الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية التكنولوجيا

قسم: الإعلام الآلي

Mémoire de Master

Spécialité :

Sécurité Informatique et Cryptographie (SIC)

Thème

La détection de maliciels Android

Présenté par :

SMAHI FEDWA

GUENACHI IBRAHIM YOUNES

Dirigé par :

Dr.MEBARKA YAHLALI

Promotion 2021 - 2022

Dédicace

Je dédie ce modeste travail :

A mes chers parents : *Que nulle dédicace ne puisse exprimer ce que je leur dois, pour toute leurs efforts et leurs sacrifices durant ma vie, leur bienveillance, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mon étude. Que ce travail soit témoignage de mon profond amour et ma grande reconnaissance "Que Dieu vous garde".*

A mes chères soeurs et mes chers frères : *pour leur encouragement permanent et leur soutien moral, je leur dédie ce modeste travail en témoignage de mon grand amour et ma gratitude infinie.*

A toute ma Famille : *notament la famille SMAHI, DJOUDI et MOHIEDDINE, pour leur soutien tout au long de mon parcours universitaire. Que ce travail soit l'accomplissement de vos voeux tant allégués, et le fruit de votre soutien infaillible.*

A toutes mes amies : *Pour leur aide et leur soutien moral durant l'élaboration du travail de fin d'études et en souvenir des bons moments qu'on a vecu ensemble que notre amitié durera éternellement.*

Smahi Fedwa

Dédicace :

Je dédie ce travail à :

Mes chers parents que nul dédicace ne peut exprimer mes sincères sentiments, je les remercie pour leur patience illimitée, leur encouragement continu ainsi que leur aide précieuse, en témoignage de mon profond amour et respect pour leur grands sacrées.

Mes chers frères et sœurs.

A mon trés cher ami Hassen merci pour tes orientations précieuses et tes conseils inestimables.

Mon chère binôme Fedwa tu es été trés collaboratrice et c'est un honneur pour moi d'avoir partagé ce travail avec toi.

Mes chers ami(e)s qui sans leur encouragements ce travail n'aura jamais vu le jour.

Et en fin à toute ma famille et à tous ceux que j'aime.

Guenachi Ibrahim Younes

Remerciments

Avant tout, nous remercions le bon DIEU de nous avoir aidé à accomplir ce modeste travail.

Nous tenons à saisir cette occasion et adresser nos profonds remerciements et nos profondes reconnaissances à :

Notre encadreur : Madame Mebarka Yahlali pour sa disponibilité, ses conseils, son orientation et ses encouragements tout au long de notre recherche.

Les membres du jury qui ont accepté d'évaluer et d'examiner ce travail.

Monsieur Abdelkader Majdoubi pour ses conseils et ses idées pour la réalisation.

Dr. Amel Belhadj pour son aide, ses conseils et ses encouragements.

Nos familles et nos amis qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Enfin, nous ne saurions terminer ces remerciements sans n'y associer toute personne qui a participé de près ou de loin à l'exécution de ce modeste travail.

ملخص

أصبحت تطبيقات الهاتف المحمول أداة مهمة في مجالات مختلفة فهي مرتبطة بمواقع الويب ونظم المعلومات وخدمات تخزين البيانات السحابية. الاندرويد هو نظام تشغيل الهاتف المحمول الأكثر شعبية المثبت على ملايين الأجهزة (الهواتف الذكية والأجهزة اللوحية وأجهزة التلفزيون والساعات الذكية)، هذه الإتجاهات الجديدة تتيح تقاسم الموارد وتبادل المعلومات السرية عن طريق الرسائل الإلكترونية مما يؤدي إلى محاولات لخرق السياسات الأمنية عن طريق الوصول الكيدي. تطبيقات هاتف الأندرويد بشكل كبير لهذه السياسات نظرا لشعبية هذا النظام. التعرف على البرامج الضارة هو مشكلة ثنائية تشير إلى إحتواء البرامج على برمجيات ضارة أم لا.

في هذا السياق تم إدخال العديد من التقنيات والمناهج لفهم وتحليل والتعرف على تصنيف برامج الأندرويد الضارة ورغم ذلك لا يزال الأندرويد يواجه العديد من التحديات والمشاكل الأمنية.

الغرض من عملنا هو تقييم بعض الأساليب الحالية وإقتراح نهج يعتمد على خوارزميات تعديل البيانات التقليدية للكشف عن البرامج الضارة.

Abstract

Mobile applications have become an important tool in different areas, they are connected to websites, information systems and cloud data storage services. Android is the most popular mobile operating system installed on millions of devices (smartphones and tablets, TVs, and smart watches). These new trends allow for the sharing of resources and the exchange of confidential information on electronic messaging. These effects are accompanied by attempts to breach security policies by malicious accesses. Android mobile apps are also highly exposed to malicious attacks and access. Malware recognition is a binary issue that indicates whether a program is malware or not. In this context several techniques and approaches have been introduced to understand, analyze, recognize and classify Android malware, but the threat is constantly effervescence, Android still faces many security challenges and problems. The purpose of our work is to evaluate some existing approaches and propose an approach based on traditional data mining algorithms for malware detection.

Résumé

Les applications mobiles sont devenues un outil important dans différents domaines, elles sont connectées avec les sites web, les systèmes d'information et les services de stockage de données dans les nuages. Android est le système d'exploitation mobile le plus populaire installé sur des millions d'appareils (Smartphones et les tablettes, téléviseurs, et montres intelligentes). Ces nouvelles tendances permettent le partage des ressources et l'échange des informations confidentielles sur les messageries électroniques. Ces effets sont accompagnés par des tentatives de violations des politiques de sécurité par des accès malveillants. Les applications Android mobile sont également très exposées aux attaques et accès malveillants.

La reconnaissance des logiciels malveillants est un problème binaire qui indique si un programme est un logiciel malveillant ou non.

Dans ce contexte plusieurs techniques et approches ont été introduites pour comprendre, analyser, reconnaître et classer les logiciels malveillants Android, mais la menace est constamment défervescence, Android doit encore faire face à de nombreux défis et problèmes de sécurité.

Le but de notre travail est d'évaluer quelques approches existantes et de proposer une approche basée sur les des algorithmes de fouille de données classiques pour la détection des logiciels malveillants.

Table des matières

Liste des tableaux

Table des figures

Liste des acroymes

AA	Apprentissage Automatique.
APK	Android Package Kit.
API	Application programming interface.
CSS	Cascading Style Sheets.
CNN	réseau Neuronal Convolutif.
CND	Contrôle Non Destructif.
DOS	Disk Operating System.
DLL	Dynamic Link Library.
DVM	Dalvik Virtual Machine.
FP	False Positive.
FN	False Negative.
GID	Group Identifier.
GUI	Graphical User Interface.
HTC	High Tech Computer.
HTTPS	Hypertext Transfer Protocol Secure.
HTML	HyperText Markup Language.
Ios	iphone operating system.
JIT	Just In Time.
KNN	K nearest neighbor.
MID	Mobile Internet Device.
MMS	Multimedia Message Service.
NN	Neural Networks.
OS	operating system.
PDA	Personal Digital Assistant.
RAM	Random Access Memory.
RIM	Research In Motion.
RNN	réseau Neuronal Récurrent.

SE	Système Exploitation.
SDK	Software Development Kit.
SMS	Short Message Service.
SVM	Support Vector Machine.
TUANDROMD	Tezpur University Android Malware Dataset.
TP	True Positive.
TN	True Negative.
UID	User Identifier.
USB	Universal Serial Bus.
VPN	Réseau Privé Virtuel.

Contexte et problématique

Les technologies de l'information et de la communication ont été la révolution la plus importante et la plus innovante de ces dernières décennies. En fait, ces technologies sont loin d'être un phénomène volatile et nous ont réconfortés dans la vie quotidienne car elles sont capables de traiter les informations dans un délai raisonnable. Cette révolution a permis l'émergence du concept de transférabilité et de mobilité, qui permet un accès à distance et immédiat et un flux ininterrompu d'informations. Cela est en fait symbolisé par la présence de différents appareils de haute technologie, tels que les Smartphones et les tablettes, qui ont des applications pratiques différentes, indépendamment du fait que : gratuit ou payé.

Les applications mobiles sont devenues un outil important dans différents domaines, elles sont connectées avec les sites web, les systèmes d'information et les services de stockage de données dans les nuages. Android est le système d'exploitation mobile le plus populaire installé sur des millions d'appareils (Smartphones et les tablettes, téléviseurs, et montres intelligentes).
Ces nouvelles tendances permettent le partage des ressources et l'échange des informations confidentielles sur les messageries électroniques. Ces effets sont accompagnés par des tentatives de violations des politiques de sécurité par des accès malveillants. Les applications Android mobile sont également très exposées aux attaques et accès malveillants.

Android est un système d'exploitation séparé de privilèges où chaque application a sa propre identité système, à savoir, Group-ID et Linux user-ID. Chaque application d'Android s'exécute dans un sandbox de procédure et accède aux autorisations d'utiliser les ressources qui ne sont pas présentes dans son sandbox. Selon la sensibilité des autorisations, le système accorde automatiquement ou peut inviter les utilisateurs à approuver ou à rejeter les demandes d'autorisation. En profitant de ces autorisations, les cybercriminels ciblent la vie privée des utilisateurs. Comme indiqué dans, G-Data Security expert dénombré $3,246,284$ applications malveillantes jusqu'à la fin de l'année 2018 et a découvert plus de $7,50,000$ nouvelles applications malveillantes à la fin de 2019.

La protection contre les attaques, les dommages ou l'accès non autorisé nécessite un antivirus ou un autre outil pour détecter et protéger les logiciels malveillants, tels que Windows Defense. La reconnaissance des logiciels malveillants est un problème binaire qui indique si un programme est un logiciel malveillant ou non.

Plusieurs méthodes ont été proposées pour contrer les attaques contre les logiciels malveillants. Ceux-ci sont basés sur un certain nombre de techniques qui peuvent être classées comme suit : l'analyse de données d'appel automatique et le contrôle flots, la similitude de code et la recherche de signature dans le code.

L'objectif de travail

Comme il à été mentionné, plusieurs techniques et approches ont été introduites pour comprendre, analyser, reconnaître et classer les logiciels malveillants Android, mais la menace est constamment effervescences ; Android doit encore faire face à de nombreux défis et problèmes de sécurité.

Le but de notre recherche est de proposer une approche basée sur les des algorithmes de fouille de données classiques pour la détection des logiciels malveillants sur la plateforme Android. Ce travail à été réalisé en deux parties :

- Dans la première nous avons appliqué quelques algorithmes de fouille de données et un algorithme d'apprentissage profond ensuite nous avons comparé les résultats.

- La deuxième étape consiste à tester une hybridation de deux algorithmes de fouille de données.

Organisation du mémoire

Nous allons détailler le projet dans ce rapport sur trois chapitres : **Chapitre** 1 **les systèmes d'exploitations mobiles.**
Ce chapitre présent les différents systèmes d'exploitation mobiles d'une manière générale, et Android en particulier, nous présentons aussi des statistiques sur les OS existants sur le marché ainsi que l'architecture de système Android, ces versions et ces outils.
Chapitre 2 **La sécurité des applications mobiles**
Ce chapitre présente la sécurité des applications mobiles et le modèle de sécurité Android et ces caractéristiques. Ainsi que les autorisations des applications Android, Les menaces à la sécurité des applications mobiles et les travaux de sécurité Android contre les malices qui détecter les mobiles.
Chapitre 3 **: Contribution et Implémentation**
Nous montrons dans ce chapitre une description générale sur les data set utilisées, les différents résultats obtenus ainsi que l'outil développé.

Chapitre 1

Les systèmes d'exploitation mobiles

Introduction :

Dans ce chapitre, nous aborderons l'importance des systèmes d'exploitation qui sont efficaces dans l'espace médiatique automatisé et indispensable et qui nous aident à utiliser et à découvrir plusieurs applications technologiques modernes et sophistiquées dans le téléphone mobile. Différentes entreprises ont donc lancé des plateformes pour développer ces systèmes. Ces plateformes sont : Symbian OS de Nokia, iOS de Appel's, BlackBerry OS de RIM's, Windows Phone de Microsoft, Ubuntu MID Edition d'Ubuntu, Bada de Samsung, Palm, webOS et Android de Google. En outre le système Android et son architectures les versions de cette système avec ces outils.

1.1 Les systèmes d'exploitation mobiles

1.1.1 Symbian OS :

Symbian OS est un système d'exploitation mobile développé par Symbian pour le système de messagerie. Il existe plusieurs technologies telles que Blackberry via BlackBerry Connect et Microsoft via Exchange ActiveSync support Symbian. Sur le marché des systèmes d'exploitation pour Smartphones, Symbian occupe une position très importante et démontre qu'il a fait de Nokia l'un des plus grands Smartphones au monde et qu'il a contribué à sa contribution la plus importante. Cette réduction du marché du système

d'exploitation a conduit Nokia, en janvier 2008, à envisager des plans pour améliorer le niveau de Symbian en tant que plate-forme open source. Selon le plan Symbian-Launch, le code source sera publié en juin 2010 et publié sous la licence "Licence Publique Eclipse 1.0". Symbian offre un support de niveau OS pour la plupart des fonctionnalités de Palm, Windows Mobile et Blackberry. C'est une plate-forme flexible qui permet aux développeurs d'ajouter leurs technologies et infrastructures à la plate-forme Symbian. En outre, il est soutenu par de grandes entreprises de téléphonie mobile telles que Nokia, Sony Ericsson, Motorola [1].

FIGURE 1.1 – Figure 1.1 : Logo du système Symbian OS.[6].

1.1.2 iOS :

iOS est un système d'exploitation mobile développé par Apple uniquement pour ses propres appareils mobiles (iPhone, iPad et iPod touch). Connu pour sa fluidité, son ergonomie et son intuition, c'est le système d'exploitation le plus performant à ce jour. Il dispose du portail App Store, qui a été établi avec un catalogue de plus de 200.000 applications comme référence dans les kiosques d'applications mobiles. Ce système iOS utilise le langage Objective-C. En 1976, Apple a publié la première version de son système d'exploitation après 2008, appelé iPhone OS (iOS1). L'iPhone 3G et iPod (Touch Release) sont des transitions vers la deuxième version (iOS 2). Certains changements, tels que la découpe, la copie, l'adhésif et l'intégration chimique, ont été introduits dans la troisième version (iOS 3) en 2009.Après 2010, Apple a apporté la quatrième version (iOS4) avec de nouvelles fonctionnalités en multitâche et en Face- time. La cinquième version indique la présence d'un centre de messagerie, pas de changements majeurs dans la sixième version. iOS 7, publié en 2013, introduit un nouveau style de conception de l'interface graphique caractérisé par l'utilisation de formes simples, des couleurs plus vives

et le respect du principe d'un site Web adaptatif 1. Parmi les nouvelles fonctionnalités de l'iOS 8 présenté en 2014, nous mentionnons : libération familiale et un nouveau design pour l'App Store. En 2015, iOS propose neuf nouvelles versions pour ses applications existantes (notes, plans, carnet de passeports, etc.), offre de nouveaux modes de multitâche et permet une autonomie de plus en plus rapide. L'iOS 10 annoncé en 2016 offre, entre autres, un kit de développement Siri 2 pour les développeurs et un nouveau design. La première version 64 bits est iOS 11, qui contient une application de gestion de fichiers pour accéder directement aux fichiers stockés localement et à iCloud 3 la dernière version [2].

FIGURE 1.2 – Figure 1.2 : Logo du système iOS.[7]

1.1.3 BlackBerry OS :

BlackBerry OS est un système d'exploitation mobile développé par la société canadienne Research In Motion (RIM) pour sa série de Smartphones Blackberry. La plateforme BlackBerry est populaire auprès des utilisateurs professionnels, car dans certaines entreprises, il offre la synchronisation avec Microsoft Exchange et d'autres programmes professionnels très importants. Les solutions RIM intègrent du matériel, des logiciels et des services qui permettent un accès facile et transparent à des informations importantes telles que le courrier électronique, le téléphone, les SMS et les applications Internet et intranet. Les produits, services et technologies RIM ont reçu de nombreux prix et sont utilisés par des milliers d'entreprises du monde entier. Avec le temps, Blackberry OS est devenu agréable et de plus en plus grand. Le système est plein de bonnes idées, à commencer par la possibilité de séparer votre lieu de travail de votre vie personnelle sur un téléphone. Blackberry prend également en charge la technologie de remorquage de différents types, tels que les fichiers avec des extensions zip, html, doc, dot, ppt, pdf, etc. Équipés d'un

bouton, clavier QWERTY et interface utilisateur, les Blackberry sont d'excellents appareils pour faciliter l'utilisation du système de messagerie. Maintenant, Blackberry est de plus en plus populaire dans le monde et offre plusieurs fonctions au grand public, telles que le multimédia (appareil photo, appareil photo, etc.) [3].

FIGURE 1.3 – Figure 1.3 : Logo du système BlackBerry OS.[8]

1.1.4 Windows Phone :

Windows Phone est un système d'exploitation développé par Microsoft pour ses appareils mobiles : Smartphones (Lumia, Asha, etc.), tablettes (Microsoft Surface) et PDA (Cortana). C'est une plateforme privée, pas open source. En 1975, Microsoft a créé sa propre entreprise et s'est spécialisée dans le développement de systèmes d'exploitation (Windows 3, Windows 95,..., Windows 10) et de logiciels (MS Office...). En 2001, Microsoft a introduit son premier système d'exploitation mobile appelé Windows Mobile, qui a été remplacé en 2010 par Windows Phone, en introduisant une toute nouvelle interface pour les terminaux tactiles. Entre 2010 et 2015, Windows Phone a fait l'objet de plusieurs mises à jour importantes, y compris la transition de Windows CE-Kernel (Windows Phone 7) à Windows NT-Kernel (Windows Phone 8) et l'introduction de Windows Cortana 8.1. À partir de novembre 2015, Windows Phone disparaîtra progressivement et sera remplacé par Windows 10 Mobile. Il existe maintenant plusieurs appareils dans le monde sous Windows Mobile qui fonctionnent comme une paire de modèles Motorola, Samsung, T-Mobile, etc. [4].

FIGURE 1.4 – Figure 1.4 : Logo du système Windows phone.[9]

1.1.5 Ubuntu MID :

Ubuntu MID3 Edition est un système open source développé par la communauté Ubuntu parrainé par Canonical 4 et Intel. Les Mids (Mobile Internet Device) peuvent être des appareils mobiles multimédias avec ou sans fonctions téléphoniques créées sous l'idée "Mini - Laptop". Comme Android, Ubuntu MID Edition est en sécurité grâce au noyau Linux. Il est très flexible, adaptable et facile à utiliser. Il prend également en charge le Web 2.0 variantes d'applications comme les navigateurs Web, e-mail, appareil photo, Voip, messagerie instantanée, GPS, blog, télévision, jeux électroniques, etc. Pour la technologie de communication, il prend en charge le Wifi, Bluetooth, GPS et Wimax. Ubuntu MID Edition est maintenant utilisé sur certains modèles de périphériques utilisant une plate-forme Intel Atom, une tablette Nokia N800 Web, etc. Finalement, Ubuntu MID Edition contient les mêmes services Internet que ceux trouvés sur les ordinateurs, mais ils sont généralement utilisés pour Mids et non pour les téléphones mobiles [5]

FIGURE 1.5 – Figure 1.5 : Logo du système Ubuntu MID.[10]

1.1.6 Bada OS :

Bada est un système d'exploitation propriétaire pour les Smartphones Samsung. Il a été développé au début de 2010, après avoir été présenté la même année au Congrès mondial mobile de Barcelone. Le Bada peut fonctionner de différentes manières selon

l'appareil dans lequel il est utilisé [5]. En 2011, Bada été le troisième SE mobile en part de marché en France et équipe plus d'un million de Smartphones. En 2013, Samsung a annoncé qu'il arrêterait le développement de Bada pour se concentrer sur le développement d'une nouvelle plate-forme appelée Tizen [5].

FIGURE 1.6 – Figure 1.6 : Logo du système Bada OS.[11]

1.1.7 Palm OS :

Palm OS est le système d'exploitation développé par la société Palm, il est facile à utiliser et à apprendre. Palm OS est passé de versions de groupes de travail simples (5.0 et supérieures) à des versions multitâches (6.0). Cette nouvelle version est basée sur des versions précédentes. Elle offre diverses améliorations, telles que la communication et les médias[6].

FIGURE 1.7 – Figure 1.7 : Logo du système Palm OS.[12]

1.1.8 WebOS :

Webos est un système d'exploitation open source basé sur un noyau Linux développé comme la dernière version de Palm. Il est très différent des versions précédentes, il est basé sur des applications avec HTML, Javascript et CSS. Une application Webos est lancée par un utilisateur ou une autre application. Il prend en charge le multitâche, mais peut être conçu pour fonctionner uniquement en arrière-plan. Cette application de fond interagit avec l'utilisateur principalement en affichant des alertes et des notifications.

FIGURE 1.8 – Figure 1.8 : Logo du système webOS.[13]

1.1.9 Android :

Android est un système d'exploitation open source pour les téléphones mobiles de nouvelle génération, Pads et autres appareils mobiles (tablettes). Il a été développé en 2007 par Android, une start-up achetée par Google, basée sur un noyau Linux 2.6. Android contient tous les services nécessaires pour un fabricant de les distribuer sur un téléphone mobile. Il est offert à tous les fabricants de téléphones mobiles pour faciliter leur adoption. Il est conçu pour intégrer des applications Google telles que Gmail, Google $Map2^{+}$, Google Agenda, Google Talk ou You Tube. Android est un système d'exploitation libre et entièrement ouvert, ce qui signifie que le code source et les API sont ouverts. Par exemple, les fabricants de matériel peuvent ajouter et adapter leurs propres extensions propriétaires à Android pour distinguer leurs produits des autres. Android est un système presque équivalent à l'IOS, tant dans les fonctions que dans les applications. Là où Apple recherche la cohérence et la sécurité, Android offre plus d'options et de paramètres. Pour développer une application mobile sur ce support, le langage de programmation utilisé sera Java. En d'autres termes, les outils de développement utilisés seront Android SDK. Le langage C++ peut être utilisé aussi avec le kit de développement natif du CND [2].

FIGURE 1.9 – Figure 1.9 : Logo du système Android.[14]

1.2 Comparaison des systèmes d'exploitation mobile :

Cette section présente des comparaisons de quelques SE mobiles.

1. Le tableau suivant présente une comparaison en termes de configuration, de déploiement et de constructeur.

	Ios	BlackBerry	Windows Phone	Android
Langage de programmation	Objective-C	Java	C, C++	Java
	Intégré à Xcode	Gratuit	Gratuit	Gratuit
Disponibilité de l'environnement de développement	Xcode	JDE	Visual Studio, eMbeddeb VC++	Eclipse, Netbeans, Android Studio
Multiplateforme de déploiement	iPhone, iPod touch, iPad	BlackBerry seulement	Windows Mobile, Windows CE	Android seulement
Cout d'outils de développement	Gratuit	Gratuit	Gratuit	Gratuit
Magasin enLigne	App Store	App World	Windows Market Place	AndroidMarket
Open source	Non	Oui	Non	Oui
Constructeur	Apple	RIM	Microsoft	Google

TABLE 1.1 – Une comparaison entre les systèmes d'exploitation mobile[15].

2. Une comparaison en termes de taux d'utilisation est illustrée dans le diagramme suivant :

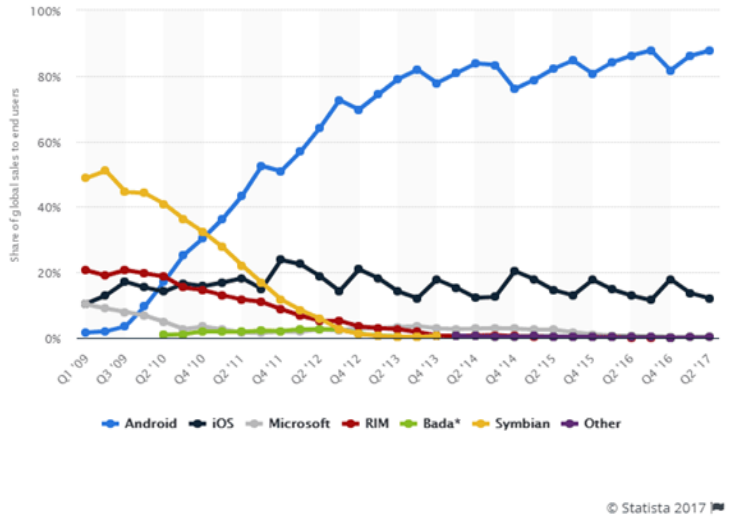

FIGURE 1.10 – Les grands OS mobile.[16]

3. **Part du marche** Dans cette section, nous présentons les statistiques publiées par Statista [1] sur les principaux système d'exploitation mobiles.

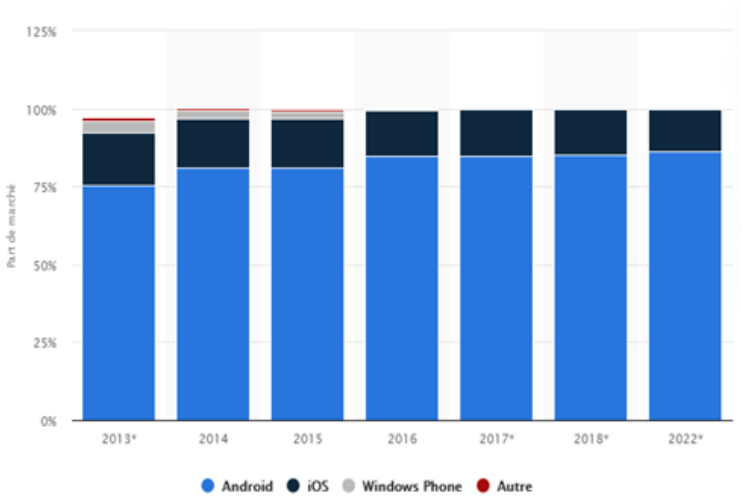

FIGURE 1.11 – Répartition des expéditions de smartphones dans le monde par système d'exploitation entre 2013 et2022.[17]

1. https ://fr.statista.com le portail de statistiques pour les données de marché

La figure 1.11 présente la part de marché détenue par les systèmes d'exploitation pour Smartphones en expéditions d'unités entre 2013 et 2022, on remarque qu'Android occupe une grande place sur le marché mondial. En 2022, le système d'exploitation Android devrait représenter environ 86% des expéditions mondiales de Smartphones, il est devenu leader du marché.

1.3 Le système d'exploitation Android

Nous avons décrit en détail les différents systèmes d'exploitation utilisés dans le monde. Dans cette partie, nous nous concentrons sur la plateforme que nous avons choisie pour développer notre application. Nous avons choisi le système d'exploitation Android pour les raisons suivantes :

- Android est une plateforme open source, puissante, moderne, sécurisée et ouverte. En ouvrant le code source et les API, les développeurs peuvent intégrer, développer et modifier des composants existants. Les utilisateurs peuvent adapter les applications à leurs besoins.
- Android est basé sur le noyau Linux. Il existe plusieurs avantages, tels que les grands magasins, la gestion des processus, le modèle de sécurité, la bibliothèque de soutien partagée, etc.

SDK Android offre une API complète pour le développement de l'application Android. Grâce à Android Develop Challenge, grâce au fait que Google offrira 10 millions de dollars de prix inconditionnels pour les applications sur la plateforme Android, les développeurs ont la possibilité de gagner beaucoup d'argent. [19]

1.3.1 La naissance d'Android :

- La création d'Android remonte en 2003, avec une société américaine appelée Android,
- Il a été acheté par Google deux ans plus tard (2005). L'objectif principal était de développer un système d'exploitation facile qui permettrait à l'utilisateur d'interagir

avec l'utilisateur. Dans le passé, chaque fabricant a développé son propre système d'étiquetage. Par conséquent, il était impossible de concevoir une application compatible pour tous les appareils, sans parler des bibliothèques de développement qui avaient été mises à disposition et qui étaient verrouillées, de sorte que les secrets de la marque n'ont pas été révélés. [19]

1.3.2 Les versions d'Android

Le système de Google n'aurait pas eu autant de succès s'il n'avait pas changé en six ans. Nous voyons ici la puissance d'un tel système d'exploitation, capable de s'adapter aux besoins des utilisateurs avec chaque version principale et enrichi de nouveautés. [19]

Nom de code	Version	Les fonctions importantes qui ajouter	Logo
Apple pie	1.0	Seulement ou presque connu des développeurs car c'est la version du SDK qui a été déployée avant la sortie du premier téléphone Android.	
Bananas split	1.1	• Beta, version sur le premier téléphone, qui est HTC G1 / Dream	
Cupcake	1.5	• Nouvelles fonctionnalités et mises à jour GUI	
Donut	1.6	• Nouvelles fonctionnalités et mises à jour GUI	
Eclair	2.0	• Nouvelles fonctionnalités et mises à jour GUI	
Froyo	2.2	• Vitesse améliorée, nouvelles fonctionnalités et mises à jour GUI	
Gingerbread	2.3	• La dernière version dédiée uniquement aux Smartphones. Cette version est parfois utilisée sur les petites tablettes	
Marshmallow	6.0	• Supports USB de type C. • Support pour l'authentification des empreintes digitales. • Meilleure autonomie de l'accumulateur en mode sommeil profond. • Panneau de commande pour contrôler les autorisations d'utilisation.	
Nougat	7.0	• Meilleur support multitâche. • Comptage multiple. • Améliorations des systèmes (par le biais de la double partition du système). • Capacité de code améliorée et performance avec un nouveau compilateur JIT.	
Nougat	7.1	• Éclairage en mode nuit. • Gestionnaire de mémoire amélioré. • Meilleure gestion de l'écran et du toucher. • Option pour activer les gestes sur le capteur d'empreintes digitales. • Mises à jour du système "transparentes".	

Oreo	8.0	• Mode Picture in Picture. • Gestion multitâche et démarrage rapide. • Gestion des notifications. • Les API d'auto complétion.	
Oreo	8.1	• Ajout de l'affichage du pourcentage de batterie d'un appareil Bluetooth dans les paramètres (et paramètres rapides) du Bluetooth. • Ajout de l'affichage des vitesses de transmission des réseaux Wifi disponibles. • Ajout de l'API « Neural Networks » (NNAPI) afin de permettre à des opérations hors-ligne de machine learning d'utiliser l'accélération matérielle d'un appareil • Amélioration des performances pour les appareils à faible RAM.	
Pie	9.0	• Nouvelle navigation système à gestes, elle introduit des déférents systèmes intelligents comme la gestion de luminosité, économiseur de la batterie.	
Pas de Nom	10	• Le système d'autorisations est mise à jour. • Offre un bon contrôle sur les autorisations des applications sur vos données.	android
Pas de Nom	11	• - Google se localise sur la confidentialité, surtout sur la mise à jour des autorisations sur un accès limité sur l'appareille photo, le microphone et l'emplacement de l'appareil, seules les applications légitimes selon Google peuvent accéder aux emplacements des appareilles d'utilisateurs.	android 11

TABLE 1.2 – les versions Android. [19] Ce tableau présente tout les versions Android à partir 2011 à 2020

.

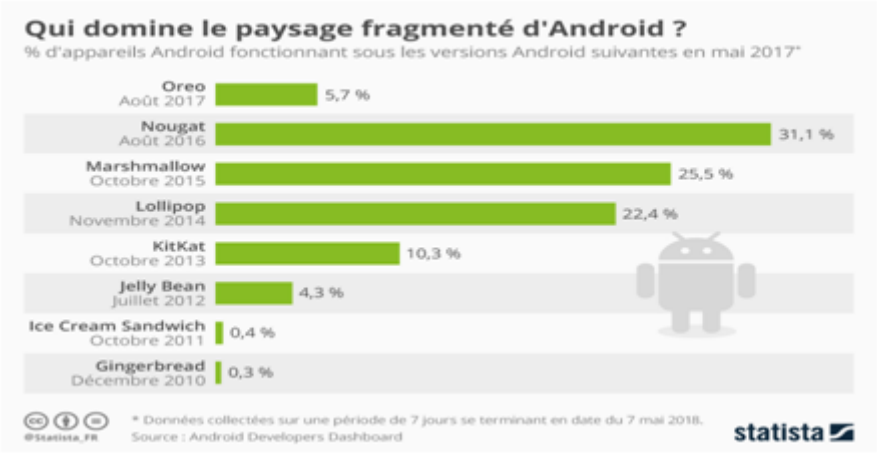

FIGURE 1.12 – les versions Android les plus utiliser dans le monde 2018. [20]

La Figure 1.12 nous donnons les dernières statistiques qui datent du 7 Mai 2018 concernant la répartition des différentes versions Android.

1.3.3 Architecture logicielle Android

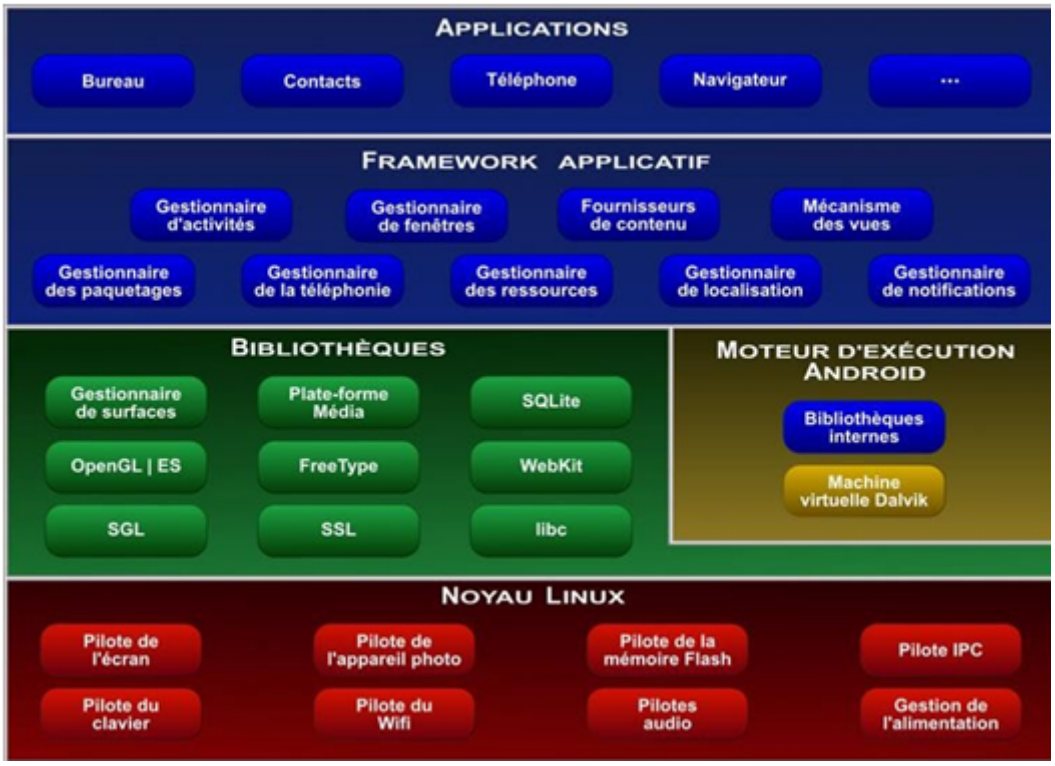

FIGURE 1.13 – Architecture logiciel Android. [19]

Le système Android est représenté comme un système d'exploitation en temps réel avec différents niveaux tels que le niveau des applications, le niveau des applications Framework, le niveau des Libraries, Linux Kernel, Moteur d'exécution Android :

- Le niveau "applications" comprend toutes les applications téléphoniques telles que les navigateurs, les contacts, les appareils photo, les calendriers..., c'est le plus haut niveau. [21]
- Le niveau "applications Framework" c'est le cadre d'application en français Il comprend un certain nombre de classes et de services qui permettent aux applications d'accéder aux données. Les services sont disponibles en tant que gestionnaire de tâches, gestionnaire de ressources, gestionnaire d'alertes...etc [21]
- Le niveau "Libraries" en français est le niveau de la bibliothèque, qui représente le

bas niveau de logiciel, qui contient plusieurs bibliothèques en C / C + + langue, bibliothèque web, bibliothèque de formats multimédias (image, audio, vidéo) ...etc [21]

- La couche "Linux Kernel" en français la couche de base de Linux, est la couche la plus importante de l'architecture Android, est au coeur de cette architecture, car il fournit des services importants avec le matériel logiciel, ces services comprennent la sécurité, la gestion de l'énergie, la gestion du stockage...etc
- La couche " Moteur d'exécution Android " en français, la couche Android Runtime, contient la Dalvik Virtual Machine (DVM) et les comptoirs principaux. [21]

1.3.4 Kit de développement ou SDK

Le SDK est un acronyme anglais pour l'outil de développement logiciel. Le kit de développement Android est une plate-forme avec laquelle les développeurs peuvent programmer des applications Android mobiles. Il contient un ensemble de projets, de codes source, d'API et de documents. Les applications sont écrites dans le langage de programmation Java et exécutées sur la machine virtuelle Dalvik. Une application Android est regroupée dans un package Android sous l'extension APK. [22]

Conclusion

Grâce à l'étude, on peut soutenir que, malgré la multiplicité des systèmes d'exploitation, Android a balayé l'arène et devenir le leader mondial des systèmes de téléphonie mobile. La facilité, la flexibilité et l'ouverture du système Android en fait une gamme d'attraction pour tous les utilisateurs.

Chapitre 2

La sécurité des applications Android

Introduction

Les développeurs comprennent l'importance de la sécurité des applications mobiles, mais n'est pas universellement compris.

Au-delà d'un taux croissant de fraude mobile, les institutions financières devraient prendre au sérieux la sécurité des applications mobiles et s'engager à élaborer une stratégie globale. Les consommateurs doivent se méfier de l'information qu'ils divulguent et des données qu'ils téléchargent lorsqu'ils naviguent sur Internet, mais les professionnels des affaires doivent aussi être vigilants. Les appareils mobiles sont presque toujours allumés, toujours à proximité des utilisateurs, et stockent des quantités stupéfiantes de renseignements personnels ainsi que des données et des documents sensibles. Cela peut en faire un trésor pour les attaquants. Les applications mobiles peuvent être présomptueuses dans les autorisations qu'elles demandent.

Pour analyser et reconnaître les applications malveillantes sur la plate-forme Android, il est important de comprendre comment les applications Android fonctionnent et d'étudier les différentes techniques de détection disponibles. Nous avons présenté le système Android, son architecture et les différentes composantes des applications dans la chapitre I. Ce chapitre analyse la sécurité des applications mobiles Android, les secteurs d'attaque utilisés par les auteurs de logiciels malveillants pour infecter les appareils mobiles

et identifie différentes techniques d'analyse des applications Android.

2.1 La sécurité des applications mobiles

La sécurité des applications mobiles consiste à protéger les applications mobiles de grande valeur et l'identité numérique contre toute attaque frauduleuse sous toutes ses formes. Cela comprend la falsification, la rétro ingénierie, les logiciels malveillants, les enregistreurs de frappe et d'autres formes de manipulation ou d'interférence. La sécurité des applications mobiles a rapidement pris de l'importance, car les appareils mobiles ont proliféré dans de nombreux pays et régions. [13]

2.2 Les caractéristiques de sécurité mobiles Android

Les applications de sécurité mobiles pour la plateforme Android de Google aident à protéger les Smartphones et tablettes Android contre les menaces de logiciels malveillants ainsi que l'accès non autorisé suite à la perte ou au vol accidentel de l'appareil. Les autres caractéristiques de sécurité fréquemment offertes par les applications de sécurité mobiles Android comprennent :

- La sécurisation des données sur l'appareil,
- La connectivité VPN pour protéger les données en transit,
- La numérisation des sites Web pour détecter d'éventuels stratagèmes d'hameçonnage ou d'autres activités frauduleuses.
- La localisation : aide l'utilisateur de localiser son appareil s'il est perdu ou volé. Les applications de sécurité mobiles Android sont disponibles auprès de Google ainsi que des fournisseurs de sécurité tiers bien connus tels que Lookout, Avast, Kaspersky, Symantec et Qihu. [14]

2.3 Menaces de sécurité des applications mobiles

Les menaces en sécurité des applications mobiles se produisent principalement sous la forme de la distribution de code malveillant qui exploite les vulnérabilités du système d'exploitation et des applications par des méthodes d'ingénierie sociale [1] telles que le courrier indésirable et les messages SMS, ce qui entraîne une fuite de renseignements personnels ou un gain financier. [15]

La Menace	Explication
Malware	Menaces installées sur le terminal pour comportement malveillant.
Spam	Menaces utilisées pour distribuer des publicités et des logiciels malveillants qui peuvent être envoyés à un nombre indéterminé de personnes.
Application vulnérabilités	Menaces qui exécutent des actions malveillantes telles que l'élévation des privilèges en utilisant la vulnérabilité de l'application développée.
Informations personnelles extrusion	Menace de fuite d'informations personnelles en raison de la négligence de l'utilisateur lors du développement applications installées.
Contournement d'authentification	Menaces qui contournent ou volent l'authentification au hasard pour les applications qui nécessitent une authentification.
DoS	Menaces qui rendent le service fourni par l'application inutilisable

TABLE 2.1 – Menaces à la sécurité dans les applications mobiles [15]

Les menaces de sécurité peuvent conduire à une vulnérabilité dans la plupart des applications. Les vulnérabilités dans l'application peuvent contourner l'authentification et afficher des renseignements non autorisés. Comme il est possible de prendre des actions malveillantes telles que le téléchargement et l'installation de certaines applications avec l'autorité au niveau du système en enracinant et emprisonnant le terminal par la force en exploitant une vulnérabilité déjà annoncée. En outre, re-packaging, l'une des techniques

1. L'ingénierie sociale est un mot-parapluie qui désigne une variété de méthodes et techniques employées par les pirates et autres cybercriminels cherchant à tromper les victimes innocentes et leur faire partager leurs données personnelles, ouvrir des liens vers des sites web infectés ou, sans le savoir, permettre aux pirates d'installer des maliciels sur leurs ordinateurs. [https ://softwarelab.org/fr/ingenierie-sociale/]

utilisées par les codes malveillants, modifie les applications normales pour ajouter des fonctions malveillantes, ou supprime les fonctions de sécurité et les redistribue de sorte que de nombreux utilisateurs installent des applications avec des codes malveillants sans soupçon. [15]

2.4 La vérification des vulnérabilités :

La Korea Internet & Security Agency a distribué des guides pour vérifier la vulnérabilité de sécurité des applications mobiles dans le cadre du système de vérification de la sécurité de l'application de service e-gouvernement mobile afin de fournir un service qui effectue des contrôles de vulnérabilité de code source pour les applications mobiles développées par l'État et les institutions publiques [15].

Menaces de sécurité	Explication
Octroi du moindre privilege	Accorde uniquement les privilèges minimaux requis pour le fonctionnement de la fonction.
Validation des entrées externs	Lorsqu'une fonction est utilisée sur la base d'informations d'entrée externes, la validité de l'information d'entrée est vérifiée pour voir si la longueur spécifiée est dépassée et le code malveillant est inclus.
Gestion sécuritaire des informations sensibles	Confirmation du chiffrement lors du stockage et de la transmission de renseignements importants (renseignements personnels, localisation personnelle, renseignements commerciaux, etc.).
Confirmation de la violation de modèle de sécurité de plateforme mobile	Confirmation de l'existence de fonctions de modulation de plateforme telles que rooting et jailbreak [2].
Identification des vulnérabilités de sécurité du code source	Confirmation de l'existence de vulnérabilités connexes selon la classification des vulnérabilités du code source telles que la vérification et l'expression des données d'entrée, l'utilisation abusive des API et les caractéristiques de sécurité.
Assurance de la sécurité pour modules commerciaux et publics.	Vérification de l'adéquation de la finalité et des fonctions commerciales ou modules publics.
Vérification de l'utilisation des fonctions communes de sécurité de l'infrastructure	Confirmation de l'utilisation des fonctions fournies par l'infrastructure de sécurité établie selon l'infrastructure commune.
Confirmation de l'existence	Vérification de l'existence de vulnérabilités de sécurité connues.

TABLE 2.2 – Guide de vérification des vulnérabilités liées à la sécurité des applications mobiles [15]

2.5 Les autorisations des applications Android

Android dispose du système d'habilitation depuis le lancement du 6.0 appelé Marshmallow, l'utilisateur vérifie les autorisations des applications. Une autorisation d'application définie l'accès aux données trouvées dans le téléphone portable, par exemple l'accès aux contacts, aux fichiers multimédias, à l'emplacement, à la caméra ou au microphone. Les applications ne peuvent pas accéder aux données automatiquement. L'utilisateur doit accorder l'accès et l'autorisation pour les données que l'application a l'intention d'utiliser. Une application peut tomber en cas de mise à jour manquante ou ne pas fonctionner correctement en raison d'un rejet d'autorisations. Le but de l'utilisation des autorisations est de s'assurer l'intégrité des données de l'utilisateur, c'est-à-dire le refus d'accorder des autorisations pour des applications douteuses, est essentielle pour protéger les données contre les attaques malveillantes. L'utilisateur soit toujours lire les spécifications avant de les installer en consultant leur description dans le Play Store. Il doit toujours connaître les autorisations nécessaires pour chaque application, par exemple pour une application de messagerie, il a besoin de contacts et d'autorisations SMS, mais il n'a pas besoin d'avoir accès aux données de santé. [16]

2.5.1 Quelques permission Android

- **Calendrier :** cette autorisation permet de modifier, créer, lire ou supprimer un événement dans le calendrier utilisateur. [16]
- **Appareil photo :** avec cette autorisation, on peut enregistrer des photos et des vidéos. [26]
- **Contacts :** avec cette autorisation, on peut accéder à la liste de contacts et modifier, ajouter ou supprimer un contact. [16]
- **Localisation :** Cette autorisation on peut 'accéder à la position avec le GPS pour un haut niveau de sécurité de détection ou avec le WI-FI ou les données de téléphonie mobile pour une détection approximative. [16]
- **Microphone :** avec cette autorisation, on peut enregistrer du son (note vocale).

[16]

- **Téléphone :** Cette autorisation permet d'accéder aux données d'appel, permet le détournement d'appel, la modification des journaux d'appels, l'accès à la messagerie vocale... [16]
- **SMS :** avec cette autorisation, on peut lire, écrire ou envoyer des SMS ou des MMS. [16]
- **Stockage :** cette autorisation nous permet de lire, d'écrire des fichiers dans la mémoire interne ou externe de votre appareil mobile. [16]

FIGURE 2.1 – Exemple des Permissions pour l'application whatApp. [17]

2.6 Modèle de sécurité Android

Le modèle de sécurité Android n'a pratiquement pas évolué depuis le début et a déjà été largement commenté. Cependant, son étude reste une condition indispensable pour toute analyse des risques sur cette plateforme. [18]

2.6.1 Signature numérique

En pratique, tous les systèmes de signature numérique récemment apparus sur les marchés grand public (par exemple, le programme Symbian Signed, la signature de l'application iPhone, la signature du pilote 64 bits de Windows) n'offrent aucune sécurité a priori. C'est un procédé technique et juridique permettant à des individus d'apporter consentement et approbation à des documents numériques. Les coûts d'obtention d'un certificat sont marginaux et les audits effectués sont pratiquement nuls. Le certificat sert essentiellement d'identifiant unique pour la révocation ultérieure. Pour éviter les problèmes de procédure et d'extension de clé, Google impose l'utilisation de certificats de signature dont la date d'expiration est supérieure au 22 octobre 2033. [18]

2.6.2 Cloisonnement

Selon ANSSI[3], *" Un mécanisme de cloisonnement permet de compartimenter un environnement d'exécution en plusieurs parties ne comportant pas les mêmes éléments et ne bénéficiant ni des mêmes droits ni des mêmes ressources. Intuitivement, il s'agit de découper un environnement monolithique à la manière d'un puzzle, sans impact sur le service rendu. L'avantage d'une telle démarche tient alors dans la possibilité de restreindre chaque partie de l'environnement aux actions dont elle a besoin. En d'autres termes, l'intérêt du découpage découle de l'application du principe de moindre privilège sur chaque sous-partie de l'environnement. Une fois ceci mis en oeuvre, la compromission d'une sous-partie devient plus difficile car sa surface d'attaque est réduite. De plus, une corruption ne peut avoir que des conséquences limitées. ". [18]*

2.7 Les travaux existants

La plateforme Android a évolué rapidement, c'est le système d'exploitation le plus populaire et est utilisée le plus souvent dans de nombreux appareils. Cette croissance rapide en a fait la cible de nombreuses applications malveillantes qui tentent de voler des

3. Agence Nationale de la Sécurité des Système Informatique

informations et des données sensibles. Plusieurs travaux on été proposés pour la détection de logiciels malveillants sur les appareils mobiles. Des approches surveiller l'utilisation de la puissance des applications et faire rapport consommation anormale. D'autres surveillent les appels système et tenter de détecter des schémas d'appel de système inhabituels. Autre les approches utilisent une comparaison plus traditionnelle avec des logiciels malveillants connus ou d'autres heuristiques. Le domaine plus général de la détection des logiciels malveillants est l'hôte d'un plus large éventail d'approches :

- Les approches d'analyse statique traditionnelles telles que, qui se concentrent sur la comparaison des programmes aux logiciels malveillants connus basés sur le code du programme, la recherche des signatures ou en utilisant des heuristiques.
- Autres approches, mettre l'accent sur l'apprentissage automatique et l'exploration de données approches de détection de logiciels malveillants. [19]

Nous présentons dans ce qui suit quelques travaux de détection de malice Android [20] [21] [22] [23] :

- **Tesauro et al :** Train un réseau neuronal pour détecter les virus du secteur de démarrage, basé sur l'octet trigrammes de cordes.
- **Schultz et al :** Compare trois machines algorithmes d'apprentissage formés sur trois caractéristiques : DLL (dynamic link library), et système appels effectués par le programme, chaînes de caractères trouvées dans le programme binaire, et une représentation hexadécimale brute du binaire.
- Kolter et Maloof : Forme plusieurs machine learning ,algorithmes sur la chaîne d'octets n-grammes.
- **Justin Sahs :** a introduit un système qui ap-Sahs pour la préparation de commandes automatique pour identifier les applications malveillantes sur les appareils Android, son système extrait un certain nombre de fonctions et pousse un SVM dans une classe hors ligne (périphérique hors ligne) pour exploiter la plus grande puissance de calcul d'un serveur ou d'un cluster de serveurs.
- **Asaf Shabtai :** a présenté un cadre pour la reconnaissance des applications malveillantes sur les appareils Mobiles Android Malware basé sur l'hôte qui surveille

constamment diverses fonctions et événements reçus par l'appareil mobile, puis applique des détecteurs automatiques d'erreurs d'apprentissage pour classer les données collectées comme normales (bénignes) ou anormales (malignes).

- **Naser Peiravian :** a suggéré de combiner les autorisations et les appels API, en utilisant des méthodes d'apprentissage automatique pour identifier les applications Android malveillantes.
- **Zhenlong Yuan :** a proposé une méthode AA qui offre plus de 200 fonctions extraites à la fois de l'analyse statique et de l'analyse dynamique de l'application Android aux logiciels malveillants. La comparaison des résultats de modélisation montre que la technologie Deep Learning est particulièrement adaptée à la reconnaissance des logiciels malveillants Android et peut atteindre un taux de réussite élevé de 96% avec de véritables ensembles d'applications Android [24]
- **Gianluca Dini :** décrit un détecteur d'anomalies à plusieurs niveaux pour Android. Ce détecteur surveille également Android de manière centralisée et au niveau de l'utilisateur pour détecter les véritables infections de logiciels malveillants à l'aide de techniques d'apprentissage machine, afin de distinguer les comportements standard des comportements malveillants [25].
- **Jaemin Jung :** a proposé une méthode de malware-machine-learning qui identifie le sous-ensemble de l'API Android, qui est e ?cace comme des fonctions, et classe les applications Android comme des applications bénignes ou malignes. La méthodologie proposée repose d'abord sur deux API Android populaires, l'une avec des applications bénignes et l'autre avec des applications malveillantes, puis applique l'algorithme Random Forest à un ensemble de données en utilisant chaque liste comme caractéristique de classe, pour évaluer la méthodologie proposée. [26]
- **Shaikh Bushra Almin :** a proposé un système pour la reconnaissance et la suppression des logiciels malveillants sur l'appareil Android de l'utilisateur.[27]
- **Chenglin Li :** a proposé une nouvelle classi ?eur très puissant pour la reconnaissance de logiciels malveillants Android basée sur l'architecture machine permettant d'affaisser et d'extraire les fonctions des applications Android à partir de manifeste

et de code source [28].

- **Hossein Fereidooni :** a proposé un système d'identification des applications Android malveillantes, soit par une analyse statique du comportement de l'application, soit par une couverture plus large des comportements en matière de sécurité, suivi d'un système d'identification de la capacité et d'une utilisation acceptable du système de vérification positive. [29]
- **Tieming Chen :** a proposé un nouveau modèle Light Static Detection (Tinydroid) avec des instructions simples et l'apprentissage automatique, et un mètre Classifim est formé pour la reconnaissance des logiciels malveillants et les tâches classification. [30]

Conclusion

Dans ce chapitre, nous découvrons que la sécurité des Android est très importante, car elle protège le téléphone contre les virus et les cyberattaques. Les informaticiens ont étudié le développement de logiciels de téléphonie et ont trouvé des solutions pour protéger l'utilisateur contre le vol de ses propres informations.

Chapitre 3

Contribution

Introduction :

Dans ce chapitre nous allons analyser et détecter les logiciels malveillants. La première partie de ce travail consiste à utiliser quelques algorithmes de fouille de données (naïve base, arbre décision, KNN et SVM) et les algorithmes d'apprentissage profond. Dans la deuxième partie nous présentons une introduction à une nouvelle approche de détection des malices Android. L'objectif est de faire des hybridations des algorithmes et des approches existants afin d'améliorer les performances du détecteur. Nous terminons ce chapitre par une comparaison des résultats obtenus.

3.1 Description des data sets

Nous avons utilisé deux corpus TUANDROMD et Android malware dataset for machine learning 2.

– TUANDROMD (Tezpur University Android Malware Dataset) Data Set : ce corpus contient 241 colonnes et 4465 ligne la dernière contient le type d'application Android, les autres 240 colonnes contiennent des différents types d'autorisations. Les applications malware prennent la valeur "1", et les applications normales prennent la valeur "0". Ce corpus a été créé par orah, Parthajit, D. K. Bhattacharyya, and J.

K. Kalita grâce à leurs recherches en 'Malware Dataset Generation and Evaluation.' 2020 IEEE 4th Conference on Information & Communication Technology (CICT). IEEE, 2020.. [32]

– Android malware dataset for machine learning 2 : contient 215 attributs extraits de 15 036 applications (5 560 applications malveillantes du projet Drebin et 9 476 applications bénignes). Ce Dataset a été utilisé pour développer et évaluer une approche de fusion de classificateurs multiniveaux pour la détection de logiciels malveillants Android, publiée dans l'article IEEE Transactions on Cybernetics " DroidFusion : A Novel Multilevel Classifier Fusion Approach for Android Malware Detection ". [35]

3.2 Les mesures de performance de classifieurs

3.2.1 Matrice de confusion :

ou tableau de contingence est un moyen d'évaluation des performances d'un modèle de Machine Learning en vérifiant notamment à quelle fréquence ses prédictions sont exactes par rapport à la réalité dans des problèmes de classification. Elle présente un récapitulatif des résultats de prédictions. On classe les résultats en 4 catégories :

– **True Positive (TP) ou Vrai Positif :** la prédiction et la valeur réelle sont positives.
– **True Negative (TN) ou Vrai Négatif :** la prédiction et la valeur réelle sont négatives.
– **False Positive (FP) ou Faux Positif :** la prédiction est positive alors que la valeur réelle est négative.
– **False Negative (FN) ou Faux Négatif :** la prédiction est négative alors que la valeur réelle est positive.

		Classe estimée par le classifier	
		Application malicieuse	Application non malicieuse
Classe réelle	Application malicieuse	Vrai positif (TP)	Faux négatif (FN
	Application Non malicieuse	Faux positif (FP)	Vrai négatif (TN)

FIGURE 3.1 – Matrice de Confusions

3.2.2 Les métriques :

Plusieurs métriques peuvent être calculées à partir de la matrice de confusion afin d'en faciliter l'interprétation.

- L'exactitude (accuracy) : L'exactitude est le rapport entre le nombre d'échantillons prédits correctement et le nombre total de prédictions réalisées. Cette métrique mesure la proportion d'applications correctemnent classées par le classifieur.

$$Accuracy = \frac{TP + TN}{N}$$

  Ou $N = TP + TN + FP + FN$.
- Précision : la précision mesure le ratio des observations positives bien classées par rapport à toutes les observations positives prévues :

$$Prcision = \frac{TP}{TP + FP}$$

- Sensibilité :la sensibilité (ou rappel) mesure le rapport entre les observations positives bien classées et toutes les observations positives réelles.

$$Sencevity = \frac{TP}{TP + FN}$$

- La F1-Mesure : La F1-Mesure est la moyenne harmonique de la précision et du rappel.

$$F - measure = \frac{(2 \times Precision \times Recall)}{Precision + Recall}$$

3.3 Travaux réalisés :

L'objectif de ce travail est d'étudier la problématique de détections de malices Android. Pour atteindre le but nous avons commencé par l'entraînement d'un ensemble d'algorithmes de la fouille de données et de l'apprentissage profond sur deux corpus. Ensuite et afin d'améliorer les performances des détecteurs nous avons proposé un nouveau modèle basé sur l'hybridation de deux algorithmes de fouille de données.

3.3.1 Partie 1 : Application algorithmes de fouille de données et de l'apprentissage profond

Présentation des algorithmes

- **Naïve base :** un algorithme de classification supervisé basé sur le théorème de Bayes fonder sur les probabilités conditionnelles avec une hypothèse d'indépendance entre les attributs. L'hypothèse d'indépendance implique immédiatement que la probabilité peut être décomposée en un produit de probabilités par dimension. En termes simple, un classificateur NBs suppose que la présence d'une caractéristique particulière dans une classe est sans rapport avec la présence de toute autre caractéristique. [33]
- **Arbre de décision :** est une technique populaire de la classification supervisée utilisée dans diverses applications parce qu'elle est efficace, universelle et simple à implémente. Il est utilisé pour prédire la classe d'un nouvel exemple ainsi que pour la régression quand la classe dans la base d'apprentissage est numérique. Son principe général est de construire l'arbre (un modèle prédictif) à partir de la base d'apprentissage dont chaque chemin depuis la racine jusqu'à une feuille correspond à une règle de classification. Les règles extraites sont ensuite utilisées pour classer de nouveaux textes dont la classe est inconnue. On peut distinguer plusieurs versions de l'algorithme ID3, C4.5, C5, ect. [33]
- **KNN :** Le KPPV appelé en anglais K nearest neighbor (KNN) est un algorithme de classification supervisée simple et naïve. L'objectif c'est de classer chaque nouvel

exemple (de la base de test) sur la base de leur distance avec les exemples de la base d'apprentissage. Il nécessite la présence des paramètres suivants :

- Une base d'apprentissage.
- La valeur du K.
- Une mesure de distance.

Pour prédire la classe d'un nouvel exemple " X " l'algorithme calcule la distance de X avec chaque exemple de la base d'apprentissage afin de trouver les " K " plus proches voisin de X. Enfin la classe majoritaire Parmi les K classes sera attribuée à X. [33]

- **SVM :** Une machine à vecteurs de support (SVM) est un algorithme d'apprentissage automatique qui analyse les données pour la classification et l'analyse de régression. SVM est une méthode d'apprentissage supervisé qui examine les données et les trie dans l'une des deux catégories. Un SVM produit une carte des données triées avec les marges entre les deux aussi éloignées que possible. Les SVM sont utilisés dans la catégorisation de texte, la classification d'images, la reconnaissance de l'écriture manuscrite et dans les sciences. [34]
- **Deep Learning :** Le deep learning est un type d'intelligence artificielle dérivé de la machine learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, par opposition à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées. [36]
- **CNN (réseau neuronal convolutif) :** Un réseau de neurones convolutionnels (CNN, ou ConvNet) est une autre classe de réseaux de neurones profonds spécialisés pour le traitement de données ayant une topologie semblable à une grille. [36]

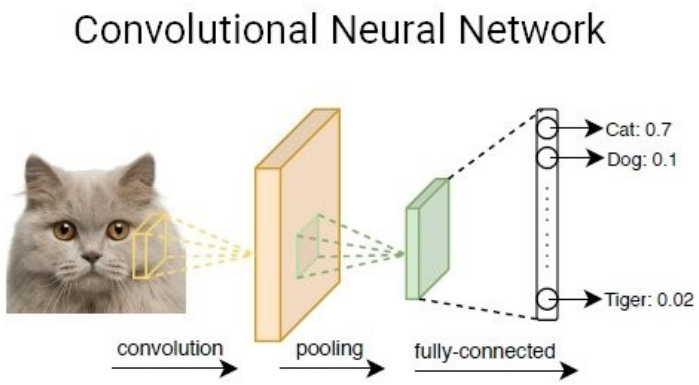

FIGURE 3.2 – Concept de réseau neural de convolution (CNN). [36]

– **RNN (Réseau neuronal récurrent) :** un réseau neuronal récurrent (RNN) est une autre classe de réseaux neuronaux artificiels qui utilisent l'alimentation séquentielle de données. Les RRN ont été élaborés pour régler le problème des séries chronologiques de données d'entrée séquentielles. [36]

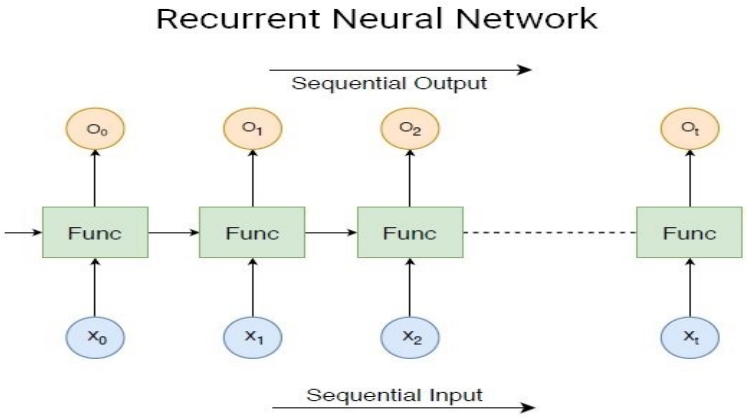

FIGURE 3.3 – Concept de réseau neuronal récurrent (RNN). [45]

Résultat expérimentaux

Nous avons utilisé le mode d'évaluation croisé, le principe de ce mode est de divisé dataset sur un nombre entré comme paramètre, et d'évaluer chaque segment comme base

de test en alternant sur tous les segments.

Les algorithmes	Dataset 1				Dataset 2			
	Rappel	précision	accuracy	F-mesure	Rappel	précision	accuracy	F-mesure
naïve base	0.93	0.93	0.93	0.93	0.92	0.92	0.92	0.92
arbre de décision	**0.98**	**0.98**	**0.98**	**0.98**	0.96	0.96	0.96	0.96
KNN	0.97	0.97	0.97	0.97	**0.97**	**0.97**	**0.97**	**0.97**
SVM	0.97	0.97	0.97	0.97	**0.97**	**0.97**	**0.97**	**0.97**
RNN	0.98	0.91	0.91	0.91	0.93	0.94	0.75	0.77
CNN	0.98	0.98	0.97	0.98	**0.98**	**0.98**	**0.98**	**0.98**

TABLE 3.1 – Comparaison des résultats du système classique pour les deux dataset.

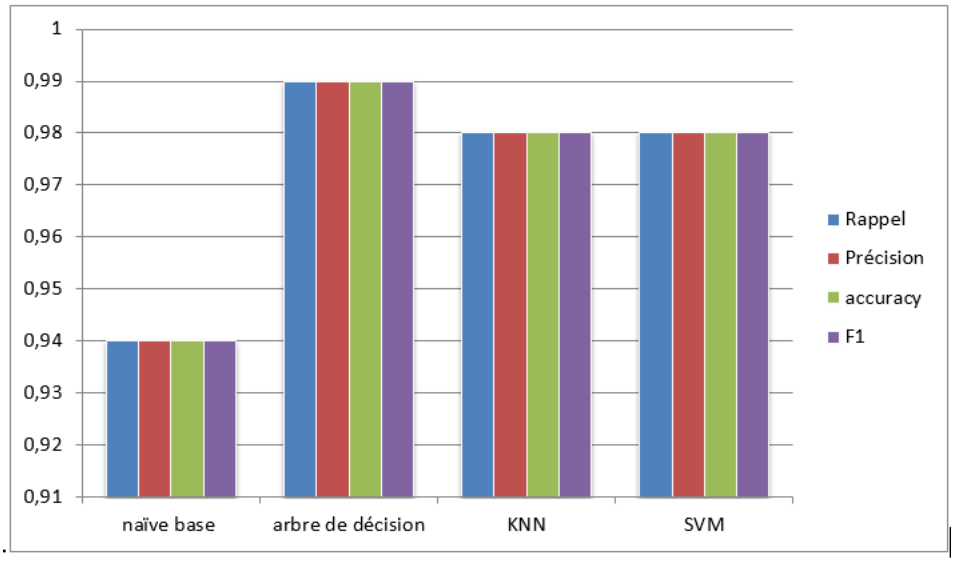

FIGURE 3.4 – Histogramme de comparaison des résultats du système classique et deep learning pour dataset 1.

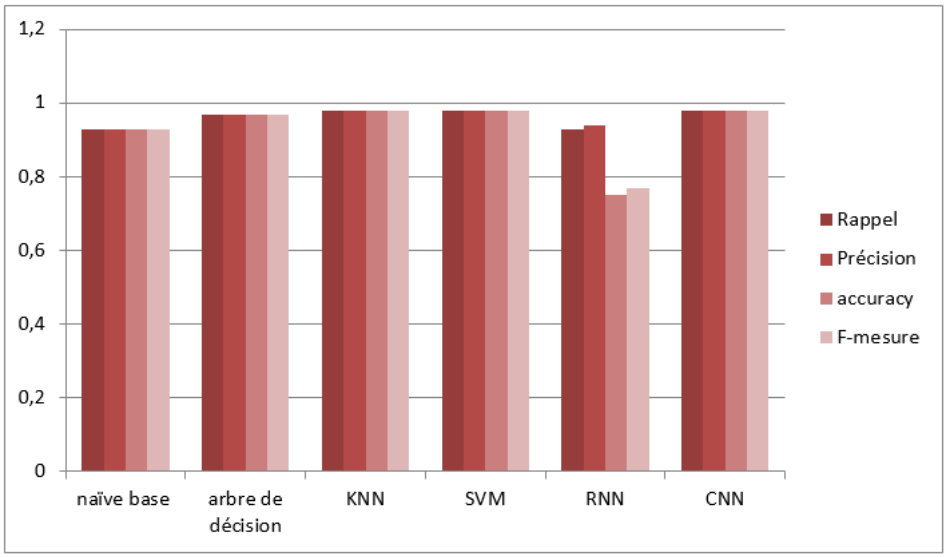

FIGURE 3.5 – Histogramme de comparaison des résultats du système classique et deep learning pour dataset 2.

Sur le tableau 3.1 et les figures 3.4 et 3.5 nous avions effectué une étude comparative sur des différents algorithmes utilisés dans d'autres études. Sur les lignes du tableau nous avions cité les algorithmes et sur les colonnes nous avions cité les différentes mesures pour évaluer les résultats de chaque algorithme. Après l'illustration des résultats, arbre de décision a donné des meilleurs résultats dans dataset 1 par rapport aux autres études, commençant par le rappel. Pour deuxième dataset KNN et SVM a donné des meilleurs résultats par rapport aux autres études notre algorithme a donné un meilleur rappel avec 0,98 et une précision avec 0,98 et un taux de succès (Accuracy) avec 0,98 et un F-mesure avec 0,98.

3.3.2 Partie 2 : Approche proposée

Après avoir entrainé les algorithmes de data mining et les deep learning sur deux corpus. Notre aidée consiste à appliquer deux filtres (algorithmes) d'une façon séquentielle sur les data sets . L'architecture du modèle est présentée dans la figure suivante :

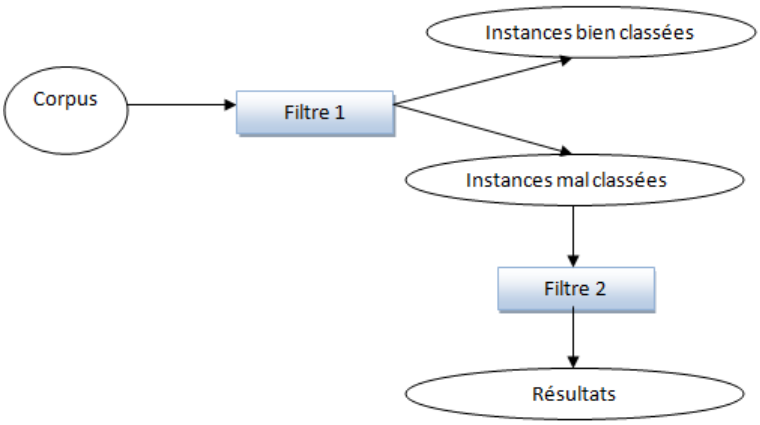

FIGURE 3.6 – Le modèle proposé

Les algorithmes utilisés sont SVM avec arbre de décision. Le choix a été effectué selon les résultats obtenus. Le tableau présente les résultats obtenus :

Hybridation	Dataset 1				Dataset 2			
	Rappel	précision	accuracy	F1-mesure	Rappel	précision	accuracy	F1-mesure
SVM avec arbre de décision	0.998	1	0.998	0.998	0.995	0.993	0.996	0.994
Arbre de décision avec SVM	0.994	1	0.995	0.997	0.986	1	0.995	0.993

TABLE 3.2 – Le modèle proposé

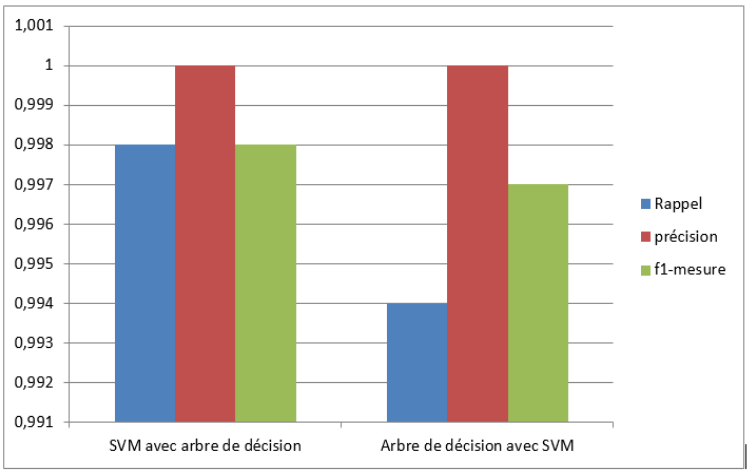

FIGURE 3.7 – Histogramme de comparaison des résultats d'hybridation testé pour dataset 1.

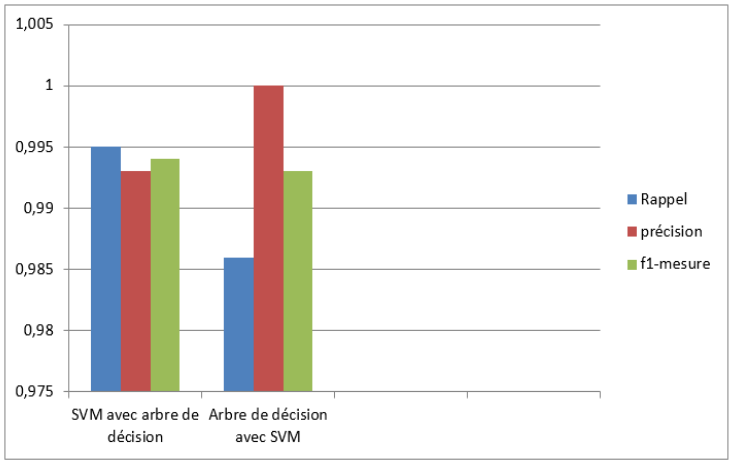

FIGURE 3.8 – Histogramme de comparaison des résultats d'hybridation testé pour dataset 2.

Sur le tableau 3.2 et les figures 3.7 et 3.8 nous avions effectué une étude comparative sur deux hybridation qui nous avions proposé. Après l'illustration des résultats, l'hybridation a donné des meilleurs résultats par rapport aux autres études, commençant par le rappel , notre hybridation a donné un meilleur rappel avec 0,998 et une précision avec 1 et un et un F-mesure avec 0,998 .

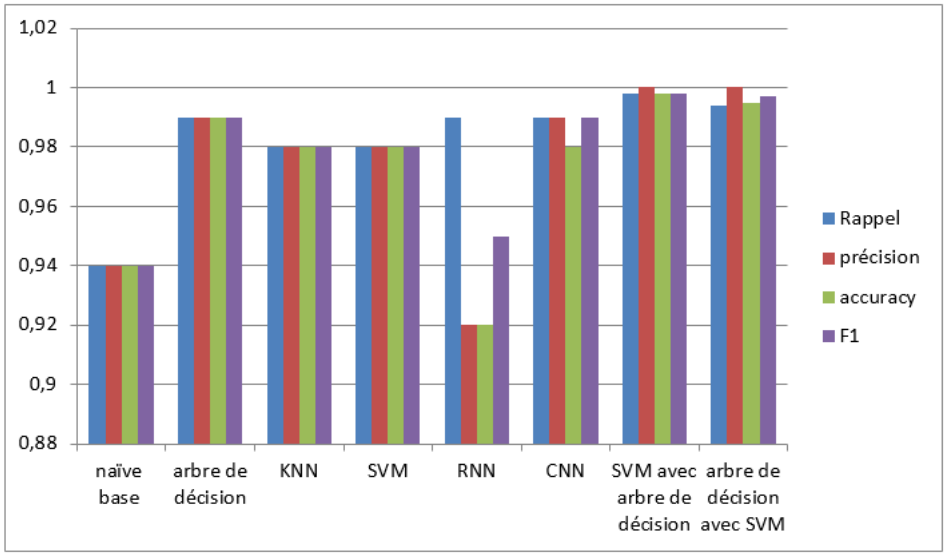

FIGURE 3.9 – Histogramme de comparaison des résultats les algorithmes classique et deep learning et hybridation pour dataset 1.

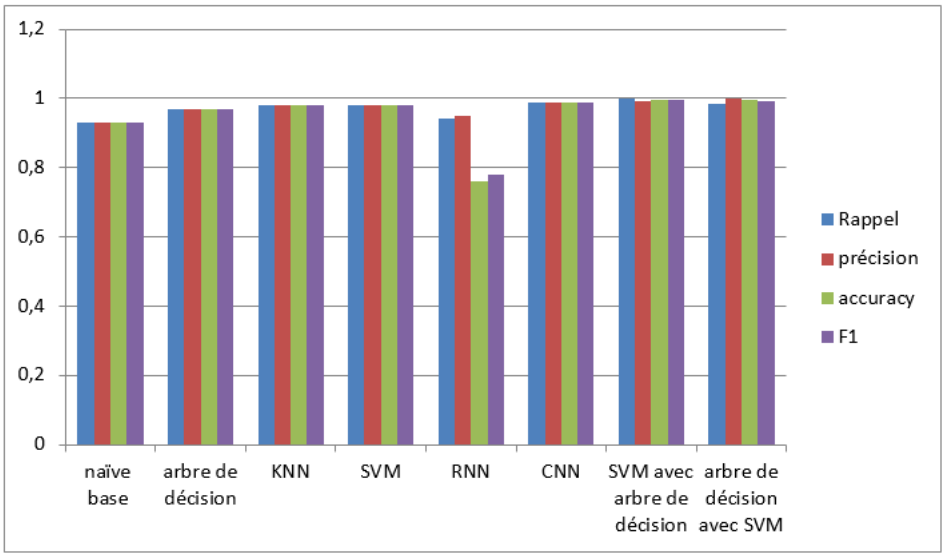

FIGURE 3.10 – Histogramme de comparaison des résultats les algorithmes classique et deep learning et hybridation pour dataset 2.

Sur le les figures 3.9 et 3.10 nous avions effectué une étude comparative sur les algorithmes classiques, deep learning et hybridation. Après l'illustration des résultats, l'hybridation a donné des meilleurs résultats par rapport aux autres études,

3.4 Implémentation :

3.4.1 Langage et Environnement de Travail.

Dans notre travail, en utilisant l'environnement python :

- **Python :** est un langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.[37]
- **Jupyter notebook :** est un notebook de calcul (computational notebook) open source, gratuit et interactif. C'est une application web basée client permettant de créer et de partager du code, des équations, des visualisations ou du texte.[38]
- **Google Colab :** est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation.[39]
- **Tensorflow :** une bibliothèque open source de Machine Learning, créée par Google, permettant de développer et d'exécuter des applications de Machine Learning et de Deep Learning. Découvrez tout ce que vous devez savoir à son sujet.[40]

3.4.2 Description de notre application

Dans ce qui suit nous présentons l'interface qui a nommé Classification Desktop App qui constitué de :

- Bouton titré à " import the dataset " : Ce bouton importer le dataset qui est utilisé.

– Bouton titré à " apply " : Nous utilisons ce bouton pour exécuter les algorithmes (SVM, KNN, NB, arbre de décision).

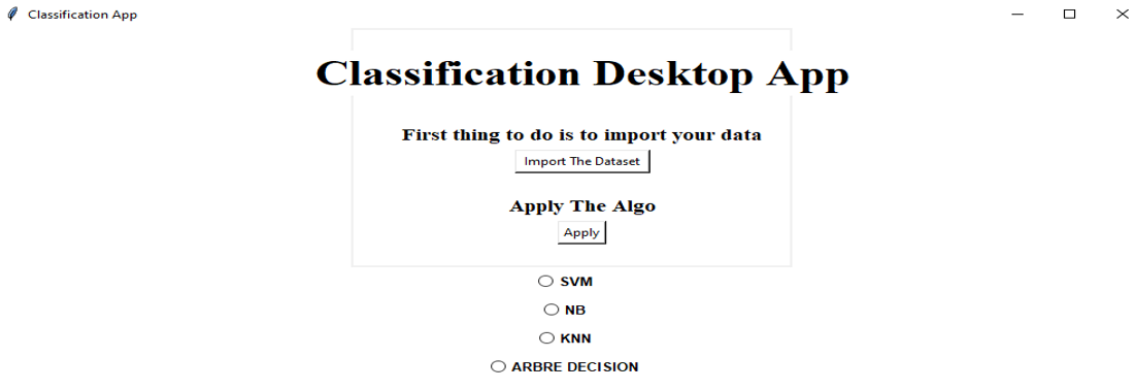

FIGURE 3.11 – Notre interface.

Conclusion :

Dans ce chapitre, nous avons entrainé algorithmes de fouille de données (naïve base, arbre décision, KNN et SVM) et les algorithmes d'apprentissage profond sur deux corpus d'une part et nous avons présenté notre part de contribution au problème détection de malice Android , représentant les outils et les jeux de données utilisées .

Conclusion générale

L'Android est probablement le système d'exploitation mobile le plus commun, une popularité qui a beaucoup attiré les développeurs de logiciels malveillants. Le grand nombre d'appareils exportés vers Android et la variété des données contenues dans ces appareils en ont fait une source importante de données sensibles. Android fait face à de nombreux défis et problèmes de sécurité, de sécurité sur Android Un thème de recherche qui a attiré l'attention de nombreux chercheurs. Pour résoudre les problèmes de sécurité sur Android, il devient de plus en plus important de mettre en oeuvre de nouvelles approches pour la reconnaissance des logiciels malveillants sur la plate-forme mobile. Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis de confronter et d'améliorer nos compétences et nos connaissances dans le domaine de la programmation et la recherche. Afin d'atteindre ces résultats, nous avons passé beaucoup de temps à lire et réviser des publications, des articles et des livres pour voir et comprendre les concepts et comment appliquer les différents modèles et algorithmes utilisé. Nous avons commencé le travail par l'entrainement de quelques algorithmes de fouille de données et un algorithme d'apprentissage profond sur deux corpus. Nous avons remarqué que quelques algorithmes ont donné un très bon résultat (Précision CNN=0.98). La deuxième étape consiste à tester l'hybridation deux algorithmes. Cette aidé à améliorer les résultats d'une façons remarquable (précision =1) . Ces résultats ont conduit à un certaines perspectives. Afin de proposer un modèle de détection de malice Android performons nous pensons à :

– utiliser d'autres corpus et d'autres paramètres de simulation pour confirmer les

résultats obtenus.

– tester d'autres types d'hybridation (tester plusieurs algorithmes et ajouter d'autre niveau).

Annexe

- **La confidentialité :** Seules les personnes habilitées doivent avoir accès aux données. Toute interception ne doit pas être en mesure d'aboutir, les données doivent être cryptées, seuls les acteurs de la transaction possèdent la clé de compréhension. [41]
- **L'intégrité :** Il faut garantir à chaque instant que les données qui circulent sont bien celles que l'on croit, qu'il n'y a pas eu d'altération (volontaire ou non) au cours de la communication. L'intégrité des données doit valider l'intégralité des données, leur précision, l'authenticité et la validité.[41]
- **La disponibilité :** Il faut s'assurer du bon fonctionnement du système, del'accès à un service et aux ressources à n'importe quel moment. La disponibilité d'un équipement se mesure en divisant la durée durant laquelle cet équipement est opérationnel par la durée durant laquelle il aurait dû être opérationnel.[41]
- **La non-répudiation :** Une transaction ne peut être niée par aucun des correspondants. La non-répudiation de l'origine et de la réception des données prouve que les données ont bien été reçues. Cela se fait par le biais de certificats numériques grâce à une clé privée.[41]
- **L'authentification :** Elle limite l'accès aux personnes autorisées. Il faut s'assurer de l'identité d'un utilisateur avant l'échange de données.[41]
- **Risque :** C'est la probabilité qu'une menace exploite une vulnérabilité. Autrement dit, c'est une possibilité qu'un fait dommageable se produise.[41]
- **Vulnérabilité :** C'est une faiblesse inhérente à un système (software ou hardware).

Appelée parfois faille ou brèche, elle représente le niveau d'exposition face à la menace dans un contexte particulier. [41]

- **Menace :** C'est le danger (interne ou externe) tel qu'un hacker, un virus, etc.[41]
- **Contre-mesure :** C'est un moyen permettant de réduire le risque dans une organisation.[41]
- **La maladresse :** Commettre des erreurs ou exécuter de traitement non souhaité, ou effacer involontairement des données ou des programmes ; etc. [41]
- **L'ingénierie sociale :** Une méthode pour obtenir d'une personne des informations confidentielles, que l'on n'est pas normalement autorisé à obtenir, en vue de les exploiter à d'autres fins. [41]
- **Logiciel malveillant :** Est un programme développé dans le but de nuire à un système informatique, sans le consentement de l'utilisateur dont l'ordinateur est infecté. [41]
- **Virus :** Est un programme informatique malveillant qui perturber le fonctionnement normal d'un système informatique à l'insu de son propriétaire.[42]
- **Ver :** Est un programme malveillant qui infecte un maximum d'appareils en se propageant sur un réseau tel qu'internet.[43]
- **Cheval de Troie(Trojan) :** Est un programme malveillant. Bien qu'en apparence inoffensif, le logiciel renferme en fait un malware qui infecte la machine cible.[44]

Bibliographie

[1] N. Tien Thinh. Système d'exploitation pour les mobiles, Rapport final (Travail personnel Encadré), Hanoi, Juillet 2009.

[2] H.B. Bekkaye & S. Meziane. Développement d'une application de géolocalisation des médecins de la wilaya de Tlemcen sous Android, Université Abou Baker Belkaid, Tlemcen, 2016-2017.

[3] F. Boudjadi & L.Cheklat.Conception et réalisation d'une application mobile sensible au contexte pour un musée. Université A. Mira, Bejaïa, Juillet 2013.

[4] Thèse Chapitre 1 Introduction aux Applications Mobiles,Dr. KOUAH SOFIA .Université de Larbi Ben mhidi,Oum lbouaghi, 2019-2020.

[5] W. Salem & K. Messeguem. Réalisation d'une application sous Android pour le suivi des diabétiques UNIVERSITE MOHAMED BOUDIAF ,M'SILA, 2020.

[6] `www.wikipedia.org`, consulté le 15 janvier 2022 à16.00.

[7] H.B. Bekkaye & S. Meziane. Développement d'une application de géolocalisation des médecins de la wilaya de Tlemcen sous Android, Université Abou Baker Belkaid, Tlemcen, 2016-2017.

[8] www.statista.com, consulté le 17 janvier 2022 à 18.00.

[9] H.B. Bekkaye & S. Meziane. Développement d'une application de géolocalisation des médecins de la wilaya de Tlemcen sous Android, Université Abou Baker Belkaid, Tlemcen, 2016-2017.

[10] www.statista.com, consulté le 10 février 2022 à 18.00.

[11] Shibly, F. Android Operating System : Architecture, Security Challenges and Solutions. Lecturer in IT, South Eastern University of Sri Lanka, Oluvil, Sri Lanka. 2016.

[12] J. Raphael, Android versions : A living history from 1.0 to 11, 2020.https://www.computerworld.com/article/3235946/android-versionsa-living-history-from-1-0-to-today.html, Consulté le 23mars2022 à 10.00.

[13] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6992244, consulté le 20avril2022 à 22.00.

[14] https://www.webopedia.com/definitions/android-mobile-security, consulté le 19 avril 2022 à 18.30

[15] https://www.koreascience.or.kr/article/JAK0202034465346164.pdf, consulté le avril,22,2022.

[16] R.Triggs, What android app permissions mean and how to use them, 2018. https://www.androidauthority.com/app-permissions-886758/. Consulté le 24mars2022 à17.10

[17] www.wikipedia.org, consulté le mars 24.2022

[18] N. Ruff EADS, Thèse Sécurité du système Android, Innovation Works nicolas.ruff(@)eads.net.

[19] S.K. Justin. A machine learning approach to android malware detection. European Intelligence and Security Informatics Conference IEEE, 2012.

[20] S.K. Justin. A machine learning approach to android malware detection. European Intelligence and Security Informatics Conference IEEE, 2012.

[21] S.K. Justin. A machine learning approach to android malware detection. European Intelligence and Security Informatics Conference IEEE, 2012.

[22] S.K. Asaf. A behavioral malware detection framework for android devices. Journal of Intelligent Information Systems Springer, 2012.

[23] P.Z.Naser. Machine learning for android malware detection using permission and api calls. IEEE 25th International Conference on Tools with Artificial Intelligence, 2013.

[24] Y.Zhenlong. Droid-sec : deep learning in android malware detection. SIGCOMM '14 Proceedings of the ACM conference on SIGCOMM Chicago, Illinois, USA, 2014.

[25] D. M.Gianluca. A multi-level anomaly detector for android malware. International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS : Computer Network Security Springer,2012.

[26] J.Jaemin, K. Android malware detection based on useful api calls and machine learning. First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE) IEEE, 2018.

[27] B.Shaikh, M. A. novel approach to detect android malware. Procedia Computer Science 45 , Peerreview under responsibility of scientific committee of International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), Elsevier, 2015.

[28] C. K.Li, Android malware detection based on factorization machine. Cryptography and Security, 2016.

[29] H.Fereidooni. Android malware detection using static analysis of applications. IFIP International Conference on New Technologies, Mobility and Security (NTMS) IEEE, 2016.

[30] T.Q. Chen,. Tiny droid : a lightweight and efficient model for android malware detection and classification. Mobile Information Systems, 2018.

[31] `https://android-developers.googleblog.com/2010/06/exercising-our-remote-application.html`, consulté le 01/06/2022 à 14h30.

[32] P.Borah, D. Bhattacharyya & J. K. Kalita, Malware Dataset Generation and Evaluation. In 2020 IEEE 4th Conference on Information & Communication Technology (CICT), IEEE. pp. 1-6, 2020.

[33] H.A. Bouarara. Fouille de données (Data Mining) Université Dr Moulay Tahar de Saida, 2018-2019.

[34] https://dataanalysticpost.com/Lexique/svm/, consulté le aout.03.2022

[35] https://dataanalyticspost.com, consulté le aout 03.2022

[36] L.Bastien,"Deep Learning ou apprentissage profond". https://www.lebigdata.fr/deeplearning-definition, consulté le aout.04, 2022.

[37] https://ieeexplore.ieee.org/document/8853349, consulté le aout 06,2022

[38] https://www.lebigdata.fr/jupyter-notebook, consulté le aout 06.2022

[39] https://research.google.com/colaboratory/faq.html?hl=fr, consulté le aout 06.2022

[40] https://www.lebigdata.fr/tensorflow-definition-tout-savoirconsulté

[41] R. Yende. Support de cours de sécurité informatique et crypto. 2018.

[42] https://www.cybermalveillance.gouv.fr, consulté le 09 Juin 2022.

[43] https://www.serenicity.fr, consulté le 09 Juin 2022.

[44] https://www.lemagit.fr/definition/Cheval-de-Troie