République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Larbi Tébessi – Tébessa

Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie

Département : Mathématiques et Informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Option : Réseaux et sécurité d'information

Thème

Malware detection and classification for GCS of UAV

Présenté Par :

Kemache Amina

Devant le jury :

Mr. M. Amroune	*MCA*	*Univérsité Larbi Tébessa*	*Président*
Mme. S. Bourougaa Tria	*MCB*	*Univérsité Larbi Tébessa*	*Examinateur*
Mr. A. Gattal	*MCA*	*Univérsité Larbi Tébessa*	*Encadreur*
Mr. M.Gasri	*Doctorant*	*Univérsité Larbi Tébessa*	*Co - Encadreur*

Date de soutenance : 21/06/2021

Résumé

Récemment, l'utilisation de véhicules aériens sans pilote (UAV), qui est un aéronef télé piloté s'est développée. Les drones sont utilisés dans divers domaines tels que le sauvetage d'urgence, la prise de photos, la météo, ainsi que le domaine militaire et d'autres domaines…,

En raison de son importance et de son développement, il est devenu vulnérable à de nombreuses attaques de pirates informatiques, parmi les méthodes de pirate l'utilisant des fichiers et des programmes malveillants, ils sont concentrés sur les canaux de communication et les sensor, ainsi que la station de contrôle au sol (GCS).

Dans notre travail, nous avons concentré sur les attaques sur la station de contrôle au sol, nous avons utilisé plusieurs modèles pour la détection et la prédiction des fichiers malveillant**.**

Mot clé : station de contrôle au sol (GCS), aérons sans pilot (UAV), interface de programmation d'application (API), Apprentissage, L'extraction des caractéristiques, classification, Sélection de caractéristique

Abstract

Recently, the use of unmanned aerial vehicles (UAV), which is a remotely piloted aircraft, has developed. UAV are used in various fields such as emergency rescue, photo taking, weather, as well as military field. And other areas.

Due to its importance and development, it has become vulnerable to many hacker attacks, among the hacker methods using it files and malware; they are focused on communication channels and sensors, as well as the ground control station (GCS).

In our work we focused on attacks on the ground control station, we used several models for the detection and prediction of malicious files.

Keyword: Ground control station (GCS), unmanned aerial vehicle (UAV), application programming interface (API), learning, feature extraction, classification, feature selection.

ملخص

انتشر في الاونة الاخيرة استعمال الطائرات بدون طيار و هي طائرة توجه عن بعد. تستعمل الطائرات بدون طيار في مختلف المجالات مثل الإنقاذ في حالات الطوارئ، التقاط صور، معرفة أحوال الطقس و كذا المجال العسكري وغيرها من المجالات.

ونظرا لاهميتها و تطورها اصبحت عرضة للكثير من الهجمات من قبل المخترقين اللذين استطاعوا اختراقها باستعمال الملفات و البرامج الضارة من خلال قنوات الاتصال و المستشعرات وكذلك محطة التحكم الارضية.

ركزنا في عملنا على الهجمات على محطة التحكم الأرضية ، واستخدمنا عدة نماذج لاكتشاف الملفات الضارة والتنبؤ بها.

Dédicace

À mes parents pour leur amour, leur confiance Et leur soutien et la mise à disposition de conditions adaptées à mes études, leur aide et pour cette heureuse vie qui ont su parfaitement nous assurer. Qu'ils trouvent ici mon profond amour et ma profonde estime.

Remerciements

Nous remercions Allah de nous avoir donné la volonté et le courage qui nous ont permis de réaliser ce travail.

Je tiens à remercier en particulier ***Mr AB. GATTAL*** *ainsi que* ***M. Mohamed Elarabi Gasri*** *pour leur grande disponibilité durant ce travail, et pour leur encouragement constant envers moi face aux difficultés, et leur donnant suffisamment de temps sans s'ennuyer, et aussi pour leur gentillesse, leur moralité et leur hospitalité.*

Nous remercions aussi les membres de jury pour avoir accepté de juger notre travail.

Table des matières

Liste des tableaux

Liste des figures

Introduction Générale

" La théorie, c'est quand on sait tout et que rien ne fonctionne. La pratique,

C'est quand tout fonctionne et que personne ne sait pourquoi."

Albert EINSTEIN

Contexte et problématique

Les véhicules aériens sans pilote (UAVs), également appelés drones, sont un avion volant sans pilote à bord [65] En raison de la facilité de déploiement, de la configuration dynamique, de la mobilité élevée et de la réponse la plus rapide, il rend les drones vulnérables à plusieurs types d'attaques [6].
Ils existent plusieurs type d'attaque.

-Attaque sur (UAV) sensor [66].
-Attaque sur les canaux de communication [6].
-Attaque sur les (GCS)le poste de contrôle au sol [67]

L'ancienne méthode de détection des malwares et la méthode de signature (La signature digitale est un mécanisme cryptographique qui permet d'assurer la non répudiation de l'origine, Ce mécanisme repose sur un système cryptographique asymétrique, calculée en utilisant la clé privée de l'émetteur et vérifiée en utilisant la clé publique de l'émetteur [68]), mais les pirates ont trouvé des solutions pour éviter cette méthode. Parmi elle l'ajout ou le chiffrement de opcode ou l'API…etc. Dans ce cadre, on a proposé une méthode de crée un nouveau mécanisme, basée sur classification et prédiction.

2. Objectif

Notre objectif est de crée une méthode de classification et prédiction des fichiers malveillant en utilisant plusieurs modèles et nous allons choisir la meilleur parmi elle.

3. Organisation du manuscrit

La suite de ce manuscrit est structurée en trois (3) chapitres :

Chapitre 1. Etat de l'art (UAV, GCS et Les Malware)

Ce chapitre se divise en trois parties essentielles. La première partiefournit une vue d'ensemble sur les drones (UAV) en décrivant la définition l'historique, les classifications des drones, les type et les composantset les attaques réelles contre un UAV. La seconde partie sera concentrée sur la Station de contrôle au sol(GCS), les troisièmes parties Nous parlons de virus et de leur objectif

Chapitre 2. Etude du domaine

Ce chapitre sera consacré à la présentation les différents Modelés de classification utilisée et une description générale de notre processus.

Chapitre 3 : Etude expérimentale

Nous conclurons ce manuscrit par une conclusion générale qui discute les apports de notre travail.

Chapitre 1
Etat de l'art (UAV, GCS, Malware)

" La science sans religion est boiteuse,
La religion sans science est aveugle."
Albert EINSTEIN

Introduction

Ce chapitre se compose de trois sections principales. La première section est prévue pour fournir une vue d'ensemble sur la notation UAV (UnmannedAerialVehicle), nous décrivons la terminologie et les notations utilisées. Les stations de contrôle au sol (Ground Control System, GCS) seront présentées dans la deuxième section et Nous verrons également les malwares dans la troisième section.

1. Véhicules aériens sans pilote UAV

1.1. Definition de (UAV)

Les véhicules aériens sans pilote (UAV), également appelés drones, sont un avion volant sans pilote à bord. Ils sont contrôlés à distance soit par des systèmes informatiques seuls (avions autoprogrammés), soit par un opérateur (individuel ou groupe de personnes) via une communication sans fil et utilisés dans plusieurs domaines [1].

1.2. Histoires

La première chose que les scientifiques considéraient comme un drone était un ballon gonflable [2], en 1918, Etats-Unis se développe le projet Hewitt-Sperry Automatic-Airplane. Ils ont également travaillé sur le développement d'un système automatisé expérimental [3].

- En 1920, le succès des avions radiocommandés, en tentant de lancer des avions télécommandés à l'aide d'ondes de radiotélégraphie [3].
- Entre 1939 et 1945, l'Allemagne développe le bombardier « V-1 » pendant la seconde guerre mondiale [2].
- Les premiers drones apparaissent en France dans les années 1960 [3].
-Les découvertes et de nombreux développements se sont poursuivis jusqu'au début du XXIe siècle, Global Hawk est apparu dans les opérations militaires [2].
- Enfin, à l'instar de nombreuses technologies que nous utilisons aujourd'hui pour des applications civiles (Internet, GPS, etc.) et grâce aux progrès de la microélectronique, des technologies de routage sans fil et des systèmes optoélectroniques, les drones commencent à apparaître dans la vie civile.
- Ainsi, en 2006, la première société spécialisée dans la fabrication d'avions sans pilote a vu le jour ses activités dans ce secteur. La société est DJI et a son siège à Shenzhen (une ville chinoise similaire à SiliconValey).

Initialement, cette société proposait des drones capables de capturer des photos avec un appareil photo intégré. A titre d'exemple de modèle, nous pouvons citer le Phantom 2 [4].

1.3 Classification d'UAV

Après d'avoir étudié la plupart des classifications des drones selon les études existantes et après les avoir analysées, nous avons constaté qu'avec le nombre croissant de drones développés et transportés ces dernières années, il y a un problème avec la classification de ce nouveau drone. Étant donné que les UAV sont utilisés dans une variété d'applications, il est difficile de développer un système de classification qui inclut tous les UAV. Dans [5], la classification des drones dépend de plusieurs facteurs, dont la mission et les objectifs pour lesquels les drones ont été utilisés, les chercheurs classent les drones en fonction de la hauteur à laquelle les drones volent, ils classent donc les drones en deux parties Figure 1.1.

1.3.1 Plates-formes à haute altitude (High altitude Platform, HAP) :

Ce sont des drones qui volent au-dessus de 17 km et sont généralement semi-tropicales, plus durables et sont conçus pour des opérations à longue portée, sont généralement préférés pour fournir une large couverture sans fil à de vastes zones géographiques [6].

1.3.2 Plate-forme basse altitude (Low Altitude Platform, LAP) :

C'est le drone qui peut voler à des altitudes de dizaines de mètres jusqu'à quelques kilomètres, et peut se déplacer à une vitesse qui est flexible, voler librement sans aucun permis est de 400 pieds [6], De plus déployés plus rapidement et ainsi les rendre plus adaptés aux applications urgentes (telles que les urgences), et utilisés pour collecter des données à partir de capteurs au sol [6].

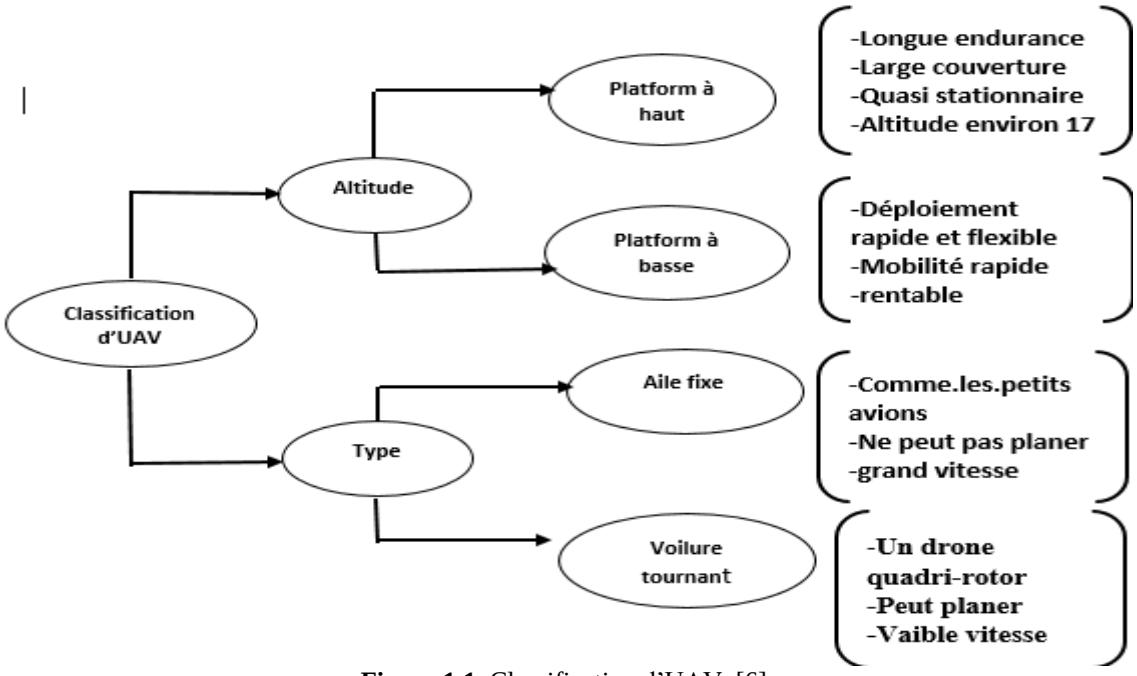

Figure 1.1. Classification d'UAV. [6]

Les chercheurs se sont également concentrés sur le type d'aile (méthode de vol) pour la classification des drones.

1.3.3 UAV à voilure fixe

Les drones à voilure fixe comme les petits avions ont de petits poids et une vitesse plus élevée, et doivent avancer pour rester haut. Voir la Figure 1.2

Figure 1.2. UAV à voilure fixe [22]

1.3.4 UAV à voilure tournante

Les drones à ailes tournantes planent et restent fixes sur une certaine zone. Voire la figure 1.3.

Figure 1.3. UAV à voilure tournante [23]

1.3.5 Poids d'UAV

Le tableau 1.1 montre la classification en fonction du poids :

	Type				
	Micro Poids<10g	Plus petit 100g<Poids<2kg	Petit 2kg<Poids<25kg	Moyen 25kg<Poids<150kg	Large Poids>150kg
Poids	16g	750g	3,3kg	90kg	2223kg

TABLEAU 1.1. Classification d'UAV en fonction du poids [6]

1.4 Types et caractéristiques

1.4.1 Poids de la charge (charge utile)

Le poids de la charge ou la charge utile signifie la capacité de l'UAV à transporter des objets ou la capacité de l'UAV à soulever des charges allant de dizaines de grammes à des centaines de kilogrammes [7], lorsque la capacité de levage augmente, la charge utile et la taille des drones augmente, les charges utiles sont des caméras vidéo et tous types de capteurs [6].

1.4.2. Mécanisme de vol

Il existe plusieurs modes de vol et selon les mécanismes de vol. Les drones peuvent être classés en :

1.4.2.1 UAV multi-rotor

Les drones multi-rotors permettent le décollage et l'atterrissage vertical, ils peuvent voler de manière stable sur des emplacements fixes. Le drone multi-rotor a une mobilité limitée et une grande consommation d'énergie car il doit lutter contre la gravité tout le temps [6].

1.4.2.2 UAV à voilure fixe

Les drones à voilure fixe permettent :

- Le décollage et l'atterrissage vertical a ne sont pas possibles pour les drones à voilure fixe et par conséquent nécessitent une piste pour le décollage et l'atterrissage. [6]
- Impossible de survoler un emplacement fixe. Les drones fixes sont également plus chers que drones multi-rotor

1.4.2.3 UAV hybride à voilure fixe/tournante

Décoller verticalement et un chariot et l'atteindre à sa destination en glissant à travers l'air, puis passage au défilement avec quatre gammes de rotors et hauteur [6].

1.4.3. Portée et hauteur

La distance de portée varie de plusieurs dizaines de mètres pour les petits drones à des centaines de Kilomètres pour les gros drones [6].

1.4.4. Vitesse et temps de vol

Les petits drones ont généralement une vitesse inférieure à 15 m / s, tandis que les gros drones atteignent une vitesse de 100 m / s. La comparaison entre la vitesse de l'UAV et l'agilité de rotation a été étudiée dans [6].

1.4.5. Courant électrique

Le courant électrique est la principale source d'énergie des drones et il est considéré comme le plus grand facteur déterminant l'endurance des drones. Les drones utilisent de nombreuses sources d'énergie comme les batteries rechargeables, le carburant [8] et l'énergie solaire est également une technologie prometteuse [6].

1.5 Composants d'UAV

Les drones sont des systèmes techniques très exposés, pour analyser les vulnérabilités des drones, il est important de comprendre les composants des drones et le fonctionnement de chaque composant, nous décrivons donc les drones en termes de modèles de composants. Figure 1.4 représenté ensemble de modules de base de drone [6] [36].

- **Le châssis:** ils peuvent être en X ou en H

Figure 1.4. Châssis en X

Figure 1.4. Châssis en H

- **Le moteur :**

Figure 1.5

- **Les contrôleurs :**

Figure 1.6. Contrôleur ESC (Electronic Speed Control)

- **Les hélices :**

Figure 1.7

- **La batterie :**

Figure 1.8

- **La radiocommande :**

Figure 1.9

- **Le contrôleur de vol :**

Figure 1.10

- **La caméra :**

Figure 1.11

Un drone incorpore un ensemble de modules de base qui peuvent être regroupés en deux parties, la partie drone et le poste de commande au sol (GCS) partie.

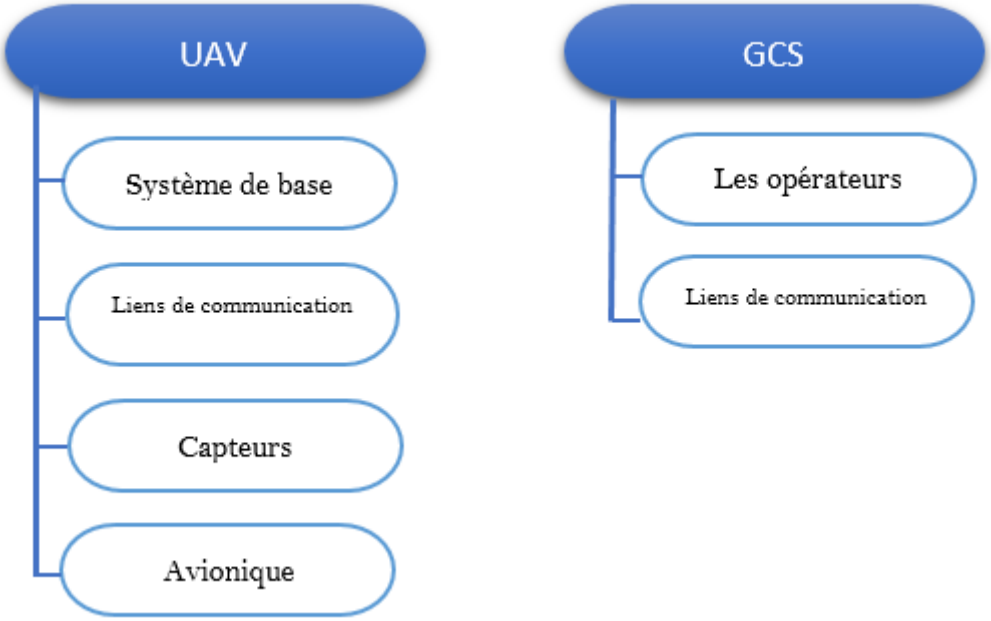

Figure 1.12. Droite : modèle de composant général d'un UAV. Gauche : modèle de composant simple d'une station au sol GCS [6]

La Figure 1.12 montre également un modèle général d'UAV, sans outils secondaires ni armes d'aviation [25]. Le modèle décrit dans les composants de base qui devraient être dans UAV. Le système de base des drones est la base des drones qui connectent des composants pour faire voler les drones en toute sécurité.

Dans la partie UAV, le module du système de base constitue la base et le système d'exploitation de l'UAV. Il relie différents modules entre eux en prenant en charge les communications inter modules (voir Figure 1.13). Le module de capteur se compose de divers capteurs ainsi que des fonctionnalités de prétraitement nécessaires. Les capteurs couramment équipés sont les capteurs de pression, les capteurs d'attitude et les accéléromètres, qui sont essentiels pour voler en toute sécurité à une vitesse et à un niveau constant. [6]

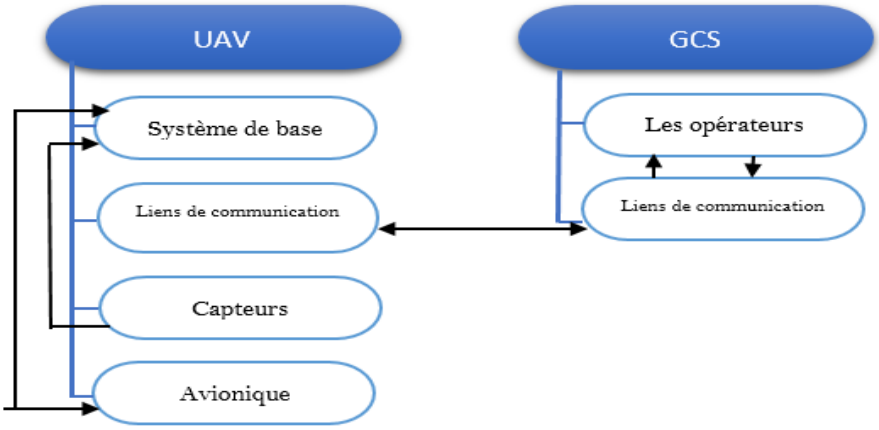

Figure 1.13. Modèle de composant d'UAV étendu [6]

Le système de positionnement global (Global Positioning System, GPS) a pour mission de fournir l'emplacement et les coordonnées de vitesse des contrôleurs au sol pour localiser la zone de l'UAV. Le responsable du transfert des ordres de contrôle reçus aux ordres de moteur, de plaque, de gouvernail, de fixations et de spoilers est le système UAV [6]. Les drones doivent communiquer avec GCS via des canaux de communication qui sont des canaux sans fil. Il communique avec GCS pour recevoir les commandes de base et envoie les données collectées à GCS. Le flux d'informations entre les composants de l'UAV varie selon le type d'UAV [6]. Il existe de nombreux flux d'informations entre les drones et leur environnement, comme le montre la figure8, les deux liens opérationnels les plus importants sont :

- Flux d'informations bidirectionnel entre le système de télécommunication et la station de contrôle au sol (GCS).
- L'information circule de l'environnement vers les capteurs.

GCS contrôle et coordonne le comportement des drones [6], traite les données reçues des drones et les renvoie à nouveau. Les fonctions de communication sont fournies à l'aide d'une norme sans fil, choisie parmi 3G, 4G, 5G, WiFi, WiFi Direct, Bluetooth et WiMAX, via l'unité de communication UAV. Figure 1.14.

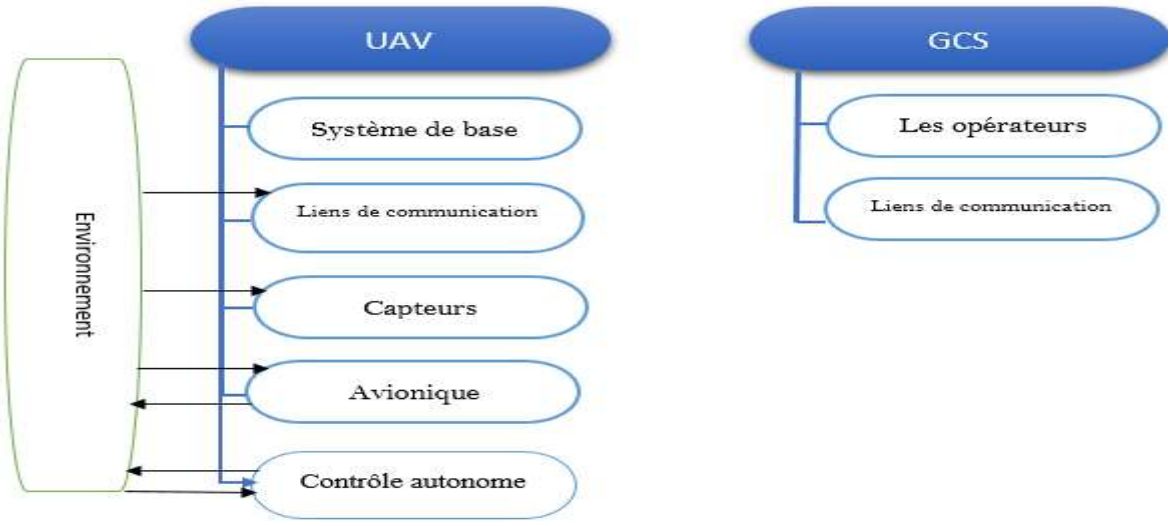

Figure 1.14. Modèle de composant d'UAV étendu avec flux d'informations [6].

2. Station de contrôle au sol(GCS)

2.1 Définition de GCS

Le poste de contrôle au sol (GCS) est la partie centrale de l'UAV, et le GCS conjoint est une intégration du système de commande et de contrôle, et de suivi. GCS peut être installé dans un camion qui transporte tout l'équipement au sol pour communiquer avec son drone. GCS peut être réduit à un appareil portable [10]

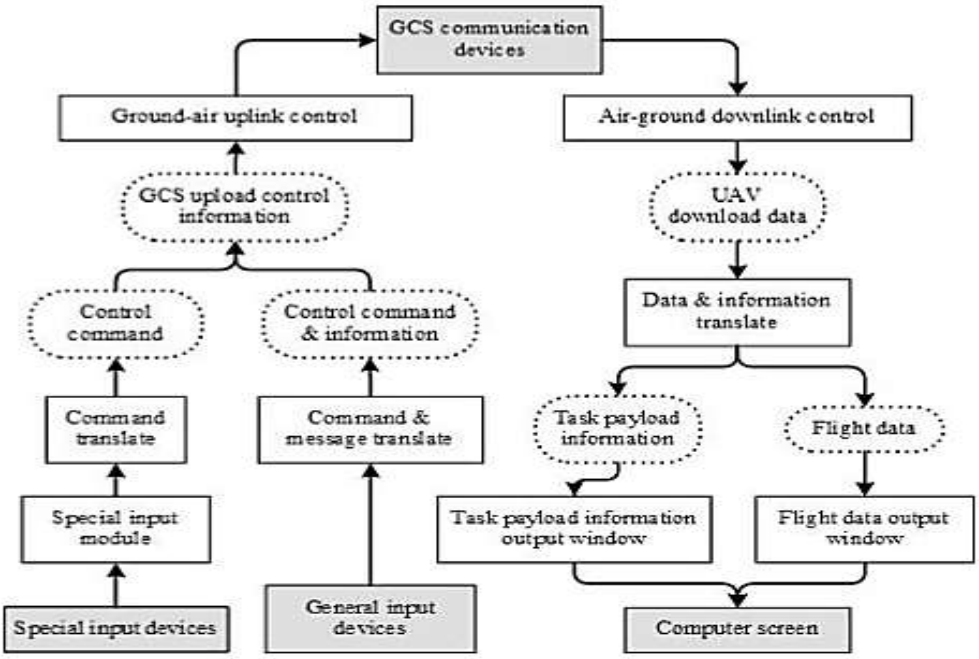

Figure 1.15. Structure typique du logiciel GCS [12]

Le GCS comprend un PC de laboratoire et une radio moderne pour communiquer avec le système embarqué. Il peut recevoir des informations de vol d'UAV en temps réel, telles que la position, l'altitude, la vitesse, le cap, la trajectoire de vol, l'état de santé des capteurs et les commandes de liaison ascendante. D'autre part, il peut envoyer les commandes de contrôle au processeur embarqué si nécessaire [11].

2.2 Système de poste de contrôle au sol

2.2.1 Station Hardware « Matériel »

Ce désigne principalement l'ordinateur GCS et ses périphériques secondaires.

2.2.1.1. Liaison de données air-sol

La communication air-sol entre le drone et le système de contrôle de mesure échange un grand volume d'informations multi-espèces. Il nécessite une vitesse de transmission élevée et une fiabilité élevée. Cependant, limité par la faible capacité du matériel du système, le GCS de type miniature ne peut fournir qu'une seule liaison de données sans sauvegarde pour les connexions de données. Par conséquent, le logiciel GCS doit être en mesure d'effectuer une communication multi-type, à grand volume et bidirectionnelle, tout en utilisant le même support de communication. La nouvelle génération de format d'échange de données, par exemple, XML (Extensible Markup Language) aidera à résoudre ce problème [13]. Afin d'améliorer la fiabilité de l'interconnexion, le

logiciel GCS doit également être en mesure de détecter les erreurs et les défauts de liaison de données, d'alerter correctement et de corriger automatiquement les défauts.

2.2.1.2 Dispositif d'affichage

Il comparé au GCS de grand type, en raison de la petite taille et du petit nombre de projecteurs, il y a un manque apparent d'espace d'affichage d'ordinateur disponible dans le GCS de type mini, ce qui conduit à de grandes difficultés dans le graphique interface utilisateur (utilisateur graphique) Conception d'interface. Par conséquent, la planification globale du processus de conception de l'interface graphique est très essentielle. Selon différents niveaux de données, l'importance de l'information, l'anxiété et les exigences en temps réel, appliquer une méthode différente et appropriée de sortie à différentes données [14], afin d'améliorer l'efficacité de l'interaction des informations.

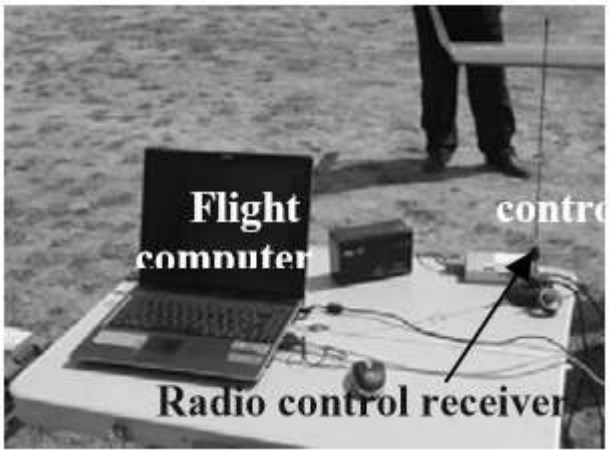

Figure 1.16.b. Matériel de support au sol

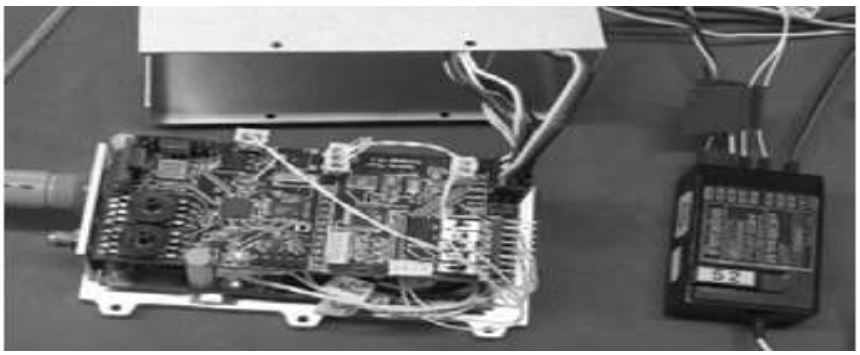

Figure 1.16.a. Système embarqué

Figure 1.16. Système embarqué et matériel de support au sol [11]

2.2.2 Station Software

Le développement de la station de contrôle au sol est principalement basé sur Microsoft Visual C ++ 6.0 qui est utile pour afficher toutes les données le plus rapidement possible. Le protocole de communication entre UAV et GCS doit être connu avant la transmission des données par quête. La trame est dirigée par les chaînes « @@@ », puis 35 groupes de données circulent, comme la position, l'altitude et l'attitude. Toutes les données seront enregistrées dans un fichier texte contenant les mauvaises données. Après avoir vérifié les données restantes, elles seront toutes converties en différents ensembles d'informations et affichées sur le tableau de bord à l'écran. S'il y a quelque chose qui ne va pas dans les données, le système donnera immédiatement un avertissement avec une lumière rouge et un bip sonore pour nous en informer. La figure 1.17 affiche le panneau du poste de commande au sol [11].

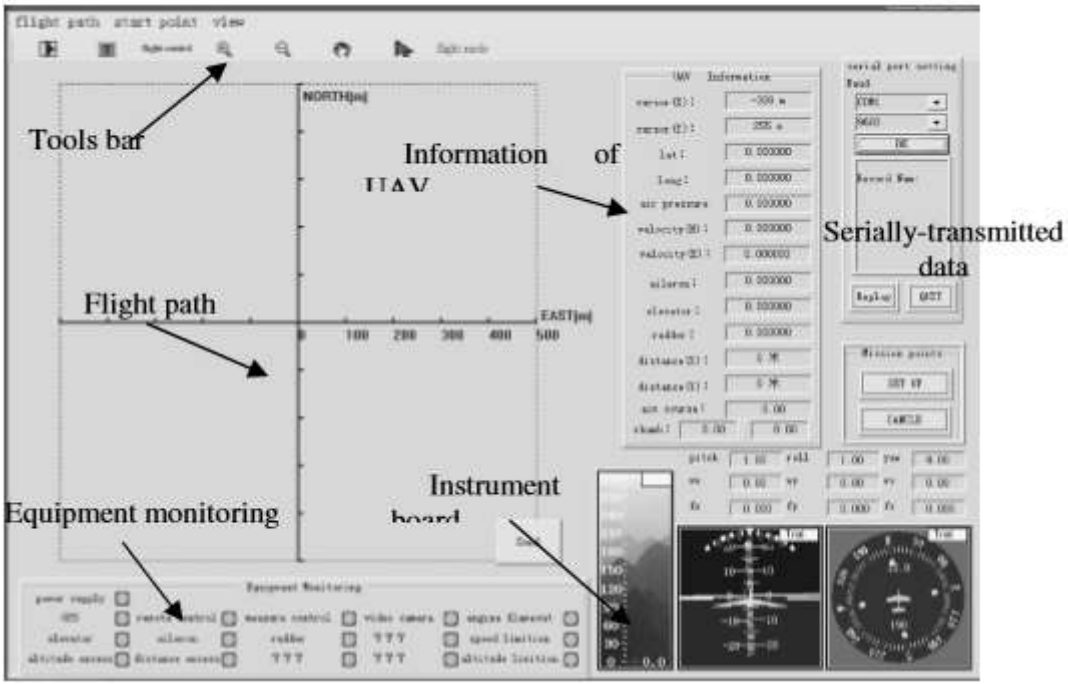

Figure 1.17. Panneau du poste de commande au sol [11]

2.3 Intérêt et motif de l'attaque

Le but de l'attaque du poste de contrôle au sol de l'appareil est de :

- Perturber le fonctionnement de l'appareil pour empêcher le contrôle de l'UAV.
- Contrôle de gain de la station de contrôle au sol de l'appareil intelligent pour contrôler le drone.
- Accéder aux données utiles à l'attaquant.

Une attaque réussie nécessite une violation d'au moins une des informations de sécurité.

Parmi ses objectifs : (confidentialité, intégrité ou disponibilité).

- Une perte de confidentialité est la divulgation non autorisée de données transmises et stockées (écoute clandestine d'un réseau de communication) [15].
- Une perte d'intégrité est la modification intentionnelle ou non intentionnelle des données transmises et stockées (logiciels malveillants) [16].

• Une perte de disponibilité est la perte de la capacité d'accéder aux ressources en utilisant des appareils mobiles chaque fois que nécessaire [17] (appareil de brouillage).

3. Logiciels malveillants

3.1. Définition de logiciels malveillants

Malware, abréviation de logiciel malveillant, est un terme générique désignant les virus, vers (Worm), chevaux de Troie (Trojan) et autres programmes informatiques nuisibles que les pirates utilisent pour détruire et accéder à des informations sensibles. Comme le dit Microsoft, « [malware] est un terme fourre-tout pour désigner tout logiciel conçu pour endommager un seul ordinateur, serveur ou réseau informatique ». En d'autres termes, un logiciel est identifié comme un logiciel malveillant en fonction de son utilisation prévue [9].

Cela signifie que la question de, par exemple, quelle est la différence entre un malware et un virus ce qui manque un peu le point : un virus est un type de malware, donc tous les virus sont des logiciels malveillants (mais tous les logiciels malveillants ne sont pas des virus) [18].

3.2. Types de malwares

Il existe plusieurs façons de classer les logiciels malveillants. Parmi celles-ci figurent la façon dont les logiciels malveillants se propagent et les logiciels malveillants peuvent infecter les ordinateurs cibles de différentes manières : [18]

• **Worm (Un ver**) est un logiciel malveillant autonome qui se reproduit et se propage d'un ordinateur à l'autre (des programmes qui se répliquent automatiquement sur un réseau) [18] [21].

• **Un virus** est un morceau de code informatique qui s'insère dans le code d'un autre programme autonome, puis oblige ce programme à entreprendre une action malveillante et à se propager [18].

• **Trojan** (Un cheval de Troie) se fait passer pour des programmes utiles, mais contiennent du code malveillant pour attaquer le système ou fuir des données [18].

• **Backdoor** (Les portes dérobées) ouvrent le système aux entités externes en subvertissant les politiques de sécurité locales pour permettre l'accès et le contrôle à distance sur un réseau [21].

Figure 1.18. Type de malware [20]

3.3. Techniques d'analyse des logiciels malveillants

L'analyse des malwares est une étape vers la détection. Premièrement, il est nécessaire d'analyser comment les malwares remplissent leur fonction et le but pour lequel ils sont développés afin que cette compréhension des malwares facilite la mise en œuvre de la fonction défensive par les développeurs d'appareils de détection. Ces technologies se divisent en trois catégories: statique, dynamique ou hybride [24] [28].

Une approche statique tente de détecter les logiciels malveillants avant que le programme inspecté ne s'exécute. À l'inverse, une approche dynamique tente de détecter les comportements malveillants lors de l'exécution du programme ou après l'exécution du programme [43].et les techniques hybrides qui combinent les deux approches [25].

L'approche ou l'analyse spécifique d'une technique déterminée par la manière dont la technique rassemble les informations pour détecter les logiciels malveillants [43].

	Avantage	**Inconvénient**
Analyse statique	Rapide et sur Faible taux de faux positifs Malware multi-chemins	Difficulté à analyser Malware inconnu
Analyse dynamique	Bon à détecter Malware inconnu	Ni rapide ni sur Difficulté à analyser Malware multi-chemins

TABLEAU 1.2. Résumé des avantages et des inconvénients de l'analyse statique et dynamique [24]

3.4 Techniques de détection des logiciels malveillants

Il y a eu une augmentation rapide du nombre d'études universitaires sur la détection des logiciels malveillants. Dans un premier temps, la méthode de détection basée sur la signature était largement utilisée. Cette méthode fonctionne rapidement et efficacement contre les malwares connus, mais ne fonctionne pas bien contre les malwares zero-day, [11]. Au fil du temps, les chercheurs ont commencé à utiliser des techniques telles que la détection basée sur la vérification du comportement, l'heuristique et des modèles; et de nouvelles techniques telles que la détection basée sur l'apprentissage en profondeur, le cloud, les appareils mobiles et l'IoT. Figure 1.19.

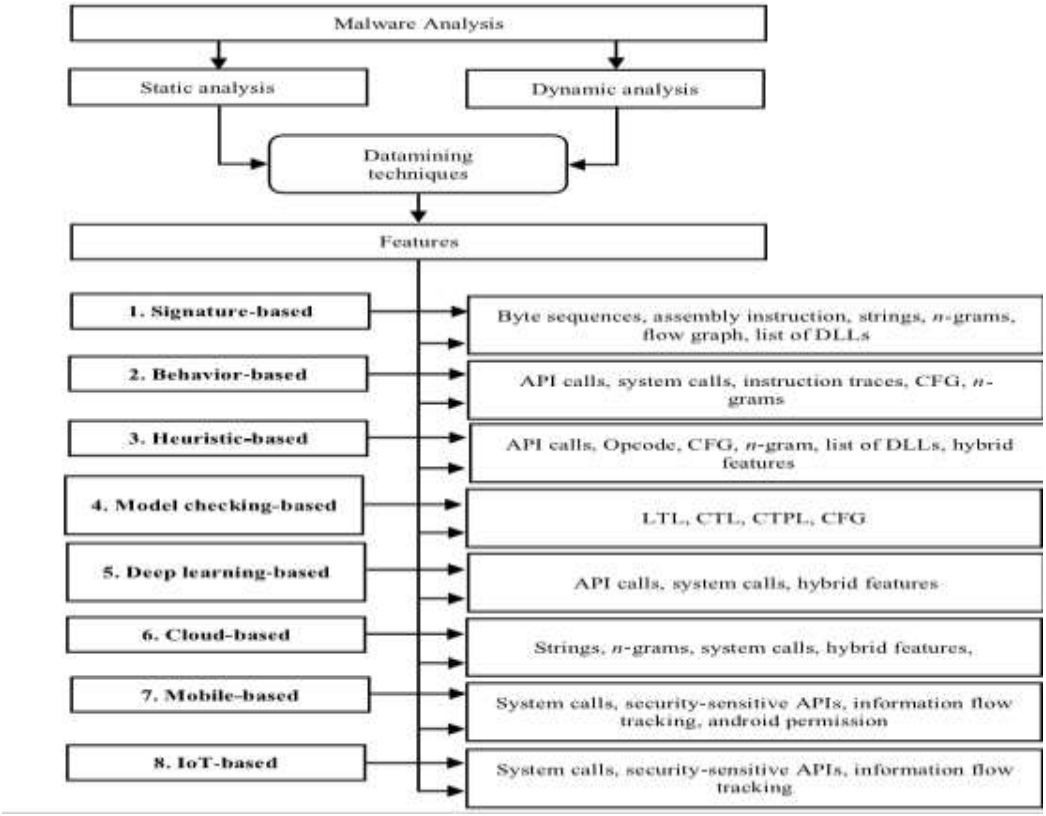

Figure 1.19. Méthodes de détection des logiciels malveillants et leurs fonctionnalités [29].

3.4.1 Détection basée sur les signatures (Signature-based detection)

Basé sur la signature (ou parfois appelé détection d'abus), la base de données de la technique d'intrusion connue (signature d'attaque) sera maintenue et détectera l'intrusion en comparant le comportement à la base de données. Il faudra moins de ressources système pour détecter les intrusions. Cependant, l'inconvénient de cette technique est inefficace contre les attaques inédites et, par conséquent, elle ne peut pas détecter de nouvelles méthodes d'intrusion inconnues car aucune signature n'est disponible pour de telles attaques. [26] [27]. Et Pour protéger les utilisateurs contre les menaces de logiciels malveillants, les produits logiciels des sociétés anti-malware (par exemple, les produits de Comodo, Kaspersky, Kingsoft, MacAfee et Symamtec) constituent la principale défense [30].

3.4.1.1 Processus de génération de signature

Lors de la génération de la signature, les premières fonctionnalités sont extraites des exécutables (Figure 1.20). Le moteur de génération de signatures crée des signatures et les stocke dans la base de données de signatures. L'échantillon de signature pertinent est extrait de la même manière avant et comparé aux signatures de la base de données. Sur la base de la comparaison, l'exemple de logiciel est distingué comme malveillant ou inoffensif. Il existe de nombreuses techniques différentes pour créer une signature, telles que le balayage de la chaîne, le balayage du haut et de la queue, le balayage du point d'entrée et le contrôle de sécurité [29].

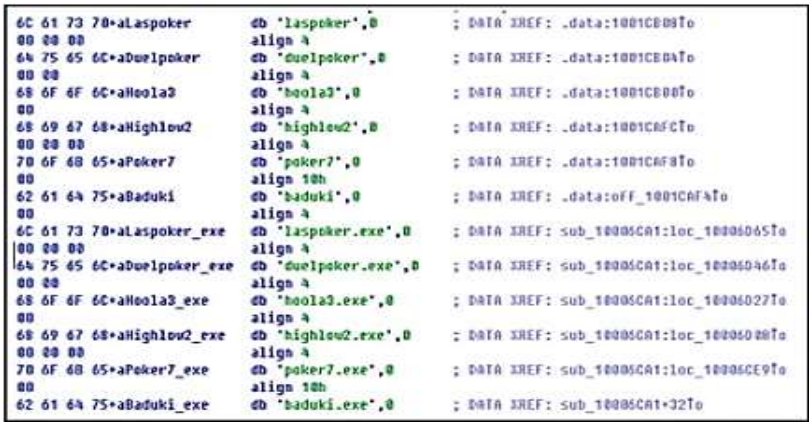

Figure 1.20. Un exemple de signature utilisée pour la détection d'un cheval de Troie de jeu en ligne [30]

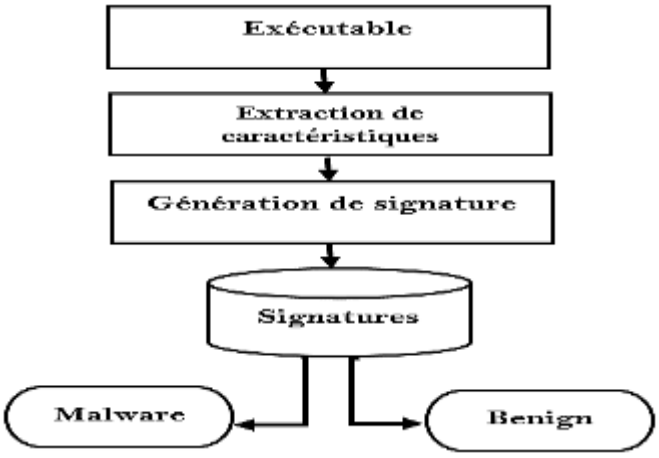

Figure 1.21. Schéma de détection des malwares basé sur les signatures. [29]

3.4.1.2 travaux connexes pour la détection basée sur la signature

Papier	Représentations des caractéristiques	Objectif / Sucées	Année
Schultz at al. [29]	Chaines imprimables appels systèmes DLL et API	Détection de nouveaux logiciels malveillants	2001
Karnik et al.[29]	Instruction d'assemblage similitude cosinus	Identifier différentes formes des malwares	2007
Cha et al .[29]	Transformation vectorielle des hachages de fichiers	Capturer plusieurs logiciels malveillants avec une seul signature	2011
Baldangombo et al.[29]	Information sue l'entête de fichiers noms de DLL et API	Détection des malwares connus avec un taux de réussite de 99%	2013
Aashima and Bajaj.[29]	Technique d'extraction de texte basée sur hybride pour extraction l'instruction	Peut prédire les types de malwares avec une faible surcharge	2016
F.Zolkipli et Jantan[29][31]	Technique basée sur signature, un algorithme génétique (GA) et un générateur de signature	Détection de nouveaux logiciels malveillants	2010

TABLEAU 1.3. Approche basée sur la signature. [29]

3.4.2 Détection des malwares basée sur le comportement (Behavior based malware detection)

Les techniques de détection basées sur le comportement se concentrent sur l'analyse du comportement du code malveillant connu et suspecté [21], avec des outils de surveillance et déterminent si le programme est un programme malveillant ou bénin, et quels que soient les changements de code du programme, le comportement sera similaire et ainsi de nouveaux logiciels malveillants peuvent être détectés dans Par ici. Et les logiciels malveillants peuvent être marqués comme bénins dans un environnement virtuel [29] [32].

3.4.2.1 Processus de détection de comportement

Lors de la mise en place d'un système de détection basé sur le comportement, les comportements sont obtenus en utilisant l'une des procédures suivantes :

Analyse automatique à l'aide du bac à sable

- Surveillance des appels système [34].
- Suivi des changements de fichier.
- Comparaison des instantanés de registre.
- Surveillance des activités du réseau [29].
- Surveillance de processus.

Dans la détection basée sur le comportement, tout d'abord, les comportements sont déterminés en utilisant l'une des techniques utilisées ci-dessus et le jeu de données est créé en soustrayant les caractéristiques à l'aide de l'exploration de données. Ensuite, des caractéristiques spécifiques de l'ensemble de données sont obtenues et la classification est effectuée à l'aide d'algorithmes ML. [29]

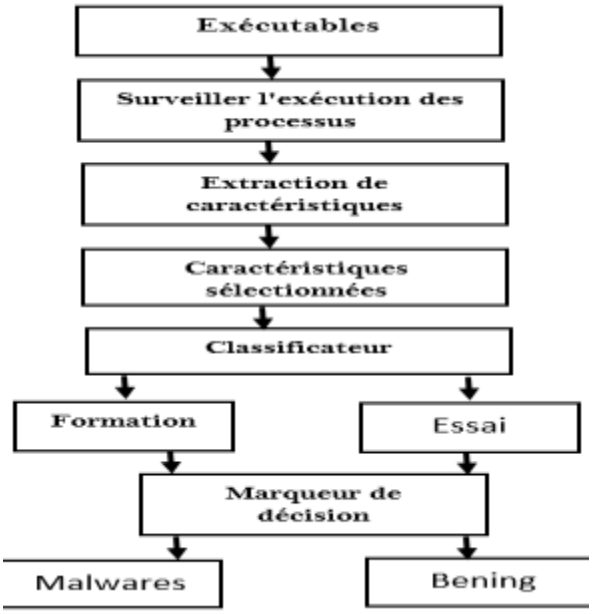

Figure 1.22. Schéma de détection des logiciels malveillants basé sur le comportement. [29]

3.4.2.2 Travaux connexes pour la détection basée sur le comportement

Papier	Représentations des caractéristiques	Objectif / Sucées	Année
Moser et al.[29]	Instruction de branchement conditionnelle	Identifier plusieurs chemins d'exécution	2007
Wagener et al.[29]	Appels système, arbre phylogénétique à distance hellinger	Identifier les nouveaux logiciels malveillants et les différentes formes de logiciels malveillants	2008
Park et al[19][29]	Création de diagrammes d'appels système	Identifier différentes formes de malwares	2013
Das et al.[29][37]	Fréquences d'appel système, n-gramme	Identifier les nouveaux logiciels malveillants et les différentes formes de logiciels malveillants	2016
Monire et al[29]	Matrice non conforme du fichier XML de l'historique exécutif	Identifier les logiciels malveillants traditionnels et nouveaux	2016

TABLEAU 1.4. Approche basée sur le comportement. [29]

3.4.3 Détection de malwares basée sur l'heuristique (Heuristicbased malware detection)

Une approche de détection basée sur l'heuristique a été utilisée pour différencier le comportement normal et anormal d'un système afin qu'en fin de compte, les attaques de logiciels malveillants connues et inconnues puissent être identifiées et résolues [28] [29]. Le processus de détection heuristique comprend deux étapes. Dans un premier temps, le comportement du système est observé et un enregistrement est conservé des informations qui peuvent être vérifiées en cas d'attaque. Cette différence est comparée à la deuxième étape de détection des logiciels malveillants pour une famille spécifique. Le détecteur de comportement utilisé dans cette technique consiste en :

- **Collecte de données** : ce composant traite de la collecte de données, qu'elles soient statiques ou dynamiques
- **Interprétation** : Interprétation des données collectées à partir du composant de collecte de données et conversion en une forme intermédiaire.
- **Algorithme de correspondance** : fait correspondre la signature du comportement avec les informations converties dans le composant d'interprétation. Le détecteur de comportement est illustré à la figure qui explique la fonction de la façon dont tous ces composants fonctionnent ensemble.

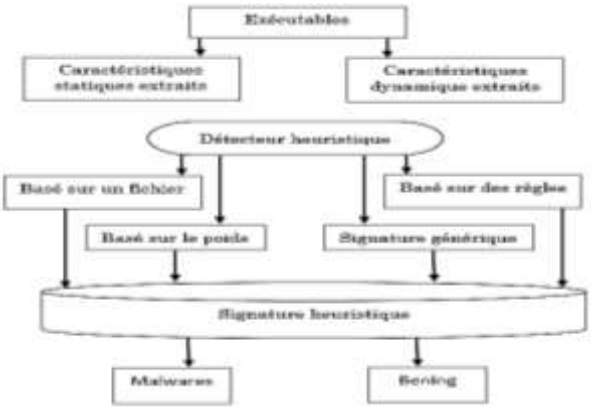

Figure 1.23. Schéma de détection des malwares basé sur l'heuristique [29]

3.4.3.1 travaux connexes pour la détection basée sur l'heuristique

Papier	Représentations des caractéristiques	Objectif / Sucées	Année
Zhang et al [29]	Les séquences d'octets de n grammes	Identifier les formes traditionnelles et différentes de logiciels malveillants	2007
Griffin et al[38][29]	Séquenced'octets	Identifier différentes formes de malwares	2009
Anderson et al.[29]	La chaîne et les diagrammes de markov n-gramme	Identifier différentes formes de malwares avec 94,41%	2011
Islam et al[39]	Fréquences de la méthode API des chaînes imprimables	Identifier les formes nouvelles et différentes de malwares avec 97%	2013
Naval et al [35]	Schéma des appels système et des relations	Détecter les attaques par insertion de code	2015

TABLEAU 1.5. Approche basée sur l'heuristique. [29]

3.4.4 Détection des logiciels malveillants basée sur la vérification des modèles

Les comportements malveillants sont extraits manuellement et les blocs de comportement sont codés à l'aide d'un système de temps linéaire (LTL) pour afficher une fonctionnalité spécifique. Les comportements de programme sont créés en examinant la relation de flux d'un ou plusieurs appels système et des comportements spécifiques à l'aide de propriétés telles que le masquage, la diffusion et l'injection. En les comparant, il est déterminé si le programme est nocif ou inoffensif. Mais il ne peut pas détecter toute la nouvelle génération de malwares [29].

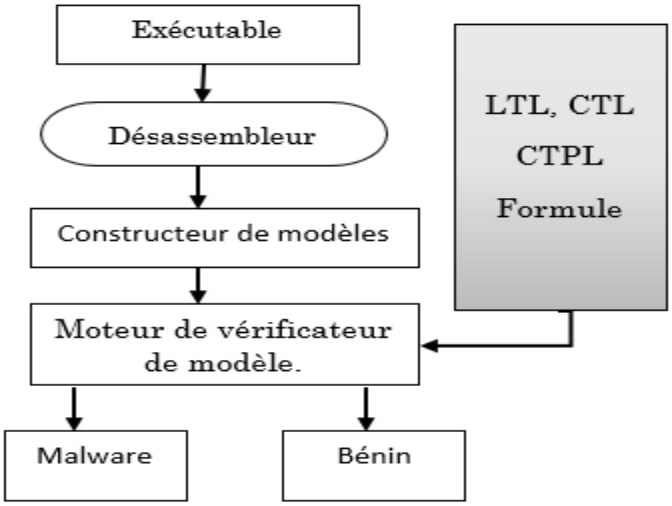

Figure 1.24. Schéma de détection des logiciels malveillants basé sur la vérification du modèle [29]

3.4.4.1 Travaux connexes pour la détection basée sur la vérification des modèles

Papier	Représentations des caractéristiques	Objectif / Sucées	Année
Singh and Lakhotia [40]	Formules LTL (logiquelinéaire)	identifier les formes nouvelles et différentes de malwares	2003
Kinder et al [29]	Logique de validation d'arbre de calcul	identifier différentes formes de malwares	2005
Beaucams and Marion.[41]	Formules LTL	identifier différentes formes de malwares	2009
Song et al[29]	Déterminer le comportement de la pile du programme à l'aide de systèmes de refoulement	identifier les nouveaux logiciels malveillants	2012
Cimitile et al[29]	Logique Mu-Calculus électif et arbres phylogénétiques	identifier différentes formes de malwares	2017
Holzer et al [29]	Technologie qui expliqué la chaîne d'outils de détection des logiciels malveillants qui intègre le processus de développement des spécifications et permet une future analyse automatisée des logiciels malveillants avec extraction des spécifications.	spécifier et détecter les logiciels malveillants	2007

TABLEAU 1.6. Approche basée sur la vérification des modèles [29]

3.4.5 Détection des logiciels malveillants basée sur l'apprentissage en profondeur (Deep Learning)

L'apprentissage en profondeur est un sous-domaine du ML hérité des réseaux de neurones industriels (ANN). Utilisé pour le traitement d'image, les voitures sans conducteur et la commande vocale; Il n'est pas utilisé efficacement pour détecter les logiciels malveillants. Et ne combattez pas les attaques en esquivant. [29]

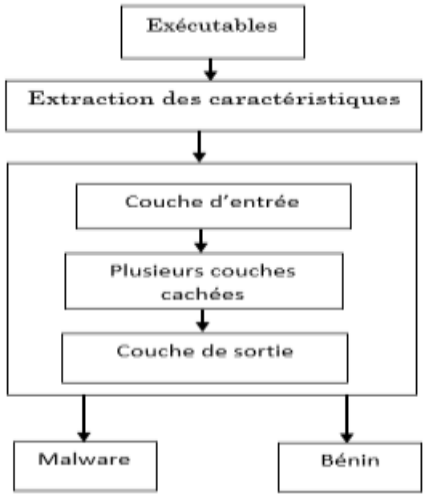

Figure 1.25. Schéma de détection des malwares basé sur l'apprentissage en profondeur. [29]

3.4.5.1 Travaux connexes pour la détection basée sur l'apprentissage en profondeur

Papier	Représentations des caractéristiques	Objectif / Sucées	Année
Saxe and Berlin	Fonctionnalités d'octet contextuelles, fonctionnalités d'importation PE, modèle d'étalonnage de score	95% DR avec 0,1% FP	2015
Huang and W.Stockes	Octets bruts, attaque basée sur le gradient	Montrez l'inefficacité de l'apprentissage profond	2016
Dali et al	Fonctionnalités hybrides, technique DeepFlow, modèle DBN	Score F1 de détection élevé de 95,05%, qui a surpassé l'apprentissage traditionnel basé sur le ML	2017
Yanfang et al	Opérations de formation gourmandes par couches	Amélioration des performances globales par rapport aux techniques d'apprentissage traditionnelles	2018

TABLEAU 1.7. Approche basée sur l'apprentissage en profondeur [29]

3.4.6 Détection de malware basée sur le cloud (Cloud based)

Utilisé pour détecter les logiciels malveillants. Pour les ordinateurs et les appareils mobiles dotés de bases de données de logiciels malveillants beaucoup plus volumineuses et de ressources de calcul étendues. La découverte basée sur le cloud utilise différents types d'agents de découverte sur les serveurs cloud et fournit la sécurité en tant que service. Elle fournit également une protection à l'utilisateur s'il télécharge un type de fichier [29].

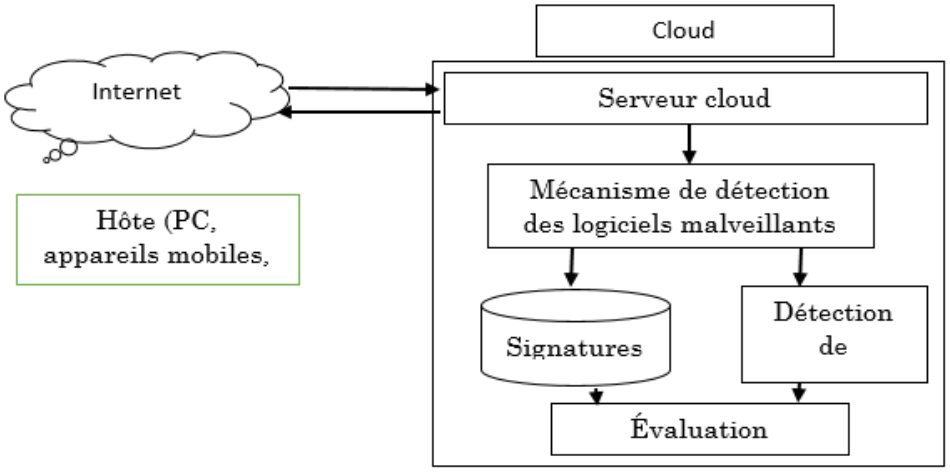

Figure 1.26. Schéma de détection des malwares basé sur le cloud. [29]

3.4.6.1 Travaux connexes pour la détection basée sur le cloud

Papier	Représentations des caractéristiques	Objectif / Sucées	Année
Martignoni et al [29][42]	Traces d'API, appels système, fonctionnalités dynamiques	Pour obtenir plusieurs traces d'exécution du même malware	2009
Hao et al [43]	Fonctions de hachage multiples, structure d'esquisse réversible, méthode de filtrage croisé des seaux	Surpassez les autres existants avec moins de temps et de communication	2015
Xiao et al [44]	Traces d'application, architecture Dyna, stratégie Qlearning	pour augmenter la précision de détection, réduisez le délai de détection	2017
Yanfang et al [29]	Schéma basé sur le cloud qui combine le contenu des fichiers et les relations entre les fichiers	améliorer les résultats de détection des logiciels malveillants	2011

TABLEAU 1.8. Approche basée sur le cloud. [29]

Conclusion :

Dans ce chapitre, nous concentrons sur les drones, leur histoire, les composants et caractéristiques les plus importants, ainsi que le GCS et le système de contrôle dans la deuxième partie. Et dans la troisième partie Nous avons présenté: Un aperçu des fichiers malveillants, de leurs types et de leur mode de fonctionnement et les techniques d'analyses et de détections. Dans le chapitre suivant, nous présenterons notre system pour créer une classification à partir d'un modèle d'apprentissage.

Chapitre 2
Etude de domaine

" La connaissance s'acquiert par l'expérience,
Tout le reste n'est que de l'information."
Albert EINSTEIN

Introduction

Lorsque nous collectons des fichiers malveillants à partir de la source prévue et disponible, nous les téléchargeons dans un environnement de développement pour extraire et définir les caractéristiques qui nous permettent ensuite de créer des modèles de classification et de choisir le modèle le plus approprié et idéal.

1. Extraction de caractéristiques

Extraction des caractéristiques ou Feature extraction est un processus de l'extraction des caractéristiques pertinentes et de réduction de la dimensionnalité par lequel un ensemble initial de données brutes est réduit à des groupes plus gérables pour le traitement [46].

2. Sélection de caractéristiques

Sélection de caractéristiques ou Feature selection est l'une des étapes de prétraitement les plus importantes de l'exploration de données. L'idée de base de l'algorithme de sélection des caractéristiques permet de rechercher dans toutes les combinaisons possibles d'attributs dans les données pour trouver quel sous-ensemble d'entités fonctionne le mieux pour la prédiction. Ainsi, les vecteurs d'attributs peuvent être réduits en nombre par lequel les plus significatifs sont conservés et les non pertinents ou redondants sont supprimés [47].

3. API

API (Application Programming Interface) est un ensemble de protocoles, de procédures et d'outils qui permettent l'interaction entre deux applications. C'est l'intermédiaire logiciel qui délivre une requête au serveur puis relaie une réponse vers le client. [48]. Lorsque vous connectez à une application ou que vous posez une question via un navigateur, vous effectuez un appel API (API call) [49].

4. Méthodologie

L'idée principale de notre système proposé est d'extraire des informations, de créer des ensembles de caractéristiques, de classer les échantillons dans une classe de malware ou une classe bénigne et de travailler sur ces catégories pour extraire une classification idéale**.**

Dans un premier temps, tous les programmes s'exécutent dans l'environnement sécurisé sous l'outil développé en interne. Après avoir surveillé les profils comportementaux des programmes, les appels d'API et les arguments sont extraits et des ensembles de caractéristiques (fonctionnalités) sont construits.

4.1 Construire la base de données

Nous avons constaté qu'à mesure que l'ensemble de données se développe, le problème du temps d'exécution, des exigences de stockage et les performances du système. Dans la phase de construire la base de données, nous collectons des échantillons des fichiers malveillants à partir de la source prévue et disponible.

L'une des méthodes disponibles pour résoudre ce problème est la sélection de caractéristiques (FS ; Feature Selection). La sélection de caractéristiques est utilisée pour sélectionner les attributs (entités 'feature') les plus pertinents dans la base de données. Cette approche est très efficace pour minimiser les données car elle filtre et nettoie les données ennuyeuses. Ce qui réduit la complexité de stockage et de temps et Amélioration de la précision des travaux [50].

4.2 Les Modèles de classification

Après les phases construire la base de données et la sélection des caractéristiques, le nombre d'attributs sera considérablement réduit et sera plus précis pour l'utilisation dans la construction du modèle de classification. Pour la phase de classification.

L'apprentissage supervisé (ou classification) consiste à construire un modèle basé sur un base d'apprentissage et des étiquettes (labels nom des catégories ou des classes) et à l'utiliser pour classer des données nouvelles.

Il existe plusieurs algorithmes et techniques utilisés pour la classification supervisée telles que :

- Naïve Bayes (NB)
- Régression logistique (Logistic Regression)
- SVM linéaire (Linear SVM)
- Forêt aléatoire (Random Forest)

5. Techniques de classification utilisées

5.1 Naïve Bayes (NB)

Le classifieur Naïve Bayes est l'instance la plus simple d'un classifieur probabiliste. La sortie Pr(C|d) d'un classifieur probabiliste est la probabilité qu'un document d appartienne à une classe C. Chaque document contient des termes auxquels sont attribuées des probabilités en fonction de son nombre d'occurrences dans ce document particulier. Avec la formation supervisée, Naïve Bayes peut apprendre le modèle d'examen d'un ensemble de documents de test bien catégorisés et donc de comparer le contenu de toutes les catégories en construisant une liste de mots ainsi que leur occurrence. Ainsi, une telle liste d'occurrences de mots peut être utilisée pour classer les nouveaux documents dans leurs bonnes catégories, selon la probabilité postérieure la plus élevée [51].

5.2 Régression logistique

L'analyse de régression logistique **(Logistic Regression)** est l'un des modèles de Machine Learning (Apprentissage automatique) les plus simple et interprétable [52] et d'analyse multivariée [53], étudie l'association entre une variable dépendante catégorielle et un ensemble de variables indépendantes (explicatives).

La régression logistique est un modèle statistique permettant d'étudier les relations entre un ensemble de variables qualitatives Xi et une variable qualitative Y. [52] Un modèle de régression logistique permet aussi de prédire la probabilité qu'un événement arrive. Ce résultat varie toujours entre 0 et 1.

- Y est une variable binaire
- 0 en cas de non occurrence de l'évènement
- 1 si occurrence
- Y aléatoire et Xi non aléatoires
- (Y,X1,X2,…,Xk) les variables de la population dont on extrait un échantillon de n individus i (yi,xi) est le vecteur des réalisations de (Yi,Xi)
- K variables explicatives [54]

5.3 SVM linéaire

Les Support Vector Machines **(SVM)**sont une classe d'algorithmes d'apprentissage. [55] L'idée est de rechercher une règle de décision basée sur une séparation par hyperplan de marge optimale, Méthode relativement récente qui découle de premiers travaux théoriques de Vapnik et Chervonenkis en 1995 [56].

Le but des SVM est de trouver un hyperplan qui va séparer et maximiser la marge de séparation entre deux classes [57].

L'idée est alors de trouver de frontières de séparation linéaire qui maximise la marge dans l'espace de grande dimension, trouver une frontière de séparation optimale dans un espace de grande dimension revient à trouv er une frontière non linéaire dans l'espace de départ. Voir figure suivante :

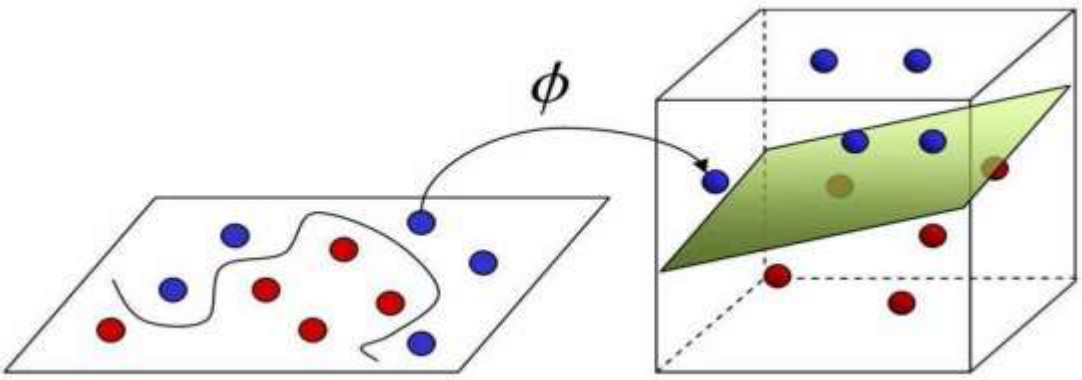

Figure 2.1. SVM : une séparation linéaire dans un espace de grande dimension est une séparation non linéaire dans l'espace de départ

L'objectif des SVMs est de trouver un hyperplan permettant de séparer les données d'apprentissage de sorte que tous les points d'une même classe soient d'un même côté de l'hyperplan.

Modèle Linéaire :

$$F(x) = w \cdot x + b$$

Définition de l'hyperplan (frontière de décision) :

$$w \cdot x + b = 0$$

La distance d'un point au plan est donnée par :

$$d(x) = |w \cdot x + b| / ||w||$$

Ou w est la normale de l'hyperplan et b paramètre de l'hyperplan.

5.4 Forêt aléatoire

La Forêt aléatoire (Random Forest) est un algorithme crée en 1995, Il est particulièrement efficace en termes de prédictions dans le domaine du machine learning, du deep learning et de l'intelligence artificielle (IA). Composé de plusieurs arbres de décision, travaillant de manière indépendante sur une vision d'un problème [58].
Un Random Forest fonctionne sur 3 étapes :

1- Découper une base de données en sous-ensembles (arbres de décision).
2- Proposer un modèle d'entraînement à chacun de ses groupes.
3- Combine les résultats de ces arbres afin d'obtenir la prévision la plus solide.

6. Métrique d'évaluation (Matrices de confusion)

Une matrice de confusion est un tableau souvent utilisé pour décrire les performances d'un modèle de classification sur un ensemble de données de test dont les vraies valeurs sont connues [64].
Pour tester et évaluer les modèles, 70 % de la base de données est utilisé, Les échantillons sont extraits puis utilisées comme ensemble de données d'analyse comparative pour les problèmes d'apprentissage automatique. En comparant la classe réelle des échantillons avec celle prédite (c'est-à-dire générée par le modèle de classification), les performances du système peuvent être mesurées en termes de rappel, de précision et de F-mesure.

- **True Positives(TP) :** ce sont les valeurs positives correctemnt prédites, ce qui signifie que la valeur de la classe réelle est oui et la valeur de la classe prédite est également oui.
- **True Negatives (TN) :** Ce sont les valeurs négatives correctement prédites, ce qui signifie que la valeur de la classe réelle est non et que la valeur de la classe prédite est également non.
- **False Positives (FP) :** Lorsque la classe réelle est non et la classe prédite est oui.
- **False Negatives (FN) :** Lorsque la classe réelle est oui mais la classe prédite est non.

Ensuite, nous pouvons calculer la Accuracy, la précision, le rappel et le score F1: [54]

Accuracy : La précision est la mesure de performance la plus intuitive et il s'agit simplement d'un rapport entre les observations correctement prédites et les observations totales. La précision est une excellente mesure, mais uniquement lorsque vous avez des ensembles de données symétriques où les valeurs des faux positifs et des faux négatifs sont presque les mêmes. Par conséquent, vous devez regarder d'autres paramètres pour évaluer les performances de votre modèle.

$$\textbf{Accuracy} = TP+TN/TP+FP+FN+TN$$

Précision: La précision est le rapport entre les observations positives correctement prédites et le total des observations positives prévues. La haute précision est liée au faible taux de faux positifs.

$$\textbf{Precision} = TP/TP+FP$$

Recall : Le rappel est le rapport entre les observations positives correctement prédites et toutes les observations de la classe réelle - oui.

$$\textbf{Recall} = TP/TP+FN$$

F1 score : Le score F1 est la moyenne pondérée de la précision et du rappel. Par conséquent, ce score prend en compte à la fois les faux positifs et les faux négatifs. Intuitivement, ce n'est pas aussi facile à comprendre

que la précision, mais F1 est généralement plus utile que la précision, surtout si vous avez une distribution de classe inégale. La précision fonctionne mieux si les faux positifs et les faux négatifs ont un coût similaire. Si le coût des faux positifs et des faux négatifs est très différent, il vaut mieux examiner à la fois la précision et le rappel.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Conclusion

Dans ce chapitre, nous avons discuté des étapes de notre système et des concepts généraux. Dans le dernier chapitre, nous parlerons en détail de chaque étape et des résultats obtenus et les comparerons pour obtenir une classification idéale.

Chapitre 3

Etude d'expérimentale

" La connaissance s'acquiert par l'expérience,
Tout le reste n'est que de l'information."
Albert EINSTEIN

Introduction

Ce dernier chapitre sera consacré à l'étude expérimentale en appliquant plusieurs modèles sont Naïves Bayes régression logistique SVM forêt aléatoire démarches de notre approche va être présenté dans les sections suivante.

Description de l'approche et méthodologie

Pour générer un modèle de classifieur, la figure 21 décrit la méthodologie en commençant par le prétraitement des données jusqu'à l'évaluation du modèle. En effet, certaines données sont inutiles (c'est-à-dire n'affectent pas le résultat de la classification même en les supprimant, comme les mots vides) donc une phase de prétraitement a été à effectuer en premier [10].

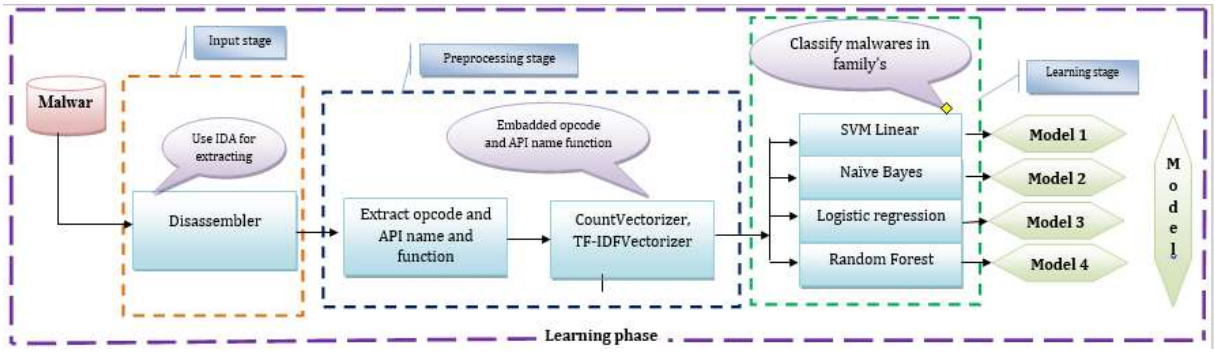

Figure 3.1. Phase d'apprentissage

Tout d'abord, nous commençons par la base de données utilisée :

Cette base est Kaspersky Anti-Virus est créé par la société russe Kaspersky Lab.[60] Contenant des logiciels malveillants, nous avons téléchargé la base de données sur les logiciels malveillants de Kaspersky et collecté un programme bénin, pour la plateforme Windows. Les échantillons des logiciels malveillants seront vérifiés à partir de logiciel Kaspersky installé au niveau de laboratoire de notre université. Ces données de logiciels malveillants ou bénins sont au format de fichier .exe.

Parmi le nombre total des échantillons de 37420 fichiers, nous sélections 6 échantillons par 9 classe, cette base est divisée en deux partie, 4 pour l'apprentissage et 2 pour le test.

Notre approche se compose en trois sections :

1.Phase d'entrée

IDA PRO pour désassembler :

À cette partie, nous prenons le fichier PE (Exécutable portable) et le mettons dans le programme IDA PRO car il peut tout démonter types de fichiers non exécutables et exécutables (tels que ELF, EXE, PE, etc.) pour désassembler, et qui accomplir, une analyse de code automatique, Il permet également la reconnaissance des références croisées à l'intérieur des sections de code ainsi que les paramètres de connaissance pour les appels aux API et d'autres informations. Lorsque on a charge le fichier(.exe) spécifié dans IDA PRO, Le résultat est le suivant :

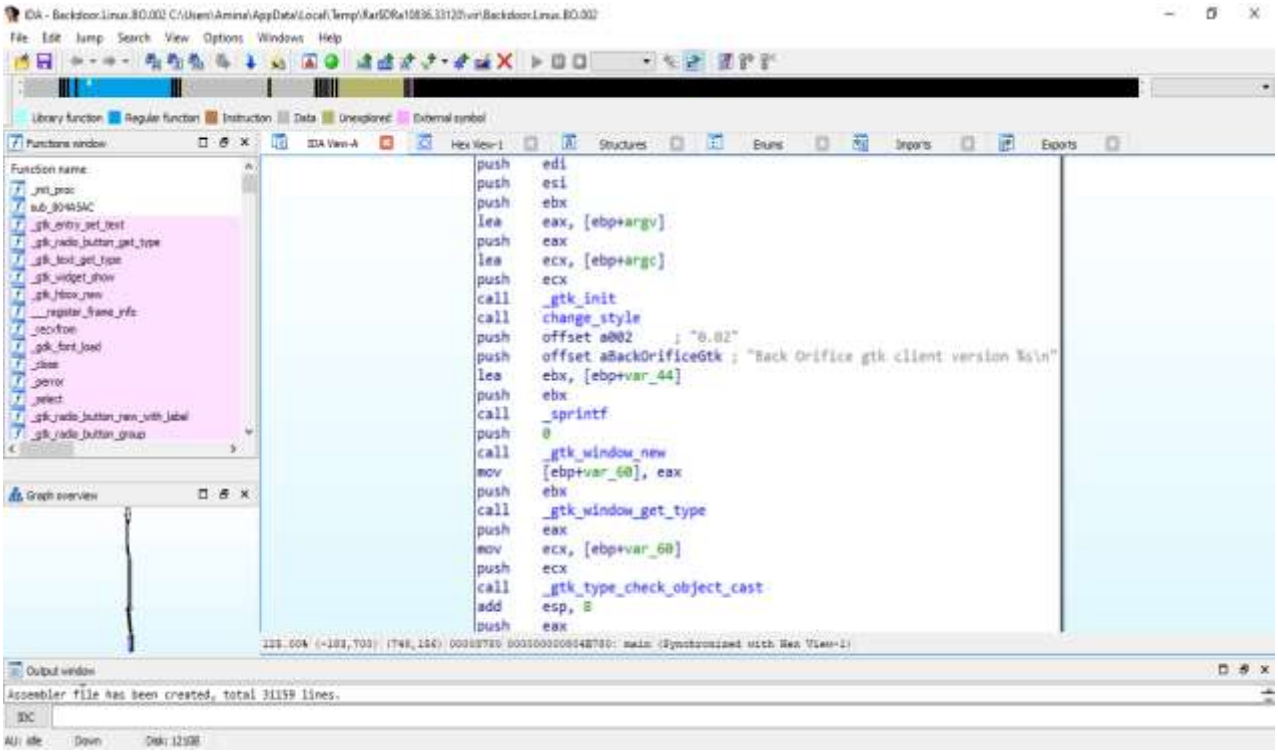

Figure 3.2. Exécution de fichier.exe

Et nous avons ici la possibilité de produire plusieurs fichiers parmi elle le fichier ASM qui contient l'opcode, l'API et les fonctions pour l'extraction des caractéristiques (feature extraction). Ce qui permet de réduire le nombre de caractéristiques dans la base de donné [61]. Ce nouvel ensemble de caractéristiques(fonctionnalités) sera en mesure de résumer la plupart des informations contenues dans l'ensemble de caractéristiques d'origine et Ceci est un exemple du fichier ASM.

```
Elf32_Sym struc ; (sizeof=0x10, align=0x4, mappedto_1)
                        ; XREF: LOAD:080487AC/r
                        ; LOAD:080487BC/r ...
st_name dd ?            ; offset (08049540)
st_value dd ?           ; offset (00000000)
st_size dd ?
st_info db ?
st_other db ?
st_shndx dw ?
Elf32_Sym ends

Elf32_Rel struc ; (sizeof=0x8, align=0x4, copyof_2)
                        ; XREF: LOAD:0804A2D4/r
                        ; LOAD:0804A2DC/r ...
r_offset dd ?
r_info dd ?
Elf32_Rel ends

Elf32_Dyn struc ; (sizeof=0x8, align=0x4, copyof_3)
                        ; XREF: LOAD:stru_805AA74/r
                        ; LOAD:0805AA7C/r ...
d_tag dd ?
d_un Elf32_Dyn::$A263394DDF3EC2D4B1B8448EDD30E249 ?
Elf32_Dyn ends

Elf32_Dyn::$A263394DDF3EC2D4B1B8448EDD30E249 union ; (sizeof=0x4, align=0x4, copyof_4)
                        ; XREF: Elf32_Dyn/r
d_val dd ?
d_ptr dd ?
Elf32_Dyn::$A263394DDF3EC2D4B1B8448EDD30E249 ends
```

Figure 3.3. Exemple de fichier ASM

2. Phase de prétraitements

2.1. Préparation de base et extraction de caractéristiques

Tout d'abord, nous nettoyons le contenu du fichier ASM, et utilisons d'abord les fonctions python pour supprimer la ponctuation et convertir toutes les majuscules en minuscules, et construisons la table de transition, c'est-à-dire définissant la liste des caractères qui doivent être remplacés dans la chaîne ou les caractères entiers qui devrait être supprimé de la chaîne [62]. Après avoir divisé une chaîne en une liste où chaque mot est un élément de liste et placé le résultat dans une liste, nous mettons le résultat (texte propre) dans un fichier texte (.txt).

```
['\ufeff', 'scopetableentry', 'struc', 'sizeof0xc', 'align0x4', 'copyof1', 'xref', 'rdatastru422220r', 'rdatastru423030r',
b401970fdr', 'wmonth', 'dw', 'xref', 'sub4019705dr', 'sub40197014dr', 'wdayofweek', 'dw', 'wday', 'dw', 'xref', 'sub4019704
infoa', 'struc', 'sizeof0x44', 'align0x4', 'copyof14', 'xref', 'sub403190r', 'cb', 'dd', 'lpreserved', 'dd', 'offset', 'lpd
pporthexrayscom', 'freeware', 'version', 'input', 'sha256', '84a84bc0af78b88d84fbd2c2bf6c86d94d880f5fec8f57f5d768ead4060444
'use32', 'assume', 'cstext', 'org', '401000h', 'assume', 'esnothing', 'ssnothing', 'dsdata', 'fsnothing', 'gsnothing', 'db'
', 'endp', 'align', '40h', 's', 'u', 'b', 'r', 'o', 'u', 't', 'i', 'n', 'e', 'attributes', 'bpbased', 'frame', 'sub401040',
e', 'attributes', 'bpbased', 'frame', 'sub4010b0', 'proc', 'near', 'code', 'xref', 'sub401014?j', 'var158', 'byte', 'ptr',
l', 'dsgetmodulefilenamea', 'cmp', 'esi', 'esp', 'call', 'sub4017d0', 'mov', 'esi', 'esp', 'push', '0', 'lpdwdisposition',
h', 'cmp', 'ebp', 'esp', 'call', 'sub4017d0', 'mov', 'esp', 'ebp', 'pop', 'ebp', 'retn', 'sub4010b0', 'endp', 'db', '39h',
, 'ecx', 'ebpbuffer', 'push', 'ecx', 'lpbuffer', 'call', 'dsgetsystemdirectorya', 'cmp', 'esi', 'esp', 'call', 'sub4017d0',
db', '3bh', 'dup0cch', 's', 'u', 'b', 'r', 'o', 'u', 't', 'i', 'n', 'e', 'attributes', 'bpbased', 'frame', 'sub4012f0', 'pr
bpvar320', 'push', 'edx', 'call', 'sub40100f', 'add', 'esp', '4', 'lea', 'eax', 'ebpvar320', 'push', 'eax', 'lea', 'ecx', '
'sub4017d0', 'loc4013f3', 'code', 'xref', 'sub4012f0b6?j', 'push', 'offset', 'asc422124', 'lea', 'ecx', 'ebpvar104', 'push'
a', 'edx', 'ebpnewfilename', 'push', 'edx', 'lpnewfilename', 'lea', 'eax', 'ebpfilename', 'push', 'eax', 'lpexistingfilenam
```

Figure 3.4. Le résultat de fichier ASM après le désassembler

2.2. Prétraiter l'ensemble de données

L'une des tâches principales avant de développer un modèle sur des données textuelles est de nettoyer le texte. Cela inclut généralement la mise en minuscules du texte, la suppression des mots vides, la suppression des ponctuations, des caractères alphanumériques et / ou spéciaux, etc. Les fonctions preprocess_corpus () et clean_text () nettoieront notre ensemble de données.

2.3. Préparer la base de données

Effectuer l'ingénierie des fonctionnalités et la préparation des données. Nous convertissons le texte en représentation vectorielle en utilisant la méthode tfidf :

- Supprimer les mots vides
- Utiliser des unigrammes
- Utilisez toutes les fonctionnalités (mots) ou un sous-ensemble d'entre elles
- Encodez les étiquettes de classe en nombres

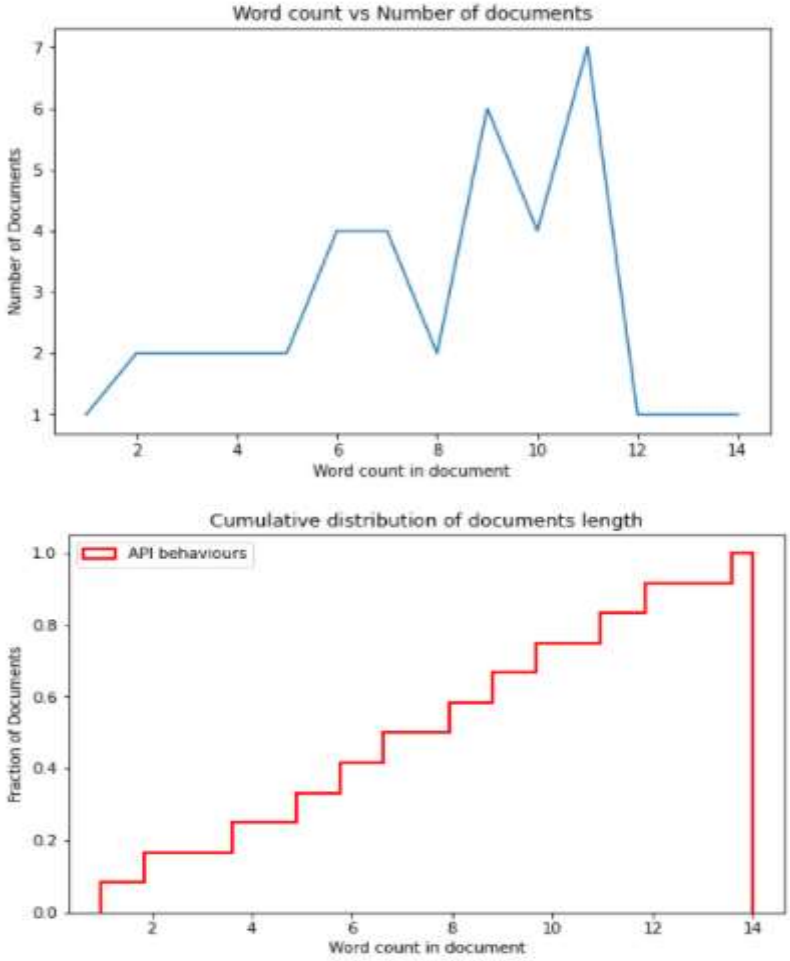

Figure 3.5. Nombre de mots dans document

2.4. Diviser l'ensemble de données en trains et ensembles de test

Nous utilisons la fonction train_test_split () stratifiée. Cette fonction déterminera les distributions des classes et maintiendra la même distribution pour le train et les ensembles de test.

2.5. Charger et explorer les données

Une fois les données chargées, nous devons examiner et obtenir un aperçu des données.

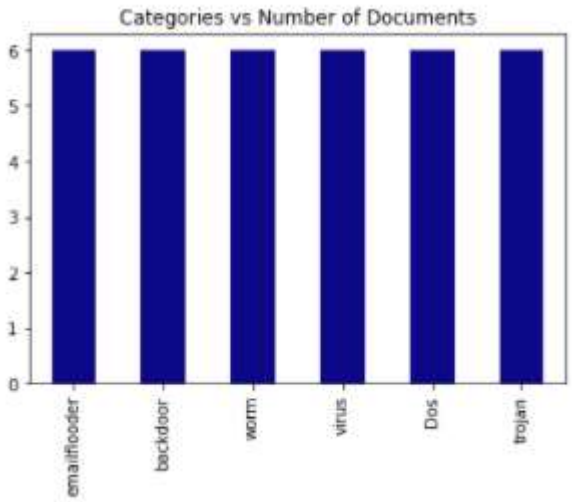

Figure 3.6. Catégories vs nombre de documents

Le graphique à barres montre que l'ensemble de données équilibré, c'est-à-dire que le nombre d'observations par classe est uniformément réparti. Par exemple, le nombre de documents relatifs à "backdoor" est le même que "trojan ».

2.6. Sélection de caractéristique

Ici nous comparons cette fichier texte avec une liste de comportement d'API « l'API behaviors » (extraction les appels API de fichier texte selon tous les appels de fonction ce que soit le type de comportement) ; Qui sont utilisés pour créer un modèle et distinguer les logiciels malveillants des fichiers bénins, et le résultat est une sélection de caractéristiques.

Behaviour	Malware Category	API Function Calls
Behaviour 1	Search Files to Infect	FindClose, FindFirstFile, FindFirstFileEx, FindFirstFileName, TransactedW, FindFirstFileNameW, FindFirstFileTransacted, FindFirstStream, TransactedW, FindFirstStreamW, FindNextFile, FindNextFileNameW, FindNextStreamW, SearchPath.
Behaviour 2	Copy/Delete Files	CloseHandle, CopyFile, CopyFileEx, CopyFileTransacted, CreateFile, CreateFileTransacted, CreateHardLink, CreateHardLink, Transacted, CreateSymbolicLink, CreateSymbolic, LinkTransacted, DeleteFile, DeleteFileTransacted.
Behaviour 3	Get File Information	GetBinaryType, GetCompressed, FileSize, GetCompressedFile, SizeTransacted, GetFileAttributes, GetFileAttributesEx, GetFileAttributes, Transacted, GetFileBandwidth, Reservation, GetFileInformation, ByHandle, GetFileInformation, ByHandleEx, GetFileSize, GetFileSizeEx, GetFileType, GetFinalPathName, ByHandle, GetFullPathName, GetFullPathName, Transacted, GetLongPathName, GetLongPathName, Transacted, GetShortPathName, GetTempFileName, GetTempPath.
Behaviour 4	Move Files	MoveFile, MoveFileEx, MoveFileTransacted, MoveFileWithProgress.
Behaviour 5	Read/Write Files	OpenFile, OpenFileById, ReOpenFile, ReplaceFile, WriteFile, CreateFile, CloseHandle.
Behaviour 6	Change File Attributes	SetFileApisToANSI, SetFileApisToOEM, SetFileAttributes, SetFileAttributesTransacted, SetFileBandwidthReservation, SetFileInformationByHandle, SetFileShortName, SetFileValidData

Figure 3.7. Liste de comportement d'API [63]

3. Phase de classification

Après la sélection des caractéristiques, l'étape suivante consiste à trouver le meilleur classifieur pour la détection des logiciels malveillants avancés. Nous avons étudié quatre classifieurs : le SVM, Naïve Bayes, Régression logistique (logistic regression), Forêt aléatoire (Random forest).

Nous téléchargeons des bibliothèques et les ensembles de données(DATASETS).

3.1. Fonction utilisés

1- Load_data (chemin fichier) : charger le fichier csv et renvoyez un Dataframe.

2- clean_text(**text**) : nettoyer le texte en supprimant les caractères spéciaux, les ponctuations, etc.

3- preprocess_corpus(**df**, **column**=*'text'*) : Prétraitez l'ensemble du corpus, y compris le nettoyage des documents texte et renvoyez la trame de données mise à jour.

4- encode_labels(**labels**) : Encodez les étiquettes de classe en nombres.

5- compute_tfidf(**corpus**,**stop_words**=*'english'*, **ngram_range**=*(1,1)*, **max_features**=*None*) : Calculez les fonctionnalités tfidf pour tous les documents texte et renvoyez une matrice (documents, fonctionnalités).

6- train_test_model(**model**, **X_train**, **X_test**, **y_train**, **y_test**, **labels**) : Entraînez et testez le modèle à l'aide des ensembles de données d'entraînement et de test. Renvoyer les prévisions, la précision et les rapports métriques.

3.2. Construisez le modèle

Dans cette section, je vais plusieurs modèles différents pour notre tâche de classification multiclasse et les comparer à la fin. Il existe une grande variété de techniques qui peuvent être utilisées. J'ai décidé d'appliquer les méthodes suivantes:

- Naive Bayes (NB)
- Régression logistique (Logistic Regression)
- SVM linéaire (Linear SVM)
- Forêt aléatoire (Random Forest)

J'ai implémenté :

3.2.1 Naïve Bayes

```
Number of documents = 21  |  Number of features = 17
Start training...done!
Start testing...done!
Total time: 2.89s
accuracy: 0.26666666666666666
==================================================
              precision    recall  f1-score   support

    backdoor       0.00      0.00      0.00         2
emailflooder       0.00      0.00      0.00         4
        worm       1.00      1.00      1.00         3
      trojan       0.00      0.00      0.00         4
         Dos       0.00      0.00      0.00         1
       virus       0.17      1.00      0.29         1

    accuracy                           0.27        15
   macro avg       0.19      0.33      0.21        15
weighted avg       0.21      0.27      0.22        15
```

Figure 3.8 matrice de confusion naïve bayes

3.2.2. Régression logistique (logistic regression)

```
Number of documents = 21  |  Number of features = 17
Start training...done!
Start testing...done!
Total time: 0.76s
accuracy: 0.26666666666666666
=====================================================
              precision    recall  f1-score   support

    backdoor       0.00      0.00      0.00         2
emailflooder       0.00      0.00      0.00         4
        worm       1.00      1.00      1.00         3
      trojan       0.00      0.00      0.00         4
         Dos       0.00      0.00      0.00         1
       virus       0.20      1.00      0.33         1

    accuracy                           0.27        15
   macro avg       0.20      0.33      0.22        15
weighted avg       0.21      0.27      0.22        15
```

Figure 3.9. Matrice de confusion régression logistique

3.2.3. Le SVM:

```
Number of documents = 21  |  Number of features = 17
Start training...done!
Start testing...done!
Total time: 0.04s
accuracy: 0.26666666666666666
=====================================================
              precision    recall  f1-score   support

    backdoor       0.00      0.00      0.00         2
emailflooder       0.00      0.00      0.00         4
        worm       1.00      1.00      1.00         3
      trojan       0.00      0.00      0.00         4
         Dos       0.00      0.00      0.00         1
       virus       0.20      1.00      0.33         1

    accuracy                           0.27        15
   macro avg       0.20      0.33      0.22        15
weighted avg       0.21      0.27      0.22        15
```

Figure 3.10. Matrice de confusion SVM

3.2.4. Forêt aléatoire (Random forest)

```
Number of documents = 21  |  Number of features = 17
Start training...done!
Start testing...done!
Total time: 0.80s
accuracy: 0.26666666666666666
=====================================================
              precision    recall  f1-score   support

    backdoor       0.00      0.00      0.00         2
emailflooder       0.00      0.00      0.00         4
        worm       1.00      1.00      1.00         3
      trojan       0.00      0.00      0.00         4
         Dos       0.00      0.00      0.00         1
       virus       0.25      1.00      0.40         1

    accuracy                           0.27        15
   macro avg       0.21      0.33      0.23        15
weighted avg       0.22      0.27      0.23        15
```

Figure 3.11. Matrice de confusion Forêt aléatoire

3.3.Résultats d'évaluation pour le modèle de classification

3.3.1 Comparaison des performances

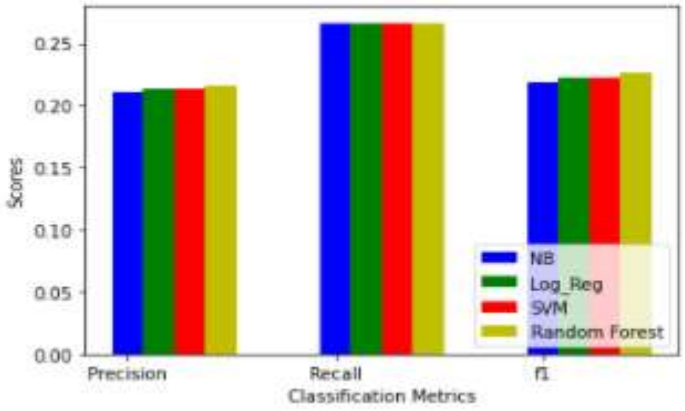

Figure 3.12. Matrices de classement

3.3.2 Comparaison du temps de formation. (Temps d'exécution)

Ce qui suit démontre que le SVM est incroyablement rapide et que NB le suit en ayant des temps d'apprentissage de 0,03 et 0,02 s

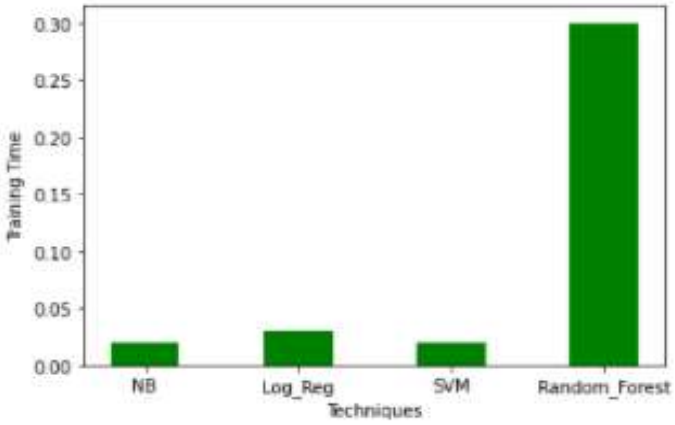

Figure 3.13. Comparaison du temps d'exécution

Conclusion

Grâce aux modèles que nous avons extraits et à travers les résultats d'évaluation des modèles de classification, nous avons constaté que la méthode optimale et précise avec un pourcentage élevé et dans un temps très court (temps d'exécution) pour la classification est la SVM.

Références Bibliographiques

[1] Jun Li, Yifeng Zhou, and Louise Lamont, Communication Architectures and Protocols for Networking Unmanned Aerial Vehicles, Communications Research Centre Canada, 3701 Carling Ave. Ottawa, ON. K2H 8S2 Canada , Globecom 2013 Workshop - Wireless Networking and Control for Unmanned Autonomous Vehicles.

[2] Edouard Finokki, commande vol non linéaire d'un drone à voilure fixe par la méthode du backstepping, école de technoligie supérieure université du québec mémoire présenté à l'école de technologie supérieure comme exigence partielle à l'obtention de la maitrise en génie, concentration génie aérospatial M.Sc A montréal, le 20 mai 2015.

[3] Prof. Tullio Joseph TANZI, Drone Autonome pour l'Intervention Humanitaire Institut Mines-Telecom, Telecom ParisTech, LTCI CNRS, 06904 Sophia Antipolis cedex, France, 06904 Sophia Antipolis cedex, France, Journées scientifiques URSI : L'HOMME CONNECTÉ. 25 et 26 MARS 2014.

[4] Colas Antoine, Damour Benjamin, Hespel Audouard, Nguyen Tri Nam, Sow Papa Libasse, Les Drones 3 eme Année, Electronique et Informatique Industrielle.

[6] Gasri Mohamed Larbi, New Attacks Classification on UAVs, 3rd ICSTR Paris - International Conference on Science & Technology Research, 28-29 November 2021, 28-Nov- 2021 to 29-Nov- 2021(accepté).

[7]] Paul Fahlstrom and Thomas Gleason. Introduction to UAV systems. John Wiley & Sons, 2012

[8] Magdalena Dudek, Piotr Tomczyk, Piotr Wygonik, Mariusz Korkosz, Piotr Bogusz, and Bartłomiej Lis. Hybrid Fuel Cell – Battery System as a Main Power Unit for. Small Unmanned Aerial Vehicles (UAV). Int. J. Electrochem. Sci, 8:8442–8463, 2013.

[9] fr.wikipedia.org/wiki/Microsoft.

[10] Jun, Xie Shaorong, Gong Zhenbang, Rao Jinjun, Unmanned Surveillance Aircraft and its Ground Control Station for Security, Proceedings of the 2005 IEEE International Workshop on Safety, Security and Rescue Robotics Kobe, Japan, June 2005 Subminiature School of Mechatronics Engineering and Automation, Shanghai University,Shanghai200072,China Email:wqlj1228@263.net.

[11] Ye Hong, Jiancheng Fang, and Ye Tao, Ground Control Station Development for Autonomous UAV, Key Laboratory of Fundamental Science for National Defense, Novel Inertial Instrument & Navigation System Technology, Beijing, 100191, China

[12] Yiqi Kang, Mei Yuan, Ground Control Station of UAV, School of Automation Science and Electrical Engineering, Beijing University of Aeronautics & Astronautics Beijing 100191, P.R.China.

[13] Robert H.Klenke, Jefferson McBride, and Hoan Nguyen, Flight Control System for Small UAVs, A Reconfigurable, Linux-based, In: AIAA infotech@Aerospace 2007 Conference and Exhibit[C]. Rohnert Park, California: AIAA, 2007. 1-10.

[14] XIONG Zi-ming, Ge Wen, The Design and Realization of Ground Monitor and Navigation System for UAV Based on GIS[J], Hydrographic Surveying and Charting, 2007, Vol.27 No.4: 54-56(in Chinese)..

[15] Steven AP Quintero, Francesco Papi, Daniel J Klein, Luigi Chisci, and Joao P Hespanha, Optimal UAV coordination for target tracking using dynamic programming, In Decision and Control (CDC), 2010 49th IEEE Conference on, pages 4541–4546. IEEE, 2010.

[16] Jun Li, Yifeng Zhou, and Louise Lamont, Communication Architectures and Protocols for Networking Unmanned Aerial Vehicles,Communications Research Centre Canada, 3701 Carling Ave. Ottawa, ON. K2H 8S2 Canada , Globecom 2013 Workshop - Wireless Networking and Control for Unmanned Autonomous Vehicles

[17] Kim Hartmann, Christoph Steup, The Vulnerability of UAVs to Cyber Attacks - An Approach to the Risk Assessment, 5th International Conference on Cyber Conflict K. Podins, J. Stinissen, M. Maybaum (Eds.) 2013.

[18]https://www.csoonline.com/article/3295877/what-is-malware-viruses-worms-trojans-and-beyond.html May 17, 2019 3:00 am By Josh Fruhlinger

[19]https://www.securiteinfo.com/attaques/malwares-virus-spam-logiciels-indesirables/malwares.shtml Arnaud Jacques 06 juin 2005-2016

[20]https://www.google.com/search?q=different+type+de+malware&client=firefox-b-d&sxsrf=ALeKk01Vhl2eOiMeEThU6d6SnFy2vXIe9A:1618864845798&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi0xcrAlYvwAhWEwuYKHftXC3kQ_AUoAXoECAEQAw&biw=1138&bih=545#imgrc=awPJhsxpWQU26M

[21] Ammar Ahmed, E. Elhadi, Mohd Aizaini Maarof, and Ahmed Hamza Osman, Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph Information Assurance and Security Research Group, Faculty of Computer Science and Information Systems, University Technology, Malaysia American Journal of Applied Sciences 9 (3): 283-288, 2012 ISSN 1546-9239 © 2012 Science Publications.

[22] https://img.directindustry.fr/images_di/photo-m2/160571-12715893.jpg

[23]https://www.google.com/imgres?imgurl=http%3A%2F%2Fguide.directindustry.com%2Fwp-content%2Fuploads%2F182307-10815143.jpg&imgrefurl=http%3A%2F%2Fguide.directindustry.com%2Ffr%2Fbien-choisir-un-drone%2F&tbnid=YUKqDFOJ7Ql0zM&vet=12ahUKEwivgqLbyLfwAhVHEhoKHTbPDWsQMygCegUIARCfAQ..i&docid=ExBy5M4dwc1vJM&w=1666&h=1062&q=drone%20%C3%A0%20voilure%20tournante&client=firefox-b-d&ved=2ahUKEwivgqLbyLfwAhVHEhoKHTbPDWsQMygCegUIARCfAQ

[24] Ammar Ahmed, E. Elhadi, Mohd Aizaini Maarof ,and Ahmed Hamza Osman Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph Information Assurance and Security Research Group, Faculty of Computer Science and Information Systems, University Technology, Malaysia American Journal of Applied Sciences 9 (3): 283-288, 2012 ISSN 1546-9239 © 2012 Science Publications

[25]J.Rabek,R.Khazan,S.Lewandowski,andR.Cunningham.Detectionofinjected, dynamically generated, and obfuscated malicious code. In Proceedings of the 2003 ACM Workshop on Rapid Malcode, pages 76–82, 2003

[26] Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiware, A., & Yang, H., "Specification-based Anomaly Detection: A New Approach for DetectingNetwork Intrusions", ACM Computer and Communication Security Conference, 2002

[27] Robiah Y, Siti Rahayu S., Mohd Zaki M, Shahrin S., Faizal M. A., Marliza R A New Generic Taxonomy on Hybrid Malware Detection Technique Faculty of Information Technology and Communication Univeristi Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia (IJCSIS) International Journal of Computer Science and Information Security, Vol. 5, No. 1, 2009.

[28] I.J. Education and Management Engineering, 2018, 2, 20-30 Published Online March 2018 in MECS (http://www.mecs-press.net) DOI: 10.5815/ijeme.2018.02.03 Available online at http://www.mecs-press.net/ijeme A Study on Malware and Malware Detection Techniques Rabia Tahir Department of Computer Science, Virtual University of Pakistan Received: 09 October 2017; Accepted: 19 December 2017; Published: 08 March 2018.

[29] ÖMER ASLAN 1,2 AND REFIK SAMET 1, A Comprehensive Review on Malware Detection (Member, IEEE) 1Computer Engineering Department,

[30] YANFANG YE, A Survey on Malware Detection Using Data Mining Techniques, West Virginia University TAO LI, Florida International University & Nanjing University of Posts and Telecommunications DONALD ADJEROH, West Virginia University S. SITHARAMA IYENGAR, Florida International University

[31]M.F.Zolkipli, and A. Jantan, ‘‘A framework for malware detection using combinationtechniqueandsignaturegeneration,’’inProc.2ndInt.Conf. Comput. Res. Develop., May 2010.

[32] Y. Tang, B. Xiao, and X. Lu, ‘‘Using a bioinformatics approach to generate accurate exploit-based signatures for polymorphic worms,’’ Comput. Secur., vol. 28, no. 8, pp. 827–842, Nov. 2009

[33] O. Aslan and R. Samet, ‘‘Investigation of possibilities to detect malware using existing tools,’’ in Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA), Oct. 2017

[34] M. Sikorski and A. Honig, Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. San Francisco, CA, USA: No Starch Press, 2012.

[35] J.Kinder,S.Katzenbeisser, C.Schallhart, and H.Veith, ‘‘Detectingmalicious code by model checking,’’ in Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment. Berlin, Germany: Springer, 2005.

[36] http://www.afcadillac.net/_serveurs/drone/composition_dun_drone.html.

[37] S. Das, Y. Liu, W. Zhang, and M. Chandramohan, ‘‘Semantics–based online malware detection: Towards efficient real–time protection against malware,’’ IEEE Trans. Inf. Forensics Security, vol. 11, no. 2,pp. 289–302, Feb. 2016.

[38] K. Griffin, S. Schneider, X. Hu, and T.-C. Chiueh, ‘‘Automatic generation of string signatures for malware detection,’’ in Proc. Int. Workshop Recent Adv. Intrusion Detection. Berlin, Germany: Springer, 2009.

[39] R. Islam, R. Tian, L. M. Batten, and S. Versteeg, ‘‘Classification of malware based on integrated static and dynamic features,’’ J. Netw. Comput. Appl., vol. 36, no. 2, pp. 646–656, Mar. 2013.

[40] P. Singh and A. Lakhotia, ‘‘Static verification of worm and virus behavior in binary executables using model checking,’’ in Proc. IEEE Syst., Man Soc. Inf. Assurance Workshop, Mar. 2003.

[41] P. Beaucamps and J. Marion, ‘‘On behavioral detection,’’ in Proc. EICAR, vol. 9, 2009.

[42] L. Martignoni, R. Paleari, and D. Bruschi, ‘‘A framework for behaviorbased malware analysis in the cloud,’’ in Proc. Int. Conf. Inf. Syst. Secur. Berlin, Germany: Springer, 2009

[43] H. Sun, X. Wang, J. Su, and P. Chen, ‘‘RScam: Cloud-based anti-malware via reversible sketch,’’ in Proc. Int. Conf. Secur. Privacy Commun. Syst. Cham, Switzerland: Springer, 2015.

[44] L. Xiao, Y. Li, X. Huang, and X. Du, ‘‘Cloud–based malware detection game for mobile devices with offloading,’’ IEEE Trans. Mobile Comput., vol. 16, no. 10, pp. 2742–2750, Oct. 2017...

[46]https://deepai.org/machine-learning-glossary-and-terms/feature-extraction.

[47] S.L. Ting, W.H. Ip, Albert H.C, Is Naïve Bayes a Good Classifier for Document Classification?, International Journal of Software Engineering and Its Applications Vol. 5, No. 3, July, 2011. Tsang Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hum, Kowloon, Hong Kong jacky.ting@polyu.edu.hk.

[48]https://www.redhat.com/fr/topics/api/what-are-application-programming-interfaces

[49] https://rapidapi.com/blog/api-glossary/api-call/.

[50] C. Rama Krishna2 and Sanjay K., Detection of Advanced Malware by Machine Learning Techniques Sanjay Sharma1, Sahay3 1M.E. Scholar, Department of Computer Science and Engineering, 2Professor and Head, Department of Computer Science and Engineering, 3Assistant Professor, Department of Computer Science and Information System, 1,2National Institute of Technical Teachers Training and Research, Chandigarh, India 3BITS, Pilani, Goa Campus, India 1sanjay.cse@nitttrchd.ac.in,2 rkc_97@yahoo.com, 3ssahay@goa.bits-pilani.ac.in

[51] S. Chakrabarti, S. Roy, and M.V. Soundalgekar, "Fast and accurate text classification via multiple linear discriminant projection", The VLDB Journal The International Journal on Very Large Data Bases, 2003, pp. 170–185

[52] https://datascientest.com/regression-logistique-quest-ce-que-cest.

[53]https://www.em-consulte.com/article/842576/comprendre-la-regression-logistique

[54] ECKER, S., LINCOLN, P., MARTI-OLIET, N., MESEGUER, J., VERDIJO, A., "Deduction, strategies, and rewriting.", *In 6th International Workshop on Strategies in Automated Deduction, STRATEGIES 2006,* Electronic Notes in Theoretical Computer Science, Vol. 174(11), pp. 3–25, Elsevier, Amsterdam, 2007.

[55] Introduction aux "Support Vector Machines" (SVM) Olivier Bousquet,Centre de Math´ematiques Appliqu´ees,Ecole Polytechnique, Palaiseau, Orsay, 15 Novembre 2001.

[54] Hyeoun-Ae Park Seoul National University An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain April 2013,Journal of Korean Academy of Nursing 43(2):154-164, DOI:10.4040/jkan.2013.43.2.154, Source PubMed

[56] Sebastien Gadat Seance 12: Algorithmes de Support Vector ´ Machines Laboratoire de Statistique et Probabilites´ UMR 5583 CNRS-UPS www.lsp.ups-tlse.fr/gadat

[57] Fares Menasri. Thèse de doctorat « Contributions à la reconnaissance de l'´écriture arabe manuscrite » université de Paris , France , 2008

[58] https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501905-random-forest-ou-foret-aleatoire-definition-et-cas-d-usage/ Mis à jour le 28/05/21 16:24..

[60] Kaspersky http://vx.zedz.net/kav/doc/release_notes_wsfs_en.html.

[61] https://deepai.org/machine-learning-glossary-and-terms/feature-extraction

[62] https://www.geeksforgeeks.org/python-maketrans-translate-functions/ Difficulty Level: Medium Last Updated: 15 Oct, 2020.

[63] 2010 Second Cybercrime and Trustworthy Computing Workshop Towards Understanding Malware Behaviour by the Extraction of API calls Mamoun Alazab Internet Commerce Security Laboratory (ICSL) University of Ballarat , Sitalakshmi Venkataraman Internet Commerce Security Laboratory (ICSL) University of Ballarat, Paul Watters Internet Commerce Security Laboratory (ICSL) University of Ballarat

[64]https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/September 9, 2016

[65] Jun Li, Yifeng Zhou, and Louise Lamont, Communication Architectures and Protocols for Networking Unmanned Aerial Vehicles,Communications Research Centre Canada, 3701 Carling Ave. Ottawa, ON. K2H 8S2 Canada , Globecom 2013 Workshop - Wireless Networking and Control for Unmanned Autonomous Vehicles.

[66] Alireza Abbaspoura *, Kang K. Yena , Shirin Noeib , Arman Sargolzaei Detection of Fault Data Injection Attack on UAV Using Adaptive Neural Network Complex Adaptive Systems, Publication 6 Cihan H. Dagli,

Editor in Chief Conference Organized by Missouri University of Science and Technology 2016 - Los Angeles, CA Procedia Computer Science 95 (2016) 193 – 200.

[67]https://www.futura-sciences.com/sciences/definitions/aeronautique-drone-6174/ ©2001-2021Futura-Sciences, tous droits réservés - Groupe MadeInFutura.

[68]https://moodle.utc.fr/pluginfile.php/16777/mod_resource/content/0/SupportIntroSecu/co/CoursSecurite_15.html.