, Support vector machine ainsi que les algorithmes de l'arbre de décision.

Enfin le dernier chapitre « Application des algorithmes de classification sur l'ensemble des données et comparaison des résultats » illustre des représentations graphiques obtenue en appliquant les algorithmes de classification support vector machine et random forest sur l'ensemble des données, et ses significations ainsi que des résultats d'évaluation de performance des ces algorithmes.

CHAPITRE 1

SÉCURITÉ DES RÉSEAUX

1 introduction

Dans ce chapitre nous allons parler sur les réseaux informatique et leurs architecteurs OSI et TCP/IP, les réseaux pear to pear et l'internet des objets.

2 Réseaux informatiques

Les réseaux informatiques sont nés de la nécessité de connecter des terminaux distants à un site central et de relier les ordinateurs entre eux et enfin les terminaux, tels que les stations de travail ou les serveurs. Initialement, ces communications étaient destinées au transport de données informatiques. Aujourd'hui, l'intégration de la parole et de la vidéo est répandue dans les réseaux informatiques, même si ce n'est pas sans difficulté [22].

3 Architecture OSI

L'ISO (International Normalisation Organizations) a normalisé sa propre architecture sous l'appellation d'OSI (Open System Interconnexion). L'architecture ISO est la première à avoir été définie, et ce de manière partiellement parallèle à celle d'internet. La séparation entre les deux est que l'architecture ISO définit catégoriquement les différentes couches, alors que l'architecture internet s'exécute à faire un contexte pragmatique.

3.1 Couche physique

La couche physique est très compliquée. Plusieurs normes décrivent comment encoder et Envoyer un signal physique sur une ligne de communication.

3.2 Couche de trame

La couche de trame fournit les fonctions et les moyens de traitement nécessaires pour établir, maintenir et libérer des connexions entre les entités de réseau et pour transporter des unités de données du service de liaison.

3.3 Couche paquet

le rôle de la couche paquet (niveau de transport) est le moyen de fournir Établir, maintenir et libérer les connexions réseau entre les systèmes ouverts, et D'autre part, fournir les moyens fonctionnels et les procédures nécessaires à l'échange, Entre entités de transport et les unités de service réseau.

3.4 Couche message

La couche message (niveau transport) doit assurer le transfert de données entre entités Session. Ce transport doit être transparent, c'est-à-dire indépendant de la succession Signes transportées et même des éléments binaires.

3.5 Couche session

le rôle de la couche session est de donner aux entités de présentation les ressources requis à l'organisation et à la synchronisation de leur dialogue. en vue de cela, la couche 5 propose les prestations permettant l'établissement d'une connexion, son maintien et sa libération, et ceux permettant de contrôler les interactions entre les entités de présentation

3.6 Couche présentation

La couche présentation s'occupe de la syntaxe des renseignements que les entités d'application se échangent. Deux formes additionnels sont définis dans la norme :

- La représentation des données transférées entre entités d'application.
- La représentation de la structure de données à laquelle les entités se réfèrent au cours de leur communication et la représentation de la totalité des actions faites sur cette structure de données.

3.7 Couche application

La couche application est la dernière du modèle reconnu. Elle propose aux processus applicatifs le moyen d'avoir accès à l'environnement réseau. Ces processus échangent leurs informations via des entités d'application [22].

4 Architecture TCP/IP

Dans les années 1970, le département de la défense américain, ou DOD (Department Of Defense), choisisse, face au foisonnement d'équipements se servant des protocoles de communication différents et incompatibles, de établir sa propre architecture. Cette architecture, intitulée TCP/IP, est à la source de réseau internet. Elle est aussi adoptée par plusieurs réseaux privés, nommés intranets. Les deux principaux protocoles établis dans cette architecture sont ceux-ci :

• IP (Internet Protocol), de niveau réseau , qui assure une prestation sans liaison.

• TCP (Transmission Control Protocol), de niveau transport, qui propose une prestation efficace avec liaison.

TCP/IP définit une architecture en couches qui comprend également, sans que cela soit définie explicitement, une interface d'accès au réseau. En effet, plusieurs sous-réseaux différents peuvent être pris en considération dans l'architecture TCP/IP, de type aussi bien local qu'étendu Cette architecture a pour socle le protocole IP, qui correspond au niveau paquet (couche 3). En fait, il ne correspond que en partie à ce stade.

4.1 Protocole TCP

Le protocole TCP réunit les fonctions de niveau message (couche 4). C'est un protocole suffisamment difficile, qui comprend une multitude de possibilités permettant de solutionner toutes les erreurs de perte de paquet dans les niveaux plus bas. Notamment, un fragment perdu peut être rapatrié par retransmission sur le flot d'octets. Le protocole TCP est en mode avec liaison, à l'inverse de UDP. Ce dernier protocole UDP est inclut dans l'architecteur tcp/ip et il se place aussi au niveau transport mais au sein d'un mode sans connexion et ne fournit quasiment aucune fonctionnalité . Il ne peut considérer que des logiciels qui requièrent peu de prestation de la part de la couche transport[22].

4.2 Protocole IP

Le protocole IP a été élaboré comme protocole d'inter liaison, définissant un bloc d'informations d'un format bien défini et renfermant une adresse, mais sans autre fonctionnalité. Sa fonction était de transporter ce bloc d'informations à l'intérieur d'un paquet d'après toute autre technique de transfert de paquet . Cela vaut pour la première génération du protocole IP, dénommée IPv4, qui reste toujours massivement employée. La seconde version du protocole IP, IPv6, joue sans aucun doute un rôle de niveau paquet, avec de nouvelles fonctionnalités permettant de transporter les paquets d'une extrémité du réseau à une autre avec une certaine sécurité. Les paquets IP sont autonomes les uns des autres et sont routés de façon individuelle dans le réseau via les routeurs. La qualité de prestation fournie par le protocole IP est très faible, sans détection de paquets perdus ni de option de relance sur erreur[22].

5 Réseau P2P

5.1 Définition

Un réseau P2P consiste en un ensemble de pairs et de liens virtuels entre ces pairs. En différents termes, des nœuds et des liens de recouvrement définissent tout un réseau mis à un échelon supérieur à l'infrastructure physique. Chaque pair hôte comprend un ensemble d'objets (fichiers de musique, articles, etc.) et de ressources physiques (mémoire de stockage, bande passante, unité centrale, etc.).

Le terme P2P fait indexe aux systèmes et applications qui utilisent ces ressources réparties avec la mission d'accomplir des actions critiques d'une manière décentralisée. Le terme P2P émane aussi de la relation réciproque entre entités de même état pour faire des échanges d'objets. Les utilisateurs des systèmes P2P ont besoin de mécanismes qui repèrent et récupèrent ces objets dans le réseau[22].

5.2 Classification des réseaux P2P

Trois grandes rubriques de réseaux P2P peuvent être identifiées : centralisé, décentralisé et hybride. La catégorie décentralisée peut toujours être scindée en décentralisé mais structuré et décentralisé et non structuré. La différence principale entre ces systèmes est le mécanisme utilisé pour rechercher des ressources dans le réseau Peer to Peer[22].

5.2.1 Réseaux P2P centralisés

Dans les systèmes P2P centralisés, la présentation et l'adresse des ressources sont stockées dans un annuaire d'un serveur central. Les nœuds transmettent des requêtes au serveur central afin de dénicher quels nœuds ont les ressources désirées. Le serveur ne fournit que la capacité de prospection et négocie les téléchargements entre clients. Le contenu reste du côté client, ne passant jamais par le serveur. Après avoir reçu une requête d'un pair, l'index central recherche le meilleur pair dans son annuaire pour répondre à la requête. Le meilleur pair est celui qui est le plus économique, le plus rapide ou le plus disponible selon les besoins de l'utilisateur. Lorsque le nombre de participants est trop grand, ce modèle comprend divers problèmes dus à son infrastructure centralisée, principalement la surcharge de l'annuaire chargé de recueillir des renseignements sur tous les participants. Cette catégorie de réseaux Peer to Peer ne passe malheureusement que mal à l'échelle et peut subir des avaries en raison d'un seul nœud. Napster a été le cas le plus connu se basant sur ce modèle[22].

5.2.2 Réseaux P2P décentralisée

Dans l'architecture P2P décentralisée, tous les pairs sont fonctionnellement parfaitement équivalents. C'est un modèle P2P pur Cette architecture est caractérisée par une décentralisation de l'index, qui devient local à chaque pair, faisant effet de table de routage. La décentralisation rend le système autonome et répartit la charge équitablement. L'architecture décentralisée est donc plus robuste que l'architecture centralisée, mais le temps de découverte des ressources est évidemment plus long[18]. dans cette architecture on trouve deux types de réseaux :

1. Réseaux P2P décentralisés et non structurés Les systèmes décentralisés et non structurés sont ceux parmi lesquels il n'existe ni guide centralisé, ni contrôle sur la topologie du réseau, ni adresse sur l'emplacement des fichiers. Gnutella ou FreeNet en sont des exemples. Ce réseau est formé de noeuds, qui rejoignent le réseau selon plusieurs normes simples. L'emplacement des fichiers n'est créé sur aucune connaissance de la topologie. Afin de trouver un fichier, un noeud réclame à ses voisins. La façon la plus fréquente est l'inondation, dans laquelle la demande est propagée à la totalité des voisins au sein d'un certain rayon. Ces architectures non structurées sont très résistantes aux noeuds entrant et sortant du système. Toutefois, l'actuel mécanisme de prospection passe mal à l'échelle et produit une charge considérable pour les les participants au réseau[22].
2. Réseaux P2P décentralisés structurés Les systèmes décentralisés mais structurés ne disposent pas de serveur central de répertoire mais ont une topologie strictement contrôlée, dans laquelle les fichiers sont archivés à des sites spécifiques pour simplifier les recherches. Ils sont dits structurés parce qu'au-dessus de l'infrastructure physique sous-jacent, les nœuds sont reliés par un réseau de recouvrement créé sur plusieurs contraintes et reliant les pairs selon une structure précise (anneau, espace cartésien multidimensionnel, etc.). Ce genre de réseau ne fournit que des services et des cadres de travail communs basé sur le P2P (routage et localisation) mais ne forme pas une application P2P (partage de fichiers, etc.). Les systèmes P2P décentralisés structurés mettent en œuvre un algorithme de recherche au travers duquel une ressource donnée est assignée de manière univoque à un pair pour un état donné du système. L'assignation est faite en réduisant une métrique prédéfinie sur un espace numérique d'identifiants partagé par les pairs et les ressources. L'algorithme de recherche est totalement déterministe, et les liens entre les pairs sont établis selon des normes bien définies[22].

5.2.3 Réseaux P2P hybrides

Les réseaux hybrides permettent de solutionner certains des soucis de l'approche purement distribuée, sans perdre pour autant la performance de la solution centralisée. Les réseaux fondés sur un tel système de localisation et de routage de données peuvent supporter un nombre de pairs de l'ordre du million. Cette solution cumule les spécifications des modèles centralisé et décentralisé. La décentralisation assure notamment l'extensibilité, l'indulgence aux fautes et le passage à l'échelle.

La centralisation partielle implique divers centres serveurs qui contiennent des données importantes pour le système. Tout usager élit son « superpair », qui sert de serveur central pour des nœuds locaux et qui peut rentrer en contact avec différents « super-pairs ». FastTrack, KaZaA, BitTorrent, eDonkey/eMule sont des exemples de ce système[22].

6 Services de sécurité des réseaux P2P

Dans le domaine de sécurité des réseaux P2P, il y a 5 services de sécurité pertinents : disponibilité, authenticité de fichier, anonymat, contrôle d'accès et l'équité.

6.1 Disponibilité

Chaque pair doit pouvoir communiquer avec les autres pairs et est capable d'offrir l'accès aux ressources avec qui il collabore.

6.2 L'authenticité du fichier

elle peut être évaluer de différents manières :

6.2.1 Plus ancien document

avec cette approche, on a confiance à la plus ancienne réponse d'une requête pour l'authentifier.

6.2.2 Basé sur des experts

dans cet état les experts donnent leurs points de vue sur l'authenticité du fichier.

6.2.3 Basé sur le vote

dans l'approche fondée sur des votes, différents experts donnent leurs points de vue sur l'authenticité du fichier ensuite ces opinion réunis nous informent sur l'authenticité.

6.2.4 Basé sur la réputation

l'approche fondée sur la renommée est semblable à l'approche précédente à la différence que celle-ci prend en considération la réputation des experts.

6.3 Anonymat

il rend compliquée aux pairs de trouver qui a construit un fichier, qui l'a hébergé, qui en a l'accès et quels sont les documents hébergés sur un pair. Les objectifs de cette anonymisation sont

à la fois le combat contre la censure et la peur des pouvoirs publics. Le premier réseau de ce type à être proposé est Freenet. Il cache les communications par des mécanismes de Proxy et chiffre les liens afin de fuir l'écoute.

6.4 Contrôle d'accès

c'est-à-dire la restriction d'accès aux ressources et aux informations, aux seuls sujets qui sont autorisés.

6.5 Équité

les systèmes P2P reposent sur la collaboration de la totalité des participants. Il est essentiel d'empêcher que d'autres pairs bénéficient plus qu'ils ne coopèrent , cela veut dire qu'un pair qui bénéficie de réseau doit participer aussi à cela[26].

7 Définition IoT

L'IoT est l'infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physiques ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution[21]. pour faciliter et simplifier le travail des programmeurs d'applications et des fournisseurs de services.

8 Protocoles dédiés à l'IoT

De nombreuses normes IoT sont proposées pour faciliter et simplifier le travail des programmeurs d'applications et de fournisseurs de services. Différents groupes ont été créés pour fournir des protocoles prenant en charge l'IdO, notamment les efforts du Consortium World Wide Web (W3C), du Groupe de travail d'ingénierie Internet (IETF), de EPCglobal, de l'Institut des ingénieurs électriciens et électroniciens (IEEE) et des normes européennes de télécommunication. Institut (ETSI). La figure présente un résumé des protocoles les plus importants définis par ces groupes.Les protocoles IoT sont classé en quatre grandes catégories, à savoir : protocoles d'application, protocoles de découverte de services, protocoles d'infrastructure et autres protocoles influents. Cependant, tous ces protocoles ne doivent pas nécessairement être regroupés pour fournir une application IoT donnée. De plus, en fonction de la nature de l'application IoT, il n'est pas forcément nécessaire de prendre en charge certaines normes dans une application. [17]. Dans les sous-sections suivantesnous fournissons une vue d'ensemble de certains de ces protocoles et de leurs fonctionnalitées principales.

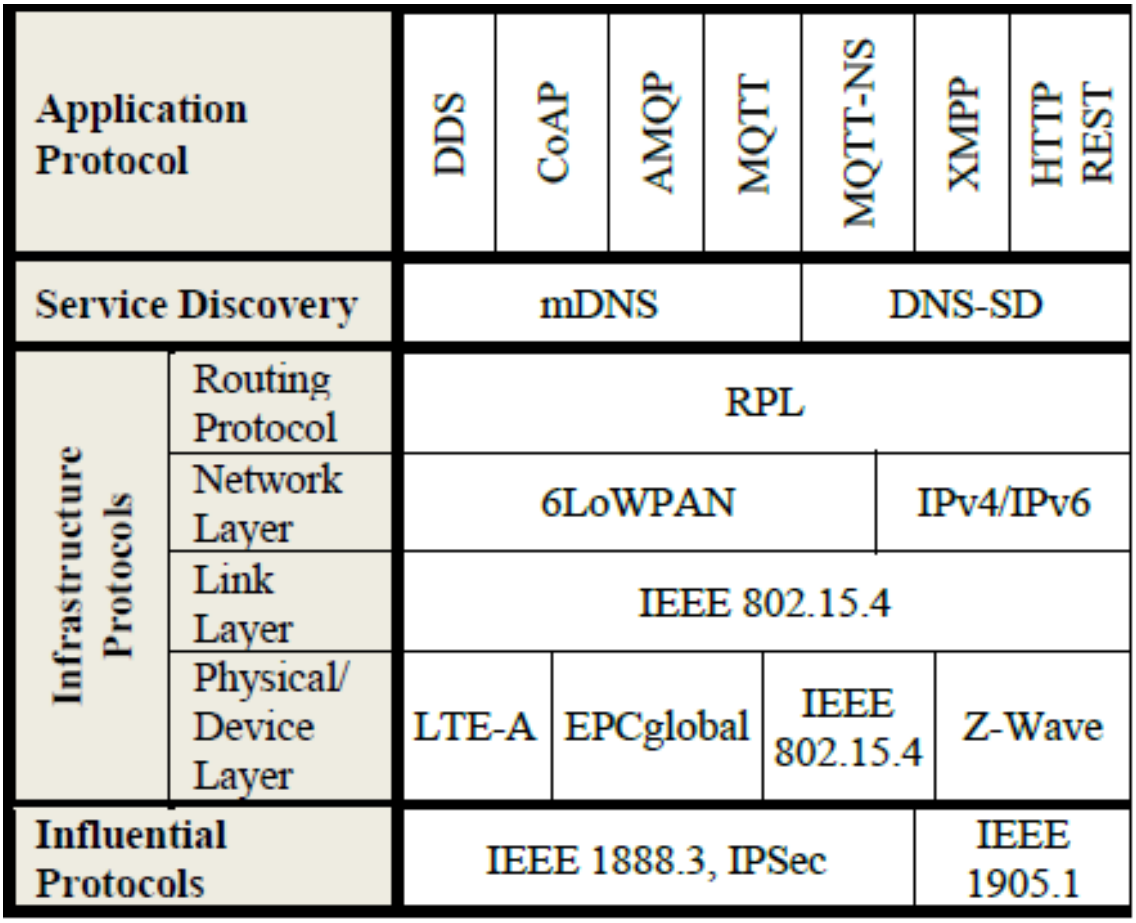

FIGURE 1.1 – Les protocoles dédiés à l'IoT[17]

8.1 Constrained application protocol (CoAP)

Le CoAP a été mis au point au sein de l'IETF et publié en 2014, c'est un protocole "Canada dry" de HTTP, client/serveur comme lui, mais destiné à fonctionner essentiellement sur UDP (bien que son installation sur TCP soit possible), avec des frais généraux réduits et un bloc de données limité à 1 024 octets. Il utilise les spécifications originelles d'HTTP, notamment l'URI (Uniform Resource Identifier) comme identifiant des ressources et des verbes (méthodes) et des codes de réponse issus d'HTTP. Le passage d'HTTP en CoAP (et vice versa) se fait de façon simple au niveau d'un proxy situé en bordure du réseau local qui garde en cache toutes les informations locales nécessaires à la traduction. Il est ainsi possible de bâtir, en associant CoAP à HTTP, des architectures satisfaisant aux critères REST (Representational State Transfer) caractéristiques du Web et bien adaptées aux systèmes distribués. CoAP apparaît comme une bonne solution pour des applications IoT dans les domaines de la domotique et du comptage intelligent. Comme HTTP, il peut être considéré comme un protocole généraliste[24]. La fonctionnalité globale du protocole CoAP est illustrée à la Figure.

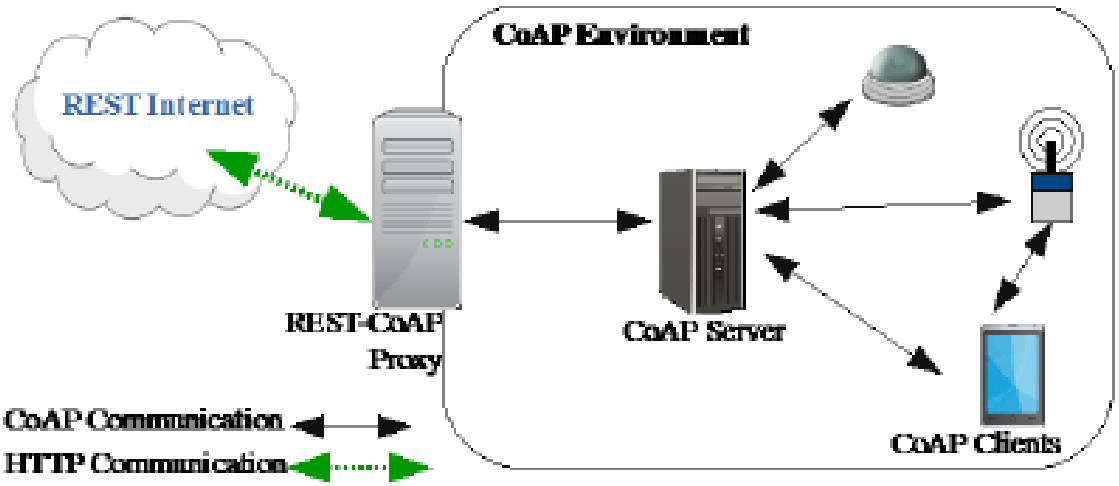

FIGURE 1.2 – Protocole d'application contraint (CoAP)[17]

8.2 Multicast Domain Name System (mDNS)

Un service de base pour certaines applications IoT telles que la discussion est la résolution de nom. mDNS est un service qui peut effectuer la tâche de serveur DNS unicast [84]. mDNS est flexible du fait que l'espace de noms DNS est utilisé localement, sans dépenses ni configuration supplémentaires. mDNS est un choix approprié pour les périphériques Internet intégrés en raison du fait que :

— Il n'est pas nécessaire de procéder à une reconfiguration manuelle ni à une administration supplémentaire pour gérer les périphériques.

— Il est capable de fonctionner sans infrastructure.

— Il est capable de continuer à fonctionner en cas de défaillance de l'infrastructure.

Le mDNS interroge les noms en envoyant un message de multidiffusion IP à tous les nœuds du domaine local, comme illustré à la figure suivante. Cette requête permet au client de demander aux périphériques portant le nom donné de répondre. Lorsque la machine cible reçoit son nom, elle multidiffuse un message de réponse contenant son adresse IP. Tous les périphériques du réseau qui obtiennent le message de réponse mettent à jour leur cache local en utilisant le nom et l'adresse IP donnés[17].

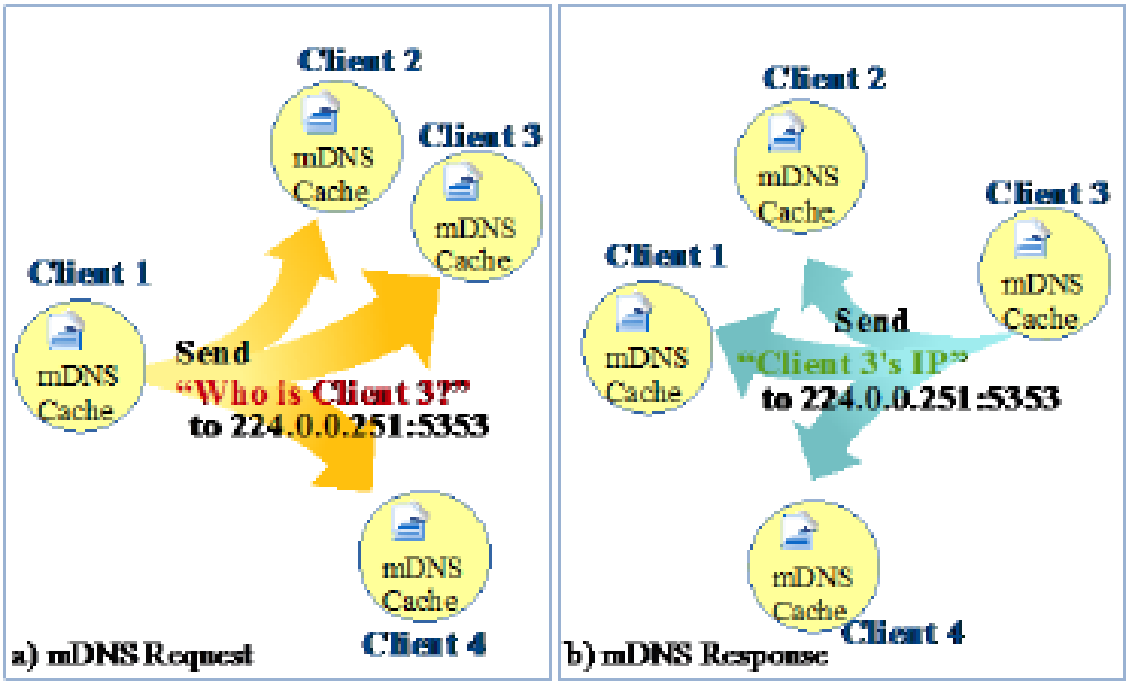

FIGURE 1.3 – protocole mdns[17]

8.3 6LowPAN

Les réseaux personnels sans fil (WPAN) de faible puissance sur lesquels de nombreuses communications IoT peuvent s'appuyer présentent des caractéristiques particulières différentes des technologies de couche liaison précédentes, telles que la taille de paquet limitée (par exemple, 127 octets maximum pour IEEE 802.15.4), différentes longueurs d'adresses et une faible bande passante. Il était donc nécessaire de créer une couche d'adaptation qui adapte les paquets IPv6 aux spécifications IEEE 802.15.4. Le groupe de travail IETF 6LoWPAN a élaboré une telle norme en 2007. 6LoWPAN est la spécification des services de mappage requis par IPv6 sur WPAN de faible puissance pour maintenir un réseau IPv6. La norme prévoit la compression d'en-tête pour réduire le temps système de transmission, la fragmentation pour répondre aux exigences de l'unité de transmission maximale (MTU) IPv6 et le transfert à la couche liaison pour prendre en charge la livraison multi-sauts. Les datagrammes enveloppés par 6LoWPAN sont suivis d'une combinaison de certains en-têtes. Ces en-têtes sont de quatre types qui sont identifiés par deux bits : (00) en-tête NO 6LoWPAN, (01) en-tête d'envoi, (10) adressage de maillage et (11) fragmentation. Par l'en-tête NO 6LoWPAN, les paquets ne correspondant pas à la spécification 6LoWPAN seront supprimés. La compression des en-têtes IPv6 ou la multidiffusion est effectuée en spécifiant l'en-tête Dispatch. L'en-tête d'adressage de maillage identifie les paquets IEEE 802.15.4 à transmettre à la couche liaison. Pour les datagrammes dont la longueur dépasse une seule trame IEEE 802.15.4, l'en-tête de fragmentation doit être utilisé. 6LoWPAN supprime beaucoup de temps système IPv6 qu'un petit datagramme IPv6 peut être envoyé sur un seul IEEE 802.15.4 hop dans le meilleur des cas.

Il peut également compresser les en-têtes IPv6 sur deux octets[17].

8.4 Protocole IEEE 1905.1

Les divers périphériques dans les environnements IoT reposent sur différentes technologies de réseau. Il est donc nécessaire d'interopérer les technologies sous-jacentes. La norme IEEE 1905.1 était conçu pour les réseaux domestiques numériques convergents et technologies hétérogènes. Il fournit une abstraction couche qui masque la diversité des topologies de contrôle d'accès au support, comme indiqué sur la figure 4, sans nécessiter de modification des couches sous-jacentes. Ce protocole constitue une interface avec les technologies communes des réseaux domestiques, permettant ainsi de combiner des protocoles de liaison de données et de couche physique, notamment IEEE 1901 sur lignes électriques, WiFi / IEEE 802.11 sur différentes bandes RF, Ethernet sur câbles à paires torsadées ou à fibres, et MoCA 1.1 sur les câbles coaxiaux peuvent coexister les uns avec les autres[17].

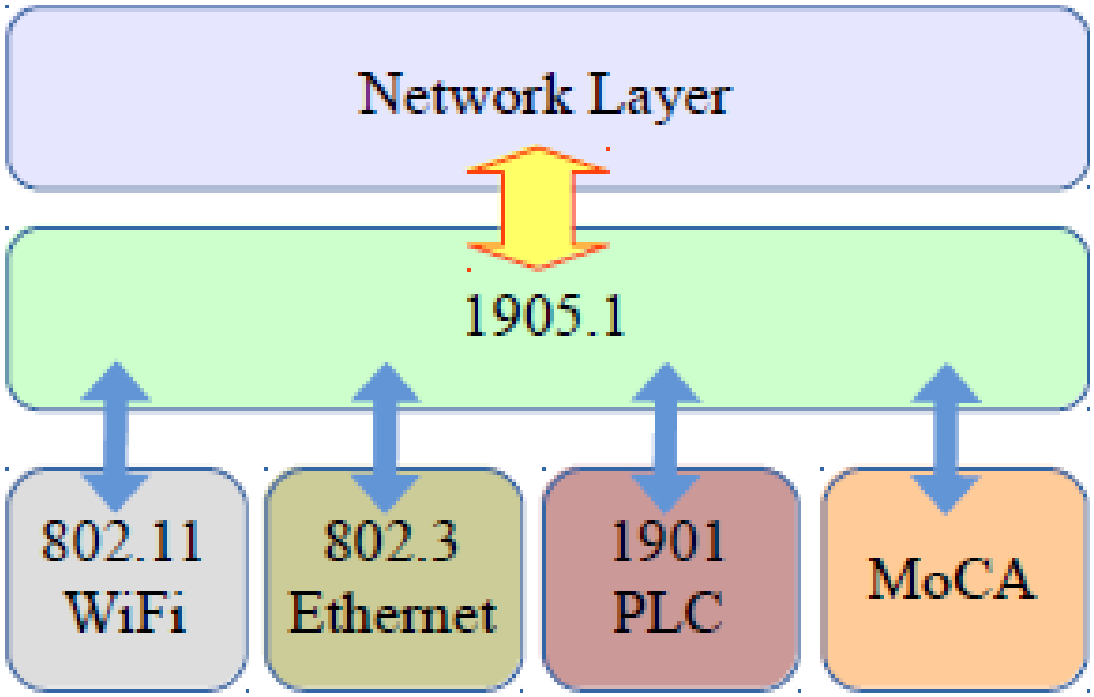

FIGURE 1.4 – protocole IEEE 1905.1[17]

9 Architecture de l'IOT

L'Ido devrait être capable d'interconnecter des milliards d'objets hétérogènes sur Internet, il existe donc un besoin critique d'une architecture en couches flexible. Le nombre sans cesse croissant d'architectures proposées n'a pas encore convergé vers un modèle de référence. Dans le

même temps, des projets tels que IoT-A tentent de concevoir une architecture commune basée sur l'analyse des besoins des chercheurs et du l'industrie. Parmi les modèles proposés, le modèle de base est un modèle architecture de 3 couches consistant en l'application, Couches de réseau et de perception. Dans la littérature récente, Cependant, certains autres modèles ont été proposés et qui ajoutent plus d'abstraction à l'architecture IoT. La Figure 5 illustre certaines architectures communes entre eux. Dans ce qui suit nous fournissons une brève discussion sur le modèle a cinq couches(five-layer)[17].

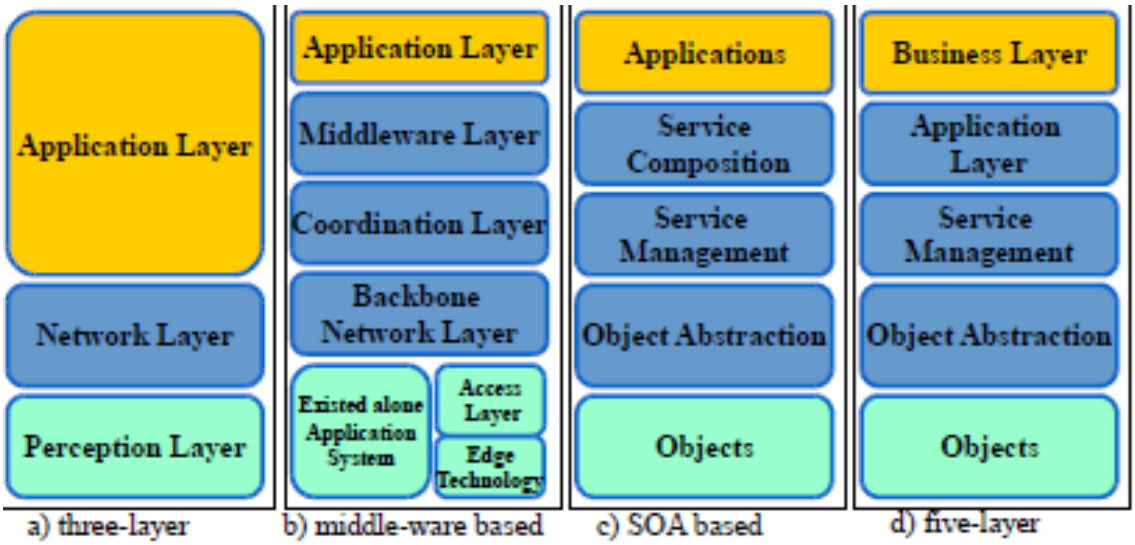

FIGURE 1.5 – Architecture de l'IDO[17]

9.1 Couche perception

La première couche, la couche perception, représente les capteurs physiques de l'IoT qui visent à collecter et traiter les informations. Cette couche comprend des capteurs et actionneurs permettant d'exécuter des fonctionnalités différentes telles que l'interrogation de l'emplacement, la température, le poids, le mouvement, les vibrations, l'accélération, l'humidité, etc. Des mécanismes brancher et utiliser (plug-and-play) standardisés sont nécessaires à utiliser par la couche de perception pour configurer les objets hétérogènes. La couche de perception numérise et transfère les données vers la couche d'abstraction d'objet via des canaux sécurisés. Les mégadonnées créées par l'IoT sont initialisées sur cette couche.

9.2 Couche abstraction

L'abstraction d'objet transfère les données produites par la couche objet vers la couche de gestion des services par le biais de canaux. Les données peuvent être transférées via diverses technologies tels que RFID, 3G, GSM, UMTS, WiFi, Bluetooth faible Energie, infrarouge, ZigBee, etc. De plus, d'autres fonctions comme l'informatique en nuage (cloud computing) et les processus de gestion de données sont traités par cette couche.

9.3 Couche de service management

La couche Gestion des services ou Middleware (appariement) associe un service à son demandeur en fonction des adresses et des noms. Cette couche permet aux programmeurs d'applications IoT de travailler avec des objets hétérogènes sans tenir compte d'une plate-forme matérielle spécifique. Cette couche traite également les données reçues, prend des décisions et fournit les services requis via les protocoles filaires du réseau.

9.4 Couche application

La couche d'application fournit les services demandés par les clients. Par exemple, la couche application peut fournir les mesures de la température et de l'humidité de l'air au client qui demande ces données. L'importance de cette couche pour l'IoT est qu'elle est en mesure de fournir des services intelligents de haute qualité pour répondre aux besoins des clients. La couche d'application couvre de nombreux marchés verticaux tels que la maison intelligente, la construction intelligente, les transports, l'automatisation industrielle et les soins de santé intelligents.

9.5 Couche Business

La couche métier (gestion) gère l'ensemble des activités et des services du système IoT. Les responsabilités de cette couche sont de construire un modèle d'entreprise, des graphiques, des

organigrammes, etc. basés sur les données reçues de la couche application. Elle est également censée de concevoir, analyser, mettre en œuvre, évaluer, surveiller, et développer des éléments liés au système IoT. La couche métier permet de prendre en charge les processus de prise de décision basés sur l'analyse Big Data. En outre, la surveillance et la gestion des quatre couches sous-jacentes sont obtenu à cette couche. De plus, cette couche compare la sortie de chaque couche avec la sortie attendue pour améliorer les services et maintenir la vie privée.

10 Sécurité de l'IOT

L'Iot est une technologie caractérisée par une importante ubiquité dans le monde physique et une omniprésence autour de ses utilisateurs. Les diverses applications éventuelles de l'IOT, l'hétérogénéité de ses technologies habilitantes et sa forte dimension humaine et socioéconomique rendent sa sécurité un thème difficile et compliquée . en plus des soucis de sécurité des technologie qui le constitueront, l'IOT augmente les problèmes de sécurité des gens qui s'en serviront, et fait émerger de nouveaux difficultés liés à la sécurité des systèmes sous sa vérification. Comme le montre la Figure suivante, la sécurité et la privacy dans l'IdO peut être abordée de trois angles complémentaires qui reflètent ses dimensions technologique, humaine et systémique. La protection de la technologie touche tout d'abord la protection des données , des communications et des infrastructures réseaux. Cette protection est primordial pour contrarier les attaques classiques et futures sur l'intégrité, l'exactitude et la confidentialité des données, et les attaques sur les infrastructures réseaux et leurs fonctionnalités. La protection des gens concernera la protection de la confidentialité des utilisateurs (« privacy ») qui requiert, en plus des solutions technologiques, une régulation adéquate qui établit les responsabilités en cas de litiges. La protection des systèmes interconnectés et hébergeant les objets de l'IOT, concernera la protection des objets eux-mêmes fournis à ces systèmes et les processus qu'ils contrôleront [20].

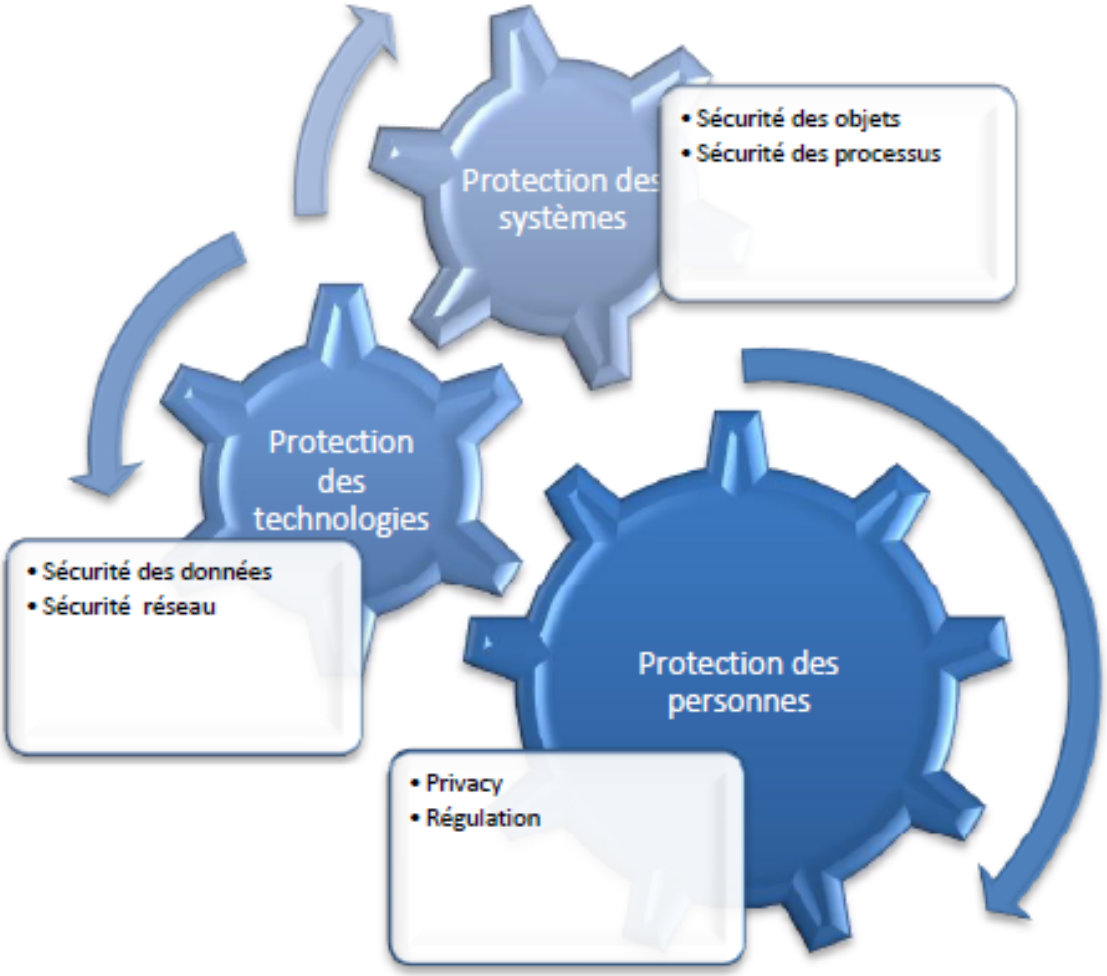

FIGURE 1.6 – Sécurité et Privacy de l'Internet des Objets[20]

11 Conclusion

dans ce chapitre nous avons présenter les différentes couches qui compose l'architecture OSI, les protocoles IP et TCP fonctionnant sur l'architecteur TCP/IP ensuite les catégories des réseaux peer to peer et leurs services de sécurité et enfin les protocoles, l'architecteur et la sécurité de l'internet des objets.

CHAPITRE 2

INTRODUCTION DU CAS D'ÉTUDE

1 Introduction

Dans ce chapitre nous allons introduire notre cas d'étude en présentant l'ensemble de données qu'on va utiliser dans notre quatrième chapitre, pour cela nous allons donner une description pour tous les fichiers ainsi que les protocoles réseaux contenus dans ces derniers.

2 Ensemble de données CTU-13

La CTU-13 est un ensemble de données du trafic botnet qui a été capturé à l'Université CTU, République tchèque, en 2011. L'ensemble de données CTU-13 consiste en treize captures (appelées scénarios) de différents échantillons de botnet. Dans chaque scénario, nous avons exécuté un malware spécifique, qui a utilisé plusieurs protocoles et effectué différentes actions. Les captures représentent le trafic botnet réel mélangé avec le trafic normal et le trafic de fond.

Chaque scénario a été capturé dans un fichier pcap qui contient tous les paquets des trois types de trafic. Ces fichiers pcap ont été traités pour obtenir d'autres types d'informations, tels que le flux net et les journaux web, etc. La première analyse de l'ensemble de données CTU-13, qui a été décrite et publiée dans l'article "Une comparaison empirique des méthodes de détection de botnet" utilise des flux nets unidirectionnels pour représenter le trafic et attribuer les étiquettes. Ces flux nets unidirectionnels ne doivent pas être utilisés car ils ont été surclassés par leur deuxième analyse de l'ensemble de données, qui a utilisé des flux nets bidirectionnels. Les flux nets bidirectionnels ont plusieurs avantages sur les directionnels [1].

2.1 Description des botnet

Ici nous allons nous focaliser sur les botnets intitulé : neris, virut, rbot, donbot et sogou.

2.1.1 Capture du botnet neris

1. Chronologie

   l'Université CTU ont capturé le malware Neris ainsi que les paquets de tout le département de la CTU qui appartient à un réseau universitaire, une machine XP virtualbox avec l'adresse IP publique 147.32.84.165 a été utilisé. La première heure de capture n'était composée que de trafic en arrière-plan, puis ils ont lancé le malware. Le malware a été arrêté 5 minutes avant la fin de la capture. La bande passante de l'expérience était limité à 20 kbps dans la sortie du bot[2].

2. Adresse IP

   Hôtes infectés

   — 147.32.84.165 : Windows XP (version anglaise), nom : SARUMAN ; label : Botnet, quantité de flux bidirectionnels : 409610 KO.

   Hôtes normaux

   — 147.32.84.170 : quantité de flux bidirectionnels : 18438 KO, étiquette : Normal-V42-Stribrek.

   — 147.32.84.164 : quantité de flux bidirectionnels : 7654 KO, étiquette : Normal-V42-Grill.

   — 147.32.84.134 : quantité de flux bidirectionnels : 3808 KO, étiquette : Normal-V42-Jist.

   — 147.32.87.36 : quantité de flux bidirectionnels : 269 KO, étiquette : CVUT-WebServer. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur Web.

   — 147.32.80.9 : quantité de flux bidirectionnels : 83 KO, étiquette : CVUT-DNS-Server. Cet hôte normal n'est pas aussi fiable car il s'agit d'un serveur DNS.

   — 147.32.87.11 : quantité de flux bidirectionnels : 6 KO, étiquette : MatLab-Server. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur matlab[2].

2.1.2 Capteur du botnet Virut

1. Chronologie

   — Le 15 août 2011 à 17 :14 :41 : Ils ont commencé la capture globale afin qu'ils puissent la laisser tourner pendant toute la nuit.

   — Le 15 août 2011 à 17 :18 :01 : Ils ont infecté la machine virtuelle avec un malware à flux rapide, donc le bot est lancé. La bande passante sera à 100kBps.

   — Le 16 août 2011 à 9 :35 :05 : Ils ont arrêté la machine virtuelle et les deux captures globales et botnet.

2. Adresses IP

   Hôtes infectés

— 147.32.84.165 : Version anglaise de Windows XP, nom : SARUMAN, label : Botnet, quantité de flux bidirectionnels : 40003 KO

Hôtes normaux

— 147.32.84.170 : quantité de flux bidirectionnels : 26846 KO, étiquette : Normal-V42-Stribrek.

— 147.32.84.134 : quantité de flux bidirectionnels : 948 KO, étiquette : Normal-V42-Jist.

— 147.32.84.164 : quantité de flux bidirectionnels : 3539 KO, étiquette : Normal-V42-Grill.

— 147.32.87.36 : quantité de flux bidirectionnels : 422 KO, étiquette : CVUT-WebServer. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur Web.

— 147.32.80.9 : quantité de flux bidirectionnels : 6 KO, étiquette : CVUT-DNS-Server. Cet hôte normal n'est pas aussi fiable car il s'agit d'un serveur DNS.

— 147.32.87.11 : quantité de flux bidirectionnels : 18 KO, étiquette : MatLab-Server. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur matlab[3].

2.1.3 capteur du botnet rbot-dos

1. Chronologie

— Le 15 août 2011 à 10 :59 :23 : Ils ont essayé de faire fonctionner une machine. Ils ont commencé la capture globale, et ils ont utilisé le botnet rbot.exe

— Le 15 août 2011 à 12 :21 :33 : Ils ont commencé à capturer le bot. Puis Ils ont effectué une DoS à l'adresse 147.32.96.69 avec des paquets UDP au port 161 et ils ont fixé la bande passante à 40kBps. L'attaque a été lancée, mais ils ne savent pas si elle a réussi en raison de la limite de la bande passante.

— Le 15 août 2011 à 13 :06 :40 : Ils ont arrêté la capture de dos, Ils ont arrêté le malware. Ils n'ont pas arrêté la capture de l'ensemble du département.

— Le 15 août 2011 à 13 :25 :40 : Ils avaient de nouveau infecté la VM, mais cette fois ils allaient attaquer avec des paquets ICMP. Ils ont commencé une nouvelle capture de Dos, mais ils ont gardé la même capture globale du département. Ils allaient de nouveau infecter avec rbot.exe La bande passante sera de 100kBps Ils ont rendu au ministère des affaires étrangères à l'adresse suivante : 147.32.96.69

— Le 15 août 2011 à 13 :40 :00 : L'attaque a pris fin, mais ils continuent à capturer des paquets dans les deux fichiers pcap. L'attaque a réussi car ils ne peuvent plus accéder à la cible depuis d'autres ordinateurs en dehors de l'université.

— Le 15 août 2011 à 13 :46 :43 : Ils ont arrêté le bot et la capture des bots.

— Analyse : Ils avaient fait une attaque de DoS de la CIPD contre une propriété intellectuelle.

— Le 15 août 2011 à 13 :47 :00 : Ils avaient commencé une nouvelle capture de bot pour attaquer à nouveau en utilisant l'icmp mais avec une bande passante de 300kbps. Ils avaient arrêté et démarré la VM et infecté à nouveau avec rbot.exe Ils allaient faire parvenir cette adresse au ministère des affaires étrangères avec l'adresse suivante : icmp :147.32.96.69.

— Le 15 août 2011 à 13 :50 :51 : Ils avaient commencé l'attaque.

— Le 15 août 2011 à 13 :58 :35 : L'attaque a pris fin avec succès car ils ne peuvent plus accéder à la cible à partir d'autres ordinateurs en dehors de l'université.

— Le 15 août 2011 à 14 :06 :29 : Ils avaient arrêté la capture du botnet.

— Le 15 août 2011 à 15 :11 :46 : La capture globale a été arrêtée pour tout.

— Analyse Ils avaient fait une attaque ICMP DoS contre une adresse IP.

2. Adresses IP

Hôtes infectés

— 147.32.84.165 : Windows XP (version anglaise), nom : SARUMAN , libellé : Botnet, quantité de flux infectés : 5160 KO.

Hôtes normaux

— 147.32.84.170 : quantité de flux bidirectionnels : 12133 KO, étiquette : Normal-V42-Stribrek.

— 147.32.84.134 : quantité de flux bidirectionnels : 10382 KO, étiquette : Normal-V42-Jist.

— 147.32.84.164 : quantité de flux bidirectionnels : 2474 KO, étiquette : Normal-V42-Grill.

— 147.32.87.36 : quantité de flux bidirectionnels : 89 KO, libellé : CVUT-WebServer. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur Web.

— 147.32.80.9 : quantité de flux bidirectionnels : 13 KO, étiquette : CVUT-DNS-Server. Cet hôte normal n'est pas aussi fiable car il s'agit d'un serveur DNS.

— 147.32.87.11 : quantité de flux bidirectionnels : 4 KO, étiquette : MatLab-Server. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur matlab[4].

2.1.4 Capteur du botnet donbot

1. Chronologie

— Le 16 août 2011 à 10 :01 :58 : Ils ont commencé la capture globale de bande passante à 100kBps.

— Le 16 août 2011 à 10 :08 :47 : Ils ont commencé le malware et la capture du malware.

— Le 16 août 2011 à 12 :10 :31 : Ils ont arrêté la VM, la capture du bot et la capture globale[5].

2. Adresses IP

   Hôtes infectés

   — 147.32.84.165 : Version anglaise de Windows XP, nom : SARUMAN, label : Botnet, Quantité de flux bidirectionnels : 9260 KO.

   Hôtes normaux

   — 147.32.84.170 : quantité de flux bidirectionnels : 10976 KO, étiquette : Normal-V42-Stribrek.

   — 147.32.84.134 : quantité de flux bidirectionnels : 1364 KO, étiquette : Normal-V42-Jist.

   — 147.32.84.164 : quantité de flux bidirectionnels : 2490 KO, étiquette : Normal-V42-Grill.

   — 147.32.87.36 : quantité de flux bidirectionnels : 68 KO, étiquette : CVUT-WebServer. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur Web.

   — 147.32.80.9 : quantité de flux bidirectionnels : 40 KO, étiquette : CVUT-DNS-Server. Cet hôte normal n'est pas aussi fiable car il s'agit d'un serveur DNS.

   — 147.32.87.11 : quantité de flux bidirectionnels : 4 KO, étiquette : MatLab-Server. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur matlab[5].

2.1.5 Capteur du botnet sogou

1. Chronologie

   — Le 16 août 2011 à 13 :51 :25 : Ils avaient commencé la capture globale, la bande passante etait de 100kBps.

   — Le 16 août 2011 à 13 :52 :53 : Ils avaient lancé la VM.

   — Le 16 août 2011 à 13 :56 :07 : Ils avaient commencé la capture et le bot.

   — Le 16 août 2011 à 14 :12 :17 : Ils avaient arrêté le malware et les deux capteurs[6].

2. Adresses IP

   Hôtes infectés

   — 147.32.84.165 : Version anglaise de Windows XP Nom : SARUMAN. Label : Botnet. Quantité de flux bidirectionnels : 126 KO

   Hôtes normaux

   — 147.32.84.170 : quantité de flux bidirectionnels : 1614 KO, étiquette : Normal-V42-Stribrek.

   — 147.32.84.134 : quantité de flux bidirectionnels : 584 KO, étiquette : Normal-V42-Jist.

   — 147.32.84.164 : quantité de flux bidirectionnels : 1040 KO, étiquette : Normal-V42-Grill.

   — 147.32.87.36 : quantité de flux bidirectionnels : 98 KO, libellé : CVUT-WebServer. Cet hôte normal n'est pas aussi fiable puisqu'il s'agit d'un serveur Web.

— 147.32.80.9 (quantité de flux bidirectionnels : 2 KO, étiquette : CVUT-DNS-Server. Cet hôte normal n'est pas aussi fiable car il s'agit d'un serveur DNS[6].

3 Protocoles réseaux

Il existe de nombreux protocoles envoyés comme étant paquets lors de la capture des botnets cités précédemment par conséquent on définit ceux qui sont souvent répétées.

3.1 Protocole ARP

Le protocole ARP (Adresse Resolution Protocol) est un protocole de la couche liaison permettant de déterminer l'adresse physique d'un hôte à partir de son adresse logique. Chaque hôte maintient à jour une table ARP (/ cache ARP) associant une adresse logique à une adresse physique. Pour pallier les changements de matériels ou d'adressage logique, cette table est dynamique et ses entrées ont une durée de vie limitée. Si une communication est à destination d'un hôte non-référencé dans la table ARP, alors une requête ARP est diffusée sur le réseau afin de déterminer son adresse physique. Seul l'hôte ayant reconnu son adresse logique y répond, l'émetteur premier peut ainsi mettre à jour sa table de correspondance. Du fait des opérations réalisées par les routeurs, les informations ARP sont limitées au même réseau physique[7]. Ce Protocole n'est pas limité à établir une correspondance entre adresses Ethernet et adresses IP (32 bits). Par exemple, les adresses logiques pourraient être des adresses CHAOS (16 bits, associées au protocole CHAOSnet) ou PUP (sur 8 bits pour le protocole de Xerox, PARC Universal Protocol). Par conséquent, le format des paquets ARP est très malléable puisque les tailles des adresses des niveaux 2 et 3 ne sont pas prédéfinies[19].

3.2 Protocole DNS

Le but de ce protocole est de fournir un mécanisme pour nommer les ressources de telle sorte que les noms soient utilisables dans différents hôtes, réseaux, familles de protocoles, internet et organisations administratives

Toutes les communications à l'intérieur du protocole de domaine sont transportées dans un format unique appelé un message. Le format de message de niveau supérieur est divisé en 5 sections (dont certaines sont vides dans certains cas) ci-dessous :

L'en-tête comprend des champs qui spécifient lesquelles des sections restantes sont présentes, et spécifient également si le message est une requête ou une réponse, une requête standard ou un autre opcode, etc.

Les noms des sections après l'en-tête sont dérivés de leur utilisation dans les requêtes standard. La section des questions contient des champs qui décrivent une question à un serveur de noms.

Ces champs sont un type de requête (QTYPE), une classe de requête (QCLASS) et un nom de domaine de requête (QNAME). Les trois dernières sections ont le même format : une liste éventuellement vide d'enregistrements de ressources concaténés (RR). La section réponse contient les RR qui répondent à la question, la section d'autorité contient des RR qui pointent vers un serveur de noms faisant autorité, la section des enregistrements supplémentaires contient des RR qui se rapportent à la requête, mais ne sont pas strictement des réponses à la question[8].

3.3 Protocole DHCP

Le protocole DHCP (Dynamic Host Configuration Protocol) est un protocole de la couche réseau (couche de modèle OSI) de type client/serveur, utilisé par un hôte afin de configurer ses paramètres réseau conformément au réseau sur lequel il est connecté, lui permettant ainsi de s'intégrer à l'ensemble de ses hôtes, et de communiquer avec eux. Ces paramètres sont déterminés et centralisés par un serveur DHCP ; le client doit donc lui envoyer une requête afin de s'auto-configurer. Les informations minimums transmises sont l'adresse IP et le masque, ce qui permet ainsi de se connecter au réseau ; mais on peut aussi préciser le(s) serveur(s) DNS afin de réaliser des translations de noms, la passerelle afin d'accéder à d'autres réseaux interconnectés avec le réseau courant, le nom de domaine, le serveur de mail (SMTP et POP3), le serveur de temps, etc. Pour l'hôte, une configuration des paramètres réseau par DHCP est dynamique, et est donc préférée à une configuration statique pour simplifier les opérations de paramétrage. De plus, cela permet un nombre d'hôtes connectables au réseau bien supérieur au nombre d'hôtes permis par le réseau (adresse IP masque), tant que ces connexions ne sont pas simultanées. En effet, la gestion des adresses IP au niveau du serveur DHCP est dynamique, et une adresse IP est allouée pour un temps donné – appelé bail –; à l'expiration du bail, l'adresse IP est à nouveau disponible pour n'importe quel hôte se connectant au réseau. Enfin, en cas de changement d'un ou plusieurs des paramètres réseau, seule la configuration du serveur DHCP doit être modifiée [7].

3.4 Protocole UDP

Ce protocole de datagramme utilisateur (UDP) est défini pour rendre disponible un mode datagramme de communication informatique à commutation de paquets dans l'environnement d'un ensemble interconnecté de réseaux informatiques. Ce protocole suppose que le protocole Internet (IP) est utilisé comme protocole sous-jacent.

Ce protocole fournit une procédure permettant aux programmes d'application d'envoyer des messages à d'autres programmes avec un minimum de mécanisme de protocole. Le protocole est orienté transaction, et la livraison et la protection en double ne sont pas garanties. Les applications nécessitant une livraison fiable et ordonnée de flux de données devraient utiliser le protocole TCP (Transmission Control Protocol)[9].

3.5 Protocole IGMP

Le protocole IGMP (Internet Group Management Protocol) est utilisé par les systèmes IPv4 (hôtes et routeurs) pour signaler leurs appartenances aux groupes de multidiffusion IP à tous les routeurs multicast voisins. Notez qu'un routeur multicast IP peut lui-même être membre d'un ou plusieurs groupes de multidiffusion, auquel cas il exécute à la fois la « partie routeur de multidiffusion » du protocole pour collecter les informations d'adhésion nécessaires à son routage multicast protocole et la "partie membre du groupe" du protocole pour informer lui-même et d'autres routeurs de multidiffusion voisins de ses appartenances.

IGMP est également utilisé pour d'autres fonctions de gestion de multidiffusion IP, en utilisant les types de messages autres que ceux utilisés pour les rapports d'appartenance à un groupe[10].

3.6 Protocole TLS

L'objectif principal du protocole TLS est d'assurer la confidentialité et l'intégrité des données entre deux applications communicantes. Le protocole est composé de deux couches : le protocole d'enregistrement TLS et le protocole de prise de contact TLS. Au niveau le plus bas, superposé à un protocole de transport fiable (par exemple, TCP), se trouve le protocole d'enregistrement TLS. Le protocole d'enregistrement TLS fournit une sécurité de connexion qui a deux propriétés de base :

- La connexion est privée. La cryptographie symétrique est utilisée pour le cryptage des données (par exemple, DES, RC4, etc.) Les clés de ce cryptage symétrique sont générées uniquement pour chaque connexion et sont basées sur un secret négocié par un autre protocole (tel que le TLS Protocole de poignée de main). Le protocole d'enregistrement peut également être utilisé sans cryptage.

- La connexion est fiable. Le transport des messages comprend un contrôle d'intégrité des messages à l'aide d'un MAC à clé. Les fonctions de hachage sécurisé (par exemple, SHA, MD5, etc.) sont utilisées pour les calculs MAC. Le protocole d'enregistrement peut fonctionner sans MAC, mais n'est généralement utilisé que dans ce mode pendant qu'un autre protocole utilise le protocole d'enregistrement comme moyen de transport pour la négociation des paramètres de sécurité.

Le protocole d'enregistrement TLS est utilisé pour l'encapsulation de divers protocoles de niveau supérieur. Un de ces protocoles encapsulés, le TLS Handshake Protocol, permet au serveur et au client de s'authentifier mutuellement et pour négocier un algorithme de cryptage et des clés cryptographiques avant que le protocole d'application ne transmette ou reçoive son premier octet de données. Le protocole TLS Handshake fournit une sécurité de connexion qui a trois propriétés de base :

- L'identité de l'homologue peut être authentifiée à l'aide d'une cryptographie asymétrique ou à clé publique (par exemple, RSA, DSS, etc.).Cette authentification peut être rendue facultative,

mais est généralement requise pour au moins un des pairs.

- La négociation d'un secret partagé est sécurisée : le secret négocié est indisponible pour les écoutes indiscrètes, et pour toute connexion authentifiée le secret ne peut être obtenu, même par un attaquant qui peut se placer au milieu de la connexion.

- La négociation est fiable : aucun attaquant ne peut modifier la communication de négociation sans être détecté par les parties à la communication[11].

3.7 Protocole IRC

Le protocole IRC (Internet Relay Chat) a été conçu sur un nombre d'années pour une utilisation avec des conférences textuelles, il a été développé sur des systèmes utilisant le protocole réseau TCP / IP, bien qu'il n'y ait aucune exigence que cela reste le seul domaine dans lequel il opère[12]. IRC lui-même est un système de téléconférence qui est bien adapté pour fonctionner sur de nombreuses machines grâce à l'utilisation de modèle client-serveur de manière distribuée. Une configuration typique implique un seul processus (le serveur) formant un point central pour les clients (ou d'autres serveurs) pour se connecter, effectuer la remise / multiplexage des messages requis et d'autres fonctions [12].

Il permet plusieurs possibilités de transfert de données entre clients, et tout comme avec d'autres mécanismes de transfert comme le courrier électronique, le destinataire des données doit faire attention à la manière dont les données sont traitées[13].

3.8 Protocole SSL

L'objectif principal du protocole SSL est d'assurer la confidentialité et la fiabilité entre deux applications communicantes. Le protocole est composé de deux couches. Au niveau le plus bas, superposé à un protocole de transport fiable (par exemple, TCP), se trouve le protocole d'enregistrement SSL. Le protocole d'enregistrement SSL est utilisé pour l'encapsulation de divers protocoles de niveau supérieur. Un de ces protocoles encapsulés, le protocole de prise de contact SSL, permet au serveur et au client de s'authentifier mutuellement et de négocier un algorithme de cryptage et des clés cryptographiques avant que le protocole d'application ne transmette ou reçoive son premier octet de données. Un avantage de SSL est qu'il est indépendant du protocole d'application. Un protocole de niveau supérieur peut se superposer au protocole SSL de manière transparente. Le protocole SSL fournit une sécurité de connexion qui a trois propriétés de base :

- La connexion est privée. Le chiffrement est utilisé après une négociation initiale pour définir une clé secrète. La cryptographie symétrique est utilisée pour le cryptage des données (par exemple, DES, 3DES, RC4).

- L'identité de l'homologue peut être authentifiée à l'aide d'une cryptographie asymétrique ou à clé publique (par exemple, RSA, DSS).

-La connexion est fiable. Le transport de message comprend une vérification de l'intégrité du message à l'aide d'un code d'authentification de message (MAC) . Les fonctions de hachage sécurisé (par exemple, SHA, MD5) sont utilisées pour les calculs MAC[14].

4 Conclusion

Dans ce chapitre nous avons fait une description des capteurs de botnets neris, virut, rbot-dos, donbot et sogou en donnant leurs chronologies ainsi que les adresses IP des hôtes qui ont contribué a l'échange de traffic réseau contenus dans ces capteurs vers la fin nous avons présenter les protocoles des paquets échanger au sein de réseau.

CHAPITRE 3

PRÉSENTATION DES ALGORITHMES DE CLASSIFICATIONS.

1 Introduction

Dans ce chapitre nous allons présenter les algorithmes de classification intituler : Naive bayes classifier, Support vector machine et les algorithmes de l'arbre de décision.

2 Naive Bayes classifier

Un classifieur Bayes naïf correspond à un réseau bayésien, comme dans l'equation suivante. Ici, une seule variable de classe C et m variables d'attribut Xi (les attributs sont discrets). Soit c une étiquette de classe et xi une valeur d'un attribut Xi. Un Bayes naïf induit une distribution : $Pr(c, xi, ..., xm) = Pr(c).\prod_{i=1}^{m} Pr(xi|c)$

où on a une classe a priori Pr(c) et des distributions conditionnelles Pr (xi | C). On peut estimer ces paramètres à partir de données (étiquetées), en utilisant le maximum de vraisemblance ou l'estimation MAP Une fois que nous avons appris un classificateur Bayes naïf à partir de données, nous pouvons étiqueter de nouvelles instances en sélectionnant l'étiquette de classe c * qui a une probabilité postérieure maximale étant donné l'observation sx1, ..., xm. Sélectionner : $c^* = argmax_c Pr(c|x1, ..., xm)$[27].

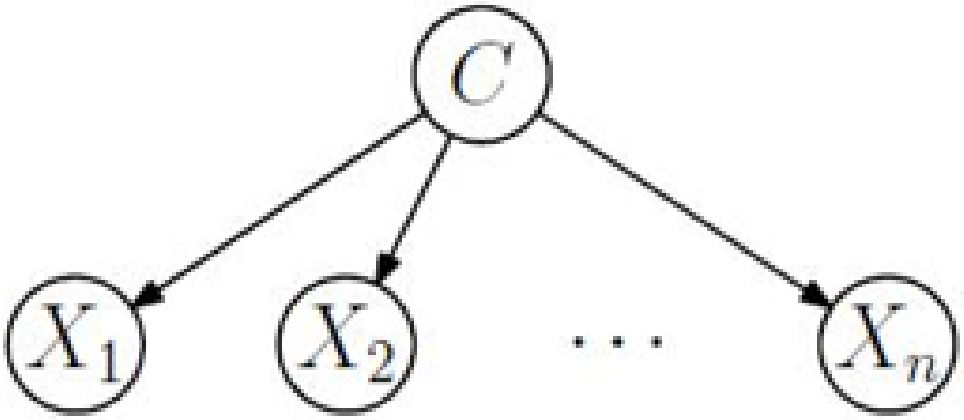

FIGURE 3.1 – Naïve Bayes Classifier[27]

2.1 Naive Bayes and augmented naive bayes

La classification est un problème fondamental dans l'apprentissage automatique . En classification, le but d'un algorithme d'apprentissage est de construire un classificateur à partir d'un ensemble d'exemples d'entraînement avec des étiquettes de classe. Régulièrement, l'exemple E est représenté par des valeurs d'attribut par un tuple (x1, x2, $\cdots$, xn), où xi est la valeur de l'attribut Xi. Soit C la variable de classification, et soit c la valeur de C. Il n'y a que deux classes ici : + (la classe positive) ou - (la classe négative).

Un classificateur est une fonction qui attribue une étiquette de classe à un exemple. Du point de vue des probabilités, selon la règle de Bayes, la probabilité qu'un exemple E = (x1, x2, $\cdots$, xn) soit de classe c est : $P(c|E) = \frac{p(E|c)p(c)}{p(E)}$

E est classé dans la classe C = + si et seulement si :$Fb(E) = \frac{p(C=+|E)}{p(C=-|E)} \geq 1$ Où Fb (E) est appelé un classificateur bayésien.

Supposons que tous les attributs sont indépendants compte tenu de la valeur de la variable de classe, pour effectuer une classification dans ce cas on doit appliquer la fonction suivante : $Fnb(E) = \frac{p(C=+)}{p(C=-)} \prod_{i=1}^{n} \frac{p(xi|C=+)}{p(xi|C=-)}$ La fonction Fnb (E) est appelée un classifieur bayésien naïf, ou simplement Bayes naïf (NB). Dans les Bayes naïfs, chaque nœud d'attribut n'a pas de parent à l'exception du nœud de classe. Naïve Bayes est la forme la plus simple de réseau bayésien, dans lequel tous les attributs sont indépendants compte tenu de la valeur de la variable de classe. C'est ce qu'on appelle l'indépendance conditionnelle. Il est clair que l'hypothèse d'indépendance conditionnelle est rarement correcte dans la plupart des applications du monde réel. Une approche simple pour contrôler la limitation des Bayes naïfs consiste à augmenter sa structure pour représenter explicitement les dépendances entre les attributs. Un bayésien naïf augmenté network, ou simplement Bayes naïf augmenté (ANB), est un Bayes naïf étendu, que le nœud de classe pointe directement

vers tous les nœuds d'attribut, et y trouve des liens entre les nœuds d'attribut. La figure suivante montre un exemple d'ANB. Du point de vue des possibilités, un ANB G montre une distribution de probabilité conjointe représentée ci-dessous. $PG(x1, \ldots, xn, c) = p(c) \prod_{i=1}^{n} p(xi|pa(xi), c)$ Où pa (xi) désigne une affectation aux valeurs des parents de Xi. The pa (Xi) est utilisé pour désigner les parents de Xi. ANB est une forme remarquable de réseaux bayésiens où aucun nœud n'est identifié comme un nœud de classe. Il a été démontré que tout réseau bayésien peut être représenté par un ANB. Ainsi, toute distribution de probabilité conjointe peut être rendue par un ANB [27].

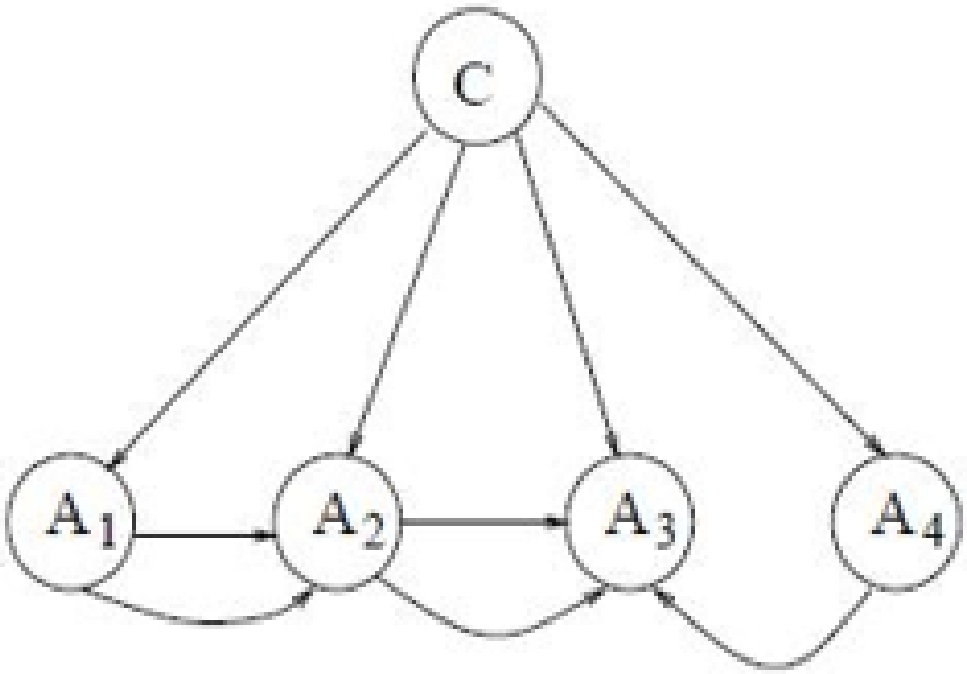

FIGURE 3.2 – exemble de ANB[27]

2.2 Applications de l'algorithmes naïve bayésienne

L'algorithme Naive Bayes est utilisé dans plusieurs scénarios réels tels que :

2.2.1 Classification du texte

Il est utilisé comme méthode d'apprentissage probabiliste pour la classification de texte. Le classifieur Naive Bayes est l'un des algorithmes connus les plus réussis en ce qui concerne la classification de documents texte, c'est-à-dire si un document texte appartient à une ou plusieurs catégories (classes).

2.2.2 Filtration du spam

C'est un exemple de classification de texte. C'est devenu un mécanisme populaire pour distinguer les courriers indésirables des courriers électroniques légitimes. Plusieurs services de messagerie modernes implémentent le filtrage bayésien des spams. De nombreux filtres de messagerie côté serveur, tels que DSPAM, Spam Bayes, Spam Assassin, Bogofilter et ASSP, utilisent cette technique.

2.2.3 Analyse des sentiments

Il peut être utilisé pour analyser le ton des tweets, des commentaires et des critiques, qu'ils soient négatifs, positifs ou neutres.

2.2.4 Système de recommandation

L'algorithme Naive Bayes en combinaison avec le filtrage collaboratif est utilisé pour créer des systèmes de recommandation hybrides qui aident à prédire si un utilisateur souhaite ou non une ressource donnée[27].

3 support vector machine (svm)

3.1 SVM d'un point de vue géométrique

Dans les tâches de classification, une technique d'apprentissage automatique discriminante vise à trouver, sur la base d'un ensemble de données d'entraînement indépendant et distribué de manière identique, une fonction discriminante capable de prédire correctement les étiquettes des instances nouvellement acquises. Contrairement aux approches d'apprentissage automatique génératif, qui nécessitent des calculs de distributions de probabilités conditionnelles, une fonction de classification discriminante prend un point de données x et l'affecte à l'une des différentes classes faisant partie de la tâche de classification. Moins puissantes que les approches génératives, qui sont principalement utilisées lorsque la prédiction implique la détection de valeurs aberrantes, les approches discriminantes nécessitent moins de ressources de calcul et moins de données d'apprentissage, en particulier pour un espace de caractéristiques multidimensionnel et lorsque seules des probabilités postérieures sont nécessaires. D'un point de vue géométrique, l'apprentissage d'un classifieur équivaut à trouver l'équation d'une surface multidimensionnelle qui sépare le mieux les différentes classes dans l'espace d'entités. SVM est une technique discriminante et, comme elle résout analytiquement le problème d'optimisation convexe, elle renvoie toujours le même paramètre optimal d'hyperplan, contrairement aux algorithmes génétiques (GA) ou perceptrons, qui sont tous deux largement utilisés pour la classification dans l'apprentissage automatique. Pour les perceptrons, les solutions dépendent fortement des critères d'initialisation et de terminaison. Pour un noyau spécifique qui transforme les données de l'espace d'entrée en espace de fonctionnalités, l'entraînement

renvoie des paramètres de modèle SVM définis de manière unique pour un ensemble d'apprentissage donné, tandis que le perceptron et GA Les modèles de classificateurs sont différents à chaque fois que la formation est initialisée. L'objectif des AG et des perceptrons est uniquement de minimiser les erreurs lors de l'entraînement, ce qui se traduira par plusieurs hyperplans répondant à cette exigence. Si de nombreux hyperplans peuvent être appris pendant la phase d'apprentissage, seul l'optimum est retenu, car l'apprentissage est pratiquement effectué sur des échantillons de la population même si les données de test peuvent ne pas présenter la même distribution que l'ensemble d'apprentissage. Lorsqu'ils sont formés avec des données qui ne sont pas représentatives de la population de données globale, les hyperplans sont sujets à une faible généralisation. La figure suivante illustre les différents hyperplans obtenus avec les classificateurs SVM, perceptron et GA sur des données bidimensionnelles à deux classes. Les points entourés de cercles représentent le vecteur de support, tandis que les hyperplans correspondant aux différents classificateurs qui sont représentés dans des couleurs différentes, conformément aux étiquettes. [25].

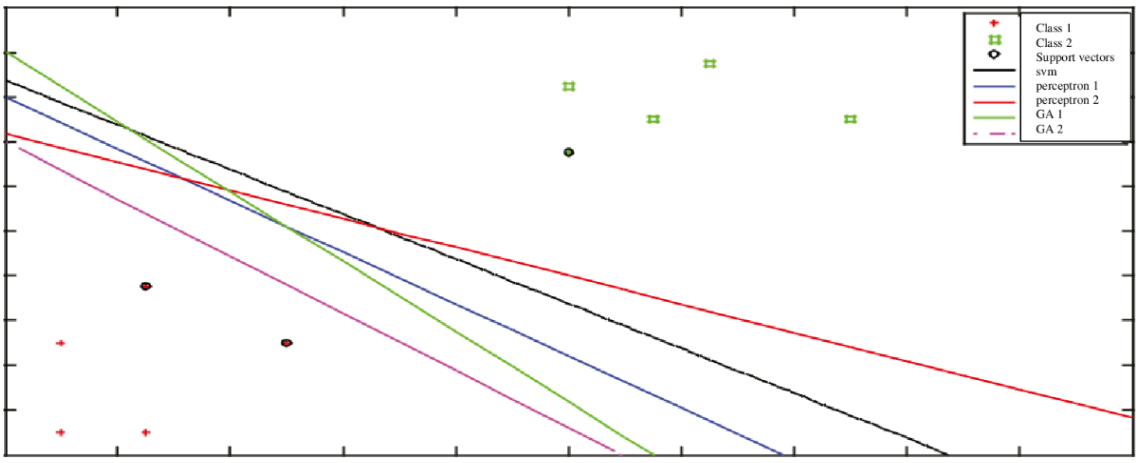

FIGURE 3.3 – Diagramme bidimensionnel à deux classes pour les hyperplans SVM, perceptron et GA.[25]

3.2 Propriétés de svm

3.2.1 SVM est une technique éparse

Comme les méthodes non paramétriques, SVM nécessite que toutes les données d'apprentissage soient disponibles, c'est-à-dire stockées en mémoire pendant la phase d'apprentissage, lorsque les paramètres du modèle SVM sont appris. Cependant, une fois les paramètres du modèle sont identifiés, SVM ne dépend que d'un sous-ensemble de ces instances d'entraînement, appelés vecteurs de support, pour les prévisions futures. Les vecteurs de support définissent les marges des hyperplans. Les vecteurs de support sont trouvés après une étape d'optimisation faisant intervenir une fonction objectif régularisée par un terme d'erreur et une contrainte, utilisant la relaxation lagrangienne.

La complexité de la tâche de classification avec SVM dépend du nombre de vecteurs de support plutôt que de la dimensionnalité de l'espace d'entrée. Le nombre de vecteurs de support qui sont finalement conservés à partir de l'ensemble de données d'origine dépend des données et varie en fonction de la complexité des données, qui est capturée par la dimensionnalité des données et la séparabilité des classes. La limite supérieure du nombre de vecteurs de support correspond à la moitié de la taille de l'ensemble de données d'entraînement, mais en pratique c'est rarement le cas[25].

3.2.2 SVM est une technique de noyau

SVM utilise l'astuce du noyau pour mapper les données dans un espace de plus grande dimension avant de résoudre la tâche d'apprentissage automatique sous la forme d'un problème d'optimisation convexe dans lequel les optima sont trouvés de manière analytique plutôt qu'heuristique, comme avec d'autres techniques d'apprentissage automatique. Souvent, les données réelles ne sont pas séparables linéairement dans l'espace d'entrée d'origine. En d'autres termes, les instances qui ont des étiquettes différentes partagent l'espace d'entrée d'une manière qui empêche un hyperplan linéaire de séparer correctement les différentes classes impliquées dans cette tâche de classification. Essayer d'apprendre une frontière de séparation non linéaire dans l'espace d'entrée cela augmente les exigences de calcul pendant la phase d'optimisation, car la surface de séparation sera au moins du second ordre. Au lieu de cela, SVM mappe les données, en utilisant des fonctions de noyau prédéfinies, dans un nouvel espace de dimension supérieure, où un séparateur linéaire serait capable de faire la distinction entre les différentes classes. La phase d'optimisation SVM entraînera donc l'apprentissage uniquement d'une surface discriminante linéaire dans l'espace cartographié[25].

3.2.3 SVM est un séparateur de marge maximal

Au-delà de la minimisation de l'erreur ou d'une fonction de coût, basée sur les jeux de données d'apprentissage (similaire à d'autres techniques d'apprentissage machine discriminantes), SVM impose une contrainte supplémentaire sur le problème d'optimisation : l'hyperplan doit être situé de telle sorte qu'il soit à une distance maximale du différentes classes. Une telle condition oblige l'étape d'optimisation à trouver l'hyperplan qui finirait par se généraliser mieux car il est situé à une distance égale et maximale des classes. Ceci est essentiel, car la formation est effectuée sur un échantillon de la population, tandis que la prédiction doit être effectuée sur des instances encore à voir qui peuvent avoir une distribution légèrement différente de celle du sous-ensemble formé[25].

3.3 SVM du noyau

Lorsqu'un problème n'est pas linéairement séparable dans l'espace d'entrée, SVM à marge souple ne peut pas trouver un hyperplan de séparation robuste qui minimise le nombre de points de

données mal classés et qui se généralise bien. Pour cela, un noyau peut être utilisé pour transformer les données en un espace dimensionnel supérieur, appelé espace noyau, où les données seront séparables linéairement. Dans l'espace noyau, un hyperplan linéaire peut ainsi être obtenu pour séparer les différentes classes impliquées dans la tâche de classification au lieu de résoudre une hypersurface de séparation d'ordre élevé dans l'espace d'entrée. C'est une méthode intéressante, car la surcharge liée au passage à l'espace noyau est insignifiante par rapport à l'apprentissage d'une surface non linéaire. Un noyau doit être une matrice semi-définie hermitienne et positive et doit satisfaire le théorème de Mercer, qui se traduit par l'évaluation du noyau ou de la matrice de Gram sur toutes les paires de points de données comme étant positives et semi-défini. La sélection du noyau dépend fortement des spécificités des données. Par exemple, le noyau linéaire - le plus simple de tous - est utile dans les grands vecteurs de données clairsemés. Cependant, il se classe derrière le noyau polynomial, ce qui évite de remettre à zéro le Hessian. Le noyau polynomial est largement utilisé dans le traitement d'images, alors que le noyau ANOVA RB est généralement réservé aux tâches de régression. Les RBF de Gauss et de Laplace sont des noyaux à usage général qui sont principalement appliqués en l'absence de connaissances préalables. Une matrice de noyau qui finit par être diagonale indique que l'espace des fonctionnalités est redondant et qu'un autre noyau doit être essayé après la réduction des fonctionnalités. Lorsque les noyaux sont utilisés pour transformer les vecteurs d'entités de l'espace d'entrée en espace noyau pour des ensembles de données linéairement non séparables, le calcul de la matrice du noyau nécessite une mémoire et des ressources de calcul massives, pour le Big Data. La figure suivante affiche les données bidimensionnelles OU exclusives (XOR), une distribution linéairement non séparable dans l'espace d'entrée (en haut à gauche) ainsi que dans l'espace des fonctionnalités. Dans ce dernier, 16 points (pour différents ensembles) sont créé pour les quatre entrées lorsque le noyau est appliqué. Le choix du paramètre gaussien de lissage du noyau RBF s2 affecte la distribution des données dans l'espace noyau. Le choix de la valeur du paramètre étant essentiel pour transformer les données d'un espace linéairement non séparable en un espace linéairement séparable, des recherches de grille sont effectuées pour trouver les valeurs les plus appropriées [25].

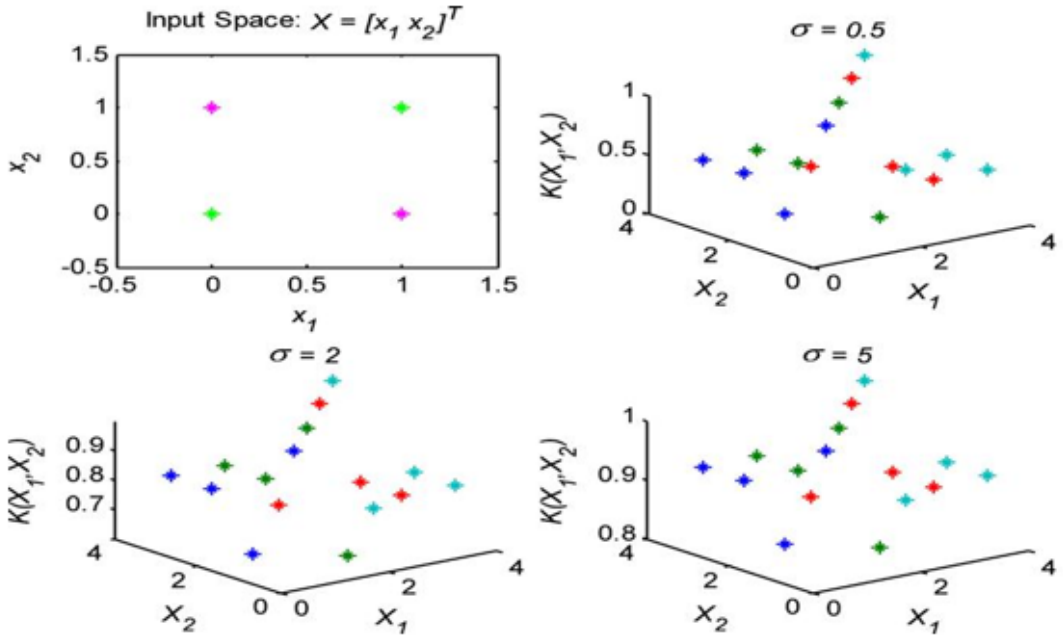

FIGURE 3.4 – Données XOR bidimensionnelles, de l'espace d'entrée à l'espace noyau.[25]

3.4 Multiclass SVM

Les premières extensions de la classification binaire SVM au cas multiclasse ont été le travail de Weston et Watkins (1999) et Platt (2000). Les chercheurs ont conçu diverses stratégies pour résoudre le problème de la multiclassification, y compris la classification un contre le reste, la classification par paires et la formulation de la multiclassification. Bien que le modèle paramétrique SVM permette des ajustements lors de la construction de la fonction discriminante, pour les problèmes multiclasses, ces paramètres ne correspondent pas toujours à l'ensemble de données. Pour cette raison, il est parfois préférable de partitionner les données en sous-groupes avec des caractéristiques similaires et de dériver les paramètres du classificateur séparément. Ce processus aboutit à un SVM à plusieurs étages (MSVM), ou SVM hiérarchique, qui peut produire une plus grande précision de généralisation et réduire la probabilité de surajustement, comme le montre Stockman (2010). Une représentation graphique d'un seul SVM et d'un MSVM est présentée à la figure suivante [25].

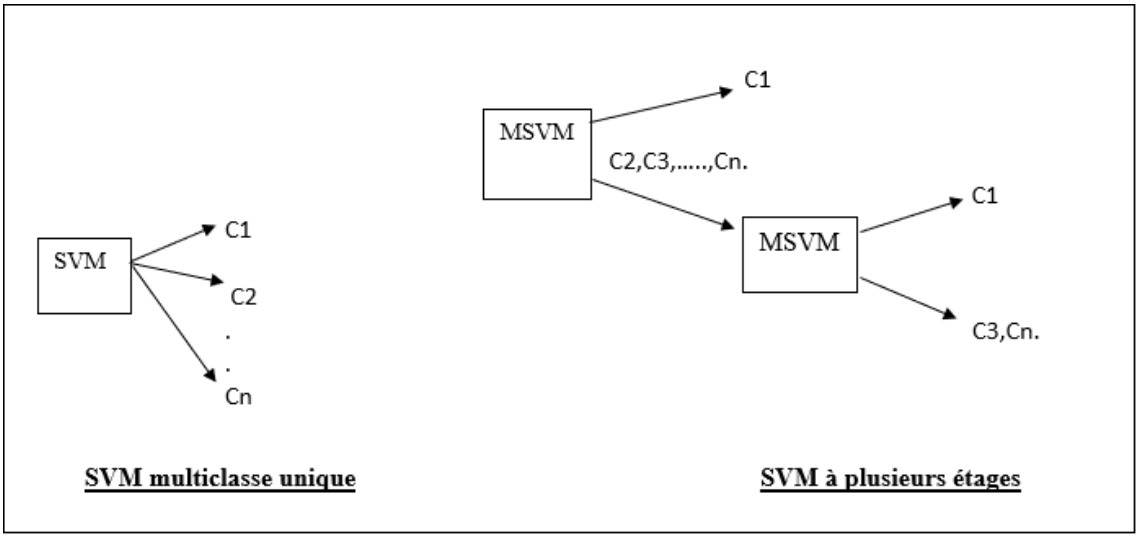

FIGURE 3.5 – SVM multiclasse unique et Flux MSVM.[25]

4 Arbre de décision

Un arbre normal comprend des racines, des branches et des feuilles. La même structure est suivie dans l'arbre de décision. Il contient le nœud racine, les branches et les nœuds feuilles. Le test d'un attribut se fait sur chaque nœud interne, le résultat du test est sur l'étiquette de branche et de classe, par conséquent sur le nœud feuille. Un nœud racine est le parent de tous les nœuds et, comme son nom l'indique, il s'agit du nœud le plus élevé dans l'arbre. Un arbre de décision est un arbre où chaque nœud montre une caractéristique (attribut), chaque lien (branche) montre une décision (règle) et chaque feuille montre un résultat (valeur catégorielle ou continue). Comme les arbres de décision imitent la pensée au niveau humain, il est si simple de saisir les données et de faire de bonnes interprétations. L'idée générale est de créer un arbre comme celui-ci pour l'ensemble des données et de traiter un seul résultat à chaque feuille [23].

4.1 Travaux connexes sur l'arbre de decission

L'arbre de décision est similaire au processus de prise de décision humaine et il est donc facile à comprendre. Il peut résoudre un problème donner dans les deux situations soit pour des données discrètes ou continues en entrée. L'exemple de l'arbre de décision est le suivant.

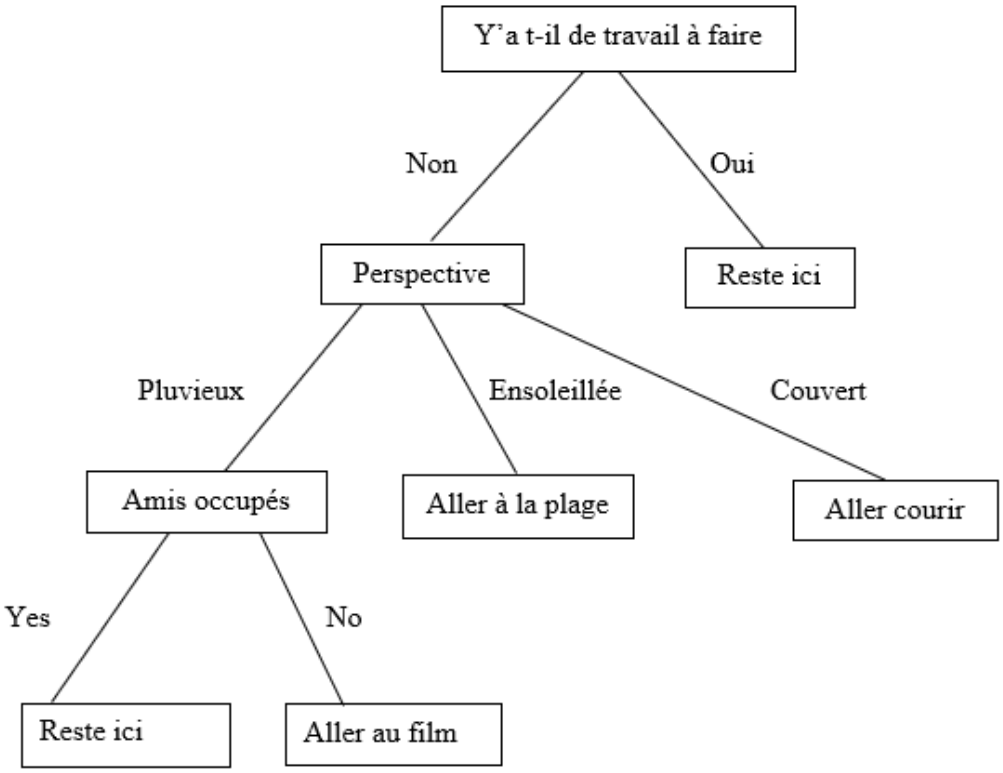

FIGURE 3.6 – Exemple d'arbre de décision sur ce qu'il faut faire lorsque différentes situations se produisent par temps.[23]

Lorsque les données n'offrent pas d'avantages lors du fractionnement, l'exécution est immédiatement arrêtée.

L'arbre de décision rend explicites toutes les alternatives possibles et trace chaque alternative jusqu'à sa conclusion dans une seule vue, pour faciliter la comparaison entre les différentes alternatives. La transparence dans la nature est l'un des meilleurs avantages de l'arbre de décision.

Un autre avantage principal est la possibilité de sélectionner la caractéristique la plus biaisée.

Il est aussi facile à classer et à interpréter les données. Le filtrage des variables et la section des caractéristiques sont suffisamment bons dans l'arbre de décision.

En parlant de ses performances, le non-linéaire n'affecte aucun des paramètres de l'arbre de décision [23].

4.2 Algorithmes de l'arbre de décision

Les algorithmes d'arbre de décision sont utilisés pour fractionner les attributs à tester à n'importe quel nœud afin de déterminer si le fractionnement est «Meilleur» dans les classes individuelles. Le résultat partitionné à chaque branche est PURE que possible, car les critères de fractionnement doivent être identiques[23]. Dans ce qui suit nous allons présenter les algorithmes de l'arbre de

décision.

4.2.1 ID3

L'algorithme ID3 (Iterative Dichotomized) est basé sur l'algorithme Concept Learning System (CLS). L'algorithme CLS est l'algorithme de base pour l'apprentissage de l'arbre de décision. La phase de croissance de l'arbre de CLS consiste à choisir l'attribut à tester à chaque nœud par le formateur. ID3 améliore CLS en ajoutant une heuristique pour la sélection d'attributs. ID3 est basé sur l'algorithme de Hunt et est implémenté en série. Cet algorithme partitionne récursivement le jeu de données d'apprentissage jusqu'à ce que les jeux d'enregistrements appartiennent à l'étiquette de classe en utilisant la technique de profondeur d'abord gloutonne. Dans la phase de croissance de la construction de l'arborescence, cet algorithme utilise le gain d'information, une mesure basée sur l'entropie, pour sélectionner le meilleur attribut de fractionnement, et l'attribut avec le gain d'information le plus élevé est sélectionné comme attribut de fractionnement. ID3 ne donne pas de résultat précis lorsqu'il y a trop de détails ou de bruit dans l'ensemble de données d'apprentissage, donc un prétraitement intensif des données est effectué avant de construire un modèle d'arbre de décision avec ID3. L'un des principaux inconvénients de l'ID3 est que la mesure Gain utilisée tend à favoriser les attributs avec un grand nombre de valeurs distinctes. Il n'accepte que les attributs catégoriels lors de la création d'un modèle d'arbre. Cet algorithme d'arbre de décision génère des branches variables par nœud [28].

4.2.2 C4.5

L'algorithme C4.5 est une version améliorée d'ID3, cet algorithme utilise le rapport de gain comme critère de division, au lieu de prendre un gain comme dans l'algorithme ID3 pour les critères de division en phase de croissance de l'arbre. Par conséquent, C4.5 est une évolution de ID3. Cet algorithme gère à la fois les attributs continus et discrets. Afin de gérer les attributs continus, C4.5 crée un seuil puis divise la liste en ceux dont la valeur d'attribut est supérieure au seuil et ceux qui lui sont inférieurs ou égaux. Comme ID3, les données sont triées à chaque nœud de l'arborescence afin de déterminer le meilleur attribut de fractionnement. Le fractionnement cesse lorsque le nombre d'instances à fractionner est inférieur à un certain seuil. Les principaux avantages de C4.5 sont que lors de la construction d'un arbre de décision, C4.5 peut traiter des ensembles de données qui ont des modèles avec des valeurs d'attribut inconnues. C4.5 peut également traiter le cas des attributs à domaines continus par discrétisation. Cet algorithme gère les données d'entraînement avec des valeurs d'attribut en permettant aux valeurs d'attribut d'être marquées comme manquantes. Les valeurs d'attribut manquantes ne sont tout simplement pas utilisées dans les calculs de gain et d'entropie. Il dispose d'une méthode améliorée d'élagage des arbres qui réduit les erreurs de classification dues au bruit ou à trop de détails dans l'ensemble de données d'apprentissage [28].

4.2.3 Classification and Regression Trees (CART)

CART se caractérise par le fait qu'il construit des arbres binaires, à savoir que chaque nœud interne a exactement deux arêtes sortantes alors que les deux algorithmes ID3, C4.5 génèrent les arbres de décision avec des branches variables par nœud. CART est unique par rapport aux autres algorithmes basés sur Hunt car il est également utilisé pour l'analyse de régression à l'aide d'arbres de régression. La fonction d'analyse de régression est utilisée pour prévoir une variable dépendante étant donné un ensemble de variables prédictives sur une période donnée. L'arbre de décision CART est une procédure de partitionnement récursif binaire capable de traiter des attributs continus et nominaux à la fois comme cibles et comme prédicteurs. Dans CART, les arbres sont construits, en utilisant l'indice gini pour la procédure de fractionnement, à une taille maximale sans l'utilisation d'une règle d'arrêt, puis élagués (essentiellement divisés par fractionnement) à la racine via l'élagage de complexité des coûts. Le mécanisme CART est destiné à produire non pas un, mais une séquence d'arbres élagués imbriqués, qui sont tous des arbres candidats optimaux. Le mécanisme CART comprend l'équilibrage automatique des classes (facultatif), la gestion automatique des valeurs manquantes et permet un apprentissage sensible au coût, la construction de caractéristiques dynamiques et l'estimation d'arbre de probabilité [28].

4.2.4 Best First Tree (BFT)

Les algorithmes standard tels que ID3, C4.5 et CART pour l'induction descendante des arbres de décision développent les nœuds dans le premier ordre de profondeur à chaque étape en utilisant la stratégie diviser-et-conquérir. L'attribut de sélection en C4.5 est basé sur le gain d'entropie dans la phase de croissance de l'arbre. L'attribut de sélection est basé sur l'index gini dans l'algorithme CART, puis divise les instances d'entraînement en sous-ensembles ; un pour chaque branche partant du nœud racine, le nombre de sous-ensembles est le même que le nombre de branches. Un ordre fixe est utilisé pour développer les nœuds (normalement, de gauche à droite) dans ces arbres de décision. Dans les arbres de décision Best-first, la sélection de la meilleure division est basée sur des algorithmes d'amplification qui sont utilisés pour étendre les nœuds dans le meilleur premier ordre au lieu d'un ordre fixe. Cet algorithme utilise à la fois l'indice de gain et l'indice gini pour calculer le meilleur nœud dans la phase de croissance de l'arbre. Cette méthode ajoute le «meilleur» nœud de fractionnement à l'arborescence à chaque étape. Le meilleur nœud est le nœud qui réduit au maximum les impuretés parmi tous les nœuds disponibles pour le fractionnement (c'est-à-dire non étiqueté comme nœuds terminaux). Bien que cela donne le même arbre entièrement développé que l'expansion standard en profondeur d'abord, cela nous permet d'étudier de nouvelles méthodes d'élagage d'arbres qui utilisent la validation croisée pour sélectionner le nombre d'expansions[28].

4.2.5 Supervised Learning In Quest (SLIC)

SLIQ a été l'un des premiers algorithmes évolutifs d'induction d'arbre de décision. Cela peut être implémenté dans un modèle série et parallèle. Il n'est pas basé sur l'algorithme de Hunt pour la classification des arbres de décision. Il partitionne un ensemble de données d'apprentissage de manière récursive en utilisant une stratégie gourmande en largeur d'abord intégrée à une technique de pré-tri pendant la phase de création de l'arborescence. SLIQ utilise un format de données vertical, ce qui signifie que toutes les valeurs d'un attribut ont été stockées sous forme de liste, qui a été triée au début de l'algorithme. Cela signifiait que les attributs n'avaient pas besoin d'être triés à plusieurs reprises à chaque nœud comme c'était le cas dans les algorithmes existants tels que CART et C4.5. Avec la technique de pré-tri, le tri aux nœuds de l'arbre de décision est éliminé et remplacé par un tri unique, avec l'utilisation d'une structure de données de liste pour chaque attribut afin de déterminer le meilleur point de partage. Le calcul de l'indice Gini pour chaque point de partage possible peut être effectué efficacement en stockant les distributions de classes dans des histogrammes, une par classe et par nœud. Cependant SLIQ utilise une structure de données résidant en mémoire appelée liste de classes qui stocke les étiquettes de classe de chaque enregistrement. Cette structure de données limite la taille des ensembles de données que SLIQ peut gérer. Lors de la construction d'un modèle d'arbre de décision, SLIQ gère à la fois les attributs numériques et catégoriels. L'un des inconvénients de SLIQ est qu'il utilise une structure de données de liste de classes résidant en mémoire, imposant ainsi des restrictions de mémoire sur les données. Il utilise le principe de longueur de description minimale (MDL) pour l'élagage de l'arbre après sa construction. MDL est une technique peu coûteuse d'élagage d'arbres qui utilise le moins de codage pour produire des arbres de petite taille. L'algorithme d'arbre de décision SLIQ produit des arbres de décision précis qui sont nettement plus petits que les arbres produits à l'aide de C4.5 et CART. Dans le même temps, SLIQ exécute presque un ordre de grandeur plus rapidement que CART[28].

4.2.6 Scalable Parallelizable Induction of Decision Tree algorithme (SPRINT)

Cet algorithme associe l'étiquette de classe avec l'identificateur d'enregistrement pour chaque valeur dans les listes d'attributs. C'est un classificateur d'arbre de décision rapide et évolutif. Il n'est pas basé sur l'algorithme de Hunt pour la construction de l'arbre de décision, mais il partitionne l'ensemble de données d'apprentissage de manière récursive en utilisant une technique gloutonne de largeur d'abord jusqu'à ce que chaque partition appartienne au même nœud feuille ou classe. Il s'agit d'une amélioration de SLIQ car il peut être implémenté à la fois en série et en parallèle pour un bon placement des données et un bon équilibrage de la charge. L'implémentation en série de SPRINT, comme SLIQ, utilise un tri ponctuel des éléments de données et il n'a aucune restriction sur la taille des données d'entrée. Contrairement à SLIQ, il utilise deux structures de données ; liste d'attributs et histogramme qui ne résident pas en mémoire rendant SPRINT approprié pour

un grand ensemble de données, ainsi il supprime toutes les restrictions de mémoire de données. Il gère à la fois les attributs continus et catégoriels[28].

4.2.7 Random forest

Random forest est un ensemble d'arbres de décision binaires non élagués, contrairement à d'autres classificateurs d'arbres de décision. random Forest cultive plusieurs arbres qui créent une forêt comme la classification. Chaque arbre est développé sur un échantillon amorcer de l'ensemble d'apprentissage (cela aide au sur-ajustement). La méthode d'apprentissage d'ensemble de random forest est une technique très prometteuse en termes de précision et offrant également un aspect distribué. Dans la phase de croissance arborescente des arbres standard (ID3, C4.5, CART, SLIQ, SPRINT, BFTree), chaque nœud est divisé en utilisant la meilleure répartition parmi toutes les variables. Dans une forêt aléatoire, chaque nœud est divisé en utilisant le meilleur parmi un sous-ensemble de prédicteurs choisis au hasard à ce nœud. Cette stratégie quelque peu contre-intuitive s'avère très performante par rapport à de nombreux autres classificateurs, y compris l'analyse discriminante, les machines vectorielles de support et les réseaux neuronaux, et est robuste contre le sur-ajustement. Random forest produit de meilleures performances par rapport à celles du classificateur d'arbre unique. Cette méthode est efficace en termes de calcul, ne surpasse pas , et robuste au bruit et peut également être appliquée lorsque le nombre de variables est beaucoup plus grand que le nombre d'échantillons. Il comprend une bonne méthode pour estimer les données manquantes et maintient l'exactitude lorsqu'une grande partie des données est manquante[28].

5 Conclusion

Dans ce chapitre nous avons présenté l'algorithme naive bayes classifier et ses domaines d'application, l'algorithme de support vector machine ses propriétés, ses types de noyau et ses extensions, enfin nous avons aborder les algorithmes de l'arbre de décision tels que : ID3, CART, SLIC et random forest. Dans le chapitre suivant nous allons utiliser les algorithmes support vector machine et random forest vue que sont plus performant par rapport aux autres algorithmes.

CHAPITRE 4

APPLICATION DES ALGORITHMES DE CLASSIFICATION SUR L'ENSEMBLE DES DONNÉES ET COMPARAISON DES RÉSULTATS

1 Introduction

Dans ce chapitre nous allons présenter notre environnement de travail et puis on va préparer les données pour qu'elles soit adapter à l'application des algorithmes de classification : support vector machine et random forest et enfin nous allons évaluer les performances des modèles obtenues en appliquant ces algorithmes.

2 Présentation de l'environement de travail

Jupyter Notebook est une application client-serveur créée par l'organisation à but non lucratif . Elle a été publiée en 2015. Elle permet la création et le partage de documents Web au format JSON constitués d'une liste ordonnée de cellules d'entrées et de sorties et organisés en fonction des versions successives du document. Les cellules peuvent contenir, entre autres, du code, du texte au format Markdown, des formules mathématiques ou des contenus médias. Le traitement se fait avec une application client fonctionnant par Internet, à laquelle on accède par les navigateurs habituels. Il est nécessaire pour cela que le serveur Jupyter Notebook soit installé et activé dans le système. Les documents Jupyter créés peuvent être exporter par exemple aux formats HTML, PDF, Markdown ou Python, ou bien se partager par email, avec Dropbox, GitHub ou un lecteur Jupyter Notebook[15]. Dans notre cas nous allons coder avec python. La figure ci-dessous nous montre un apercu sur la page d'accueil de jupyter.

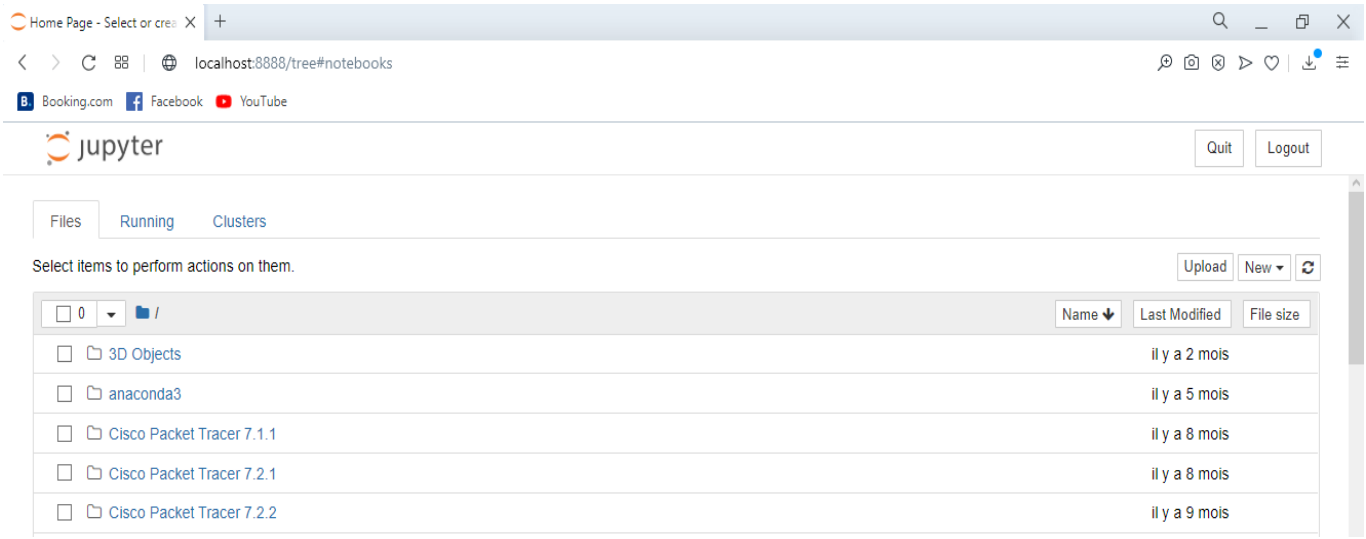

FIGURE 4.1 – un apercu sur la page d'accueil de jupyter notebook.

3 Préparation des données

Les capteurs de botnet contiennent des paquets d'un traffic de réseau universitaire dont on trouve les informations suivantes : numéro, temps, adresse source, adresse destination, protocole, longueur et information. La figure ci-dessous nous montre l'entête de fichier csv donbot.

```
In [1]: import os
        import pandas as pd
        os.chdir("C:/Users/messouaf yacine/Desktop/fichiers csv")
        donbot = pd.read_csv (r'donbot.csv')
        donbot.head()
```

Out[1]:

	No.	Time	Source	Destination	Protocol	Length	Info
0	1	0.000000	147.32.84.165	91.212.135.158	TCP	62	1040 > 5678 [SYN] Seq=0 Win=64240 Len=0 MSS=...
1	2	0.000009	147.32.84.165	91.212.135.158	TCP	62	[TCP Out-Of-Order] 1040 > 5678 [SYN] Seq=0 W...
2	3	0.062970	91.212.135.158	147.32.84.165	TCP	62	5678 > 1040 [SYN, ACK] Seq=0 Ack=1 Win=65535...
3	4	0.063292	147.32.84.165	91.212.135.158	TCP	60	1040 > 5678 [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	5	0.063304	147.32.84.165	91.212.135.158	TCP	60	[TCP Dup ACK 4#1] 1040 > 5678 [ACK] Seq=1 Ac...

FIGURE 4.2 – l'entête du fichier donbot.csv.

Pour qu'on puisse appliquer les algorithmes de classification sur les capteurs de données qui sont enregistré sous forme de fichiers csv, on doit ajouter une colonne 'state' qui indique l'état de paquet, et cela dépend de l'adresse source de l'hôte dont le paquet provient de sorte que si ce dernier provient d'un hôte infecté la valeur contenante dans la colonne 'state' est égale a 1, par

conséquent le paquet en question est malveillant, et si est égale a 0 cela implique que le paquet et l'hôte dont il provient sont sains, ces valeurs (1 et 0) vont être utiliser comme cible pour avoir un modèle de classification efficace.

Dans ce qui suit nous avons le programme qui doit être exécuter pour ajouter la colonne 'state' et la remplir.

```
from csv import writer
from csv import reader
import os
os.chdir("C:/Users/messouaf yacine/Desktop/fichiers csv")
with open('donbot.csv', 'r') as read_obj, \
open('donbot-MAJ.csv', 'w', newline='') as write_obj:
# Create a reading object
csv_reader = reader(read_obj)
head=[next(csv_reader)]
# Create a write object for the destination file
csv_writer = writer(write_obj)
head.append('state')
csv_writer.writerow(head)
# read each line as a list ignoring the header
for row in csv_reader:
#add 1 if the address is infected and 0 if it is not infected
if "147.32.84.165" in row[2]:
row.append('1')
# Add the updated row / list to the output file
csv_writer.writerow(row)
else:
row.append('0')
# Add the updated row / list to the output file
csv_writer.writerow(row)
```

En exécutant le programme ci-dessus notre fichier csv sera modifié comme nous le montre l'entête de nouveau fichier dans la figure ci-dessous.

In [1]:
```
import pandas as pd
import os
os.chdir("C:/Users/messouaf yacine/Desktop/fichiers csv")
donbot_MAJ= pd.read_csv("donbot-MAJ.csv", delimiter=',', header=None, skiprows=1,
names=['No.','Time','Source','Destination','Protocol','Length','Info','state'])
donbot_MAJ.head()
```

Out[1]:

	No.	Time	Source	Destination	Protocol	Length	Info	state
0	1	0.000000	147.32.84.165	91.212.135.158	TCP	62	1040 > 5678 [SYN] Seq=0 Win=64240 Len=0 MSS=...	1
1	2	0.000009	147.32.84.165	91.212.135.158	TCP	62	[TCP Out-Of-Order] 1040 > 5678 [SYN] Seq=0 W...	1
2	3	0.062970	91.212.135.158	147.32.84.165	TCP	62	5678 > 1040 [SYN, ACK] Seq=0 Ack=1 Win=65535...	0
3	4	0.063292	147.32.84.165	91.212.135.158	TCP	60	1040 > 5678 [ACK] Seq=1 Ack=1 Win=64240 Len=0	1
4	5	0.063304	147.32.84.165	91.212.135.158	TCP	60	[TCP Dup ACK 4#1] 1040 > 5678 [ACK] Seq=1 Ac...	1

FIGURE 4.3 – l'entête de fichier donbot.csv aprés la modification.

nous avons appliqué le même traitement sur toutes les captures de botnet pour qu'elle soit adapté à l'application des algorithmes de classification.

4 Application des algorithmes

nous avons appliqué le même traitement sur toute les captures de botnet pour qu'elle soit adapté à l'application des algorithmes de classification.

4.1 Application sur neris

Ici nous allons appliquer les algorithmes support vector machine et random forest sur la capteur de botnet neris, par conséquence on remarque que les deux classes ne sont pas parfaitement séparé en appliquant l'algorithme support vector machine comme le montre la figure 4.4 contrairement au résultat obtenu en utilisant l'algorithme random forest où on constate que les deux classes sont bien séparé comme l'illustre la figure 4.5.

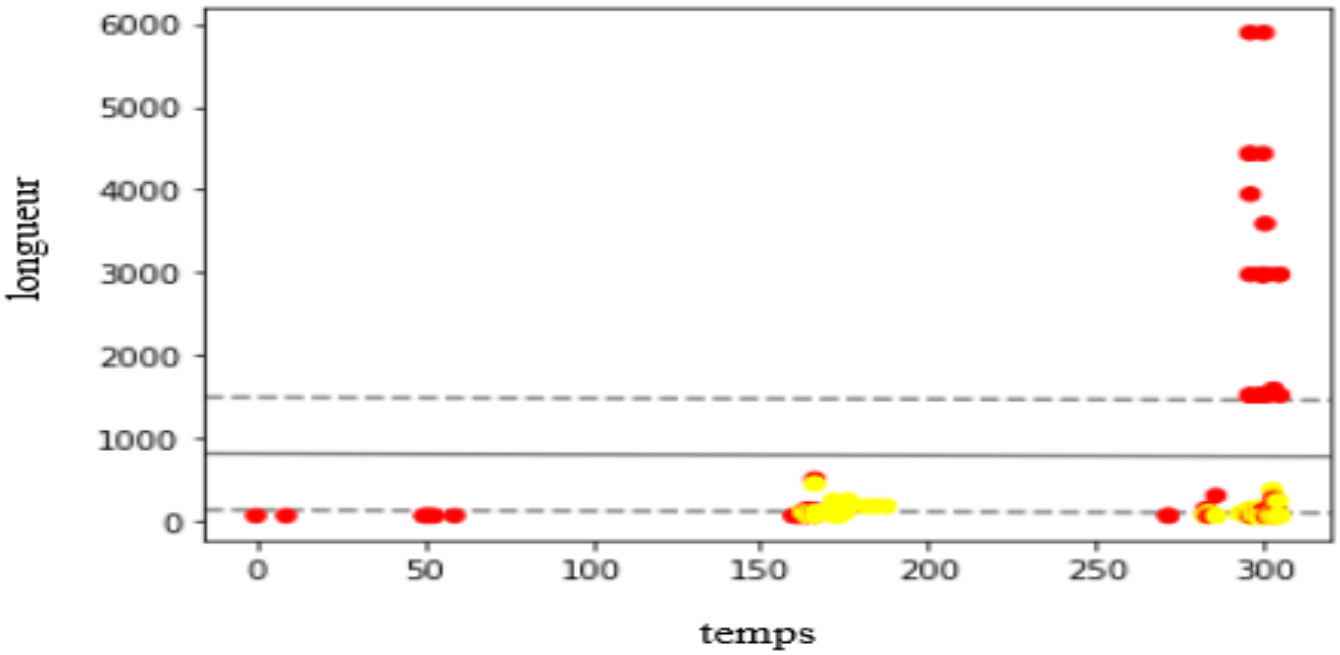

FIGURE 4.4 – svm sur neris.

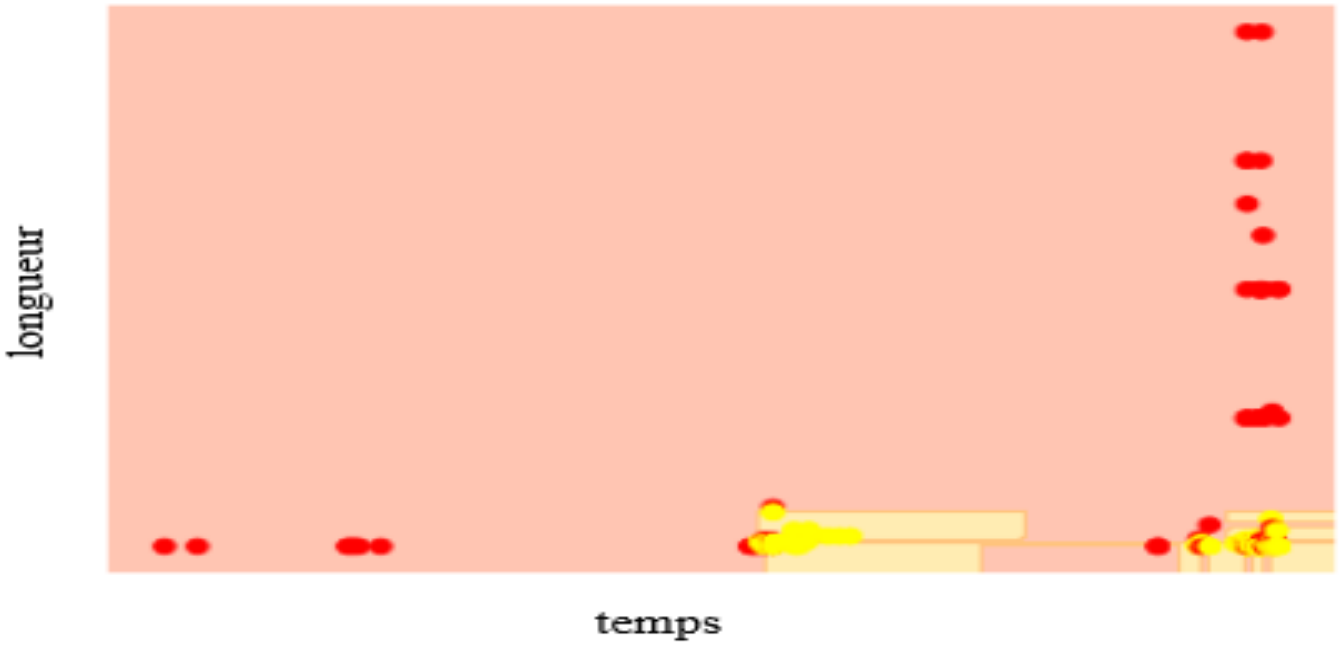

FIGURE 4.5 – random forest sur neris.

4.2 Application sur virut

Ici nous allons appliquer les algorithmes support vector machine et random forest sur la capteur de botnet virut, par conséquence on remarque que les deux classes ne sont pas parfaitement séparé

en appliquant l'algorithme support vector machine comme le montre la figure 4.6, contrairement au résultat obtenu en utilisant l'algorithme random forest où on constate que les deux classes sont bien séparé comme l'illustre la figure 4.7.

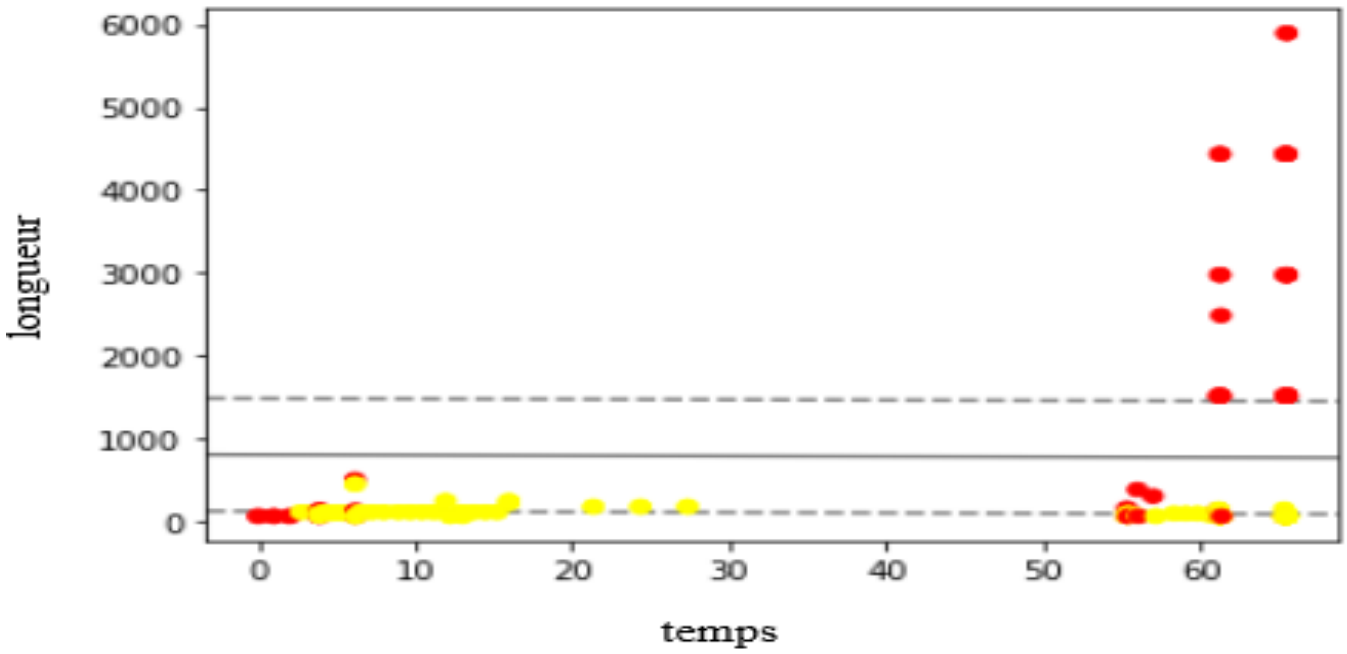

FIGURE 4.6 – svm sur virut.

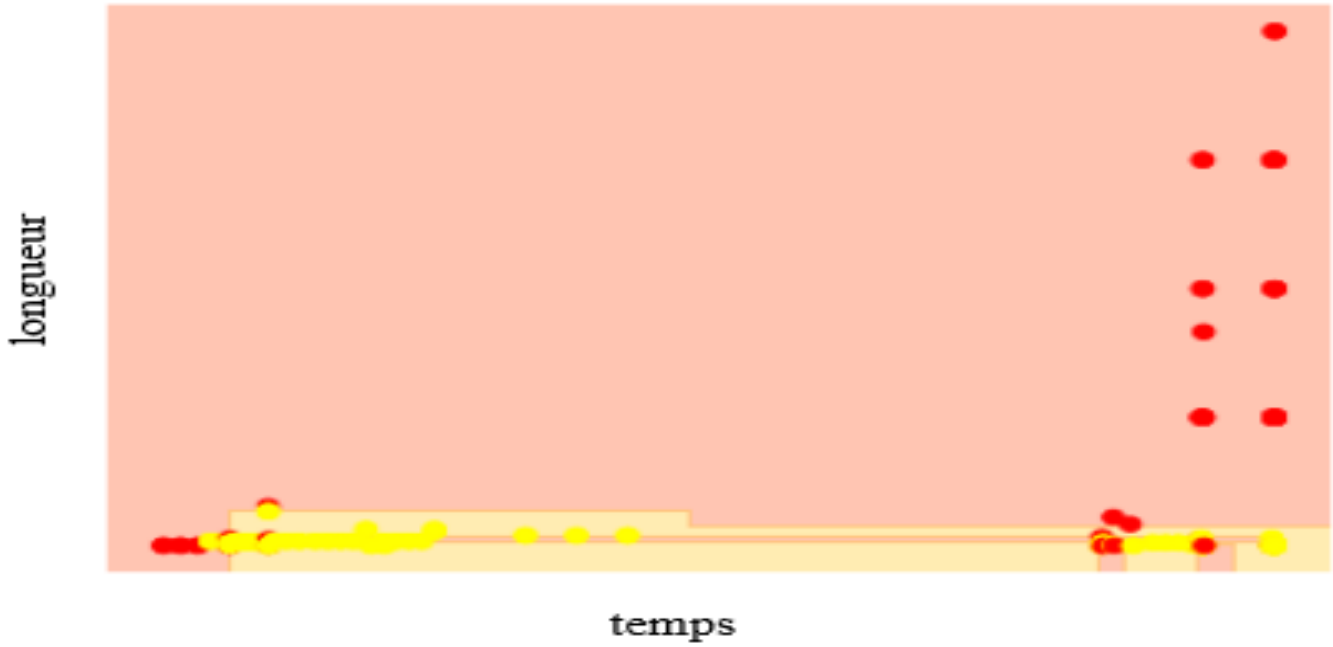

FIGURE 4.7 – random forest sur virut.

4.3 Application sur rbot-dos

Ici nous allons appliquer les algorithmes support vector machine et random forest sur la capteur de botnet rbot-dos, par conséquence on remarque que les deux classes ne sont pas parfaitement séparé en appliquant l'algorithme svm comme le montre la figure 4.8, contrairement au résultat obtenu en utilisant l'algorithme random forest où on constate que les deux classes sont bien séparé comme l'illustre la figure 4.9.

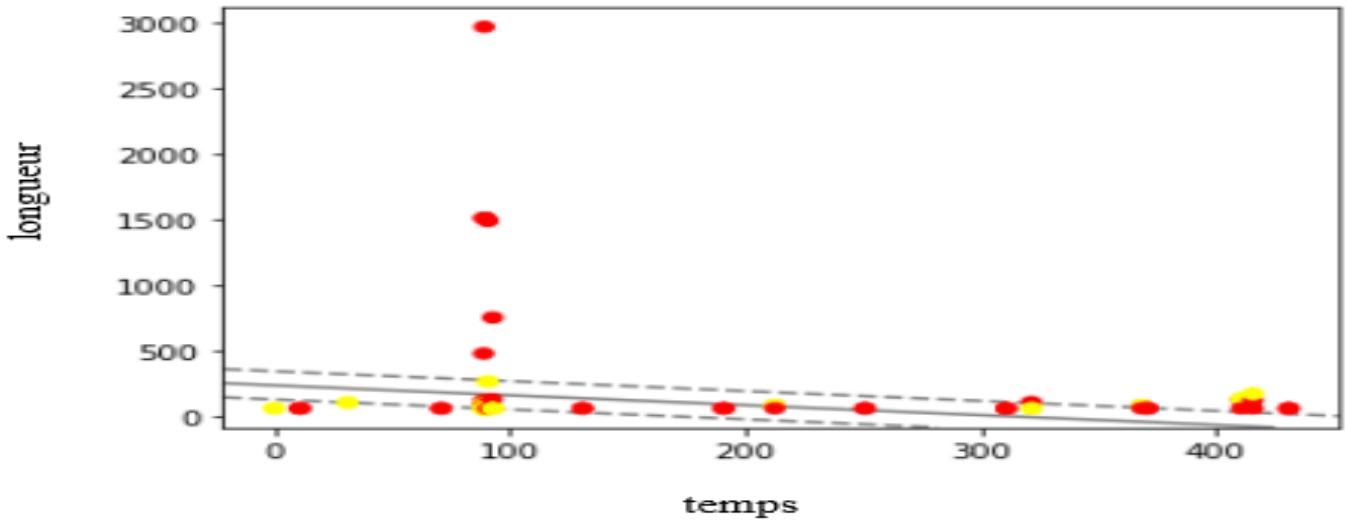

FIGURE 4.8 – svm sur rbot-dos.

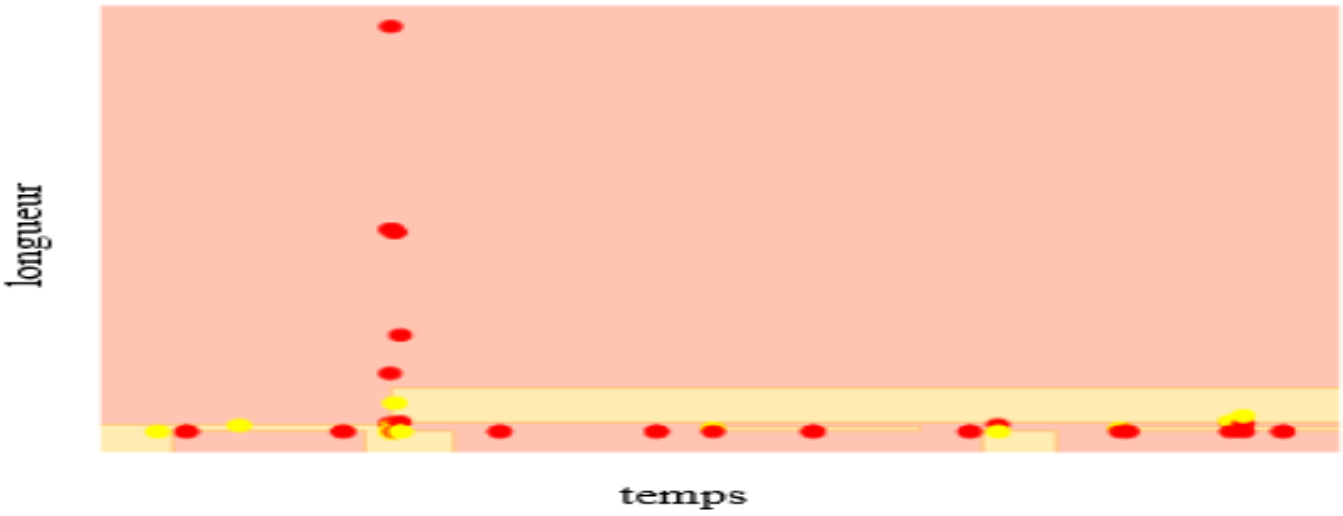

FIGURE 4.9 – random forest sur rbot-dos.

4.4 Application sur donbot

Ici nous allons appliquer les algorithmes support vector machine et random forest sur la capteur de botnet donbot, par conséquence on remarque que les deux classes sont bien séparé, a l'exception d'un seul point rouge qui a dépassé la limite de la ligne qui sépare les deux classes en appliquant l'algorithme support vector machine sur la capteur donbot comme le montre la figure 4.10.

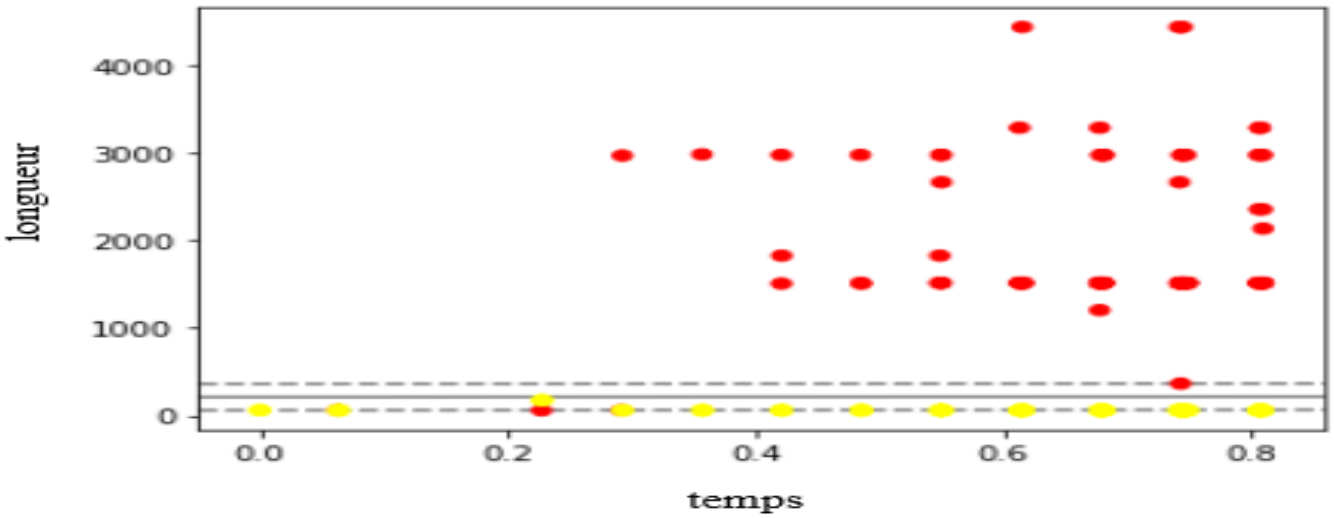

FIGURE 4.10 – svm sur donbot.

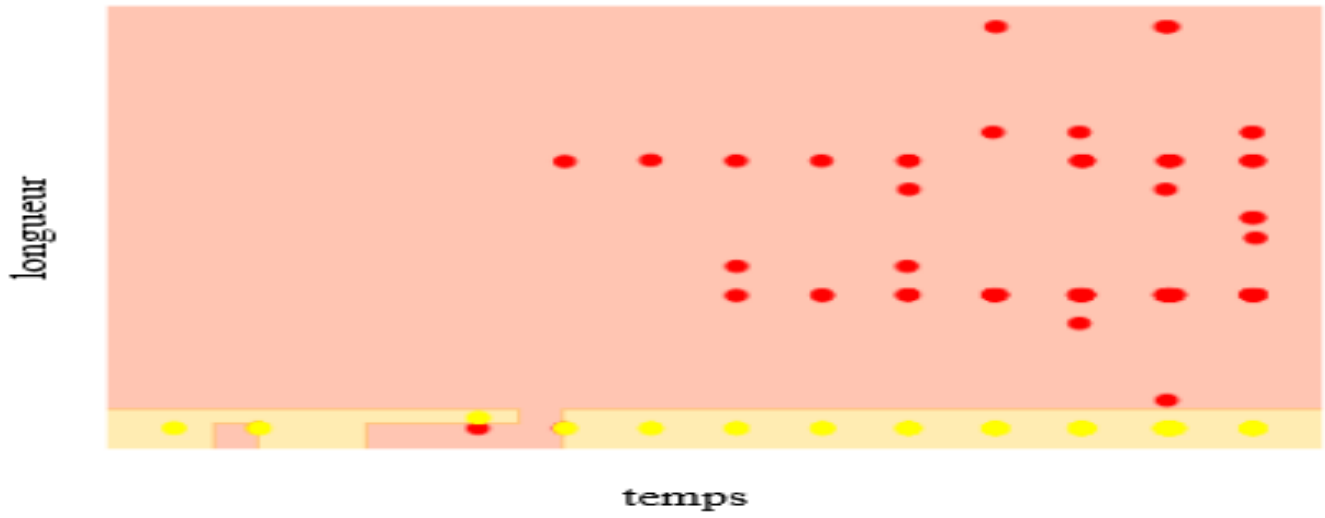

FIGURE 4.11 – random forest sur donbot.

4.5 Application sur sogou

Ici nous allons appliquer les algorithmes support vector machine et random forest sur la capteur de botnet sogou, par conséquence on remarque que les deux classes ne sont pas parfaitement séparé en appliquant l'algorithme support vector machine comme le montre la figure 4.12, contrairement au résultat obtenu en utilisant l'algorithme random forest où on constate que les deux classes sont bien séparé comme l'illustre la figure 4.13.

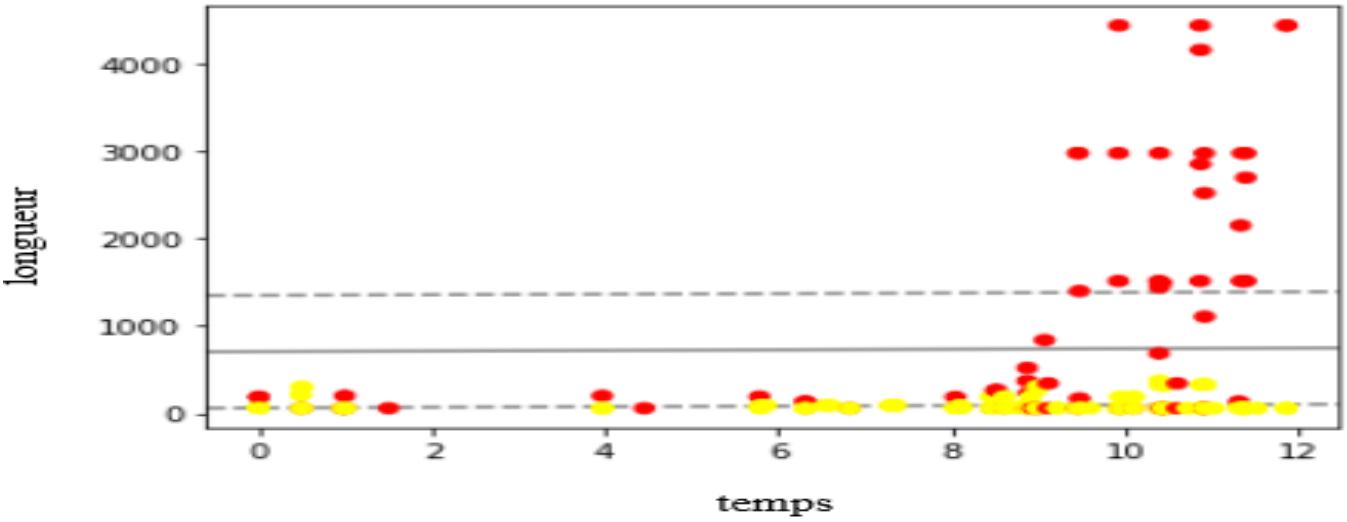

FIGURE 4.12 – svm sur sogou.

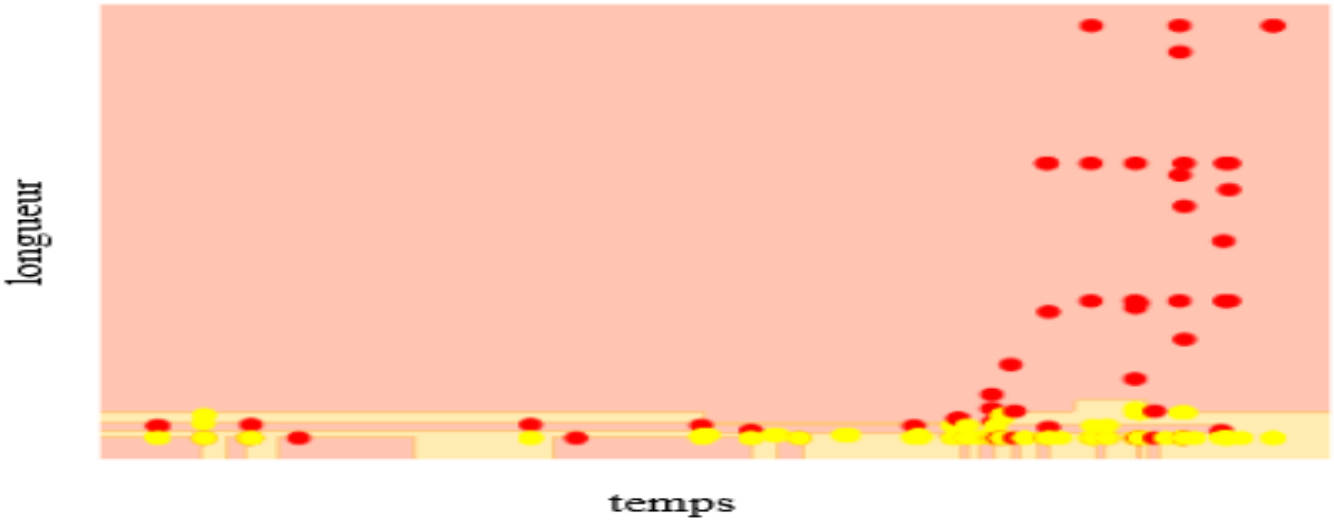

FIGURE 4.13 – random forest sur sogou.

5 Évaluation des performance

Dans cette section nous allons évaluer les performances des modèles obtenues en appliquant les algorithmes support vector machine et random forest sur les capteurs de données pour cela nous allons exécuter un code qui va déviser l'ensemble de données en données d'entrainement et données de test pour évaluer les performances de prédiction de l'appartenance des données de test a une classe bien précise(0 ou 1).

La figure ci-dessous nous montre la catégorie des résultats de prédiction des données de test qu'on peut avoir par rapport à deux types de classes : réelle et prédite. Après l'exécution d'un code

		Classe réelle	
		-	+
Classe prédite	-	True Negatives *(vrais négatifs)*	False Negatives *(faux négatifs)*
	+	False Positives *(faux positifs)*	True Positives *(vrais positifs)*

FIGURE 4.14 – catégorie des résultats de prédiction des données de test[16]

on aura un tableau d'évaluation de performances dont on trouve des valeurs telle que le rappel, la précession et le F-mesure qu'on va définir ci-dessous comme suit :

- Le rappel : "recall" en anglais est le taux de vrais positifs, c'est à dire la proportion de positifs que l'on a correctement identifiés.
- La précision : "precision" en anglais c'est la proportion de prédictions correctes parmi les points que l'on a prédits positifs.
- le "F-mesure" : "F-score" en anglais Pour évaluer un compromis entre rappel et précision, qui est leur moyenne harmonique[16].

5.1 Évaluation des resultats obtenues sur neris

Aprés l'exécution du code qui nous a permis d'obtenir un tableau d'évaluation des modèles, on constate que les resultats obtenues en appliquant l'algorithme support vector machine sur neris sont comme suit :

- pour la classe 0 : on a 0.23 de 'precision', 1.00 de 'recall' et 0.38 de 'f1-score'.
- pour la classe 1 : on a 1.00 de 'precision', 0.73 de 'recall' et 0.38 de 'f1-score'.

cela veut dire que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme support vector machine n'est pas toujours vrai.

Par contre les résultats obtenues en appliquant l'algorithme random forest sur neris sont égaux a 1 dans la 'precision', le 'recal' et le 'f1-score' a la fois ce qui implique que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme random forest est totalment vrai

Les résultats de la prédiction de l'appartenance des données sont illustré dans les deux figures ci-dessous.

	précision	**rappel**	**f1-mesure**	**support**
0	0.23	1.00	0.38	3
1	1.00	0.73	0.84	37
précision			0.75	40
macro moyenne	0.62	0.86	0.61	40
moyenne pondérée	0.94	0.75	0.81	40

TABLE 4.1 – evaluation des performances de svm appliquer sur neris.

	précision	**rappel**	**f1-mesure**	**support**
0	1.00	1.00	1.00	13
1	1.00	1.00	1.00	27
précision			1.00	40
macro moyenne	1.00	1.00	1.00	40
moyenne pondérée	1.00	1.00	1.00	40

TABLE 4.2 – evaluation des performances de random forest appliquer sur neris.

5.2 Évaluation des resultats obtenues sur virut

Aprés l'exécution du code qui nous a permis d'obtenir un tableau d'évaluation des modèles, on constate que les resultats obtenues en appliquant l'algorithme support vector machine sur virut sont comme suit :

— pour la classe 0 : on a 0.56 de 'precision', 1.00 de 'recall' et 0.71 de 'f1-score'.

— pour la classe 1 : on a 1.00 de 'precision', 0.95 de 'recall' et 0.95 de 'f1-score'.

cela veut dire que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme support vector machine n'est pas toujours vrai.

Par contre les résultats obtenues en appliquant l'algorithme random forest sur virut sont égaux a 1 dans la 'precision', le 'recal' et le 'f1-score' a la fois ce qui implique que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme random forest est totalment vrai.

Les résultats de la prédiction de l'appartenance des données sont illustré dans les deux figures ci-dessous.

	précision	**rappel**	**f1-mesure**	**support**
0	0.56	1.00	0.71	5
1	1.00	0.91	0.95	45
précision			0.92	50
macro moyenne	0.78	0.96	0.83	50
moyenne pondérée	0.96	0.92	0.93	50

TABLE 4.3 – evaluation des performances de svm appliquer sur virut.

	précision	**rappel**	**f1-mesure**	**support**
0	1.00	1.00	1.00	9
1	1.00	1.00	1.00	41
précision			1.00	50
macro moyenne	1.00	1.00	1.00	50
moyenne pondérée	1.00	1.00	1.00	50

TABLE 4.4 – evaluation des performances de random forest appliquer sur virut.

5.3 Évaluation des resultats obtenues sur rbot-dos

Aprés l'exécution du code qui nous a permis d'obtenir un tableau d'évaluation des modèles, on constate que les resultats obtenues en appliquant l'algorithme support vector machine sur rbot-dos sont comme suit :

— pour la classe 0 : on a 0.32 de 'precision', 0.67 de 'recall' et 0.43 de 'f1-score'.

— pour la classe 1 : on a 0.89 de 'precision', 0.65 de 'recall' et 0.75 de 'f1-score'.

cela veut dire que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme support vector machine n'est pas toujours vrai.

Par contre les résultats obtenues en appliquant l'algorithme random forest sur rbot-dos sont égaux a 1 dans la 'precision', le 'recal' et le 'f1-score' a la fois ce qui implique que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme random forest est totalment vrai

Les résultats de la prédiction de l'appartenance des données sont illustré dans les deux figures ci-dessous.

	précision	**rappel**	**f1-mesure**	**support**
0	0.32	0.67	0.43	12
1	0.89	0.65	0.75	48
précision			0.65	60
macro moyenne	0.60	0.66	0.59	60
moyenne pondérée	0.77	0.65	0.68	60

TABLE 4.5 – evaluation des performances de svm appliquer sur rbot-dos.

	précision	**rappel**	**f1-mesure**	**support**
0	1.00	1.00	1.00	25
1	1.00	1.00	1.00	35
précision			1.00	60
macro moyenne	1.00	1.00	1.00	60
moyenne pondérée	1.00	1.00	1.00	60

TABLE 4.6 – evaluation des performances de random forest appliquer sur rbot-dos.

5.4 Évaluation des resultats obtenues sur donbot :

Aprés l'exécution du code qui nous a permis d'obtenir un tableau d'évaluation des modèles, on constate que les résultats obtenues en appliquant l'algorithme support vector machine et random forest sur donbot sont egaux a 1 dans la 'precision', le 'recal' et le 'f1-score' a la fois, cela veut dire que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme svm et random forest sur donbot sont totalment vrai.

Les résultats de la prédiction de l'appartenance des données sont illustré dans les deux figures ci-dessous.

	précision	**rappel**	**f1-mesure**	**support**
0	1.00	1.00	1.00	24
1	1.00	1.00	1.00	36
précision			1.00	60
macro moyenne	1.00	1.00	1.00	60
moyenne pondérée	1.00	1.00	1.00	60

TABLE 4.7 – evaluation des performances de svm appliquer sur donbot.

	précision	**rappel**	**f1-mesure**	**support**
0	1.00	1.00	1.00	24
1	1.00	1.00	1.00	36
précision			1.00	60
macro moyenne	1.00	1.00	1.00	60
moyenne pondérée	1.00	1.00	1.00	60

TABLE 4.8 – evaluation des performances de random forest appliquer sur donbot.

5.5 Évaluation des resultats obtenues sur sogou

Aprés l'exécution du code qui nous a permis d'obtenir un tableau d'évaluation des modèles, on constate que les resultats obtenues en appliquant l'algorithme support vector machine sur sogou sont comme suit :

— pour la classe 0 : on a 0.36 de 'precision', 1.00 de 'recall' et 0.53 de 'f1-score'.

— pour la classe 1 : on a 1.00 de 'precision', 0.84 de 'recall' et 0.91 de 'f1-score'.

cela veut dire que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme support vector machine n'est pas toujours vrai.

Par contre les résultats obtenues en appliquant l'algorithme random forest sur sogou sont égaux a 1 dans la 'precision', le 'recal' et le 'f1-score' a la fois, ce qui implique que la prédiction de l'appartenance des données a la classe 0 ou 1 avec l'algorithme random forest est totalment vrai.

Les résultats de la prédiction de l'appartenance des données sont illustré dans les deux figures ci-dessous.

	précision	**rappel**	**f1-mesure**	**support**
0	0.36	1.00	0.53	5
1	1.00	0.84	0.91	55
précision			0.85	60
macro moyenne	0.68	0.92	0.72	60
moyenne pondérée	0.95	0.85	0.88	60

TABLE 4.9 – evaluation des performances de svm appliquer sur sogou.

	précision	**rappel**	**f1-mesure**	**support**
0	1.00	1.00	1.00	14
1	1.00	1.00	1.00	46
précision			1.00	60
macro moyenne	1.00	1.00	1.00	60
moyenne pondérée	1.00	1.00	1.00	60

TABLE 4.10 – evaluation des performances de random forest appliquer sur sogou.

6 Conclusion

Aprés la séparation des classes saine et malveillante en appliquant les algorithmes support vector machine et random forest sur l'ensemble des données comme il est illustré dans les représentations graphique précédentes on constate que la séparation était meilleur avec l'algorithme random forest du même pour l'évaluation des résultats, la prédiction de l'appartenance des paquets a une classe donné était bien précise avec l'algorithme random forest par conséquence on peut adapter le modèle de classification obtenu en appliquant l'algorithme random forest sur le trafic réseau pour détecter et arrêter le trafic malveillant.

CONCLUSION GÉNÉRALE :

Les avancées technologiques combinées à une forte demande du marché encourageront l'adoption et le déploiement massifs de l'Internet des objets, cela renforcera les menaces de sécurité traditionnelles sur les données et les réseaux. Mais en outre, le rapprochement du monde physique et virtuel a travers l'internet des objets ont ouvert la voie à de nouvelles menaces exploiter a travers des botnets qui affecteront directement l'intégrité des objets eux-mêmes, les infrastructures et processus ainsi que la vie privée des personnes.

Dans ce travail, nous avons parlé sur la sécurité des réseaux en présentant les architecteurs TCP/IP et OSI, les réseaux peer to peer ainsi que l'architecteur, les protocoles et la sécurité de l'internet des objets. Ensuite nous avons fait une description des capteurs de botnets sur lesquelles on a appliqué des algorithmes de classification pour les détecter. Dans ce qui suit nous avons présenter ces algorithmes ainsi que leurs applications sur les fichiers botnet et puis on les a évalués, c'est ce qui nous a offert l 'occasion de découvrir et d'implémenter des solutions baser sur l'apprentissage automatique.

RÉFÉRENCES WEBLIOGRAPHIQUES

[1] https ://mcfp.weebly.com/the-ctu-13-dataset-a-labeled-dataset-with-botnet-normal-and-background-traffic.html, Consulté le 15/04/2020.

[2] https ://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-42/, Consulté le 15/04/2020.

[3] https ://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-46/, Consulté le 15/04/2020.

[4] https ://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-45/, Consulté le 15/04/2020.

[5] https ://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-47/, Consulté le 15/04/2020.

[6] https ://mcfp.felk.cvut.cz/publicDatasets/CTU-Malware-Capture-Botnet-48/, Consulté le 15/04/2020.

[7] https ://info.arqendra.net /Files /Rsx+OSI+TCPIP-cours, Consulté le 20/04/2020.

[8] https ://tools.ietf.org/html/rfc1035, Consulté le 20/04/2020.

[9] https ://tools.ietf.org/html/rfc826, Consulté le 20/04/2020.

[10] https ://tools.ietf.org/html/rfc3376, Consulté le 20/04/2020.

[11] https ://tools.ietf.org/html/rfc2246, Consulté le 20/04/2020.

[12] https ://tools.ietf.org/html/rfc1459, Consulté le 20/04/2020.

[13] https ://tools.ietf.org/html/rfc2812, Consulté le 20/04/2020.

[14] https ://tools.ietf.org/html/rfc6176, Consulté le 20/04/2020.

[15] https ://tools.ietf.org/html/rfc1035, Consulté le 11/10/2020.

[16] https ://openclassrooms.com/fr/courses/4297211-evaluez-les-performances-dun-modele-de-machine-learning/4308256-evaluez-un-algorithme-de-classification-qui-retourne-des-valeurs-binaires, Consulté le 12/10/2020.

RÉFÉRENCES BIBLIOGRAPHIQUES

[17] Al-Fuqaha A, Guizani M et Mohammadi M. *Internet of things : A survey on enabling technologies, protocols, and applications.*

[18] Bentahar Atef Brakni Tahar. *Les protocoles de routage dans les réseaux pair-à-pair, Université de Larbi Tébessi.*

[19] Frédéric Raynal Cédric blancher Eric Detoisien. *Jouer avec le protocole ARP.*

[20] Yacine Challal. *Sécurité de l'Internet des Objets : vers une approche cognitive et systémique. Réseaux et télécommunications, Université de Technologie de Compiègne*. 2008.

[21] S. Debaille F. Gerin J. Hauet. *Internet des objets*. 2018.

[22] J. Nozick G. Pujolle O. Salvatori. *Les réseaux*. eyrolles, 2008.

[23] Purvi Prajapati Harsh Patel. *Study and Analysis of Decision Tree Based Classification Algorithms*. October 2018.

[24] Jean-Pierre Hauet. *Deux technologies clés : les réseaux de communication et les protocoles.*

[25] Rahul Khanna Mariette Awad. *Support Vector Machines for Classification*. January 2015.

[26] I. Nakmouche. *Modélisation de la propagation de l'attaque Eclipse dans un réseau eDonkey, Université Abderahmane Mira de Béjaia.*

[27] Mrs. Sunita Dhotre Pouria Kaviani. *Short Survey on Naive Bayes Algorithm*. 11 November 2017.

[28] Lokanatha C. Reddy Venkatadri.M. *A comparative study on decision tree classification algorithms in data mining.*

RÉSUMÉ :

La sécurité des objets connectés soulève cependant plusieurs problèmes qui peuvent constituer des obstacles sérieux au déploiement ou à l'acceptation de l'IoT. La principale cause réside dans la faiblesse des capacités de calcul des objets connectés, qui les empêche d'utiliser les techniques de sécurité classique mises en œuvre dans l'Internet. Dans ce mémoire, nous présentons les algorithmes de classification pour la détection des botnets puis on les applique sur plusieurs ensembles de données pour comparer et opter à l'algorithme le plus performant.

Mot clés : Botnet, python, P2P, IoT, apprentissage automatique.

ABSTRACT

However, the security of connected objects raises several issues that can constitute serious obs-tacles to the deployment or acceptance of IoT. The main cause lies in the weakness of the computingcapacities of connected objects, which prevents them from using classic security techniques imple-mented in the Internet. In this thesis, we present the classification algorithms for botnet detectionand we apply them to several datasets to compare and choose the best performing algorithm.

Keywords : Botnet, python, P2P, IoT, Machine learning.