**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**
وزارة التعليم العالي والبحث العلمي
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**
جامعة أبي بكر بلقايد – تلمسان –
Université Aboubakr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE


# MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

**En** : Télécommunications

**Spécialité** : Réseaux et Télécommunications

**Par** : KANDOUCI Abdelilah & BAROUDI Wadie Abderrahmane

THEME

## RENFORCEMENT DE LA PROTECTION DES SMARTPHONES PAR L'INTELLIGENCE ARTIFICIELLE

Soutenu le 17/06/2025, devant le jury composé de :

| | | | |
|---|---|---|---|
| M. MERZOUGUI Rachid | Professeur | Université de Tlemcen | Président |
| M. MOUSSAOUI Djilali | MCA | Université de Tlemcen | Examinateur |
| Mme. TALEB Sarra | MCB | Université de Tlemcen | Encadrant |
| Mme. BENAISSA Amal | MCB | Université de Tlemcen | Co-Encadrant |

2024/2025


بسم الله الرحمن الرحيم

# REMERCIEMENT

Nous tenons à exprimer notre sincère gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire. Ce travail est le fruit d'une collaboration et d'un accompagnement précieux que nous souhaitons reconnaître.

**À nos encadreurs**, pour leur disponibilité, leurs conseils éclairés et leur patience tout au long de ce parcours. Leurs orientations ont été essentielles pour structurer notre réflexion et approfondir nos analyses.

**À nos familles**, et particulièrement à nos parents, pour leur amour inconditionnel, leurs sacrifices et leurs encouragements constants. Leur soutien indéfectible a été notre plus grande source de motivation.

**À nos amis et collègues**, pour leur soutien moral, leurs échanges stimulants et les moments de partage qui ont rendu ce travail plus enrichissant.

**À tous ceux qui nous ont inspirés et soutenus**, directement ou indirectement, nous adressons notre profonde reconnaissance. Ce mémoire est aussi le vôtre.

Enfin, nous remercions l'**Université Aboubakr Belkaïd de Tlemcen** et tous les acteurs académiques qui nous ont offert l'opportunité de mener à bien ce projet.

*"Seul, on va plus vite ; ensemble, on va plus loin."*

# DÉDICACE

Je dédie ce mémoire avec une profonde gratitude :

À mes chers parents, pour leur amour inconditionnel, leurs sacrifices silencieux et leur soutien constant tout au long de mon parcours.

À ma famille, qui a toujours cru en moi, même dans les moments les plus difficiles.

À mes enseignants et encadreurs, pour leurs conseils précieux, leur patience et leur rigueur qui m'ont guidé vers l'accomplissement de ce travail.

À mes amis fidèles, compagnons de route et de longues nuits d'étude, pour leur soutien, leur humour et leur encouragement.

Enfin, à toutes celles et ceux qui, de près ou de loin, ont contribué à ce que ce projet voie le jour.

# Résumé

Les systèmes d'exploitation sont l'essence de nos appareils numériques, leur interface permettent aux utilisateurs d'interagir avec le matériel et les applications. Parmi eux, s'en est l'un des derniers leaders du marché des Smartphone Android qui s'est rapidement installé en tant que leader de marché grâce à son accès simple, sa flexibilité et un large écosystème. Mais cette popularité fait aussi une cible privilégiée pour les cybercriminels. Les malwares, applications malveillantes conçue pour vous voler vos données ou compromettre la sécurité d'appareils, sont devenus une vraie menace pour les utilisateurs. Face à ces risques, l'intelligence artificielle se présente ici pour apporter un renfort en termes de sécurité des systèmes d'exploitation. À l'aide de dernières générations d'algorithme, elle permet de repérer les conduites irrégulières, d'analyser les applications en temps réel et éviter les tentatives d'intrusion avant qu'elles ne causent de dommages. Ainsi l'IA se met en avant comme un outil indispensable pour protéger les smartphone et gagner en sécurité et en protection sur le coté des usagers.

**Mots clés :** Systèmes d'exploitation, Android, sécurité, malware, intelligence artificielle, détection, applications mobiles.

# ABSTRACT

Operating systems are the lifeblood of our digital devices, allowing users to connect with hardware and applications. Among them, Android is one of the latest market leaders, quickly establishing itself as a market leader thanks to its easy access, flexibility, and broad ecosystem. But this popularity also makes it a prime target for cybercriminals. Malware, malicious applications designed to steal your data or compromise device security, have become a real threat to users. Faced with these risks, artificial intelligence is emerging to provide reinforcement in terms of operating system security. Using the latest generation of algorithms, it can identify irregular behavior, analyze apps in real time, and prevent intrusion attempts before they cause damage. AI is thus emerging as an essential tool for protecting smartphones and increasing user security and protection.

**Keywords:** Operating systems, Android, security, malware, artificial intelligence, detection, mobile applications.

## ملخص

تشكل أنظمة التشغيل شريان الحياة لأجهزتنا الرقمية، حيث تسمح للمستخدمين بالتفاعل مع الأجهزة والتطبيقات. يعد أحد والذي أثبت نفسه بسرعة باعتباره رائدًا في Android أحدث الهواتف الرائدة في سوق الهواتف الذكية التي تعمل بنظام السوق بفضل سهولة الوصول إليه ومرونته ونظامه البيئي الكبير. لكن هذه الشعبية تجعلها أيضًا هدفًا رئيسيًا لمجرمي الإنترنت. أصبحت البرمجيات الخبيثة، وهي تطبيقات خبيثة مصممة لسرقة بياناتك أو المساس بأمان جهازك، تشكل تهديدًا حقيقيًا للمستخدمين. وفي مواجهة هذه المخاطر، أصبح الذكاء الاصطناعي هنا لتوفير التعزيزات فيما يتعلق بأمن نظام التشغيل. باستخدام أحدث جيل من الخوارزميات، يمكنه تحديد السلوك غير المنتظم وتحليل التطبيقات في الوقت الفعلي ومنع محاولات التطفل قبل أن تتسبب في حدوث ضرر. وبذلك أصبح الذكاء الاصطناعي أداة أساسية لحماية الهواتف الذكية وتحسين أمن .المستخدم وحمايته



**الكلمات المفتاحية:** أنظمة التشغيل، أندرويد، الأمان، البرمجيات الخبيثة، الذكاء الاصطناعي، الكشف، التطبيقات المحمولة

# TABLE DE MATIERS

# TABLE DE FIGURE

# LISTE DES TABLEAUX

# LISTE DES ABRÉVIATIONS :

- **ANN : Réseaux Neuronaux Artificiels (RNA)**
- **API : Interface de Programmation d'Applications (IPA)**
- **AR : Réalité Augmentée (RA)**
- **AVB : Démarrage Vérifié Android (Android Verified Boot reste souvent utilisé en anglais dans le domaine technique)**
- **CNN : Réseaux de Neurones Convolutifs (RNC)**
- **DL : Apprentissage Profond (AP)**
- **DNN : Réseau Neuronal Profond (RNP)**
- **GPU : Unité de Traitement Graphique (UTG)**
- **IA : Intelligence Artificielle (IA)**
- **IoT : Internet des Objets (IdO)**
- **KNN : k-Plus Proches Voisins (k-PPV)**
- **LSTM : Mémoire à Long Terme à Court Terme (souvent gardé en anglais car terme spécifique)**
- **ML : Apprentissage Automatique (AA) ou Apprentissage Machine (AM)**
- **OS : Système d'Exploitation (SE)**
- **PDA : Assistant Personnel Numérique (APN)**
- **PUA : Applications Potentiellement Indésirables (API)**
- **RAM : Mémoire Vive (MV)**
- **RNA : Réseau Neuronal Artificiel (RNA)**
- **RNN : Réseaux de Neurones Récurrents (RNR)**
- **RNP : Réseau Neuronal Profond (RNP)**
- **SDK : Kit de Développement Logiciel (KDL)**
- **SEM : Système d'Exploitation Mobile (SEM)**
- **SVM : Machines à Vecteurs de Support (MVS)**
- **VPN : Réseau Privé Virtuel (RPV)**
- **VR : Réalité Virtuelle (RV)**

# Introduction générale

Dans notre vie quotidienne, les smartphones sont devenus des outils indispensables, offrant une multitude de fonctionnalités allant bien au-delà de la simple communication. Aujourd'hui, ils servent à naviguer sur Internet, gérer notre agenda, accéder à nos comptes bancaires, contrôler nos objets connectés, et même piloter des tâches professionnelles à distance. Leur omniprésence dans notre société numérique moderne en fait des compagnons incontournables, aussi bien pour les particuliers que pour les professionnels. Cependant, cette montée en puissance et cette popularité croissante ne sont pas sans conséquences : elles attirent l'attention des cybercriminels. Ces derniers voient dans les smartphones, et particulièrement ceux utilisant le système Android, une opportunité idéale pour mener des attaques numériques ciblées.

Les malwares, ou logiciels malveillants, ont été conçus dans le but d'exploiter les failles de sécurité présentes dans les systèmes d'exploitation mobiles. Android, en raison de son architecture ouverte et de la diversité de ses applications disponibles, se retrouve en première ligne face à ces menaces. Ces programmes malicieux peuvent intercepter des données sensibles, espionner les utilisateurs à leurs insu, ou encore prendre le contrôle des appareils à distance, compromettant ainsi gravement la vie privée et la sécurité des utilisateurs. Cette réalité soulève des enjeux cruciaux en matière de cybersécurité. Or, les méthodes traditionnelles de détection, qui reposent principalement sur la reconnaissance de signatures connues ou sur des règles prédéfinies, se révèlent rapidement dépassées. En effet, l'évolution rapide des attaques et la sophistication croissante des malwares rendent ces approches insuffisantes pour garantir une protection efficace.

C'est dans ce contexte alarmant que l'intelligence artificielle (IA), et plus particulièrement ses deux branches majeures que sont le Machine Learning (ML) et le Deep Learning (DL), s'imposent comme des solutions innovantes et efficaces. Ces technologies permettent d'aller au-delà de la simple détection de menaces connues : elles apprennent à reconnaître des schémas inhabituels, à analyser les comportements suspects en temps réel, et à anticiper les tentatives d'intrusion avant même qu'elles ne causent des dégâts. Contrairement aux méthodes classiques figées, les systèmes d'IA sont capables de s'adapter en permanence, grâce à l'apprentissage continu basé sur l'analyse de volumes massifs de données. Cette capacité évolutive leur confère un avantage considérable dans la lutte contre des menaces de plus en plus dynamiques et sophistiquées.

Dans cette optique, le présent mémoire explore l'application de l'intelligence artificielle à la détection et à la prévention des malwares ciblant les dispositifs Android. L'étude est structurée en quatre chapitres, chacun abordant un aspect essentiel du sujet. Le premier chapitre présente un panorama complet des systèmes d'exploitation mobiles, avec une attention particulière portée à Android : son architecture interne, ses caractéristiques fondamentales, ainsi que les failles de sécurité connues qui le rendent vulnérable. Le deuxième chapitre est consacré à l'univers des malwares Android, leurs différentes catégories, les mécanismes d'infection utilisés, et un état de l'art des

techniques actuelles de détection. Le troisième chapitre introduit les concepts clés de l'apprentissage automatique et de l'apprentissage profond, tout en soulignant leur rôle stratégique en cybersécurité. Enfin, le quatrième chapitre expose la partie pratique du projet, qui comprend la préparation et le traitement des données, la mise en œuvre des modèles IA, ainsi que l'interprétation des résultats obtenus.

Ce travail vise non seulement à démontrer l'efficacité des approches basées sur l'IA dans la détection des malwares Android, mais aussi à comparer les performances de plusieurs algorithmes de ML et de DL afin d'identifier ceux les mieux adaptés à cette tâche complexe. Les résultats de cette recherche pourraient contribuer à améliorer les mécanismes de sécurité actuels et à proposer des systèmes plus robustes face aux cybermenaces.

En définitive, ce mémoire ambitionne de fournir une contribution significative au domaine de la cybersécurité mobile, en combinant une analyse théorique rigoureuse avec une approche expérimentale fondée sur des données réelles. Les conclusions tirées de cette étude pourraient servir de base à de futures avancées technologiques en matière de protection des smartphones, répondant ainsi à l'un des grands défis de notre ère numérique.

# Chapitre 1

# Système d'exploitation Android

## 1.1 Introduction :

Les systèmes d'exploitation mobiles (SEM) sont des logiciels chargés de gérer le matériel et les programmes des appareils mobiles tels que les smartphones,les iPhones, les tablettes et les assistants personnels numériques (PDA). Ils gèrent la connectivité sans fil (réseaux mobiles, Wi-Fi, Bluetooth, GPS), assurent l'exécution des applications, la gestion des ressources et fournissent l'interface utilisateur sur ses écrans tactiles. Parmi les systèmes d'exploitation mobiles de masse , sont principalement :

**Android,** développé par Google, fondé sur Linux, open source et leader du marché.

**Le système mobile propriétaire d'Apple, iOS,** utilisé uniquement sur les appareils de la marque Apple, atteignait 28,99 % du marché en 2023.

**Windows Phone,** développé par Microsoft pour les téléphones portables et les Pocket PC, présente des applications de base telles que l'e-mail, Internet, le chat et le multimédia.

Ces systèmes d'exploitation sont financés pour fournir une base fiable et sécurisée à ces applications. Les systèmes d'exploitation (comme Android ou iOS) sont effectivement conçus et financés (par les entreprises qui les développent) pour offrir un environnement stable et sûr aux applications qui y sont installées[1].


**Figure 1 : Logos des systèmes d'exploitations[2]**

## 1.2 Système d'exploitation Android :

Android est un système mobile sécurisé, équipé d'une bonne approche de sandboxing des applications, d'un démarrage authentifié (AVB) et d'un solide système de contrôle des autorisations[3].

Environ 2,5 milliards de personnes dans le monde utilisent Android, une valeur qui montre leur importance dans les domaines numériques de la mobilité. Google a inventé Android, qui non seulement a révolutionné les téléphones mobiles, mais également a permis de nouvelles opportunités au développement des applications.

Ce chapitre nous explique comment Android a changé notre vie, fait ressortir les fonctionnalités, son impact économique et l'emprise, étant donné à la fois pour les développeurs et les utilisateurs.

Android a complètement changé notre vie quotidienne, de la navigation GPS à l'utilisation des applications de plateaux de sécurité numériques pour des paiements à travers des interactions mobiles personnalisées. Il a aussi ouvert la voie à un écosystème créatif pour l'application, encadrant un million de créateurs pour faire portefeuille.

La sécurité d'Android est primordiale, avec des mises à jour régulières pour l'améliorer. Des options comme le chiffrement des données et le bac à sable (sandboxing) pour applications fournissent une sécurité essentielle pour la vie privée sur Android, protégeant ainsi les données personnelles contre les logiciels malveillants qui tentent d'accéder à des informations non autorisées.

Les utilisateurs ont la paix en possédant ces outils. Ils apportent une meilleure protection contre les programmes malveillants et les virus [4].

## 1.3 Android : Un aperçu de ses fonctionnalités clés:

La plateforme Android est devenue une condition indispensable dans le monde de la technologie. Elle possède de nombreuses fonctionnalités et un potentiel de développement énorme. Après avoir suivi presque tous ses articles, on a appris que la compréhension de ce que ces éléments font est très importante pour les développeurs, les sociétés.

Android dépasse de loin la téléphonie mobile. Il est compatible avec une foule d'appareils et d'objets connectés.

Ces fonctionnalités avancées contribuent à une intégration renforcée d'Android dans la vie quotidienne, consolidant ainsi sa position dominante dans le monde technologique. La création d'applications personnalisées, répondant aux besoins spécifiques des utilisateurs, ainsi que le support d'un large éventail de matériels, offrent une flexibilité sans précédent aux fabricants. Cela permet de couvrir des domaines variés, allant des loisirs à la gestion sécurisée des systèmes, tout en intégrant les préoccupations croissantes liées à la protection des données personnelles.

Android est également à l'avant-garde de l'innovation avec de nouvelles technologies telles que la réalité augmentée (AR) et les solutions de l'intelligence artificielle (IA). Ces innovations ouvrent ainsi de nouvelles perspectives pour l'interaction de l'utilisateur avec l'appareil [4].

| Technologie | Fonctionnalité | Impact sur le développement |
|---|---|---|
| **IA** | Automatisation des processus | Applications plus intelligentes et réactives |
| **AR** | Interactions enrichies | Expériences utilisateurs immersives et interactives |
| **IoT** | Connectivité étendue | Meilleure intégration des appareils dans l'écosystème Android |

**Table 1 : Fonctions, Impact , Technologies Android [4].**

## 1.4 Comparaison des systèmes d'exploitation mobile:

Le tableau ci-dessous présente une comparaison entre les principaux systèmes d'exploitation mobiles selon différents critères :

| | Android | IOS | HarmonyOS | Windows Phone |
|---|---|---|---|---|
| Développeur | Google | Apple | Huawei | Microsoft |
| Langage de programmation | C, C++ , Java, Kotlin | C, C++, Objective-C, Swift | C, C++, JS, ArkTS, Cangjie | Visual Studio, eMbeddeb VC++ |
| Open source | Oui | Non | Oui | Non |
| Principaux navigateurs Web | Chrome for Android, Opera, Firefox, Microsoft Edge, Samsung Internet | Safari, Chrome for iOS, Opera, Firefox, Microsoft Edge | Huawei Browser, Microsoft Edge (Blink), UC Browser (Blink, V8), Opera browser (Blink, V8), Yandex Browser (Blink), Vivaldi Browser (Blink, V8) | Internet Explorer Mobile, Microsoft Edge, UC Browser, Opera Mini, Surfy Browser |
| Multiplateforme de de déploiement | Android seulement | IPhone, iPod touch, iPad | HarmonyOS | Windows Mobile<br><br>Windows CE |

**Table 2 : Comparaison des systèmes d'exploitations mobiles[5]**

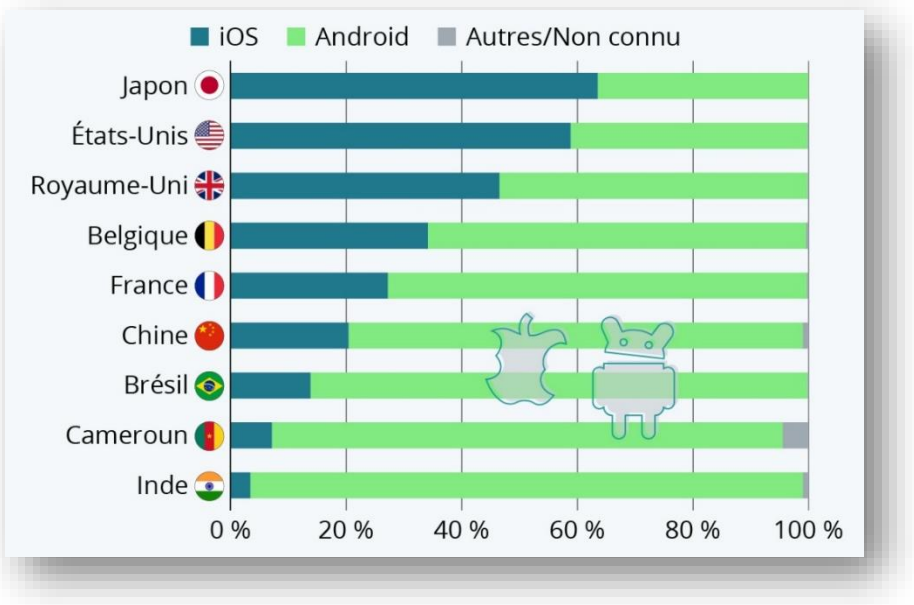



**Figure 2 : Système d'exploitation utilisé dans quelques pays (juillet 2020) [6]**

## 1.5 Les versions Android :

Depuis sa première version en septembre 2008, Android a connu une évolution constante, en introduisant régulièrement de nouvelles fonctionnalités et améliorations pour répondre aux besoins des utilisateurs et suivre les avancées technologiques. Les premières versions comme Android **1.0 (Apple Pie)** et **1.1 (Banana Bread)** ont posé les bases : support du Wi-Fi, Bluetooth, appareil photo, envoi de SMS/MMS, et accès à un magasin d'applications (Android Market).

Avec Android **1.5 Cupcake**, le système a introduit le clavier virtuel, les widgets et l'enregistrement vidéo. **Donut (1.6)** a amélioré la recherche vocale et la compatibilité avec différentes résolutions d'écran. La version **Éclair (2.0 à 2.1)** a optimisé les performances, introduit l'HTML5 dans le navigateur et le support de Microsoft Exchange. Par la suite, **Froyo (2.2)** a apporté le partage de connexion USB et le support des GIF animés, tandis que **Gingerbread (2.3)** a introduit la VoIP, le NFC et le support multi-caméra.

**Honeycomb (3.x**), pensé pour les tablettes, a offert une interface adaptée, le support USB Host, et le multi-fenêtrage. L'unification de l'interface mobile et tablette est apparue avec **Ice Cream Sandwich (4.0)**, suivi par **Jelly Bean (4.1 à 4.3)** qui a intégré Google Now, Photo Sphere et le support du Bluetooth Low Energy.

**KitKat (4.4)** a mis l'accent sur la performance et une nouvelle interface translucide. Avec **Lollipop (5.x)**, Google introduit le Material Design, une meilleure gestion de la batterie, et un support multi-SIM. **Marshmallow (6.0)** marque un tournant avec le support des empreintes digitales, Android Pay et un meilleur contrôle des permissions. **Nougat (7.0**) généralise le multi-fenêtrage et optimise les performances grâce à un nouveau compilateur

La version **Oreo (8.0**) apporte le Picture-in-Picture, les Instant Apps et des notifications enrichies. **Pie (9.0)** améliore la navigation par gestes, l'authentification biométrique, et introduit des notifications plus dynamiques.

**Android 10**, lancé en 2019, introduit un mode sombre natif, de nouveaux codecs (AV1, HDR10+), le support pour les smartphones pliables et des améliorations notables sur la gestion de la confidentialité. Android 11 continue sur cette lancée avec des bulles de conversation, la capture d'écran vidéo et des permissions temporaires.

**Android 12** (sorti le 4 octobre 2021) propose une interface totalement repensée, plus fluide et personnalisable grâce à Material You, qui adapte les couleurs du système au fond d'écran. Côté sécurité, les utilisateurs peuvent consulter un historique d'accès aux capteurs (caméra, micro, localisation) par les applications, offrant ainsi une transparence renforcée sur la gestion des données personnelles.

**Android 13**, lancé le 15 août 2022, continue à renforcer la confidentialité. Désormais, les applications doivent demander l'autorisation explicite pour envoyer des notifications, ce qui permet de limiter les interruptions non désirées. De plus, l'interface du panneau de notifications a été revue pour indiquer les applications actuellement actives, améliorant la visibilité et le contrôle des processus en arrière-plan.

**Android 14**, publié le 4 octobre 2023, introduit plusieurs évolutions subtiles mais significatives : amélioration du mode multi-utilisateur, gestion renforcée des droits d'accès par image (l'utilisateur peut choisir précisément quelles photos une application peut voir), une meilleure autonomie, et une nouveauté originale : la possibilité d'utiliser son téléphone Android comme webcam une fois connecté à un ordinateur.

Enfin, **Android 15**, prévu pour le troisième trimestre de 2024, porte le nom de code Vanilla Ice Cream. Bien que tous les détails ne soient pas encore finalisés, cette version devrait encore approfondir les améliorations en matière de sécurité, d'intelligence embarquée, et d'intégration entre appareils. L'accent est également mis sur une gestion plus intelligente des ressources énergétiques, des performances optimisées pour les jeux, et une meilleure prise en charge des nouveaux formats de matériel (comme les écrans enroulables et les objets connectés IA-natifs).


**Figure 3: Logo des versions d'android[10]**

## 1.6 Architecture d'Android :

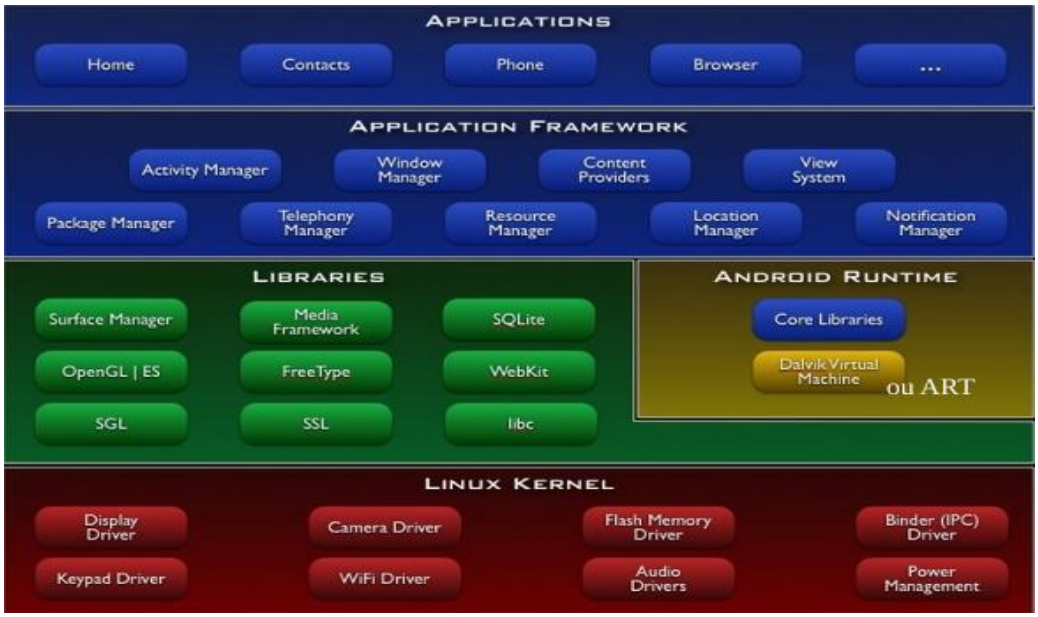



**Figure 4: Architecture d'Android [11].**

**La Figure 4** indique les différentes couches du système d'exploitation Android,Le Last-Glove comprend 5 sections divisées en 4 couches principales :

**Noyau Linux** : cette couche est le squelette du système Android. Elle embarque l'ensemble des pilotes de périphérique de bas niveau requis pour le bon fonctionnement des composants matériels d'un appareil Android.

**Bibliothèques** : elles contiennent la partie logging qui propose les services et les outils du système d'exploitation. Par exemple, la bibliothèque SQLite permet la récupération d'un système de gestion de base de données pour le stockage des informations, la bibliothèque WebKit fournit des bibliothèques spécialisées pour la navigation Web.

**Environnement d'exécution Android (Android Runtime – ART)** : Il s'agit du moteur responsable de l'exécution des applications Android. ART fonctionne en étroite collaboration avec les bibliothèques de l'environnement Android, lesquelles fournissent un ensemble d'API et de fonctionnalités nécessaires à la construction des applications. Ces bibliothèques sont majoritairement développées en Java, bien que d'autres langages puissent également être utilisés. Il contient également la machine virtuelle Dalvik, qui exécute chaque application dans son propre processus avec sa propre instance d'une machine virtuelle. Conçue à partir des filtres propres à Android, Dalvik est optimisée pour les appareils mobiles avec peu de mémoire et peu de puissance de processeur.

**Framework d'application** : cette couche s'adresse aux développeurs, aux capteurs du framework des fonctionnalités Android et à l'exploitation des ressources du système pour la création d'applications.

**Applications** : dénommée elle aussi CyanogenMod, cette couche compte les applications natives de l'architecture (comme l'application téléphone, les contacts ou le navigateur) et celles exploitées et utilisées via Play Store[11] .

## 1.7 Conclusion :

Android s'est imposé comme le système d'exploitation mobile le plus utilisé au monde grâce à sa nature open source, à sa flexibilité et à son écosystème d'applications très vaste. Sa conception en couches facilite le partage fluide des ressources entre les différents composants du système, et l'intégration avec Google Services offre une meilleure expérience de l'utilisateur. Grâce à des mises à jour quasi quotidiennes, Android continue d'améliorer ses performances et d'élargir ses options de personnalisation. La stagnation relative de l'industrie mobile reste

évidente, affectant aussi bien les entreprises manufacturières que les développeurs et les consommateurs , Le futur Android continuera l'expansion  s'ajustant aux dernières technologies et aux besoins du marché.

Après avoir présenté le système d'exploitation Android dans ses différentes versions et fonctionnalités, il est essentiel de s'intéresser aux menaces qui pèsent sur cet environnement. Le chapitre suivant sera donc consacré aux malwares Android, en mettant en lumière leurs types, leurs modes de propagation ainsi que leur impact sur la sécurité des smartphones.

# Chapitre 2

## Malwares Android

## 2.1Introduction :

Android est le système d'exploitation le plus largement adopté pour les appareils portables. Cependant, cette popularité représente aussi une vulnérabilité. En effet, Android est le système qui recense le plus grand nombre de logiciels malveillants : la quantité de codes malicieux se multiplie depuis plusieurs années (une augmentation de +472 % a été observée depuis juillet 2011) [11].

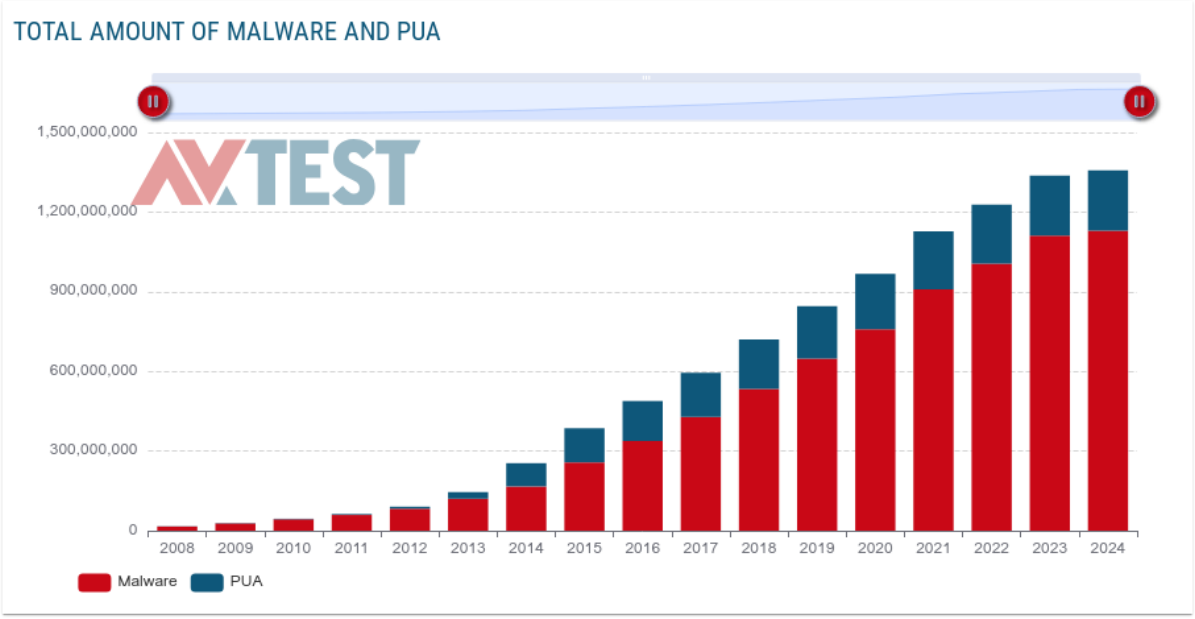



**Figure 5: Nombre total de logiciels malveillants et de PUA de 2008 à 2024 [12].**

## 2.2 Qu'est-ce qu'un malware mobile Android ?:

Un malware sur Android est un logiciel malveillant qui se télécharge discrètement sur un smartphone afin d'en nuire ou de récupérer des informations. Il peut se cacher dans des programmes douteux, des liens piégés, des bannières de propagande.

Lorsqu'il est installé sur un téléphone, il peut ralentir le système, afficher des annonces non souhaitées, suivre les activités de l'utilisateur ou transporter des données personnelles sans son accord. À l'origine, les pirates n'ont généralement d'autres buts que de réaliser un gain financier, soit en utilisant les informations volées, soit en amenant les victimes à payer du cash pour leur permettre d'accéder à nouveau à leur appareil.

Les symptômes comme une batterie qui chute rapidement ou des applications inconnues ou le comportement étrange du téléphone peuvent indiquer la présence d'un malware. À ce moment, il faut réagir aussi rapidement que possible pour éviter les dégâts [13].

## 2.3 Différents categories des malwares :

| Catégories des malwares | Définition du Malware |
|---|---|
| **Logiciel publicitaire (Adware)** | Il s'agit d'un programme malveillant qui affiche des publicités intrusives sur l'écran de l'utilisateur, principalement lors de l'utilisation de services web. |
| **Porte dérobée (Backdoor)** | Les portes dérobées sont des entrées inaperçues dans un smartphone. Autrement dit, elles permettent de contourner l'authentification d'un smartphone et d'élever les privilèges, permettant ainsi à un attaquant d'y accéder à tout moment. |
| **Infecteur de fichiers (File Infector)** | Un infecteur de fichiers est un type de logiciel malveillant qui infecte les fichiers APK. Un Android Package Kit (APK) est un fichier contenant utes les données d'une application. L'infecteur de fichiers est installé à l'aide des fichiers APK. Une fois les fichiers APK installés, le virus est exécuté. |
| **PUA** | Les PUA sont des programmes potentiellement indésirables intégrés à des logiciels authentiques et gratuits. On les appelle parfois programmes potentiellement indésirables (PPI). Les PUA ne sont pas toujours dangereux, et tout dépend de leur utilisation. Ce type de malware comprend les logiciels publicitaires, les logiciels espions et les pirates de navigateur. |
| **Ransomware** | Un ransomware est un logiciel qui chiffre les fichiers et les dossiers l'un ordinateur afin d'en interdire l'accès. Il exige une rançon importante en échange de la clé de déchiffrement permettant d'accéder aux données. |
| **Riskware** | Un riskware est un logiciel authentique qui menace les failles de sécurité appareil. qu'il s'agisse d'un logiciel légitime, il collecte des données sur l'appare et redirige les utilisateurs vers des sites web malveillants. |
| **Scareware** | Un scareware est un logiciel de peur qui incite les utilisateurs à télécharger ou à acheter des applications malveillantes en leur instillant la peur, par exemple en les persuadant d'installer de faux logiciels prétendant protéger l'appareil. |

| | |
|---|---|
| **Trojan** | Les chevaux de Troie sont des imposteurs malveillants qui se font passer pour de simples programmes. Ils peuvent voler des données de l'appareil sans être détectés. |
| **Trojan_Banker** | s programmes bancaires sont conçus pour voler vos informations de compte pour les services bancaires en ligne, les paiements électroniques et les cartes de crédit ou de débit. |
| **Trojan_Dropper** | es pirates informatiques utilisent ces programmes pour installer des chevaux e Troie ou des virus afin d'empêcher la détection de programmes dangereux. Tous les systèmes antivirus ne peuvent pas analyser tous les composants contenus dans ce type de cheval de Troie. |
| **Trojan_SMS** | us utilisez votre appareil mobile pour envoyer des SMS à des numéros surtaxés ces programmes peuvent vous coûter cher. |
| **Trojan_Spy** | Ces programmes peuvent suivre les données que vous saisissez au clavier, effectuer des captures d'écran et accéder à la liste des applications en cours d'exécution pour espionner votre utilisation de votre téléphone. |
| **Zero_Day** | Nouveaux programmes non détectés par les solutions anti-malware. |

**Table 4 : Description des catégories des malwares [14].**

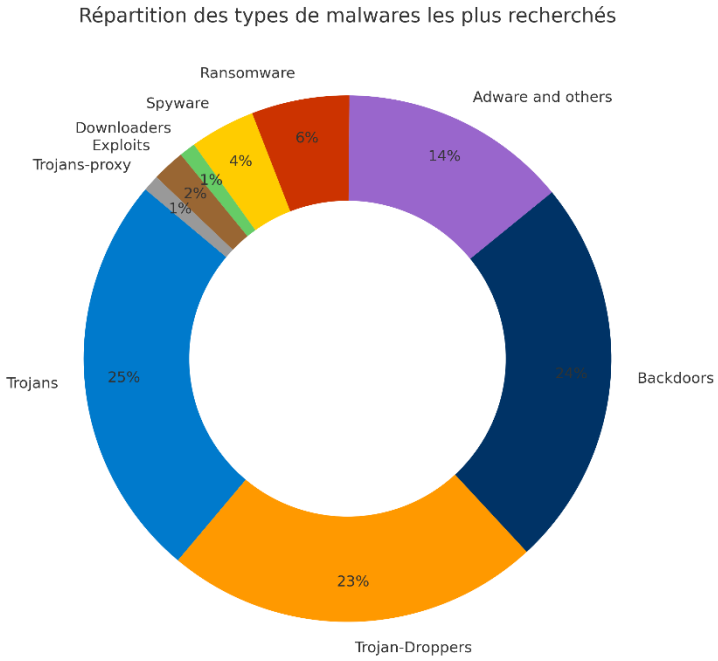



**Figure 6 : Malwares les plus recherchés[15].**

## 2.4 Techniques de détection des logiciels malveillants android :

### 2.4.1 Techniques de détection dynamique :

Certaines applications malveillantes sur les téléphones Android peuvent être détectées par leur comportement lors de leur exécution. Il s'agit de celle-ci : observer ce que l'application a l'habitude de faire en temps réel ; les données qu'elle tente de collecter, les services qu'elle utilise comme la localisation ou l'accès sur le réseau et les actions qu'elle fait sur le système du téléphone. Pour ne pas prendre un vrai risque sur un vrai téléphone, ces tests sont souvent réalisés dans un environnement communautaire et bien isolé . Une étude a prouvé que plus de mille applications dangereuses ont été détectées en utilisant un système qui fait la liste des services utilisés par les applications. Par exemple, certaines applications voulaient récupérer des informations personnelles comme, par exemple, son numéro d'identification de téléphone. D'autres attendaient que l'utilisateur soit intervenu pour aller chercher à nuire, d'autres travaillaient dès l'installation. Il a également été observé que ces applications malveillantes, elles peuvent être à même de ralentir le téléphone dans une certaine mesure. Les autorisations qu'acceptent les utilisateurs sans précautions lors du téléchargement favorisent fréquemment la venue de ces programmes nocifs[16] .

### 2.4.2 Techniques de détection statiques :

Une autre façon d'identifier les malware est de scanner leur contenu sans les faire exécuter. Cette méthode consiste en l'étude des fichiers qui font partie du programme, en particulier de celui contenant les droits demandés (comme l'accessibilité à l'appareil photo ou à la position). En analysant les spécifications, on peut deviner même des signaux d'alerte avant même de lancer l'application. Nous avons notamment vu le jour, par exemple, qu'une étude a utilisé un outil pour ouvrir le fichier, et a réussi à repérer correctement les applications potentiellement malveillantes. Cette méthode repose souvent sur les logiciels qui comparent les logiciels étudiés avec les autres appréciés d'être malveillants. Bien qu'on ne soit pas toujours en mesure de repérer une menace de surface, le fait d'explorer le code de bout en bout permet de détecter les signes de danger. Cela a été vérifié par diverses études utilisant des méthodes de comparaison et d'analyse différentes, quasiment toutes avec un rendement élevé pour différencier les applications qui sont en sécurité de celles qui ne le sont pas [17].

### 2.4.3 Techniques de détection hybrides :

Ce mode hybride offre un moyen de regarder à la fois le contenu de l'application et son comportement. En combinant ces deux aspects, il est toujours plus facile de détecter les programmes nuisibles les plus évasifs. Les études ont démontré que cette méthode avait une très bonne précision. Par exemple, un autre système appelé DroidDetector a été créé à partir de plusieurs techniques pour entraîner la reconnaissance des logiciels baveux. Dans certains cas particulièrement compliqués, il faut alors analyser les applications séparément pour ne pas mélanger les informations. La problématique a également été étudiée par d'autres recherches et montre périodiquement que certains logiciels malveillants soulèvent facilement les systèmes de détection à condition qu'ils ne soient pas trop efficaces. En utilisant des outils de données d'analyse, plusieurs études ont pu déterminer qu'une certaine application permettait de collecter secrètement des informations personnelles, telles que des photos enregistrées, la localisation ou des fichiers présents sur le téléphone. De telles menaces exposent les utilisateurs à de graves risques, notamment le vol d'identité et la surveillance non autorisée [18].

## 2.5 Outils de protection contre les Malwares :

### 2.5.1 Antivirus et antimalware avancés :

Les antivirus sont des logiciels de base pour la protection des smartphones Android contre les infections de programme. Ils aident à scanner le téléphone un peu en même temps pour détecter les applications inquiétantes ou suspectes. De plus, ils surveillent les comportements anormaux et empêchent toute menace avant qu'elle ne cause des dommages. La plupart des antivirus font également l'offre des mises à jour automatiques, ce qui leur permet de tenir en ligne les nouvelles souches. Des applications telles qu'Avast, Kaspersky ou Bitdefender figurent parmi les plus répandues et offrent une sécurité très robuste. Elles sont souvent disponibles en version gratuite, tandis que des licences payantes proposent des fonctionnalités plus avancées pour une protection accrue [13].

### 2.5.2 Pare-feu intelligents :

Un pare-feu est un outil qui permet de contrôler les accès Internet de votre téléphone. Il permet de voir quelles applications sont en lien avec Internet et de bloquer celles qui ne le nécessitent pas. Cela empêche notamment certaines applications dangereuses de vous envoyer vos données personnelles à distance. Bien qu'une telle sécurisation soit souvent intégrée à ses applications de protection, certaines applications aussi spécialisées que possible aident les utilisateurs à mieux aligner les appels sur son appareil. Cela permet de se garder les fuites de données et de surveiller ce que font vraiment les applications installées [13].

### 2.5.3 Réseaux privés virtuels (VPN):

Un VPN est un moyen d'apporter un niveau supplémentaire de sécurité à la connexion internet du smartphone. Il est surtout utile quand il est utilisé sur un Wi-Fi public, comme dans un café ou un hôtel. Le VPN empêche d'autres personnes de voir ce que l'on fait sur le net ou de voler une information comme le mot de passe. Il aide également à cacher l'endroit précis où se trouve l'utilisateur. Des descriptifs de service tels que NordVPN, ExpressVPN, ProtonVPN sont couramment employés pour cette raison. Avec un VPN, la navigation est plus privée et plus sûre [13].

### 2.5.4 Gestionnaires de mots de passe:

Beaucoup de gens utilisent des mots de passe simples ou répètent les mêmes sur plusieurs comptes, ce qui est dangereux. Les gestionnaires de mots de passe permettent de créer des mots de passe forts et différents pour chaque site ou application. Ils les enregistrent dans un espace sécurisé, et l'utilisateur n'a besoin de se souvenir que d'un seul mot de passe principal. Cela rend beaucoup plus difficile l'accès non autorisé à ses comptes. Des applications comme LastPass, Bitwarden ou 1Password sont très pratiques et faciles à utiliser sur Android [13].

## 2.6 Conclusion :

Ce chapitre présente une analyse comparative de différentes techniques de détection de logiciels malveillants sur Android. L'étude a permis d'identifier les limites et les points forts de chaque technique de détection. Les résultats obtenus confirment l'hypothèse selon laquelle les approches de détection conçues pour les logiciels malveillants Android ne garantissent pas une

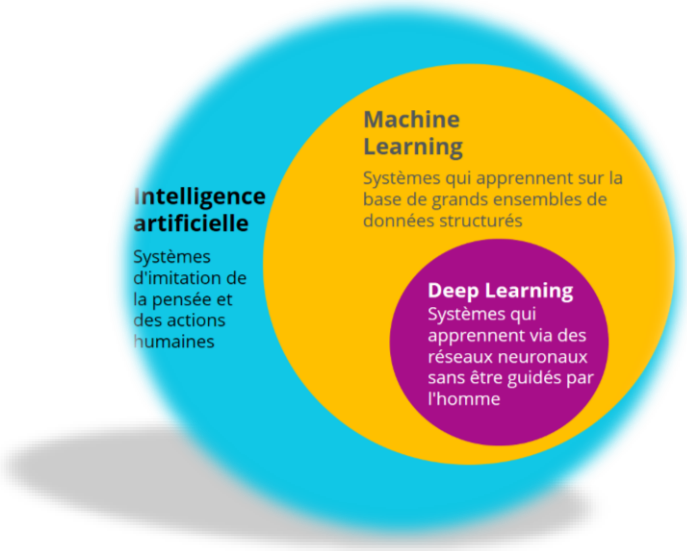
précision de détection optimale. Cette analyse comparative vise à mieux comprendre les forces et les faiblesses identifiées lors de l'étude pour les techniques de détection sélectionnées.

C'est dans ce contexte que le deep learning se révèle particulièrement prometteur. Il apporte une perspective plus flexible et adaptative en utilisant les réseaux de neurones artificiels pour traiter de très grosses quantités de données, offrant ainsi de nouvelles pistes pour surmonter les lacunes des méthodes existantes.

Dans cette optique, il devient essentiel d'explorer les approches d'apprentissage, en particulier le machine learning et le deep learning, qui ont montré un potentiel significatif dans le domaine de la cybersécurité. Le chapitre suivant sera ainsi consacré à la présentation détaillée de ces techniques, en mettant en lumière leur fonctionnement, leurs avantages respectifs et leur application dans la détection intelligente des logiciels malveillants sur Android.

# Chapitre 3

# Techniques d’apprentissage



## 3.1 Introduction :

Depuis le début des années 2000, et particulièrement après 2010, les réseaux de neurones ont subi un véritable flop avec l'apprentissage profond. Yann Lecun a prouvé l'efficacité des couches de convolution dans l'identification des chiffres. Ces progrès ont permis d'entraîner des réseaux de plus en plus nombreux, notamment en raison des progrès offerts par l'univers matériel, comme l'utilisation des GPU, mais encore plus en raison des couches de convolution qui tirent de l'image des caractéristiques abstraites.

L'apprentissage profond, ou deep learning, est un sous-domaine de l'intelligence artificielle et de l'apprentissage automatique qui permet d'extraire, d'analyser et d'interpréter de grandes masses de données. Inspiré du mécanisme neuronal humain, il fonctionne grâce à des réseaux de neurones artificiels interconnectés qui traitent l'information par des calculs mathématiques et des algorithmes complexes, simulant la transmission de signaux pour apprendre et reconnaître des modèles.

Cette technologie permet de mieux prendre des décisions en repérant des modèles très subtils dans de très grands ensembles de données. Elle ouvre ainsi de nombreuses perspectives pour les professionnels de divers domaines tels que la finance, ou la médecine. Elle facilite aussi l'amélioration de l'expérience utilisateur en analysant les comportements individuels. Les réseaux neuronaux peuvent reconnaître des modèles complexes, résoudre des modèles complexes, résoudre des problèmes ardus et agir de manière autonome.

L'apprentissage profond est très populaire dans d'autres applications comme la vision par ordinateur, la reconnaissance de la voix, la traduction automatique, le diagnostic médical et la prédiction des marchés financiers. Sa croissance rapide élargit en ouverture de nouveaux horizons pour les entreprises pour résoudre des problèmes complexes, agrandir l'exploit opérationnel et pousser l'innovation.

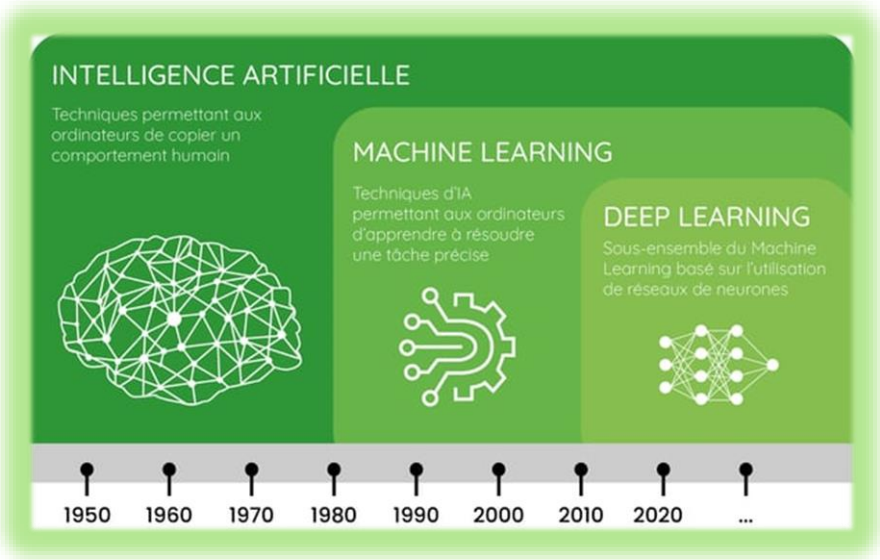



**Figure 7 : Différence entre IA,Deep learning,Machine learning en agriculture[19]**

## 3.2 Apprentissage automatique (ML) :

### 3.2.1 Définition :

Le Machine Learning, ou Apprentissage automatique, est un domaine interdisciplinaire qui combine la statistique et l'informatique. Bien que le concept soit plus ancien, le terme a été popularisé par Arthur Samuel en 1959 (plutôt que 1952). L'objectif principal du Machine Learning est de permettre aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances sur des tâches spécifiques sans être explicitement programmés pour chaque scénario. En d'autres termes, les systèmes développent leur propre "expérience" et leur capacité à s'adapter, un peu comme le font les humains.

C'est aujourd'hui l'un des secteurs technologiques de pointe qui connaît la croissance la plus rapide, et il est au cœur de l'intelligence artificielle (IA) et de la science des données.

Le machine learning est utilisé essentiellement pour aider les systèmes informatiques à traiter efficacement de grands volumes de données, et surtout que ces données ne sont pas directement compréhensibles. Suite à l'explosion des données, le rôle du machine learning s'est considérablement renforcé, et de nombreuses industries l'intègrent pour obtenir des informations pertinentes de ces données.

L’objectif principal du machine learning est de tirer des conclusions à partir des données dans le but de répondre à des problèmes complexes [20].

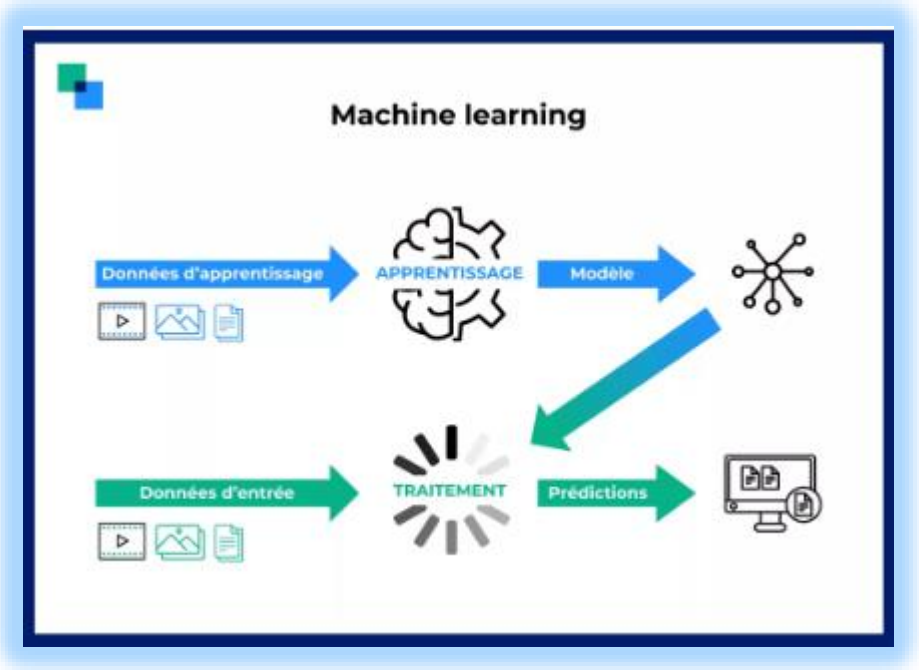



**Figure 8 : Fonctionnement du Machine learning[21]**

### 3.2.2 Types d’Apprentissage Automatique :

Il y a plusieurs catégories d’apprentissage automatique, chacune adaptée à des problèmes et des types de données différents. Voici les trois catégories les plus importantes :

#### 3.2.2.1 Apprentissage supervisé :

L'apprentissage supervisé est la forme d'apprentissage automatique la plus courante et la plus largement utilisée. Dans cette technique, les algorithmes sont entraînés à l'aide d'un ensemble de données étiquetées, où chaque donnée d'entrée est associée à la sortie souhaitée correspondante.

Le but est d'entraîner le modèle en lui présentant ces exemples afin qu'il puisse ensuite prédire ou prendre des décisions de manière fiable lorsqu'il est confronté à de nouvelles données inconnues [22].

#### 3.2.2.2 Apprentissage non supervisé :

L’apprentissage non supervisé est fait par des machines d’apprentissage automatique avec un apprentissage dans lequel les algorithmes sont formés sur des données sans étiquettes.

On a à l'opposé de l'apprentissage supervisé dans laquelle chaque entrée se fiche de sa sortie connue, l'objectif de l'apprentissage non supervisé est de trouver des modèles ou des structures cachées, des donnees. Cela peut simplifier la réduction de regroupement données dans la catégorie ou diminuer le nombre de dimensions datasets pour la fréquentation et la visualisation. Les deux principales méthodes d'apprentissage non supervisé sont les méthodes de regroupement (clustering anglais) et les règles d'association[22].

### 3.2.2.3 Apprentissage semi-supervisé :

L'apprentissage non supervisé est une catégorie de l'apprentissage automatique où les algorithmes sont entraînés sur des données non étiquetées. Contrairement à l'apprentissage supervisé, où chaque donnée d'entrée est associée à une sortie connue, l'objectif de l'apprentissage non supervisé est de découvrir des modèles ou des structures cachées au sein des données.

Cette approche permet de simplifier les données en effectuant du regroupement (clustering), c'est-à-dire en classant les données en catégories, ou en réduisant le nombre de dimensions des jeux de données (datasets) pour en faciliter l'interprétation et la visualisation [22].

### 3.2.2.4 Apprentissage par renforcement :

L'apprentissage par renforcement est un modèle d'apprentissage dans lequel un agent apprend à interagir avec son environnement et réalise une performance périodique après chaque action réussie. Le modèle d'apprentissage par récompense apprend au fur et à mesure des essais et des erreurs. Une série de résultats positifs rend le modèle apte à trouver la meilleure recommandation pour un problème donné. Cette méthode est couramment utilisée dans le monde du jeu, de la navigation, de la robotique, etc. [23].

### 3.2.3 Difference entre les méthodes supervisées :

| Apprentissage supervisé | Apprentissage non supervisé | Apprentissage semi-supervisé |
|---|---|---|
| Dans cette approche, toutes les données utilisées pour entraîner le modèle sont étiquetées, formant ainsi un jeu de données où chaque exemple est associé à une réponse ou à un label correct. Le modèle apprend en comparant ses prédictions avec ces labels pour ajuster ses paramètres. L'apprentissage supervisé est très efficace lorsque de grandes quantités de données étiquetées sont disponibles, mais il devient limité lorsque l'annotation manuelle des données est coûteuse ou difficile. | Contrairement à l'apprentissage supervisé, l'apprentissage non supervisé n'utilise aucune donnée étiquetée. Le modèle tente de trouver des structures sous-jacentes dans les données, telles que des groupes ou des motifs. Les algorithmes non supervisés sont souvent utilisés pour des tâches comme le clustering ou la réduction de dimensionnalité. Cependant, cette méthode ne permet pas d'associer directement des labels aux données, ce qui limite son utilisation pour des tâches de classification ou de prédiction. | L'apprentissage semi-supervisé combine les deux approches. Il s'appuie sur un petit ensemble de données étiquetées, qui guide l'apprentissage du modèle, tout en exploitant une grande quantité de données non étiquetées pour améliorer la généralisation et la performance. Cette méthode réduit la dépendance à des données entièrement annotées et permet au modèle d'apprendre à partir de la structure des données non étiquetées tout en s'appuyant sur des exemples étiquetés pour affiner les prédictions. |

**Table 5 : Différence entre les méthodes supervisées[24]**

### 3.2.4 L'utilisation de l'apprentissage automatique :

Les logiciels d'apprentissage automatique nous entourent souvent en arrière-plan pour améliorer notre vie quotidienne. Voici quelques exemples concrets comme tels :

- **Systèmes de recommandation :**

Les systèmes de recommandation sont l'une des principales applications de l'apprentissage automatique. Des sociétés telles que Netflix et Amazon utilisent l'apprentissage automatique pour étudier votre comportement et vous proposer des produits ou des films qui pourraient vous plaire[24].

- **Assistants vocaux :**

Les assistants vocaux comme Siri, Alexa et Google Assistant utilisent l'apprentissage machine pour traduire vos ordres vocaux et répondre par des réponses appropriées[24].

- **Détection de la fraude :**

Les banques et les sociétés de cartes de crédit utilisent l'apprentissage automatique pour identifier les paiements frauduleux. Ils peuvent aussi déclencher les activités suspectes en temps réel[24].

- **Médias sociaux :**

Les plateformes de médias sociaux s'appuient sur l'apprentissage automatique pour faire beaucoup de choses, de la personnalisation de votre flux à la filtration des contenus inappropriés[24].

### 3.2.5 Modèles d'apprentissage automatique :

Voici une liste des modèles de classification utilisés le plus souvent pour la détection des malwares Android :

#### 3.2.5.1 Réseau neuronal artificiel (RNA) :

Un réseau de neurones artificiels (RNA) est une structure de calcul qui s'inspire de la manière dont les neurones sont organisés et fonctionnent dans le cerveau humain. Cette structure est composée d'un grand assemblage d'unités (ou "neurones") interconnectées, capables de traiter l'information. Les RNA peuvent théoriquement modéliser des fonctions complexes, notamment des relations non linéaires entre les données d'entrée et de sortie.

Cependant, en raison de leur complexité structurelle et de la quantité de paramètres à ajuster, leur entraînement peut effectivement être long. Historiquement, les RNA étaient souvent entraînés à l'aide de l'algorithme de rétro-propagation du gradient. Cet algorithme, bien qu'efficace pour les réseaux de taille modeste, rencontrait des difficultés (comme le problème des gradients évanescents ou explosifs) lorsqu'il était appliqué à des réseaux très profonds (ayant de nombreuses couches).

C'est pourquoi, par le passé, un RNA "classique" était souvent qualifié de modèle peu profond, par opposition aux architectures plus récentes de l'apprentissage profond (Deep

Learning) qui sont spécifiquement conçues pour gérer et entraîner efficacement des réseaux avec un grand nombre de couches cachées[25].

### 3.2.5.2 Support Vector Machine (SVM) :

Un Support Vector Machine ou Machine à vecteur de support est un Machine Learning supervisé utilisé pour la classification, la régression, et la détection d'anomalies. Conçues par Vladimir Vapnik dans le courant des années 1990, les machines de vecteurs de support se basent sur le principe de séparer les données en classes en utilisant une frontière aussi simple que possible. Elle va donc maximiser la distance (ou marge) entre les différents points de données les plus proches de chaque classe (clusters) et leur frontière.

On surnomme d'ailleurs les SVMs « des séparateurs à large marge ». Les points de données les données les plus proches de cette frontière sont les vecteurs des supports. De manière plus précise, ces algorithmes tentent de trouver l'hyperplan maximisant cette marge tout en séparant une classe d'une autre. C'est cette notion de séparation maximale avec une ouverture large qui confère aux SVM leur excellente capacité à généraliser à de nouvelles données, tout en conservant une bonne "mémoire" des données d'entraînement [25].

### 3.2.5.3 K-means :

K-means est une méthode de partitionnement utilisée pour regrouper les données en k clusters (ou groupes). Pour l'utiliser, l'algorithme nécessite de spécifier au préalable le nombre initial de groupes (k) souhaité.

Le processus fonctionne de manière itérative : chaque cluster est défini par un centroïde (sa "moyenne" ou centre). L'algorithme vise à réduire la distance entre chaque point de données et le centroïde de son cluster, assignant ainsi les points au cluster dont le centroïde est le plus proche. Ce processus est répété jusqu'à ce que les affectations des points aux clusters ne changent plus significativement, ou qu'un critère de convergence soit atteint [25].

## 3.3 Apprentissage profond (DL) :

### .3.1 Définition :

L'apprentissage profond ( Deep Learning) est une branche de l'apprentissage automatique qui vise à exploiter l'abstraction de données de haut niveau. Pour ce faire, il utilise plusieurs

couches de traitement composées de structures complexes qui effectuent des transformations non linéaires successives sur les données. Au cœur de l'apprentissage profond se trouvent les réseaux neuronaux multicouches (ou "réseaux de neurones profonds"). Le principe est que les sorties d'une couche de neurones servent d'entrées à la couche suivante. Cette architecture en cascade permet au modèle d'apprendre progressivement des caractéristiques de plus en plus abstraites et sophistiquées à partir des données d'entraînement. C'est cette capacité à représenter les données à différents niveaux d'abstraction qui rend l'apprentissage profond si puissant pour des tâches complexes comme la reconnaissance d'images, le traitement du langage naturel ou la détection de fraudes[26] .

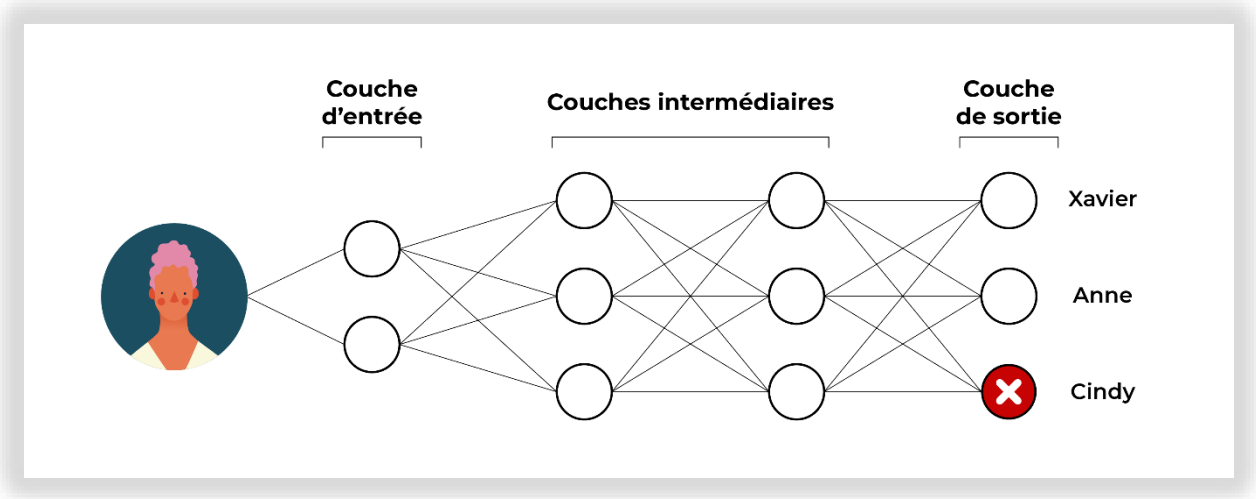



**Figure 9 : Fonctionnement du Deep learning[27]**

### 3.3.2 Fonctionnement :

Les réseaux neuronaux profonds sont caractérisés par la présence de plusieurs couches cachées de neurones, interconnectées de manière contiguë, qui s'étendent de la couche d'entrée jusqu'à la couche de sortie. Chaque couche traite l'information et transmet ses résultats à la couche suivante, permettant ainsi une progression du traitement des données. C'est ce cheminement, où les informations traversent le réseau de la couche d'entrée vers la couche de sortie pour produire des prédictions ou des classifications, qui est appelé la propagation avant (forward propagation).

Un réseau neuronal, qu'il soit profond ou non, possède deux types de couches dites "visibles" :

- La couche d'entrée (Input Layer) : C'est la première couche du réseau. Son rôle est de recevoir et de traiter les données brutes initiales qui sont fournies au modèle.
- La couche de sortie (Output Layer) : C'est la dernière couche du réseau. À partir de cette couche, le modèle produit son résultat final, qu'il s'agisse d'une prédiction (par exemple, une valeur numérique) ou d'une classification (par exemple, la catégorie à laquelle appartient une donnée).

Entre ces couches, chaque paire de couches contiguës est liée par des poids (weights). Ces poids sont des paramètres numériques qui définissent l'importance et la force des connexions entre les neurones. Leur ajustement lors de l'entraînement est crucial car il permet au modèle d'optimiser son apprentissage et d'améliorer sa capacité à faire des prédictions ou des classifications précises [28].

### 3.3.3 Classification des méthodes Deep Learning :

La classification occupe un poste important dans toutes les sciences et techniques utilisant la répartition multidimensionnelle, et ces techniques se construisent de trois manières selon sa formation et son intention d'utilisation :

#### 3.3.3.1 Apprentissage supervisé :

En apprentissage supervisé, on a un corpus de données annoté, où chaque exemple a été signalé par un enseignant ou un expert grâce à une classe. Cet ensemble d'exemples constitue la matière de base d'apprentissage. Les techniques d'apprentissage supervisé ont pour objectif de créer des fonctions de classification à partir de cette base d'apprentissage. Ces fonctions permettent, depuis la description d'un objet, de repérer un attribut particulier(29].

#### 3.3.3.2 Apprentissage non supervisé :

L'apprentissage non supervisé consiste à entraîner des modèles à partir de données non étiquetées. Dans cette approche, les algorithmes sont chargés de découvrir par eux-mêmes des structures, des motifs ou des similarités au sein des données, sans aucune intervention humaine préalable pour leur indiquer les "bonnes" réponses ou les catégories. Ils regroupent les données qui se ressemblent intrinsèquement [29].

### 3.3.3.3 Apprendre par renforcement :

L'apprentissage par renforcement est une méthode très puissante et fréquemment utilisée. Elle repose sur le principe de laisser les machines apprendre de leurs propres erreurs et succès, grâce à un système de récompenses (rewards) et de punitions (punishments). L'agent apprend ainsi de manière itérative à maximiser les récompenses cumulées sur le long terme.

Il est vrai que certains chercheurs considèrent l'apprentissage par renforcement comme une voie prometteuse pour le développement d'une intelligence artificielle générale (AGI), capable d'une intelligence similaire à celle de l'homme, en lui permettant d'apprendre et de s'adapter à une grande variété de tâches complexes[29].

## 3.3.4 Différents algorithmes de Deep Learning :

Plusieurs types d'algorithmes sont utilisés en apprentissage profond. Chaque algorithme possède ses propres spécificités et applications.

### 3.3.4.1 Réseaux neuronaux convolutifs (RNC) :

Également connus sous le nom de réseaux neuronaux convolutifs (CNN), ils se composent de plusieurs couches chargées de traiter les données et d'extraire leurs caractéristiques. Les réseaux neuronaux convolutifs sont spécifiquement utilisés pour l'analyse et la détection d'objets. Ils peuvent être utilisés, par exemple, pour la reconnaissance d'images satellites, le traitement d'images médicales, la détection d'anomalies ou la prévision de séries chronologiques[30].

### 3.3.4.2 Réseaux neuronaux récurrents (RNR):

Les réseaux neuronaux récurrents (RNR) se caractérisent par des connexions qui forment des cycles dirigés, leur permettant de conserver des informations sur les entrées précédentes et d'intégrer un "état" ou une "mémoire" au fil du temps. Cela les rend particulièrement adaptés au traitement des données séquentielles. Des variantes telles que les Long Short-Term Memory (LSTM) ou les Gated Recurrent Units (GRU) ont été développées pour améliorer leur capacité à capturer des dépendances à long terme. En pratique, les RNR sont largement utilisés dans des domaines comme le traitement du langage naturel (par exemple, la traduction automatique, l'analyse de sentiments, la reconnaissance vocale) et la prédiction de séries temporelles [30].

### 3.3.4.3 Réseau neuronaux profonds (RNP ):

Un réseau neuronal profond (RNP), ou Deep Neural Network (DNN), est un réseau neuronal artificiel composé d'une couche d'entrée, d'une couche de sortie et d'un grand nombre (au moins plusieurs) de couches cachées intermédiaires. La "profondeur" du réseau est directement liée au nombre de ces couches cachées. Chaque couche intermédiaire est conçue pour apprendre et extraire des caractéristiques de plus en plus complexes et abstraites à partir des données d'entrée. Ce processus d'apprentissage hiérarchique permet au réseau de comprendre des motifs complexes dans les données, conduisant à des capacités de représentation et de prédiction avancées [30].

## 3.3.5 L'utilisation de l'apprentissage profond :

- **Création de textes :**

  De plus le deep learning est aussi un avantage dans la génération de contenu. Et en effet, voilà, un ordinateur est désormais capable sans la moindre aide

  humaine de rédiger pour lui tout seul des textes, ou de traduire[31].

- **Assistants vocaux :**

C'est également le cas des assistants vocaux, tels que Siri, Alexa ou Google Home. Ceux-ci se fondent sur la technologie du deep learning pour développer leur compréhension du langage et leur vocabulaire[31].

- **Recherche d'objets dans les images :**

L'apprentissage profond a fait des progrès spectaculaires dans des domaines comme la détection d'objets, la reconnaissance faciale, la classification d'image ,etc. Ces modèles peuvent être entraînés sur de grandes bases de données annotées afin de les rendre plus précis et capables de généraliser à de nouvelles images [31].

- **Cybersécurité :**
- De même, il est utilisé en sécurité informatique pour identifier les évènements documentés et les risques inconnus. Il permet en effet une détection d'anomalie et renforce les sécurités.
- De plus, l'apprentissage profond est largement utilisé dans le secteur industriel, que ce soit en robotique ou en matière de maintenance [31].

## 3.4 Comparaison entre le machine learning et le Deep Learning :

| | machine learning | Deep Learning |
|---|---|---|
| Structure | Données structurées | Données non structurées |
| Taille de l'enregistrement | Petit à grand | Large (plus d'un million d'unités de données) |
| Matériel informatique | Fonctionne avec un matériel simple | Nécessite des ordinateurs puissa (avec GPU). Les réseaux neurona multiplient des matrices qui nécessitent beaucoup de temps de calcul - les GPU accélèrent le processus. |
| Temps d'exécution | De quelques minutes à quelques Heures | Jusqu'à des semaines et des mo car les réseaux neuronaux artificiels doivent calculer une énorme quantité de données |
| Domaine d'application | Tâches de routine simples | Tâches complexes |

**Table 6 : Comparaison entre le deep et le machine learning[32]**

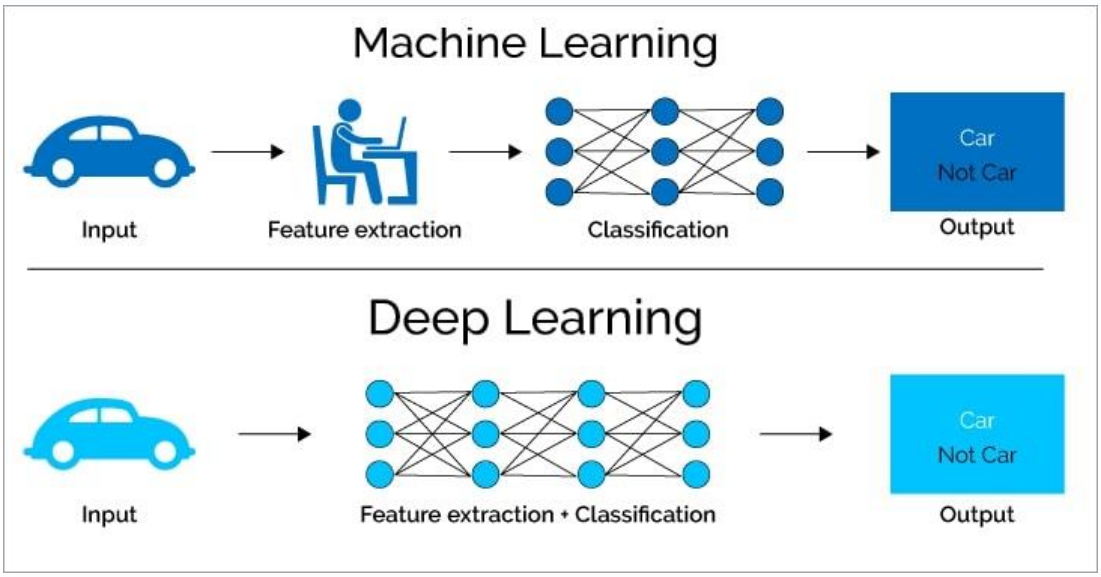



**Figure 10 : La différence entre le machine et deep learning[33] .**

## 3.5 Conclusion :

Le deep Learning constitue un progrès important en matière d'intelligence artificielle, celui des machines à assimiler des représentations des données complexes, hiérarchiques et automatiques.

Agrégeant des capacités de traitement de grandes masses de données non structurées, le deep Learning a permis de faire des applications innovantes dans divers domaines comme la vision par ordinateur, la reconnaissance vocale, l'apprentissage automatique du langage, la santé, l'argent, etc.

Dans le secteur plus spécifique de la cybersécurité, pour la détection de malwares sur les applications de type Android, de même. En appliquant des méthodes intelligentes d'apprentissage automatique, les modèles de deep learning peuvent observer le comportement des applications et identifier des actions display, afin de renforcer la sécurité de vos bénéfices. La mise en place d'un modèle basé sur deeplearning pour le filtrage des malwares dans les applications Android est une révolution en cybersécurité.

Le chapitre suivant présente un travail pratique réalisé sur la plateforme Kaggle, où un ensemble de données dédié à la détection de malwares Android a été traité. Plusieurs algorithmes d'apprentissage ont été appliqués, évalués et comparés afin d'illustrer concrètement l'efficacité des techniques étudiées dans un contexte réel.

# CHAPITRE 4

# Implémentation et discussion des résultats

## 4.1 Introduction :

Face à la complexité et à l'ampleur croissantes des logiciels malveillants, les solutions traditionnelles, telles que les antivirus basés sur les signatures, ne suffisent plus à assurer une protection efficace. De nombreuses attaques passent inaperçues, exposant les systèmes à des risques importants. Dans ce contexte, la sécurité des systèmes informatiques est devenue une préoccupation majeure, tant pour les particuliers que pour les entreprises. Une seule faille peut compromettre des données sensibles et entraîner des pertes importantes.

C'est pourquoi l'intégration des techniques d'apprentissage automatique et d'apprentissage profond est devenue essentielle. Ces approches permettent de concevoir des systèmes intelligents capables de détecter des comportements anormaux grâce à des modèles d'apprentissage. L'objectif de ce travail est donc de développer une preuve de concept pour la classification des logiciels malveillants basée sur des classificateurs robustes.

Parmi toutes les caractéristiques extraites des fichiers, seules les plus pertinentes seront retenues pour l'entraînement du modèle. L'algorithme offrant la meilleure précision distinguera efficacement les fichiers malveillants des fichiers légitimes, tout en minimisant le taux d'erreur. Pour évaluer les performances des différents modèles testés, plusieurs indicateurs seront utilisés : précision, rappel, score F1, ect…. Cette évaluation complète permettra de juger la pertinence des approches utilisées dans la lutte contre les menaces numériques.

## 4.2 Environnement d'exécution :

### 4.2.1 Plateforme Kaggle :

Pour le développement de notre projet et dans une démarche de science des données, nous avons opté pour la plateforme Kaggle, car fournit un environnement complet de développement en ligne, appelé Kaggle Kernels, qui permet d'écrire et d'exécuter du code Python sans aucune configuration préalable. Cette plateforme met également à disposition des ressources matérielles puissantes (CPU et GPU) de manière gratuite. Grâce à cet environnement, nous avons pu :

-Rechercher et importer des jeux de données publics.

-Utiliser des bibliothèques populaires en data science et IA.

-Entraîner nos modèles dans un environnement collaboratif, Visualiser et partager nos résultats avec la communauté.

Kaggle simplifie également l'accès à des bibliothèques puissantes via un

environnement prêt à l'emploi.

### 4.2.2 Bibliothèques Python utilisées :

Plusieurs bibliothèques ont été utilisées pour assurer le bon déroulement des différentes étapes :

**Pandas :** pour le chargement, le nettoyage et la manipulation des données tabulaires ;

**NumPy :** pour les opérations mathématiques et matricielles ;

**Matplotlib et Seaborn :** pour la visualisation des données et des résultats ;

**Scikit-learn :** pour la mise en œuvre des modèles de machine learning classiques (Random Forest, SVM, etc.) et les métriques d'évaluation ;

**TensorFlow et Keras :** pour la création, l'entraînement et l'évaluation des modèles de deep learning ;

**Os :** pour la gestion des chemins de fichiers, le chargement dynamique des répertoires, ou encore l'exécution de certaines tâches système utiles dans l'environnement Kaggle .

## 4.3 Mesures d'évaluation d'un modèle :

### 4.3.1 Accuracy (Précision globale) :

C'est le pourcentage de bonnes prédictions faites par le modèle sur

l'ensemble des données.

$$\text{Accuracy} = \frac{\text{Nombre de bonnes prédictions}}{\text{Nombre total de prédictions}}$$

### 4.3.2 Recall (Rappel / Sensibilité) :

C'est la capacité du modèle à retrouver tous les éléments pertinents d'une

classe donnée (ex. : tous les malwares).

$$\text{Recall} = \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux négatifs}}$$

### 4.3.3 F1-score

C'est la moyenne harmonique entre la précision et le recall, donnant un

équilibre entre les deux.

$$\text{F1-score} = 2 \times \frac{\text{Précision} \times \text{Recall}}{\text{Précision} + \text{Recall}}$$

## 4.4 Aperçu du Dataset:

Le dataset utilisé est accessible publiquement sur Kaggle et s'intitule :

**"android-malware-detection"**

Ce jeu de données contient des exemples d'applications Android répartis en quatre catégories principales :

**Android_Adware :** logiciels qui affichent des publicités excessives ou malveillantes à des fins lucratives .

**Android_Scareware :** applications qui tentent d'effrayer l'utilisateur avec de fausses alertes pour lui faire installer d'autres applications malveillantes ou payer pour des services inutiles .

**Android_SMS_Malware :** applications qui envoient à l'insu de l'utilisateur des SMS vers des numéros surtaxés .

**Benign :** applications saines, non malveillantes, servant de référence pour entraîner les modèles à faire la distinction.

Le jeu de données comprend plusieurs milliers d'échantillons, chacun étant associé à un ensemble de caractéristiques extraites statiquement de fichiers APK : appels aux API, taille de l'application, présence d'instructions suspectes, etc..

## 4.5 Préparation des données :

### 4.5.1 Réduction des dimensions :

Certaines caractéristiques du dataset sont redondantes ou peu informatives. Pour améliorer l'efficacité de l'apprentissage , nous avons appliqué une réduction dimensionnelle :

| Label | Nombre de malwares |
|---|---|
| Android_Adware | 50000 |
| Android_Scareware | 20000 |
| Android_SMS_Malware | 20000 |
| Benign | 20000 |

**TABLE 7 :Repartition des données de Dataset**

### 4.5.2 Prétraitement des données:

Cette étape représente l'un des éléments essentiels du processus : le prétraitement, le traitement et l'application de toutes les données sont réalisés avant leur utilisation par le réseau de neurones profond. Les étapes sont les suivantes :

Chaque étiquette sous forme de chaîne de caractères est remplacée par une valeur numérique :

- « Android_Adware » est remplacé par 0
- « Android_Scareware » est remplacé par 1
- « Android_SMS_Malware »est remplacé par 2
- « Benign » est remplacé par 3

La plupart des modèles de machine learning ne peuvent pas traiter directement les étiquettes sous forme de chaînes de caractères et ont besoin d'une entrée numérique.

5 Encodage des colonnes catégorielles en valeurs numériques

Transformation des variables de type texte (catégories) en valeurs numériques à l'aide d'encodeurs comme LabelEncoder ou OneHotEncoder.

6 Traitement des valeurs manquantes

Remplacement des valeurs nulles ou manquantes à l'aide de méthodes comme fillna() (par la moyenne, la médiane, une constante, etc.), ou suppression des lignes/colonnes concernées avec dropna().

7 Suppression des doublons

Élimination des lignes identiques avec drop_duplicates() pour éviter la redondance dans les données.

8 Normalisation des données

Mise à l'échelle des variables numériques (avec StandardScaler, par exemple) pour que toutes aient une moyenne de 0 et un écart-type de 1.

9 Division du dataset en ensembles d'entraînement et de test

Séparation des données (par exemple avec train_test_split) afin d'entraîner le modèle sur une partie et de l'évaluer sur une autre.

## 4.6 Implémentation et résultats :

Au cours de cette étude, le but est de réduire les taux d'erreur tout en augmentant les niveaux de précision possibles. Dans ce but plusieurs essais pour trouver les paramètres optimaux du modèle.

Une fois le modèle développé , avec un taux d'erreur minimal et une précision maximale, nous l'avons appliqué à un sous-ensemble restreint des données. Les résultats de ces tests sont illustrés dans les figures suivantes.

Cette figure représente la répartition des différents types de malwares Android détectés par ce modèle :

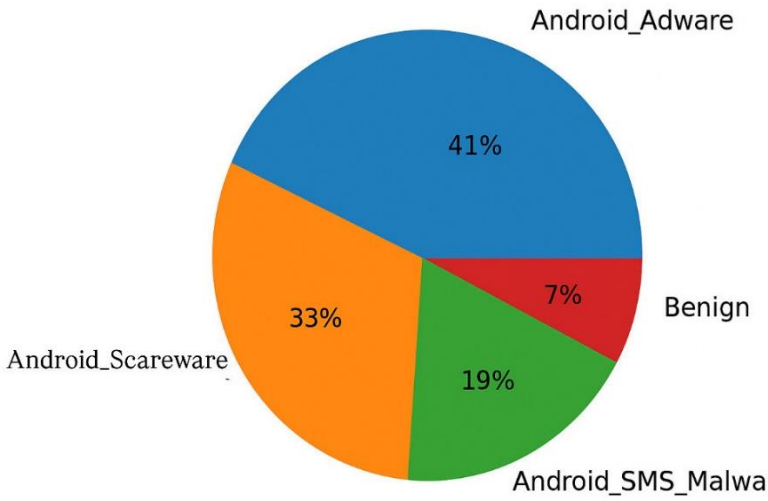



**Figure 11: Répartition des malwares du Dataset**

## 4.7 Algorithmes utilisés :

### 4.7.1 Machine learning :

Dans le cadre de cette recherche, une approche comparative a été adoptée afin d'évaluer les performances de différents algorithmes de Machine Learning appliqués à la détection de logiciels malveillants sous Android. Le script développé vise à identifier l'algorithme le plus performant en termes de précision. Plusieurs algorithmes d'apprentissage supervisé ont été retenus pour cette comparaison d'algorithme : Random Forest , Naive Bayes, XGBOOST et K-Nearest Neighbors (KNN). Nous avons choisi ces algorithmes parce qu'ils sont populaires, efficaces dans de nombreuses taches de classification, et complémentaires avec la différence de complexité et un mécanisme de décision.

Chaque algorithme est entièrement entraîné et évalué sur le même ensemble de données, en utilisant les mêmes pré-traitements . Les performances ont été évaluées grâce à des métriques standards notamment l'**accuracy**, le **f1-score**, le **recall** et les **matrices de confusion** .

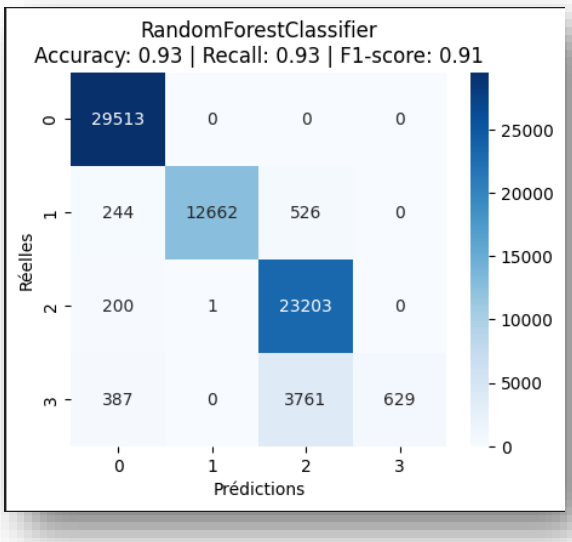



**Figure12 : Matrice de confusion du RandomforestClassifier**

Cette figure représente la matrice de confusion d'un modèle RandomForestClassifier utilisé pour une tâche de classification comportant quatre classes. Les lignes indiquent les classes réelles et les colonnes les classes prédites. On observe que la majorité des prédictions sont correctes, notamment pour les classes 0 et 2, qui présentent respectivement 29 513 et 23 203 bonnes classifications. Les erreurs de classification sont visibles dans les cellules hors diagonale, par

exemple 526 instances de la classe réelle 1 ont été prédites comme classe 2. Les métriques globales affichées en haut du graphique montrent une très bonne performance du modèle, avec une accuracy de 0,93, un rappel (recall) de 0,93 et un F1-score de 0,91, cela qui indique que le modèle est à la fois précis et robuste pour cette tâche.

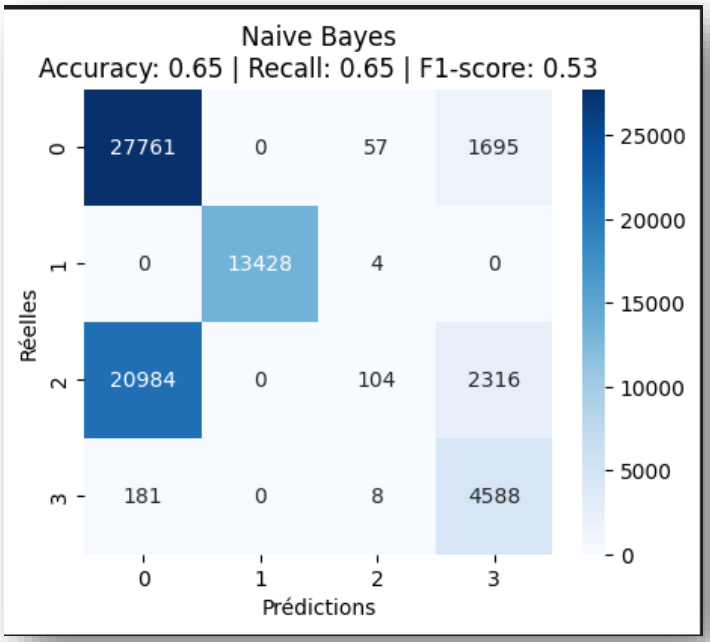



**Figure13 : Matrice de confusion du Naive bayes**

Cette Figure présente les résultats d'un classifieur **Naive Bayes** avec une précision globale (accuracy) de 65 %, un recall de 65 % et un F1-score de 0,53, cela qui indique une performance moyenne avec un déséquilibre entre l'accuracy et le recall . La matrice de confusion partiellement visible montre que certaines classes sont bien classées (comme la première avec 27 761 bonnes prédictions), tandis que d'autres ont très peu d'échantillons (comme la dernière ligne avec seulement quelques valeurs), Cela suggère un possible déséquilibre des données. Globalement, le modèle semble performant sur les classes majoritaires mais moins efficace sur les autres classes, ce qui explique le F1-score relativement faible.

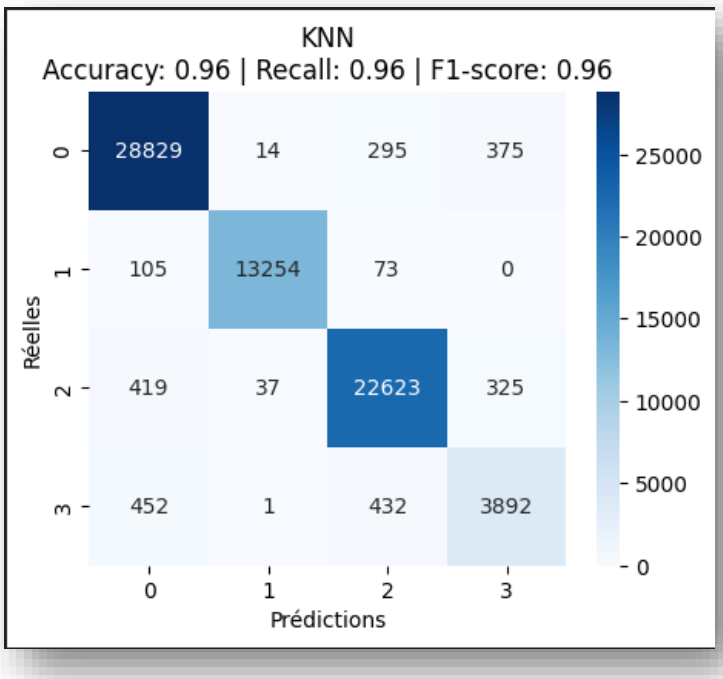



**Figure14 : Matrice de confusion du KNN**

Ce résultat montre que l'algorithme KNN (K-Nearest Neighbors) a obtenu d'excellentes performances avec une précision, un rappel et un F1-score tous à 0.96, cela signifie qu'il classe correctement 96% des données tout en maintenant un bon équilibre entre la détection des vrais positifs et la réduction des faux positifs. La matrice de confusion partiellement visible révèle que la plupart des prédictions tombent sur la diagonale principale (comme 28829, 13254 et 22623), indiquant des classifications correctes, tandis que les petites valeurs hors diagonale (comme 105 ou 419) représentent des erreurs mineures. Le KNN fonctionne remarquablement bien sur ces données.

#### 4.7.1.1 Analyse Comparative des Modèles :

10 Les matrices de confusion des modèles **Random Forest**, ,**K-Nearest Neighbors (KNN)** et **Naive Bayes** nous nous offrent une excellente vue de leur efficacité à détecter les malwares Android.

11 Pour **Random Forest**, on observe une bonne performance. La matrice montre que le modèle réussit à bien distinguer les malwares des applications normales, en limitant les erreurs.

12 Le modèle **KNN** fait encore mieux . Sa matrice de confusion indique qu'il commet très peu d'erreurs, notamment en détectant presque tous les malwares, ce qui est essentiel pour la sécurité. Même si KNN peut être sensible à certains points bruyants, ici il semble très bien adapté aux données.

**13** Enfin, **Naive Bayes** présente des résultats un peu moins convaincants. Comme il suppose que chaque caractéristique agit indépendamment, ce qui n'est pas toujours vrai pour ce type de données, il a un peu plus de mal à bien classer. Cependant, sa simplicité et sa rapidité peuvent être des atouts utiles selon les scénarios.

**14** 

### 4.7.2 Deep learning :

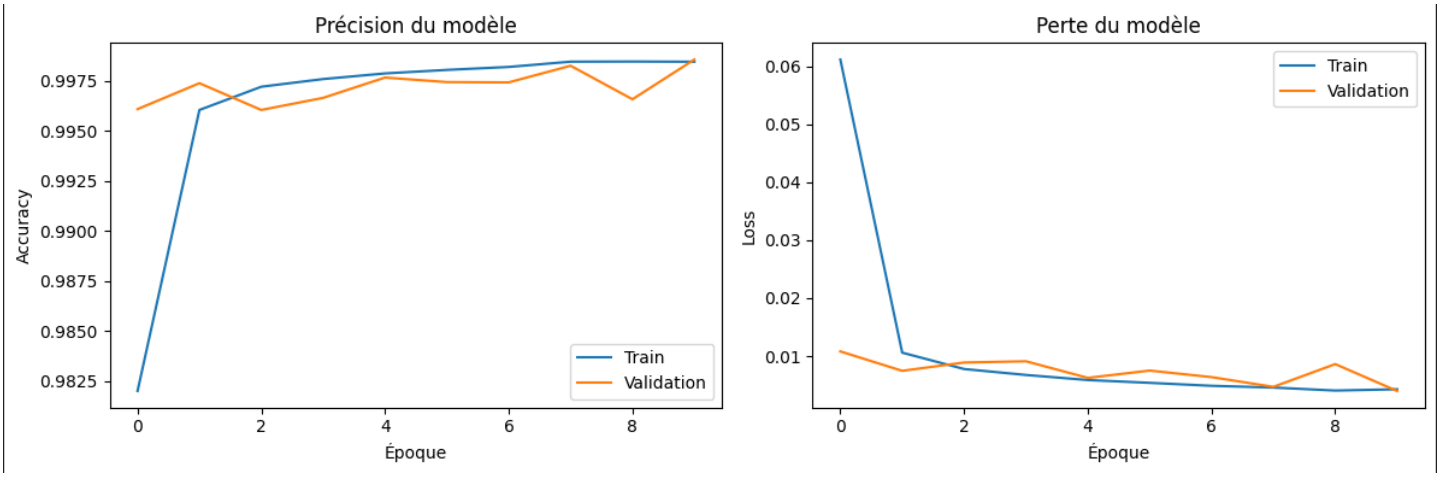



**Figure 15 : Modèle accuracy & loss du LSTM**

Les deux graphiques montrent l'évolution des performances d'un modèle LSTM au fil de 10 époques. À gauche, la courbe de précision (accuracy) indique que le modèle atteint très rapidement une précision supérieure à 99,5 % dès la deuxième époque. Cela est vrai aussi bien sur les données d'entraînement (Train) que de validation (Validation), et la précision se stabilise autour de 99,75 %. Cela signifie que le modèle fait très peu d'erreurs dans ses prédictions, tant sur les données qu'il a vues pendant l'entraînement que sur des données nouvelles. À droite, la courbe de perte (loss) montre une forte diminution dès la première époque, passant de plus de 0,06 à moins de 0,01. Par la suite , la perte reste très faible et stable pour les deux ensembles. L'absence d'écart significatif entre les courbes d'entraînement et de validation sur les deux graphiques montre que le modèle ne surapprend pas (pas d'overfitting) et qu'il généralise bien. En résumé, ce LSTM apprend très vite, atteint une excellente performance et reste stable sans signe de dégradation ou de surapprentissage.

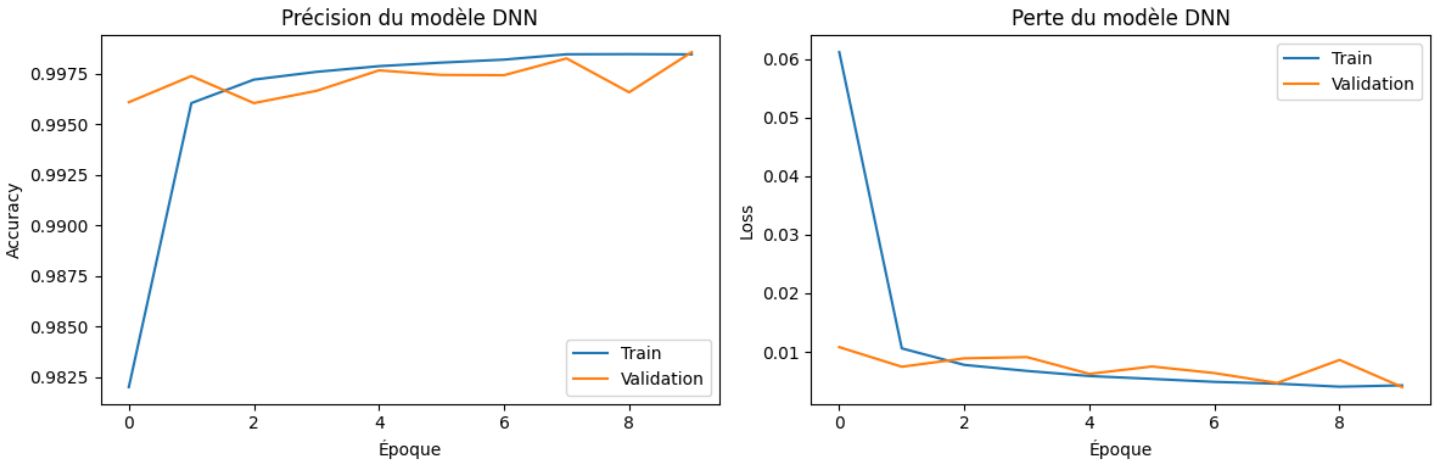



**Figure 16 : Modèle accuracy&loss du DNN**

Les deux modèles — LSTM et DNN — présentent des performances excellentes et très proches sur les données utilisées. Le modèle LSTM est particulièrement adapté aux données séquentielles et montre une stabilité remarquable, tandis que le DNN offre une structure plus simple avec des résultats comparables. Le choix entre les deux peut alors se baser sur la nature des données (séquentielles ou non) ou la complexité souhaitée du modèle.*

❖ **Matrice de corrélation :**


**Figure 17: Matrice de corrélation (heatmap)**

Celle-ci repose sur la construction d'une matrice de corrélation, représentée sous forme de heatmap Cette matrice met en évidence les relations linéaires entre les différentes caractéristiques.

Chaque cellule de la matrice indique un coefficient de corrélation compris entre -1 et 1 :

- **+1** indique une corrélation linéaire parfaite et positive (les deux variables augmentent ensemble.
- **-1** indique une corrélation linéaire parfaite et négative (l'une augmente pendant que l'autre diminue),
- **0** indique l'absence de corrélation linéaire.

La palette de couleurs utilisée dans la heatmap traduit l'intensité de ces corrélations :

**Les teintes claires** (blancs ou pastels) correspondent à des corrélations fortes, qu'elles soient positives ou négatives.

**Les teintes foncées** signalent une absence ou une faiblesse de corrélation.

Cette visualisation est utile pour les deux types de modèles (ML et DL), car elle permet de:

- Identifier les variables redondantes fortement corrélées, qui peuvent alourdir inutilement le modèle, notamment en Machine Learning où la sélection de caractéristiques est cruciale.
- Repérer les variables peu ou pas corrélées, souvent intéressantes pour les modèles de Deep Learning, car elles peuvent contenir des signaux indépendants utiles à la détection de comportements anormaux (comme des malwares).

Par exemple, les attributs "Fwd IAT Mean", "Fwd IAT Total" et "FwdIAT Max" montrent des corrélations élevées entre eux, suggérant qu'ils représentent des mesures similaires du trafic. Dans les modèles ML, il est judicieux de ne conserver qu'une seule de ces

variables pour réduire la dimensionnalité sans perte de performance. Pour les modèles DL, bien que la redondance puisse être mieux gérée automatiquement, réduire les entrées inutiles peut tout de même optimiser le temps d'entraînement et améliorer la convergence.

À l'inverse, des attributs comme "Destination IP" ou "Unnamed: 0" présentent peu de corrélation avec les autres variables, ce qui suggère qu'ils n'apportent pas d'information utile au modèle, voire qu'ils génèrent du bruit. Ces colonnes sont candidates à l'exclusion.

En résumé, la matrice de corrélation est un outil fondamental de l'analyse exploratoire des données, utile aussi bien dans le cadre des algorithmes de Machine Learning que pour les architectures de Deep Learning.

## . 4.8 Comparaison entre les algorithmes utilisés en ML et DL :

Pour comparer l'efficacité des techniques de détection, plusieurs algorithmes ont été sélectionnés : RandomForestClassifier, KNN, Naive Bayes, XGBoost, ainsi que les modèles de deep learning LSTM et DNN. Ce choix permet d'évaluer à la fois des méthodes classiques et des approches plus avancées, afin d'identifier celles offrant les meilleurs résultats pour la classification des logiciels malveillants Android , Comme le montre le tableau suivant :

| | | |
|---|---|---|
| RandomForestClassifier | 0.92 | Fiable, bonne capacité de généralisation |
| KNN | 0.96 | Bon taux de précision, simple à implémenter |
| Naive Bayes | 0.65 | Très rapide, facile à comprendre |
| Xgboost | 0.97 | Très performant, robuste face à l'overfitting |
| LSTM&DNN | 0.99 | Excellente précision, apprend des patterns complexes |

**Table 8 : Comparaison des algorithmes**

## 4.9 Comparaison entre le Machine Learning et le Deep Learning pour la détection de malwares Android :

Dans le cadre de ce projet, deux techniques d'intelligence artificielle ont été mises en œuvre afin de repérer les malwares Android : l'apprentissage automatique (ML) et l'apprentissage profond (DL).

Le Machine Learning fait appel à des algorithmes traditionnels tels que KNN, la régression logistique ou encore Random Forest. Ces modèles se caractérisent par une formation rapide, une compréhension aisée et une efficacité optimale quand les données sont correctement

préparées. Ils sont efficaces pour des missions simples et n'ont pas besoin de beaucoup de ressources.

Par ailleurs, le Deep Learning fait appel à des réseaux de neurones avancés, tels que les CNN et LSTM. Ces modèles sont capables d'apprendre de manière autonome des informations cruciales à partir des données, même lorsque celles-ci présentent une plus grande complexité. Ils proposent fréquemment de meilleures performances, en particulier avec de vastes ensembles de données, cependant leur entraînement prend plus de temps et leur explication est plus complexe.

Pour faire court, le Machine Learning se distingue par sa simplicité et rapidité, alors que le Deep Learning offre plus de puissance tout en étant plus exigeant. La décision entre les deux options est fonction des buts visés, des moyens à disposition et de la complexité des informations.

## 5 Conclusion :

Parmi les architectures testées dans cette étude, le modèle le plus performant sur le dataset "android-malware-detection" a été identifié.

Le jeu de données, composé de plus d'un million d'entrées , a été divisé en deux parties : 80% pour l'entrainement et 20% pour les tests. Avant l'entraînement du modèle, un prétraitement et un nettoyage rigoureux ont été effectués incluant la suppression des colonnes non informatives, la gestion des données manquantes (NAN/INF) et l'élimination des colonnes non numériques.

Les malwares intégrés dans le dataset ont été classés en quatre grandes champs : adware (41%), scareware (33%) , SMS malware (19%), et applications frauduleuses (7%), Cette classification a permis l'évaluation des résultats pour des modèles multiclasses.

Les courbes d'évolution de la précision et la perte indique une amélioration continue de l'apprentissage avec une perte (loss) à l'ordre de 0,38 sur les données d'entraînement et de validation. confirment la capacité du modèle à apprendre des caractéristiques distinctives de différentes classes sans surapprentissage.

En résumé, les résultats obtenus sont très prometteurs et corroborant l'efficacité de l'approche de deep learning choisie pour la détection automatique de malwares Android. Le

protocole s'applique avec fiabilité et pourrait être intégré dans un système de sécurité pour la détection, classification et l'anticipation des menaces mobiles en temps réel.

# Conclusion générale

Cette étude a examiné l'application de l'intelligence artificielle, notamment le machine learning et le deep learning, pour améliorer la protection des smartphones Android contre les malwares. Cette recherche, grâce à une étude détaillée des systèmes d'exploitation mobiles, des diverses sortes de malwares et des méthodes de détection actuelles, a prouvé que les approches classiques, même si elles sont bénéfiques, ont leurs limites face à la rapide progression des cybermenaces. Les résultats obtenus confirment confirme l'efficacité des modèles d'intelligence artificielle dans la détection des malwares Android.

Les algorithmes évalués, y compris le Random Forest, les réseaux de neurones profonds (DNN) et les LSTM, ont démontré des performances exceptionnelles en matière de précision et de robustesse .Grâce à leur aptitude à détecter des comportements douteux, à repérer des modèles malveillants et à s'ajuster aux menaces émergentes, ces outils se révèlent essentiels pour une approche proactive de la cybersécurité.

Pour aller plus loin, des axes d'amélioration pourraient inclure l'optimisation des modèles pour les dispositifs mobiles, l'incorporation de méthodes hybrides alliant analyse statique et dynamique, et l'emploi de l'apprentissage fédéré afin de protéger la confidentialité des données utilisateur. De plus, une coopération renforcée entre chercheurs, développeurs et sociétés de cybersécurité serait avantageuse pour normaliser et améliorer ces solutions.

Pour conclure , Ce projet pourrait également inclure une phase d'expérimentation en conditions réelles, sur plusieurs modèles de smartphones, afin d'évaluer la robustesse, la compatibilité, et les performances du système dans un contexte pratique. Cela ouvrirait la voie à une solution innovante, déployable dans un cadre professionnel ou grand public.

# Bibliographie

[1] Éditions Ellipses. (n.d.). *Systèmes d'exploitation mobiles*. https://www.editions-ellipses.fr/PDF/9782340025547_extrait.pdf [Consulté le : mars. 15, 2025].

[2] Dreamstime. (n.d.). *Set of top brand operating system logos isolated on white background* . Dreamstime. https://www.dreamstime.com/editorial-photo-set-top-brand-operating-system-logos-isolated-white-background-september-sri-lanka-image197346848 [Consulté le : mars. 15, 2025].

[3] PrivacyGuides. (n.d.). *Vue d'ensemble d'Android*. https://www.privacyguides.org/fr/os/android-overview/ [Consulté le : mars. 25, 2025].

[4] Fluentech Group. (2023, mars 7). *Introduction à Android : fonctionnalités, utilisations pratiques et développement d'applications*. https://www.fluentech-group.com/post/introduction-%C3%A0-android-fonctionnalit%C3%A9s-utilisations-pratiques-et-d%C3%A9veloppement-d-applications

[Consulté le : mars. 28, 2025].

[5] SocialCompare. (n.d.). *Comparaison des systèmes d'exploitation mobiles (vue développeur)*. https://socialcompare.com/fr/comparison/mobile-os-comparison-developer-view [Consulté le : Apr. 5, 2025].

[6] Statista. (2020, juillet 24). *Parts de marché des systèmes d'exploitation mobiles (iOS et Android) par pays en juillet 2020*

[7] JavaRush. (n.d.). *Historique des versions du système d'exploitation Android*. https://javarush.com/fr/groups/posts/fr.97.historique-des-versions-du-systme-dexploitation-android

[Consulté le : Apr. 5, 2025].

[8] PhonAndroid. (2021, septembre 8). *Toute l'histoire et la chronologie d'Android (dossier)*. https://www.phonandroid.com/toute-l-histoire-et-la-chronologie-d-android-dossier.html [Consulté le : Apr. 7, 2025].

[9] Google Developers. (n.d.). *Aperçu des versions Android*. https://developer.android.com/about/versions?hl=fr [Consulté le : Apr. 12, 2025].

[10] Temok. (2020, juin). *Android Versions List – Image illustrative* [Image]. https://www.temok.com/blog/wp-content/uploads/2020/06/android-feat.jpg [Consulté le : Apr. 13, 2025].

[11] Université de Picardie Jules Verne. (n.d.). *Android – Présentation du système d'exploitation* [PDF]. https://www.u-picardie.fr/ferment/android/slide_android.pdf [Consulté le : Apr. 15, 2025].

[12] AV-TEST. (n.d.). *Malware Statistics & Trends Report*. https://www.av-test.org/en/statistics/malware/ [Consulté le : Apr. 15, 2025].

[13] AMI Gestion. (n.d.).Comment Détecter, Éliminer et se Protéger des Malwares en 2025? https://ami-gestion.fr/malware/ [Consulté le : Apr. 18, 2025].

[14] CrowdStrike. (n.d.). *Quels sont les types de malwares ?*. https://www.crowdstrike.com/fr-fr/cybersecurity-101/malware/types-of-malware/ [Consulté le : Apr. 19, 2025].

[15] CrowdStrike. (n.d.). *Quels sont les types de malwares ?*. https://www.crowdstrike.com/fr-fr/cybersecurity-101/malware/types-of-malware/ [Consulté le : Apr. 19, 2025].

[16]Azmooni, H., & Rahman, S. M. M. (2018). *A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis*. ResearchGate.

[17] Sriatmaja, I., & Rajan, R. (2022). *Android malware detection based on system calls using machine learning*. Cranfield University

[18] Pratama, A. R., Santosa, P. I., & Mahmudah, I. F. (2020). *Analysis of Android malware detection techniques: A systematic review*. ResearchGate.

[19] Inter-Ligere. (n.d.). *L'intelligence artificielle dans l'entreprise : quels usages ?*. https://www.inter-ligere.fr/intelligence_artificielle_dans_l_entreprise/ [Consulté le : Apr. 22, 2025].

[20] Theodo. (n.d.). *Le machine learning : Définition, cas d'usage et explication*. https://data-ai.theodo.com/fr/parlons-data/machine-learning [Consulté le : Apr. 23, 2025].

[21] Tree Learning. (n.d.). *Qu'est-ce que le machine learning ?*. https://www.tree-learning.fr/plateforme-lms-elearning/machine-learning/ [Consulté le : Apr. 24, 2025].

[22] Wordly Fusion. (2024, 9 décembre). *Apprentissage automatique : introduction et concepts clés*. https://wordlyfusion.com/index.php/2024/12/09/apprentissage-automatique-introduction-et-concepts-cles/ [Consulté le : Apr. 27, 2025].

[23] Innovatiana. (n.d.). *Semi-supervised learning 101*. https://www.innovatiana.com/post/semi-supervised-learning-101 [Consulté le : Apr. 27, 2025].

[24] Kaizen Institute. (2024). Types et applications de l'apprentissage automatique. https://kaizen.com/fr/publications/types-applications-apprentissage-automatique/ [Consulté le : Apr. 28, 2025].

[25] Geekflare. (2024). Modèles d'apprentissage automatique. https://geekflare.com/fr/machine-learning-models/ [Consulté le : mai. 1, 2025].

[26] Red Hat. (2025). Le deep learning, qu'est-ce que c'est ? https://www.redhat.com/fr/topics/ai/what-is-deep-learning [Consulté le : Apr. 2, 2025].

[27] OpenClassrooms. (2025). *Appréhendez le Deep Learning ou l'apprentissage profond*. https://openclassrooms.com/fr/courses/6417031-objectif-ia-initiez-vous-a-lintelligence-artificielle/6823506-apprehendez-le-deep-learning-ou-lapprentissage-profond [Consulté le : mai. 5, 2025].

[28] NetApp. (2025). Le deep learning, qu'est-ce que c'est ? https://www.netapp.com/fr/artificial-intelligence/what-is-deep-learning/ [Consulté le : mai. 5, 2025].

[29] Coursera Staff. (2025, 8 janvier). *10 exemples de deep learning applications*. Coursera. https://www.coursera.org/fr-FR/articles/deep-learning-applications [Consulté le : mai. 7, 2025].

[30] Keldenich, T. (2023, 26 juin). Les différents algorithmes de Deep Learning – Meilleur Guide. Inside Machine Learning. https://inside-machinelearning.com/algorithmes-de-deep-learning/ [Consulté le : mai. 8, 2025].

[31] Inside Machine Learning. (2024). *Commencer avec le deep learning : applications et cas d'usage*. Récupéré le 6 juin 2025, de https://inside-machinelearning.com/commencer-deep-learning/ [Consulté le : mai. 11, 2025].

[32] DataCamp. (2024, 29 février). *Deep Learning (DL) vs Machine Learning (ML): A Comparative Guide*. https://www.datacamp.com/tutorial/machine-deep-learning [Consulté le : mai. 21, 2025].

[33] Alzahrani, A. (2021). *Machine learning and deep learning strategies* [Figure]. In *A survey on deep learning for human action recognition*. ResearchGate. https://www.researchgate.net/figure/Machine-Learning-and-Deep-Learning-Strategies_fig12_353827517 [Consulté le : mai. 24, 2025].

# RÉSUMÉ

Android s'est imposé ces dernières années comme le système d'exploitation mobile le plus répandu à l'échelle mondiale. Cette prédominance en fait une cible privilégiée pour les cybercriminels, ce qui soulève d'importantes questions de sécurité. Bien que la plateforme intègre des mécanismes de protection sophistiqués, elle présente néanmoins des vulnérabilités persistantes.

L'écosystème Android, caractérisé par sa grande ouverture, a connu une croissance exponentielle d'applications, dont certaines dissimulent des codes malveillants. Face à cette menace grandissante, les techniques conventionnelles de détection basées sur des signatures préétablies révèlent leurs limites. Les approches fondées sur l'apprentissage profond apparaissent comme une alternative prometteuse, capable d'analyser des comportements complexes et d'identifier des malwares inédits grâce à leur capacité d'adaptation.



**Mots clés :** Android, cybercriminalité, sécurité, malwares, apprentissage profond, détection, vulnérabilités.



# ABSTRACT

In recent years, Android has established itself as the most widely used mobile operating system worldwide. This dominance makes it a prime target for cybercriminals, raising significant security concerns. Although the platform incorporates sophisticated protection mechanisms, it nevertheless presents persistent vulnerabilities.

The Android ecosystem, characterized by its openness, has experienced exponential growth in applications, some of which conceal malicious code. Faced with this growing threat, conventional detection techniques based on predefined signatures show their limitations. Approaches based on deep learning emerge as a promising alternative, capable of analyzing complex behaviors and identifying novel malware thanks to their adaptive capabilities.



**Keywords:** Android**,** cybersecurity**,** malware**,** deep learning**,** detection**,** vulnerabilities**,** mobile applications**.**



# ملخص

في السنوات الأخيرة، أصبح **أندرويد** نظام التشغيل الأكثر انتشارًا للأجهزة المحمولة على مستوى العالم. هذه الهيمنة جعلته هدفًا رئيسيًا لمجرمي الإنترنت، مما أثار مخاوف أمنية كبيرة. وعلى الرغم من تضمين النظام لآليات حماية متطورة، إلا أنه لا يزال يعاني من ثغرات أمنية مستمرة.

يتميز نظام **أندرويد** بانفتاحه الكبير، مما أدى إلى نمو هائل في عدد التطبيقات، بعضها يحتوي على أكواد ضارة. في مواجهة هذا التهديد المتزايد، تظهر تقنيات الكشف التقليدية القائمة على التواقيع المحددة مسبقًا محدوديتها. بينما تبرز أساليب التعلم العميق كحل واعد، قادر على تحليل السلوكيات المعقدة وتحديد البرمجيات الخبيثة غير المعروفة بفضل قدراتها التكيفية.



**الكلمات المفتاحية:** أندرويد، الأمن السيبراني، البرمجيات الخبيثة، التعلم العميق، الكشف، الثغرات الأمنية، تطبيقات الهاتف المحمول.