دولة ليبيا

وزارة التعليم العالي والبحث العلمي

جامعة سبها

كلية تقنية المعلومات

قسم نظم المعلومات

بحث مقدم لاستكمال متطلبات نيل درجة البكالوريوس تحت عنوان:

تصنيف تطبيقات اند رويد الخبيثة باستخدام خوارزميات تعلم الآلة

Classification of Malicious Android Applications Using Machine Learning Algorithms

إعداد الطالبة:

الهام محمد إبراهيم العرج

الرقم الدراسي 202180709

تحت إشراف:

د. حمودة خليفة شفتر

العام الجامعي

2024-2023

إقـرار

إقرار الطالب / الطلاب

أنا الطالبة: الهام محمد إبراهيم العرج الرقم الدراسي:202180709

أُقر بأن ما ورد في هذا البحث هو من مجهودي الشخصي ماعدا الفقرات التي تم إسنادها إلى مرجع.

التاريخ:.....التوقيع:.....

إقرار المشرف

اسم المشرف :د. حمودة خليفة شفتر.

أقر بأني اطلعت على مادة البحث، وأن البحث جاهز للمناقشة.

التاريخ:.....التوقيع:.....

إقرار بالموافقة على التصحيحات وتسليم النسخة النهائية:

بعد التصحيح والاطلاع على مادة البحث، تمت الموافقة عليها، وتسليم النسخة النهائية.

اسم الممتحن الأول:.....التوقيع:.....التاريخ: .....

اسم الممتحن الثاني:.....التوقيع:.....التاريخ: .....

الآية القرآنية

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

﴿هُوَ الَّذِي جَعَلَ الشَّمْسَ ضِيَاءً وَالْقَمَرَ نُورًا وَقَدَّرَهُ مَنَازِلَ لِتَعْلَمُوا عَدَدَ السِّنِينَ وَالْحِسَابَ مَا خَلَقَ اللَّهُ ذَٰلِكَ إِلَّا بِالْحَقِّ يُفَصِّلُ الْآيَاتِ لِقَوْمٍ يَعْلَمُونَ﴾

{سورة يونس : الآية 5}

الإهـداء

إلى من أمرنا الله ببرهما، إلى من بذلا الكثير، وقدما ما لا يمكن أن يرد، إليكما تلك الكلمات

أمي وأبي الغاليان.

إلى رفيق الدرب، وصديق الأيام جميعًا بحلوها ومرها: زوجي الغالي.

إلى من حلت بركة وجودهم في حياتي، ومن ملأت ضحكاتهم الجميلة عمري، أطفالي:

رتاج وأبوعقيلة.

إلى كل من علمني حرفاً وبذل جهداً في إيصال العلم والمعرفة أساتذتي.

إلى هؤلاء جميعاً اهديكم هذا العمل

كلمـة شـكر

قال تعالى﴿وَمَن يَشْكُرْ فَإِنَّمَا يَشْكُرُ لِنَفْسِهِۦ﴾{لقمان:12}

يقول النبي ﷺ :"من لم يشكر الناس، لا يشكر الله عز و جل"

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ، والحمد لله رب العالمين، والصلاة والسلام على خاتم النبيين والمرسلين سيدنا محمَّد وعلى آله وصحبه أجمعين..

أودُّ أنْ أُظْهِرَ امتنانِي لكُلِّ مَنْ سانَدَنِي وأعانَنِي في إتْمام هذا المشروع، وأولُهُم **الدكتور**/ **حمودة خليفة شفتر** الذي رافقني في مسيرتي لإنجاز هذا البحث وكانت له بصمات واضحة من خلال توجيهاته وانتقاداته البناءة والدعم الاكاديمي فجزى الله خيرا لأستاذي الموقر وحفظة الله تعالى

فهرس المحتويات

إقرار ..... أ

الآية القرآنية ..... ب

الإهداء ..... ج

كلمة شكر ..... د

فهرس المحتويات ..... هـ

فهرس الجداول ..... ح

فهرس الأشكال ..... ط

فهرس المصطلحات ..... ي

الملخص Abstract ..... ك

الفصل الأول مقدمة البحث .....

1. مقدمة 2

1.1 مشكلة البحث Research Problem 3

2.1 أسئلة البحث Research Questions 3

3.1 أهداف البحثResearch Objectives 3

4.1 أهمية البحثResearch Significance 3

5.1 حدود البحث Research Scope 3

6.1 منهجية البحث Research methodology 4

7.1 تنسيق البحث Research Format 6

الفصل الثاني الإطار النظري والدراسات السابقة .....

2. مقدمة 8

1.2 نظام تشغيل Android 8

2.2 تطبيقات Android 9

3.2 تطبيقات Android الخبيثة Android Malware 10

1.3.2 أنواع تطبيقات Android الخبيثة 12

4.2 طرق كشف تطبيقات اندرو يد الخبيثة 13

1.4.2 الكشف الاستاتيكىStatic Detection 13

2.4.2 الكشف الديناميكي Dynamic Detection 13

3.4.2 الكشف الهجين Hybrid Detection 14
5.2 تقنيات الكشف عن برامج اندرويد الخبيثة 14
6.2 الدراسات السابقة 15
7.2 طرق المعالجة الأولية للبيانات 18
1.7.2 معالجة القيم المفقودة Missing Values 18
2.7.2 تطبيع البيانات Data Normalization 19
3.7.2 اختيار الخصائص Feature Selection 19
1.3.7.2 طريقة نسبة اكتساب المعلومات لاختيار الميزات Information Gain Ratio 20
8.2 خوارزميات التعلم الالى Machine Learning Algorithms 21
1.8.2 خوارزمية الجار الأقرب K Nearest Neighbour 22
2.8.2 خوارزمية نايف بايز Naive Bayes 22
3.8.2 خوارزمية الآلة المتجه الدعم Support Vector Machine 23
4.8.2 خوارزمية شجرة القرار Decision Tree 24
5.8.2 خوارزمية الانحدار اللوجستى Logistic Regression 24
9.2 طرق تقييم فعالية النماذج 26
1.9.2 مصفوفة الارتباك (Confusion Matrix) 26
2.9.2 دقة التصنيف Classification Accuracy 27
3.9.2 مؤشر Precision 27
4.9.2 مؤشر Recall 27
5.9.2 مؤشر Fl-score 27
الفصل الثالث تطبيق منهجية البحث .....
والنتائج والمناقشة 28
3. مقدمة 29
1.3 الأدوات البرمجية والمكتبات المستخدمة 29
2.3 تجميع البيانات 29
3.3 المعالجة المسبقة لمجموعة البيانات 31
4.3 التجربة الأولى 31
5.3 التجربة الثانية 32

الفصل الرابع الخلاصة .....
4.الخلاصة Conclusion 40
1.4 الصعوبات والعراقيل 40
2.4 التوصيات والأفاق المستقبلية.....40
المراجع 41

فهرس الجداول

جدول (1.3)مجموعة بيانات تطبيقات Android.....30

جدول (2.3) بعض خصائص مجموعة بياناتDerbin 30

جدول (3.3) نتائج تصنيف مجموعة بيانات Derbin قبل اختيار الخصائص 32

الجدول (4.3) أفضل 20 خاصية بناء على قيمة نسبة اكتساب المعلومات IGR 33

جدول (5.3) نتائج التصنيف بعد تطبيق اختيار الخصائص باستخدام IGR 34

فهرس الأشكال

الشكل (1–1) منهجية البحث 4
الشكل (1.2) البنية الأساسية لنظام تشغيل Android 8
شكل (2.2) التصنيف باستخدام خوارزمية الجار الأقرب KNN 22
الشكل (3.2) التصنيف باستخدام خوارزمية SVM 24
الشكل (4.2) خوارزمية الانحدار اللوجستى 25
الشكل (5.2) مصفوفة الارتباك Confusion Matrix 26
الشكل (1.3) نتائج التصنيف باستخدام قيم مختلفة لنسبة اكتساب المعلومات 34
الشكل (2.3) مقارنة بين نتائج التصنيف قبل و بعد اختيار الخصائص باستخدام IGR 35
الشكل (3.3) مصفوفة الارتباك للمصنف KNN على مجموعة بيانات Derbin 36
الشكل (4.3) مصفوفة الارتباك للمصنف NB على مجموعة بيانات Derbin 36
الشكل (5.3) مصفوفة الارتباك للمصنف DT على مجموعة بيانات Derbin 37
الشكل (6.3) مصفوفة الارتباك للمصنف LR على مجموعة بيانات Derbin 37
الشكل (7.3) مصفوفة الارتباك للمصنف SVM على مجموعة بيانات Derbin 38

فهرس المصطلحات

الاختصار	المصطلح باللغة الإنجليزية	المصطلح باللغة العربية
Malware	Malicious Software	برنامج خبيث
ML	Machine Learning	تعلم الآلة
KNN	K-Nearest Neighbor	خوارزمية الجار الأقرب
DT	Decision Tree	خوارزمية شجرة القرار
SVM	Support Vector Machine	خوارزمية آلة متجه الدعم
NB	Naïve Bayes	خوارزمية نايف بايز
LR	Logistic Regression	خوارزمية الانحدار اللوجستي
FS	Feature Selection	اختيار الميزات
IGR	Information Gain Ratio	نسبة اكتساب المعلومات

الملخص Abstract

يقوم مصنعو الأجهزة المحمولة بإنتاج إصدارات متنوعة من Android بسرعة في جميع أنحاء العالم. أصبحت الهواتف الذكية منصة مفتوحة المصدر لتشغيل أنواع مختلفة من التطبيقات (Apps) مثل: الخدمات المصرفية والتجارة و التعليم والصحة والترفيه وغيرها. في الوقت نفسه، يقوم مجرمو الإنترنت بتنفيذ إجراءات ضارة من خلال تطوير تطبيقات خبيثة لتنفيذ عمليات مثل: تتبع أنشطة المستخدم، وسرقة البيانات الشخصية، وارتكاب عمليات الاحتيال المصرفي. يحصل هؤلاء المجرمون على فوائد عديدة، نظرًا لأن الكثير من الأشخاص يستخدمون تطبيقات Android في أعمالهم اليومية مثل: إدارة الحسابات المصرفية والتسوق وغيرها. في هذا البحث، تم اختبار اداء خمس مصنفات للتعلم الآلي، وهي خوارزمية الجار الأقرب K Nearest Neighbour وخوارزمية شجرة القرار Decision Treesوخوارزمية نايف بييز Naïve Bays وخوارزمية الانحدار اللوجيستي Logisti Regression وخوارزمية الآلة متجه الدعم Support Vector Machine لتصنيف تطبيقات اند رويد الخبيثة. من بين هذه المصنفات، سجلت خوارزمية KNN أعلى دقة تصنيف بلغت (0.986).

الفصل الأول

مقدمة البحث

1. مقدمة

يعد نظام تشغيل اندرويد Android من أشهر أنظمة تشغيل الهواتف الذكية مفتوحة المصدر Open-Source Smartphone Operating System. بحسب إحصائيات سنة 2022، من المتوقع أن يهيمن نظام Android على سوق أنتاح أنظمة تشغيل الهواتف الذكية في جميع أنحاء العالم، بنسبة تصل الي 75.1 % من إجمالي السوق. يشكل نظام التشغيل Apple جزء أصغر بكثير من إجمالي السوق بنسبة تصل الي 24.9 % فقط. من المتوقع أن يكون هذان النظامان هما الأكثر استخداما في السنوات القادمة [3].

تمثل الزيادة المستمرة في تطوير تطبيقات البرامج الخبيثة لنظام Android خطر كبير على خصوصية وأمن معلومات المستخدمين [4]. يزداد استخدام الهواتف الذكية التي تعمل بنظام تشغيل Android من قبل عدد كبير من المستخدمين كل عام. في نفس الوقت، يعمل مجرمو شبكة الإنترنت Cybercriminals باستمرار على تطوير تطبيقات خطيرة لسرقة المعلومات الحساسة وتنفيذ هجمات ضارة. وفقًا لتقرير تهديدات الأجهزة المحمولة من Kaspersky في عام 2021 [5]، زاد مجرمو الإنترنت من فعالية هجماتهم على الأجهزة المحمولة من خلال تطوير المزيد من البرامج الخبيثة المعقدة التي تستخدم طرقًا مختلفة لسرقة معلومات المستخدمين الحساسة [4][5].

يعمل الباحثون في مجال الأمن الالكتروني على تطوير أنظمة لكشف البرامج الخبيثة Malware لحماية مستخدمي الهواتف الذكية. لتحديد وتصنيف برامج Android الخبيثة، استخدم باحثو الأمن الالكتروني طرق مختلفة تعتمد على خوارزميات التعلم الآلي. يتم تحليل ملفات تطبيقات اندرويد مثل ملف Manifest-Android والشفرة المصدرية لملف حزمة تطبيق (Android Application Package (APK بدون تشغيله. تتضمن هذه الملفات العديد من الميزات، بما في ذلك استدعاء "API "Calls، وعناوين الشبكة، والأذونات Permissions، والخصائص المادية للأجهزة وغيرها. يمكن استخدام هذه الميزات (الخصائص) Features لتحديد تطبيقات Android الخبيثة باستخدام خوارزميات التعلم الألى Machine Learning Algorithms [6][9].

1.1 مشكلة البحث Research Problem

تمثل الزيادة المستمرة في التطبيقات الضارة التي تعمل بنظام Android خطر كبير على خصوصية وأمن معلومات المستخدمين. لهذا السبب،كانت هناك حاجة ماسة إلى تطوير تقنيات فعالة وذات دقة عالية في مجال تصنيف تطبيقات البرامج الضارة.حيث تتلخص مشكلة البحث في الاتي:

- صعوبة كشف تطبيقات اند رويد الخبيثة.
- الحاجة إلى تطوير أنظمة فعالة في مجال كشف تطبيقات اندرويد الخبيثة لحماية مستخدمي تطبيقات اند رويد.

2.1 أسئلة البحث Research Questions

تتلخص أسئلة البحث في الاتي:

1. ما مدى فعالية خوارزميات تعلم الآلة في مجال تصنيف تطبيقات اندرويد الخبيثة؟
2. ما تأثير اختيار الميزات او الخصائص Feature Selection على أداء خوارزميات تعلم الآلة في مجال تصنيف تطبيقات اند رويد؟

3.1 أهداف البحث Research Objectives

يهدف هذا البحث إلى:

- تصنيف تطبيقات اند رويد الخبيثة باستخدام خوارزميات تعلم الآلة
- استغلال خوارزميات تعلم الآلة في مجال التصنيف تطبيقات اندرويد الخبيثة.

4.1 أهمية البحث Research Significance

تتلخص أهمية البحث في الاتي:

- حماية خصوصية ومعلومات مستخدمي تطبيقات اند رويد من خلال تصنيف تطبيقات اندرويد الخبيثة.
- اختبار فعالية تقنيات التنقيب عن البيانات في مجال امن المعلومات.

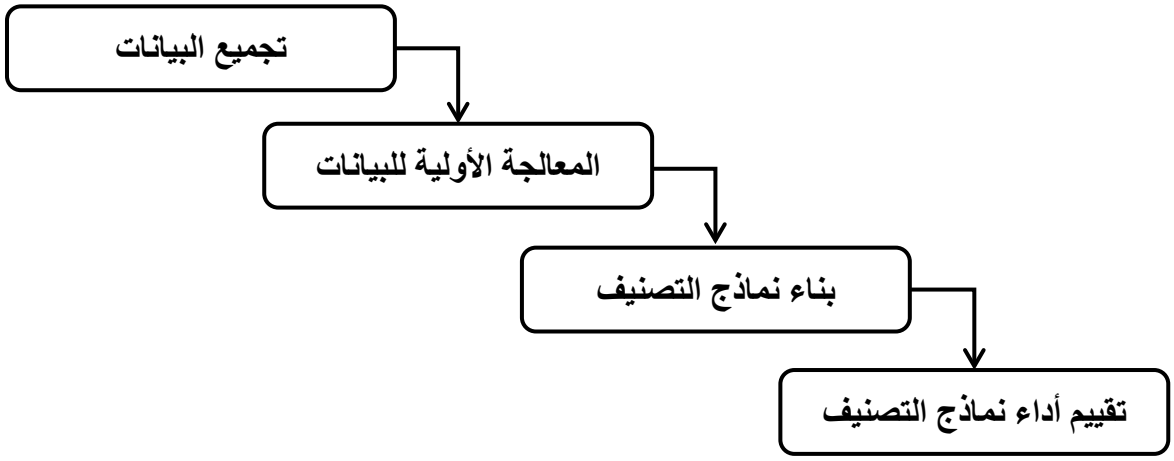
5.1 حدود البحث Research Scope

في هذا البحث، تم استخدام خمس خوارزميات للتعلم الآلي وهي (SVM، KNN، Naïve Bayes ، Decision Tree، Logistic regression) لتصنيف تطبيقات اندرويد الخبيثة Android

Malware. كذلك تم استخدام أحد طرق اختيار الميزات تسمى نسبة اكتساب المعلومات Information Gain Ratio لتحديد أفضل الميزات الممثلة لتطبيقات اندرويد التي تساعد في التمييز بين تطبيقات اندرويد الحميدة والخبيثة وبالتالي تمكن خوارزميات تعلم الآلة من تصنيف تطبيقات الخبيثة بدقة عالية. استخدمت مجموعة بيانات تسمي Derbin Dataset تضم 15036 تطبيق اندرويد (حميدة و خبيثة) ممثلة بعدد 215 ميزة Features لاختبار فعالية خوارزميات تعلم الآلة في مجال تصنيف تطبيقات اندرويد.

6.1 منهجية البحث Research methodology

تعد مرحلة تحديد منهجية البحث من أهم خطوات انجاز البحث العلمي, وهي عبارة عن سلسلة من الخطوات التي يتبعها الباحث خلال انجازه للبحث العلمي ، حيث يجب أن تتوافق المنهجية المتبعة مع مجال البحث للوصول إلى حل لمشكلة البحث. يعتمد هذا البحث على الخطوات الأساسية لمنهجية التنقيب عن البيانات Data Mining المتبعة في أبحاث التنقيب عن البيانات لتصنيف (كشف) تطبيقات اندرويد الخبيثة Android Malware [1] .يبين الشكل (1-1) المنهجية المتبعة في هذا البحث.



الشكل (1-1) منهجية البحث

- **تجميع البيانات Data Collection**

في هذه الخطوة يتم تحديد مجموعة البيانات المناسبة. بعد البحث عن مجموعة البيانات المناسبة والمتعلقة بموضوع الدراسة، تم تحديد مجموعة بيانات تسمى Drebin معدة للبحث العلمي في مجال الكشف عن تطبيقات اندرويد الخبيثة Android Malware لإجراء تجارب الكشف عن برامج (تطبيقات) اندرويد الخبيثة باستخدام خوارزميات التعلم الآلي Machine Learning.

- **المعالجة الأولية للبيانات Data Pre-processing**

يتم في مرحلة المعالجة الأولية فحص مجموعة البيانات لغرض الكشف عن المشاكل التي قد تعيق استخدام مجموعة البيانات في بناء نماذج تصنيف تطبيقات اندرويد الخبيثة وحل هذه المشاكل باستخدام الطرق المناسبة. تتمثل المعالجة المسبقة في إجراء مجموعة من العمليات على البيانات قبل استخدامها لبناء نموذج التصنيف مثل معالجة مشكلة القيم المفقودة للخصائص (Missing Values) إن وجدت أو إجراء عملية تطبيع (Normalization) لقيم الخصائص الرقمية (Numeric). كذلك تعتبر عملية اختبار الميزات او الخصائص Feature Selection احد العمليات الأساسية للمعالجة الأولية لمجموعة البيانات والتي تتم باستخدام احد طرق اختيار الخصائص. تهدف عملية اختيار الخصائص الى تحديد واختيار الخصائص المهمة التي تساهم في رفع دقة تصنيف البيانات باستخدام خوارزميات تعلم الآلة واستبعاد الخصائص التي تؤثر سلبا على دقة التصنيف. سيتم في هذا البحث استخدام أحد طرق المعروفة لاختيار الخصائص وهى طريقة نسبة المعلومات المكتسبة Information Gain Ratio لتحديد مدى أهمية السمة او الخاصية في التمييز بين تطبيقات اندرويد الحميدة و الخبيثة واختيار السمات المهمة لتصنيف تطبيقات اندرويد.

- **بناء نماذج التصنيف Building Classification Models**

سيتم في هذا البحث اختبار أداء أشهر خوارزميات تعلم الآلة المستخدمة في مجال تصنيف البيانات التي توفرها مكتبات لغة بايتون Python في تصنيف تطبيقات اندرويد إلى حميدة (غير ضارة) أو خبيثة. تشمل الخوارزميات المستخدمة كل من:

- خوارزمية الجار الأقرب K Nearest Neighbor
- خوارزمية شجرة القرار Decision Trees
- خوارزمية نايف بييز Naïve Bays
- خوارزمية الانحدار اللوجيستي Logistic Regression

- خوارزمية الآلة المتجه الدعم Support Vector Machine
- **تقييم أداء نماذج التصنيف Performance Evaluation**

بعد تدريب نموذج تعلم الآلة على جزء من مجموعة بيانات تدريب Training set تضم تطبيقات اندرويد حميدة وخبيثة. يتم اختيار أدائه باستخدام مجموعة بيانات اختبار Test Set. تقيم دقة تصنيف النموذج باستخدام عدة معايير أهمها (Precision, Recall, Accuracy, F1-score).

7.1 تنسيق البحث Research Format

الفصل الأول: يحتوي على المفاهيم الأساسية للبحث حيث يشمل المقدمة ومشكلة البحث وأهداف البحث وحدوده والمنهجية المتبعة لتحقيق الأهداف المرجوة من البحث.

الفصل الثاني: يشمل الفصل الثاني الإطار النظري للبحث والذي يتضمن التعريف بتطبيقات اندرويد الخبيثة ومدى خطورتها على مستخدمي الهواتف المحمولة, وطرق المعالجة الأولية لمجموعات البيانات. كما يتضمن أيضا خوارزميات تعلم الآلة المستخدمة في هذا البحث وفكرة عملها وطرق تقييم أدائها, وكذلك الدراسات السابقة المتعلقة بموضوع البحث والنظام المقترح.

الفصل الثالث: يتضمن هذا الفصل خطوات تطبيق منهجية البحث بشكل مفصل حيث تم توضيح الخطوات التي تم إتباعها في البحث, ويحتوي أيضا المكتبات والأدوات البرمجية المستخدمة في البحث. كذلك يحتوي على النتائج التي تم التوصل إليها بعد تنفيذ النموذج ومناقشة هذه النتائج.

الفصل الرابع: يوضح الفصل الأخير خلاصة البحث والصعوبات والتحديات التي واجهت البحث والتوصيات والخاتمة.

الفصل الثاني

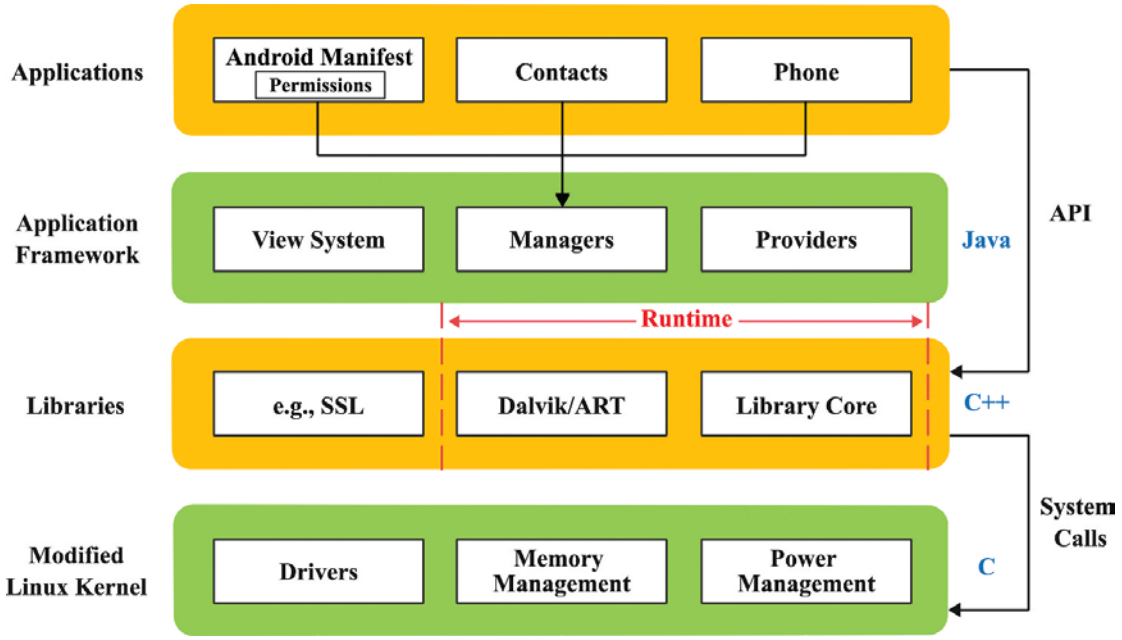
الإطار النظري والدراسات السابقة

2. مقدمة

يحتوي هذا الفصل على الإطار النظري والذي يتضمن تعريف تطبيقات اندرويد الخبيثة وطرق الكشف عنها. كذلك سيتم استعراض بعض الدراسات السابقة في مجال كشف تطبيقات اندرويد. كذالك يضم شرح طرق المعالجة الأولية للبيانات و خوارزميات تعلم الآلة المستخدمة في هذا البحث وطرق تقييم ادائها.

1.2 نظام تشغيل Android

ظهر نظام التشغيل Android للهواتف الذكية عام 2008 و هو نظام تشغيل مفتوح المصدر يعتمد على نظام التشغيل Linux للهواتف المحمولة، ثم اصداره بواسطة Google. علي الرغم من تحديث نظام Android بشكل مستمر، ظلت البنية الأساسية لنظام التشغيل Android دون تغيير. كما هو موضح في الشكل (1.2)،حيث تنقسم بنية نظام تشغيل Android إلى أربعة طبقات تضم طبقة Linux Kernel المعدلة و طبقة المكتبات Libraries و طبقة إطار التطبيق Application Framework و طبقة التطبيقات Applications Layer [12] [13].



الشكل (1.2) البنية الأساسية لنظام تشغيل Android

(1) نواة لينكس المعدلة Linux Kernel: تعتمد خدمات النظام الأساسية التي يقدمها Android مثل الأمان وإدارة الطاقة Power Management و إدارة الذاكرة Memory Management و برامج التشغيل Drivers على صيغة معدلة لنواة نظام Linux. تعمل نواة Linux المعدلة كطبقة بين الأجهزة والبرامج.

(2) المكتبات Libraries: تتضمن عدة مكتبات معتمدة على لغة Java مثل Secure Socket Layer (SSL) و الآلة الافتراضية DalVik Virtual Machine لتشغيل تطبيقات Android وغيرها.

(3) إطار العمل Application Framework: توفر طبقة إطار عمل التطبيق خدمات ذات مستوى أعلى للتطبيقات على شكل تصنيفات لغة جافا Java Classes. كذلك توفر هذه الطبقة واجهات برمجة التطبيقات (APIs) Application Programming Interfaces وهى مجموعة من البروتوكولات والأدوات والإجراءات المستخدمة لبناء التطبيقات البرمجية.

(4) طبقة التطبيقات Applications Layer: هي الطبقة العليا من بنية Android، وهي تطبيقات الهاتف المحمول التي يتفاعل معها المستخدمون. Manifest هو ملف XML يحتوي على بيانات وصفية مهمة حول تطبيق Android. تشمل اسم الحزمة و أسماء الأنشطة و النشاط الرئيسي (نقطة الدخول إلى التطبيق) و دعم إصدار Android ودعم ميزات الأجهزة hardware و الأذونات Permissions والتكوينات الأخرى.

2.2 تطبيقات Android

منذ إطلاق نظام Android في عام 2008، أصبح نظام التشغيل Android الأكثر استخداما للأجهزة المحمولة الذكية. في عام 2019، حوالي 86.6% من الهواتف الذكية المباعة عالميًا تعتمد على نظام التشغيلAndroid. وبحلول نهاية أبريل 2020، كان هناك أكثر من 2.8 مليون تطبيق على متجر Google Play Store، وهو المتجر الرسمي لتطبيقات أندرويد [4]. تشمل تطبيقات Android عدة مجالات منها على سبيل المثال:

- التطبيقات التعليمية Educationl Apps : يتم استخدام التطبيقات التعليمية لتحسين المعرفة وتحقيق الإنتاجية. يمكن للتطبيقات التعليمية أن تجعل الأشخاص أكثر تفاعلاً

وتعد تطبيقات التعلم طريقة فعالة في التعليم. من أشهر تطبيقات التعليم Google Classroom.

- تطبيقات الخدمات المصرفية Mobile Banking Apps: تمكن تطبيقات الخدمات المصرفية عبر الهاتف المحمول العملاء من سهولة الوصول إلى الحسابات المصرفية، والتحقق من الرصيد، وتحويل الأموال، ودفع الفواتير، وإيداع الشيكات، وما إلى ذلك. بشكل عام، تمكن تطبيقات الخدمات المصرفية العملاء من انجاز الخدمات المصرفية الأساسية التي تقدمها المؤسسات المصرفية وتوفر الوقت والجهد.
- تطبيقات التجارة الإلكترونية E-Commerce Apps: تعد تطبيقات التجارة الإلكترونية مثالاً لنموذج Business to Business (B2B) الأعمال التجارية كالتبادل بين الصناع وتجار الجملة أو التبادل بين تجار الجملة وتجار التجزئة أو التبادل التجاري بين شركة وأخرى. تساعد تطبيقات التجارة الالكترونية الأشخاص على بيع واستعارة العناصر المختلفة ويوفر الوقت والمال. في تطبيقات التجارة الإلكترونية، يمكن تداول السلع التجارية في الأسواق عبر الإنترنت و شراء سلع محددة وإجراء معاملات إلكترونية.
- تطبيقات وسائط التواصل الاجتماعي Social Media Apps: تتيح تطبيقات التواصل الاجتماعي للناس التواصل مع بعضهم وتبادل المعلومات. تُستخدم هذه التطبيقات بشكل أساسي لأغراض مشاركة المعلومات في عدة مجالات منها التسويق و الأعمال التجارية و إدارة الأعمال وما إلى ذلك. أشهر هذه التطبيقات Instagram وFacebook وWhatsApp وYouTube وLinkedIn و غيرها.

3.2 تطبيقات Android الخبيثة Android Malware

عادةً ما تقوم الهواتف الذكية بتخزين بيانات المستخدم الخاصة مثل الرسائل والصور والمعلومات الشخصية وما إلى ذلك. يتم استخدام الهواتف الذكية التي تعمل بنظام Android من قبل آلاف الأشخاص كل عام، يعمل مجرمي الإنترنت باستمرار على تطوير تطبيقات خطيرة من أجل سرقة المعلومات الحساسة وتنفيذ هجمات ضارة. وفقًا لتقرير Kaspersky حول تهديدات الأجهزة المحمولة في عام 2021،حيث ارتفعت فعالية الهجمات على الأجهزة المحمولة من

خلال تطوير برامج ضارة Malware معقدة للهواتف المحمولة تستخدم أساليب متعددة لسرقة المعلومات الحساسة للمستخدمين. وبحسب الدراسة، تم اكتشاف 3.5 مليون تطبيق ضار، وتم تسجيل حوالي 46.2 مليون هجمة الإلكترونية في جميع أنحاء العالم في عام 2021 [4].

البرامج الضارة التي تعمل بنظام Android هي برامج ضارة تستهدف أجهزة Android على وجه التحديد. كما هو الحال مع أي نوع من البرامج الضارة، فإن الهدف من تطوير تطبيقات اندرويد الضارة هو الوصول الغير مشروع إلى جهاز المستخدم وسرقة بياناته [19]. مقارنة مع متجر تطبيقات Apple، فإن متجر Google Play لديه إجراءات أمنية أقل صرامة. بالإضافة إلى ذلك، يمكن لمستخدمي Android تنزيل التطبيقات من مصادر مختلفة على الإنترنت. وهذا يخلق بيئة تكون فيها الهجمات السيبرانية ممكنة. تنتشر البرامج الضارة التي تعمل بنظام Android بعدة طرق أهمها [20]:

- تنزيل التطبيقات الضارة Downloading Malicious Apps: الطريقة الأكثر شيوعًا التي يستخدمها المتسللون لنشر البرامج الضارة هي من خلال تنزيل التطبيقات. عادة ما تكون التطبيقات التي يتم تنزيلها عبر المتاجر الرسمية آمنة، ولكن في بعض الأحيان، التطبيقات التي يتم تنزيلها من مصادر أقل موثوقية تحتوي على برامج ضارة.
- الثغرات الأمنية في نظام تشغيل الهاتف المحمول: استخدام جهاز به ثغرات في نظام التشغيل يمكن المتسللين Hackers من استغلال نقاط ضعف نظام التشغيل و خاصة في حال عدم تحديث البرامج بانتظام. تتيح الثغرات للمتسلل تشغيل البرامج عن بعد Remote Code Execution.
- النقر على الروابط المشبوهة في رسائل البريد الإلكتروني أو النصوص Suspicious Links: قد يؤدى النقر على رابط مشبوه إلى الانتقال إلى موقع ويب ضار يقوم بتنزيل البرامج الضارة وتثبيتها على هاتف المستخدم. يتمكن بعدها المتسلل من الوصول إلى البيانات الموجودة على الهاتف.
- استخدام شبكات Wi-Fi أو عناوين URL غير آمنة: عند زيارة مواقع ويب غير آمنة، قد يتعرض المستخدم لخطر الوصول إلى البيانات الحساسة المرسلة من جهازه عن

طريق البرامج الضارة او من خلال هجمات الوسيط Man=in–the–Middle وهو يتسلل المهاجم بين متحاورين في شبكة دون علم كل منهما.

1.3.2 أنواع تطبيقات Android الخبيثة

تشمل الأنواع الشائعة من البرامج الضارة للأجهزة المحمولة كل من [20]:

- Banking Malware: هي برمجيات خبيثة تستهدف الخدمات المصرفية لسرقة المعلومات الخاصة بالحسابات المصرفية للزبائن واستخدامها، حيث يعمل المتسللون إلى اختراق هواتف المستخدمين الذين يفضلون إجراء معاملاتهم المصرفية مثل تحويل الأموال ودفع الفواتير من أجهزتهم المحمولة.
- Mobile Ransomware: برامج الفدية الخبيثة التي تقوم بإغلاق بيانات المستخدم المهمة مثل المستندات والصور ومقاطع الفيديو عن طريق تشفير هذه المعلومات ثم المطالبة بدفع فدية لصانعي البرامج الضارة مقابل فك تشفير البيانات.
- Mobile Spyware: يتم تحميل برامج التجسس على جهاز المستخدم، وتقوم بمراقبة نشاطه، وتسجيل موقعه، ورفع المعلومات المهمة، مثل أسماء المستخدمين وكلمات المرور لحسابات البريد الإلكتروني أو مواقع التجارة الإلكترونية. في كثير من الحالات، يتم ربط برامج التجسس مع برامج أخرى تبدو حميدة وتقوم بجمع البيانات.
- MMS Malware: استغلال الاتصالات النصية كوسيلة لتوصيل البرامج الضارة. يقوم المهاجم بإرسال رسالة نصية مضمنة مع برامج ضارة Malware إلى أرقام الهواتف المحمولة.
- Mobile Adware: هي تطبيقات تقوم بإظْهار إعلانات على شاشة المستخدم بشكلٍ ذاتي بعد تنصيبها أو استعمالها. وتقوم بإعادة توجيه نتائج البحث إلى مواقع الويب الإعلانية، و تجمع بيانات المستخدم لأغراض تسويقية.
- SMS Trojans: تطبيق حصان طروادة الذي يمكنه اعتراض الرسائل النصية التي تتضمن معلومات مالية مما يمنح مجرمي الإنترنت جميع المعلومات التي يحتاجون إليها لاختراق الحسابات المصرفية للمستخدم.

4.2 طرق كشف تطبيقات اندرويد الخبيثة

لضمان أمان بيئة أنظمة Android، تم اقتراح مجموعة متنوعة من الحلول، بما في ذلك تعزيز التطبيقات و اكتشاف الثغرات الأمنية والبرامج الضارةAndroid Malware Detection. من بين خيارات تأمين أنظمة اندرويد المختلفة، تعد طريقة اكتشاف البرامج الضارة Malware Detection لنظام Android أكثر الطرق استخداما على نطاق واسع والتي يمكنها منع البرامج الضارة من الظهور في سوق تطبيقات Android أو تثبيتها واستخدامها من قبل مستخدمي الهواتف المحمولة العاملة بنظام تشغيل اندويد. بناءً على الأبحاث السابقة ، يمكن تقسيم أساليب اكتشاف البرامج الضارة لنظام Android إلى ثلاث فئات تشمل الكشف الاستاتيكى Static Detection والكشف الديناميكىDynamic Detection و الكشف الهجين Detection [3].

أساليب الكشف عن برامج اندرويد الخبيثة

بشكل عام، تنقسم أساليب الكشف عن تطبيقات اندرويد الخبيثة الى ثلاثة فروع تشمل الكشف الاستاتيكى والكشف الديناميكى والكشف الهجين.

1.4.2 الكشف الاستاتيكى Static Detection

يعتمد الكشف الثابت او الاستاتيكى على تحليل الكود المشبوه بدون تشغيل تطبيق الاندرويد. يتم إجراء التحليل الثابت من خلال تحليل ملفات Android واستخراج المعلومات مثل الأذونات المطلوبة وتسلسلات شفرة التشغيل و استدعاءات واجهة برمجة التطبيقات وما إلى ذلك. يُستخدم الكشف الثابت على نطاق واسع في مجال اكتشاف البرامج الضارة لنظام Android بسبب توفر العديد من الميزات التي يسهل استخراجها [12].

2.4.2 الكشف الديناميكي Dynamic Detection

تعتمد الطريقة الديناميكية على تشغيل التطبيق المشبوه في بيئة تشغيل محدودة Sandbox environment من خلال تتبع تسلسل استدعاء API، واستدعاء النظام، وحركة مرور الشبكة، وبيانات وحدة المعالجة المركزية لمراقبة تدفق البيانات أثناء تشغيل البرنامج، وبالتالي الكشف عن السلوك الحقيقي للبرنامج. لا يستخدم هذا الأسلوب على نطاق واسع بسبب حاجته لعدة موارد وبطء سرعة الكشف [12].

3.4.2 الكشف الهجين Hybrid Detection

يعتمد الكشف الهجين على استخدام الأسلوب الاستاتيكى والديناميكي معاً لكشف تطبيقات اندرويد الخبيثة. يمكن أن يؤدي الجمع بين التحليل الديناميكي والثابت إلى جعل اكتشاف البرامج الضارة لنظام Android أكثر دقة وكفاءة [12] .

5.2 تقنيات الكشف عن برامج اندرويد الخبيثة

بشكل عام، تضم تقنيات الكشف عن تطبيقات اندرويد الخبيثة الكشف على أساس التوقيع Signature-based detection والكشف على أساس الإذن Permission-based detection والكشف باستخدام خوارزميات تعلم الآلة Machine learning-based detection [12][13].

- الكشف على أساس التوقيع Signature-based detection: تستخرج هذه الطريقة الأنماط الدلالية وتنشئ توقيعًا فريدًا. يتم تصنيف البرامج على أنها برامج ضارة إذا كان توقيعها يتطابق مع التوقيعات الموجودة. تعتبر طريقة سريعة جدًا لاكتشاف البرامج الضارة، ومع ذلك، يمكن التحايل عليها بسهولة عن طريق تشويش التعليمات البرمجية. يمكن التعرف على البرامج الضارة الموجودة فقط وفشلها في مواجهة البديل غير المرئي للبرامج الضارة. تحتاج هذه الطريقة أيضا إلى التحديث الفوري لتوقيعات البرامج الضارة.
- الكشف على أساس الإذن Permission-based detection: تلعب الأذونات التي يطلبها التطبيق دورًا حيويًا في التحكم في حقوق الوصول. عادة، ليس لدى التطبيقات إذن للوصول إلى بيانات المستخدم والتأثير على أمان النظام. يجب أن يسمح المستخدم للتطبيق بالوصول إلى جميع الموارد المطلوبة أثناء عملية التثبيت. يجب على المطورين تحديد الأذونات المطلوبة للموارد. لكن ليس كل الأذونات المعلنة هي بالضرورة أذونات مطلوبة. يكون الاكتشاف المعتمد على الإذن سريعًا في فحص التطبيقات وتحديد البرامج الضارة، لكنه لا يقوم بتحليل الملفات الأخرى التي تحتوي على التعليمات البرمجية الضارة.
- الكشف باستخدام خوارزميات تعلم الآلة Machine learning-based detection: يقوم النظام المعتمد على تعلم الآلي بتدريب المتعلم machine learning classifier

على بيانات تضم تطبيقات ضارة (خبيثة) وعادية (حميدة). يقوم البرنامج بكشف البرامج الخبيثة يناء على المعلومات التي اكتسبها خلال التدريب. أظهرت الأبحاث الحالية أن التعلم الآلي يعد طريقة فعالة و ذات دقة عالية لكشف البرامج الضارة التي تعمل بنظام Android.

6.2 الدراسات السابقة

تقدم هذه الفقرة بعض الدراسات السابقة المعتمدة على خوارزميات التعلم الآلي Machine Learning لكشف تطبيقات اندرويد الخبيثة. على سبيل المثال، اقترحت الدراسة [7] استخدام عدد من خوارزميات تشملNaïve Bayes و KNN والآلة متجهة الدعم (SVM) والانحدار اللوجيستي Logistic Regression و شجرة القرار Decision Trees وخوارزمية الغابة العشوائية Random Forest و Gradient Boos و Cat Boost وغيرها لاكتشاف البرامج الخبيثة على نظام Android. كذلك استخدمت عدة تقنيات لتقليل الخصائص Feature Reduction مثل Principle Component Analysis و Linear Discriminat Analysis.حيث تم إنشاء مجموعة بيانات تضم 16300 سجل تمثل تطبيقات حميدة وخبيثة ممثلة بعدد 215 خاصية او ميزة. سجلت خوارزمية Cat-Boost أفضل دقة تصنيف وهى 93%.

قدمت الدراسة [8] نهجا هجينا للكشف عن تطبيقات اندرويد الخبيثة حيث استخدمت أنواع مختلفة من الميزات التي تم الحصول عليها من خلال تحليل البرامج الضارة الثابت والديناميكي. حيث تم أنشاء مجموعتين من البيانات للكشف عن التطبيقات الخبيثة. تتكون كلا من المجموعتي البيانات من 352 ميزة ثابتة و323 ميزة ديناميكية. تم استخدام خوارزمية لاختيار الميزات Feature Selection لاختيار الميزات المهمة استبعاد الميزات التى تؤثر على دقة التصنيف. من خلال هذه الخوارزمية، تم اختيار 110 و47 ميزة ثابتة في Dataset-1 وDataset-2 على التوالي و99 و35 ميزة ديناميكية في Dataset-1 و Dataset-2 على التوالي. حيث تم تطبيق مصنفات مختلفة للكشف عن Android والتعرف عليه البرمجيات الخبيثة تشمل SVM و KNN وNB وخوارزمية الشبكة العصبية Multilayer Perceptron (MLP) وخوارزمية الغابة العشوائية Random Forest.حيث أظهرت النتائج أن النهج الهجين أفضل مقارنة مع الحالات التي تم فيها استخدام الميزات الثابتة او الديناميكية وحدها. بالنسبة

لمجموعة البيانات الأولى Dataset-1، سجلت خوارزمية العابة العشوائية Random Forest أفضل دقة تصل إلى 96.5% عند استخدام الميزات الثابتة فقط و 97.01% عند استخدام الميزات الديناميكية فقط. بالنسبة لمجموعة البيانات الثانية Dataset-2، سجلت خوارزمية العابة العشوائية Random Forest أفضل دقة 86.72% عند الميزات الثابتة فقط و88.6% عند أخذ الميزات الديناميكية فقط في الاعتبار. كذلك سجلت خوارزمية العابة العشوائية Random Forest أفضل دقة في النهج الهجين (عند دمج الميزات الثابتة والديناميكية). وصلت دقة التصنيف عند استخدام Dataset-1 98.53% و90.1% عند استخدام Dataset-2.

في الدراسة [9]، اقترح الباحثون نموذجًا باستخدام التحليل الديناميكي بتطبيق خمسة خوارزميات التعلم الآلي الخاضعة للإشراف تضم شجرة القرار Decision Tree و الآلة متجهة الدعم SVM وخوارزمية الجار القريب KNN و خوارزمية نايف بايز NB و والإدراك الحسي متعدد الطبقات multi-layer perceptron. استخدمت عينات بيانات لبرامج ضارة من مشروع Android Malware Genome Project. عينة البيانات عبارة عن مجموعة من البرامج الضارة التي تم تجميعها فى الفترة بين أغسطس 2010 وأكتوبر 2011 من قبل جامعة شمال كارولينا. من بين خصائص حركة مرور الشبكة المختلفة للتطبيقات، تم اختيار ثلاث ميزات للشبكة تضم مدة الاتصال Connection duration وحجم TCP وعدد معلمات GET/POST. حصلت خوارزمية KNN على أفضل النتائج حيث بلغ معدل التصنيف الايجابي الحقيقي True Positive Rate نسبة 99.94 %.

في الدراسة [10]، تم تقييم نهج قائم على التعلم الآلي للكشف عن البرامج الضارة لنظام Android باستخدام نموذج بايزين Bayesian classification. باستخدام محلل ملفات APK لتطبيقات Android، تم استخراج مجموعة مكونة من 58 ميزة وتم اختيار أفضلها بواسطة طريقة اختيار ميزات Feature Selection. تم استخراج الخصائص من التطبيقات عن طريق أجهزة الكشف التي تبحث عن الأنماط والمراجع لاستدعاءات واجهة برمجة التطبيقات (API)، و أوامر النظام التي تظهر بشكل متكرر مع البرامج الضارة التي تعمل بنظام Android. تم استخدام 1000 عينة من 49 عائلة من البرامج الضارة التي تعمل بنظام Android بالإضافة إلى 1000 تطبيق حميد لتدريب مصنف Bayesian. بناء على التجارب التي تم إجراؤها، تم

اكتشاف أن 15 إلى 20 ميزة كافية لتوفير الأداء الأمثل لكشف تطبيقات اندرويد الخبيثة بناءً على الخصائص المكتشفة التي تعتمد عليها الميزات والتصنيف حسب وظيفة اختيار الميزة. أظهرت النتائج معدلات اكتشاف أفضل بكثير مما تم تحقيقه بواسطة برامج مكافحة الفيروسات الشهيرة القائمة على التوقيع Signature-based والتي تم اختبارها مسبقًا على نفس مجموعة عينات البرامج الضارة المستخدمة في هذه الدراسة. أظهرت النتائج أفضل TPR (معدل إيجابي حقيقي) 90.6% و بلغت اعلي دقة التصنيف 93.5%.

تقدم الدراسة [16] طريقة تعتمد على ثلاث أنواع من الميزات تشمل الثابتة Static features والديناميكية Dynamic features والجوهرية Intrinsic features للكشف عن البرامج الضارة لنظام Android. تضم مجموعة البيانات المستخدمة 30 ميزة، بالإضافة إلى 20 خاصية ديناميكية و 7 خصائص استاتيكية، تم إضافة ثلاث خصائص تضم حجم التطبيق Application size و VersionCode و عدد الملفات في ملف manifest الخاص بتطبيق اندرويد. استخدمت مجموعة بيانات Androtrack تتكون من 1552 تطبيق لاختبار الطريقة المقترحة. تم استخدام عدد من خوارزميات تعلم الآلة منها خوارزمية الجار الأقرب (k-NN) والغابة العشوائية Random Forest وشجرة القرار Decision Trees و الآلة متجهة الدعم SVM وتقنيات التعلم المجمع. بمساعدة الميزات المضافة Intrinsic Features و نهج مختلف للمعالجة المسبقة معتمد على التحليل التمييزي الخطي Linear Discriminant Analysis، تظهر النتائج أن كلا من شجرة القرار و مصنف الغابات العشوائية أنتجت دقة قدرها 99%.

استخدمت الدراسة [17] التحليل الثابت للكشف عن تطبيقات اندرويد الخبيثة بسبب التغطية الشاملة للتعليمات البرمجية وانخفاض استهلاك الموارد والمعالجة السريعة. يتطلب التحليل الثابت عدد محدود من الميزات لتصنيف البرامج الضارة بكفاءة. لذلك، تم استخدام البحث الجيني (Genetic Search)، وهو بحث يعتمد على الخوارزمية الجينية (GA)، لاختيار الميزات من بين 106 سلسلة. لتقييم أفضل الميزات التي حددتها GS، استخدمت مجموعة بيانات مكونة من 5555 تطبيق خبيث Malware و550 تطبيق حميد Benign. إضافة لذلك، استخدمت خمسة مصنفات للتعلم الآلي، وهي Naïve Bayes (NB)، والأشجار الوظيفية Functional Trees (FT)، وشجرة القرار J48، والغابة العشوائية RF، والإدراك الحسي متعدد الطبقات (MLP).

من بين هذه المصنفات، سجلت خوارزمية FT أعلى دقة 95% ومعدل إيجابي حقيقي TPR بلغ 96.7%.

في الدراسة [18]، تم اختبار أداء عدد من خوارزميات تعلم الآلة الخاضعة للإشراف تضم كل من (KNN وCNN وNB وRF وSVM وDT) لكشف تطبيقات اندرويد الضارة. استخدمت مجموعة بيانات مكونة من 17394 تطبيق ممثلة بعدد 279 خاصية. أظهرت نتائج المقارنة تسجيل خوارزمية شجرة القرار DT أفضل النتائج حيث بلغت دقة التصنيف 99% تليها خوارزمية CNN بنسبة 98.76%. سجلت خوارزمية Naïve Bayes اقل دقة تصنيف 89.71%

تبين الدراسات السابقة فعالية خوارزميات تعلم الآلة في مجال كشف تطبيقات اندرويد الخبيثة. كذلك يلاحظ إن الدراسات استخدمت مجموعات بيانات مختلفة لاختبار أداء النماذج المقترحة. نستنتج من خلال الدراسات السابقة أن أنظمة الكشف عن تطبيقات اندرويد الخبيثة المعتمدة على خوارزميات تعلم الآلة توفر دقة عالية في كشف التطبيقات الضارة.

7.2 طرق المعالجة الأولية للبيانات

تعتبر معالجة البيانات المفقودة وتطبيع قيم الخصائص في مجموعة البيانات المستخدمة في تدريب الخوارزميات من المراحل الأساسية لبناء مصنفات البيانات المعتمدة على خوارزميات تعلم الآلة.

1.7.2 معالجة القيم المفقودة Missing Values

أشهر الطرق لمعالجة مشكلة القيم المفقودة [1] :

- حذف خصائص البيانات التي تحتوى على عدد كبير من القيم المفقودة وتعتبر إستراتيجية بسيطة وفعالة لحل مشكلة القيم المفقودة في مجموعة البيانات.
- يمكن تقدير القيم المفقودة للخاصية أو السمة في مجموعة البيانات باستخدام القيم المتبقية او الموجودة. إذا كانت السمة متصلة رقميه "Numeric"، يتم استخدام متوسط القيم المتبقية للسمة للقيم المفقودة. إذا كانت الصفة منفصلة، فيمكن اعتبار القيمة المفقودة للسمة مساوية للقيمة الأكثر شيوعًا في القيم المتبقية.

2.7.2 تطبيع البيانات Data Normalization

تتعامل خوارزميات التعلم الآلي مع القيم الرقمية للخصائص او (للسمات). لذلك، إذا كان هناك اختلاف كبير في نطاق الأرقام، التي تمثل قيم الميزات فأن ذلك يؤثر علي أداء نموذج التصنيف. من المعروف ان خوارزميات التعلم الآلي تتعامل مع الأرقام ولا تعرف ما يمثله الرقم علي سبيل المثال، يبلغ وزن كائن عنصر 10 جرامات وسعره 10 دولارات، هي تمثل خاصيتين مختلفتين تمامًا، وهو أمر واضح للبشر، ولكن بالنسبة للنموذج، كلاهما يعتبر ميزة. تلعب قيم الخصائص الأرقام الأكثر أهمية دورًا أكثر مهماً في تدريب النموذج، لذلك فإن تحجيم "تطبيع" السمات مهم لجعل السمات في نطاق متقارب، تمثل طريقة المقياس المتغير البسيط و مقياس Z اهم طرق تطبيع قيم الخصائص [1][2].

في طريقة المقياس المتغير البسيطة Simple feature scaling لتطبيع قيم الخصائص، تقسيم كل قيمة حسب القيمة القصوى المتغير هذا يجعل القيم بين الصفر و الواحد.

$$\text{Xnew} = \frac{Xold}{\text{Xmax}}$$

تعتبر الدرجة z ، والتي تسمى أيضًا الدرجة القياسية واحدة من أكثر الطرق شيوعًا لتوحيد البيانات، والتي يمكن إجراؤها عن طريق طرح المتوسط وتقسيمه على الانحراف المعياري لقيمة كل سمة .معادلتها الرياضية على النحو التالي:

$$\text{z} = \frac{x - \mu}{\sigma}$$

في هذه المعادلة ، x هي القيمة قبل التطبيع ، و μ هي متوسط العينة و σ هي الانحراف المعياري للعينة . بمجرد اكتمال توحيد البيانات، سيكون لجميع السمات متوسط صفر، وانحراف معياري بواحد، وبالتالي، نفس المقياس .

3.7.2 اختيار الخصائص Feature Selection

عملية اختيار الميزات (FS) لها تأثير كبير على أداء مصنفات تعلم الآلة. في تصنيف البيانات، يتم تطبيق FS بشكل أساسي لاكتشاف وإزالة الميزات التي لا تحمل معلومات تفيد في الفصل بين فئات البيانات Classes في مجموعة البيانات والتي تؤثر سلبا على أداء مصنفات التعلم الآلي. بشكل عام، يمكن أن تكون تقنية FS إما مرشح أو التفافية. في الطريقة اختيار

الخصائص الالتفافية Wrapper FS ، تستخدم خوارزمية للبحث مثل الخوارزمية الجينية Genetic Algorithm بالتزامن مع مصنف التعلم الآلي مثل KNN لاختيار الميزات لتصنيف البيانات. في المقابل، تعتمد أساليب المرشح خوارزميات Filter FS على الطرق الإحصائية مثل مربع كايChi-Square والمعلومات المتبادلة Mutual Information في تحديد أفضل مجموعة فرعية من الميزات Best subset of features [21][12].في هذا البحث، سيتم دراسة تأثير اختيار الخصائص Feature Selection على أداء الخوارزميات المذكورة. حيث سيتم توظيف احد طرق اختيار الخصائص (الميزات) التي توفرها لغة البايثون مثل المعلومات المتبادلة Information Gain أو مربع كاى Chi-Square او نسبة اكتساب المعلومات Gain Ratio لاختيار أهم الخصائصBest subset of features و استخدامها في تصنيف تطبيقات اندرويد Android Apps.

1.3.7.2 طريقة نسبة اكتساب المعلومات لاختيار الميزات Information Gain Ratio

اكتساب المعلومات هي طريقة تعتمد على نظرية المعلومات وتستخدم في اختيار الميزات Feature Selection لتصنيف البيانات باستخدام خوارزميات التعلم الألى. تقوم تقنية اكتساب المعلومات Information Gain على قياس مدى المعلومات التي تمتلكها كل ميزة من الميزات في مجموعة البيانات. انتروبيا المعلومات Entropy هو قياس عدم اليقين في البيانات. في سياق تصنيف البيانات باستخدام خوارزميات التعلم الآلي، يستخدم الإنتروبيا فى قياس تنوع التسميات (Labels). تحسب الإنتروبيا بواسطة الصيغة التالية [15]:

$$\mathrm{H} = -\sum_{i=1}^{m} P(c_i)\ \log_2 P(c_i)$$

حيث $P(C_i)$ هي نسبة الفئة C_i في مجموعة البيانات. على سبيل المثال، يحسب Entropy لمجموعة البيانات التي تحتوي على 30 عينة إيجابية و70 عينة سلبية على النحو الأتي:

$$\mathrm{H} = -((0.3\log_2(0.3) + (0.7\log_2(0.7)) \approx 0.88$$

يتم حساب نطاق الميزة (a) نحو الانتروبيا للفئة. الميزة المهمة هي التى تقلل من إنتروبيا الفصل إلى الحد الأقصى.

$$H(c|a) = -\sum_{a \in A} P(a) * \sum_{c \in C} P(c\,|a) \log P(c\,|a)$$

$$Information\ Gain\ (a) = H(c) - H(c|a)$$

P(x) : احتمال (X).

H(x) : إنتروبيا (X).

نسبة كسب المعلومات هي طريقة أخرى لاختيار الميزات تعتمد على نظرية المعلومات. يتم تطبيع الميزة المعطاة من خلال حساب القيمة الجوهرية للميزة. تحاول نسبة الكسب تقليل انحياز اكتساب المعلومات من خلال تطبيع يسمى المعلومات الجوهرية. يتم تعريف المعلومات الجوهرية (Intrinsic Value) على أنها إنتروبيا نسب مجموعة البيانات الفرعية Sub−Class . تحسب المعلومات الجوهرية بالصيغة التالية:

$$Intrinsic\ Value = -\sum_{j=1}^{m} \frac{Dj}{D} \log_2 \frac{Dj}{D}$$

بعد الحصول على Intrinsic Value للميزة a. تحسب نسبة كسب المعلومات لميزة محددة a بالصيغة التالية:

$$Gain\ Ratio\ _a = \frac{Information\ Gain_a}{Intrinsic\ Value_a}$$

8.2 خوارزميات التعلم الالى Machine Learning Algorithms

تقدم هذه الفقرة شرح لكيفية عمل خوارزميات تعلم الآلة المستخدمة في هذا البحث لتصنيف تطبيقات اندرويد الخبيثة. تشمل الخوارزميات المستخدمة خوارزمية الجار الأقرب و خوارزمية شجرة القرار و خوارزمية الآلة متجهة الدعم و خوارزمية نايف بيز و خوارزمية الانحدار اللوجستى.

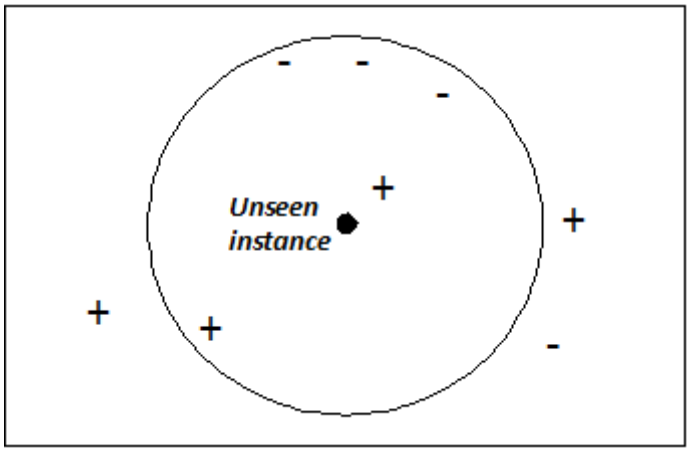
1.8.2 خوارزمية الجار الأقرب K Nearest Neighbour

تعتبر خوارزمية الجار الأقرب واحدة من أقدم وأبسط خوارزميات التعلم الخاضع للإشراف والأكثر فاعلية لتصنيف مجموعات البيانات . تعتمد خوارزمية KNN على افتراض أن الأشياء المتشابهة قريبة من بعضها البعض. بالمقارنة مع خوارزميات التصنيف الأخرى، فإن أقرب جار يستخدم أسلوب التعلم الكسول. بمعنى آخر، يقوم مصنف KNN بتخزين العينات في مرحلة التدريب ويؤخر عملية التدريب إلى حين استلام عينات الاختبار [1]. غالبًا ما تُستخدم دالة المسافة الإقليدية Euclidian Distance القياسية لقياس التشابه بين حالتين. يتم تعريفه على النحو التالي:

$$d(x,y) = \sqrt{\sum_{i=1}^{N}(a_i(x) - a_i(y))^2}$$

في الشكل التالي، إذا تم تعيين K بقيمة 1 للمصنف KNN، فسيتم تصنيف المثيل غير المرئي على أنه مثيل إيجابي (+). إما إذا خصصت القيمة 5 للمتغير K، فإن تصنيف المثيل غير المرئي يكون سالب.لان عدد الحالات السالبة اكبر من الموجبة عندما تكون قيمة K تساوى 5.

يبين الشكل (2.2) التصنيف باستخدام خوارزمية الجار الأقرب KNN.



شكل (2.2) التصنيف باستخدام خوارزمية الجار الأقرب KNN

2.8.2 خوارزمية نايف بايزNaive Bayes

نايف بايز Naive Bayes هي تقنية تصنيف بسيطة تستند إلى نظرية بايز، بافتراض أن جميع الخصائص التي تتنبأ بالقيمة المستهدفة مستقلة عن بعضها البعض. تحسب هذه التقنية احتمالية

كل فئة ثم تحدد الفئة الأكثر احتمالية (Target (Yes, No. المصنف يفترض أن خصائص الإدخال التي تدخل النموذج مستقلة عن بعضها البعض. وبالتالي، لا يؤثر تغيير ميزة إدخال واحدة على أي ميزة أخرى [1]. نفرض مجموعة من الأمثلة التدريبية حيث يتم وصف كل مثال X بواسطة مجموعة من قيم الميزات <F1,F2,...,Fn> و فئات تصنيف الهدف مثل (Yes أو No). لإسناد حالة الاختبار(t) إلى احد الفئات، استنادًا إلى قيم الميزات، تقوم NB بتعيين مثال الاختبار للفئة ذات الاحتمالية الأعلى. بناء على نظرية بييز:

$$P(C|t) = \frac{P(t|C)P(C)}{P(t)}$$

P(C|t) : احتمال وقوع الحدث C فى حال وقوع الحدث t.

P(t|C) : احتمال وقوع الحدث t فى حال وقوع الحدث C.

P(C) : احتمال وقوع الحدث C.

P(t) : احتمال وقوع الحدث t.

3.8.2 خوارزمية الآلة المتجه الدعم Support Vector Machine

تحدد خوارزمية SVM حدود القرار التي تزيد المسافة من أقرب نقاط البيانات لجميع الفئات. و يسمى حد القرار الذي تم إنشاؤه بواسطة آلات متجه الدعم الآلي بالهامش الأقصى. يعمل مصنف SVM الخطي البسيط عن طريق إنشاء خط مستقيم (فاصل) بين فئتين. هذا يعني أن جميع نقاط البيانات على جانب واحد من الخط تمثل فئة وأن نقاط البيانات على الجانب الآخر من الخط تمثل فئة مختلفة [1]. يمكن تمثيل المستوى الفائق للفصل في فضاء العينة على أنها الدالة الخطية التالية:

$$W^T.X + b = 0$$

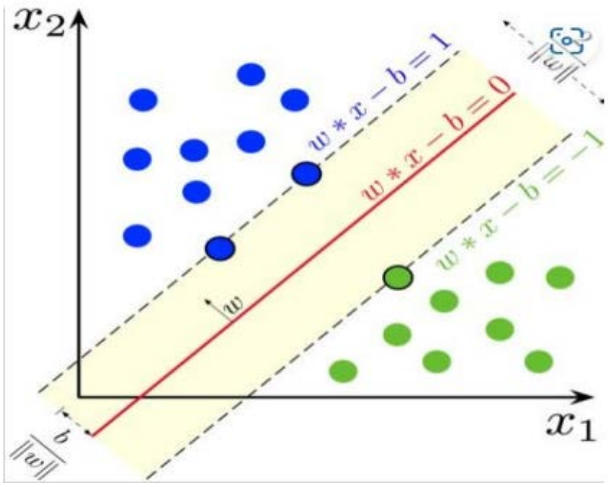
يُطلق على W اسم ناقل الوزن للمستوى الفائق الأمثل ويُعرف b باسم التحيز. تصنيف البيانات الأساسية في فضاء الإدخال وفقاً للنموذج الرياضي التالي:

الفئة الموجبة: $W^T,X_j+b >=+1$ for $d_j =+1$,$j=1,2,3,...,N$

القئة السالبة: $W^T,X_j+b >=-1$ for $d_j =-1$,$j=1,2,3,...,N$

يمثل W متجه الأوزان، x يمثل متجه الإدخال، b يمثل قيمة التحيز و d يمثل قيمة الإخراج.

يوضح الشكل (3.2) التصنيف باستخدام خوارزمية SVM.



الشكل (3.2) التصنيف باستخدام خوارزمية SVM

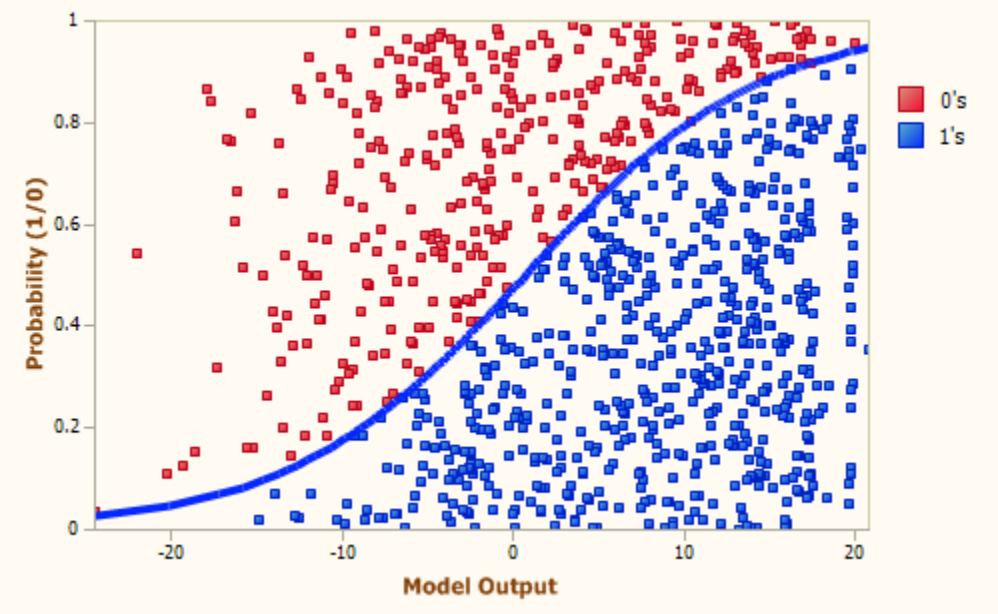
4.8.2 خوارزمية شجرة القرار Decision Tree

شجرة القرار هي أحد خوارزميات تعلم الآلة الخاضعة للإشراف التي يمكن استخدامها فى التصنيف والانحدار Classification or Regression، وتعرف بأنها بنية هيكلية تأخذ شكل الشجرة تستخدم لتجزئة مجموعة كبيرة من السجلات إلى مجموعات فرعية صغيرة باستخدام قواعد القرار البسيطة وفق تتابع محدد. تتكون شجرة القرار من عقدة رئيسية واحدة تسمى عقدة الجذر Root Node و عدة عقد وسطية إضافة إلى العقد الطرفية ، يتم تقسيم العينات في كل عقدة إلى عقد فرعية بناء على نتائج التفرع. يمثل كل مسار من عقدة الجذر إلى العقدة الطرفية تسلسل القرار، تحاول شجرة القرار تفريع مجموعة البيانات المستخدمة لبناء شجرة القرار بطريقة تجعل البيانات في كل مجموعة فرعية متشابهة قدر الإمكان وكذلك تختلف البيانات في كل مجموعة جزئية قدر الإمكان عن البيانات في المجموعات الجزئية الأخرى المكونة لشجرة القرار. تمثل العقدة الطرفية او الورقة نهاية مسار فى الشجرة وتمثل احد فئات الهدف او القرار مثل (سالبة او موجبة). الهدف هو إنتاج شجرة يمكنها تعميم العينات غير المرئية [1].

5.8.2 خوارزمية الانحدار اللوجستى Logistic Regression

الانحدار اللوجستي (Logistic Regression) هو نموذج إحصائي ينتمي لنماذج الانحدار الخطي يمكن من نمذجة متغير ثنائي الحد بدلالة مجموعة من المتغيرات العشوائية المتوقعة، رقمية

كانت أو فئوية. يستخدم الانحدار اللوجستي للتنبؤ باحتمالية وقوع حدث ما بمعرفة إضافية لقيم متغيرات يمكن أن تكون مفسرة أو مرتبطة بهذا الحدث. يبين الشكل (4.2) طريقة الفصل بين البيانات باستخدام خوارزمية الانحدار اللوجستى [14]. هناك أنواع من الانحدار اللوجستى تشمل:



الشكل (4.2) خوارزمية الانحدار اللوجستى

- الانحدار اللوجستي ثنائي الحدين: هو نوع من الانحدار اللوجستي يستخدم عندما يكون المتغير التابع (الهدف) ثنائي الحدين، أي يأخذ قيمتين مختلفتين فقط. يستخدم الانحدار اللوجستي للتنبؤ باحتمالية وقوع حدث ما بمعرفة إضافية لقيم متغيرات يمكن أن تكون مفسرة أو مرتبطة بهذا الحدث. يعمل علي بيانات تحتوي علي متغير تابع (الهدف) ثنائي الحدين، أي يأخذ قيمتين مختلفتين فقط، يمكن أن تكون هذه القيمتين مثلاً 0 و 1، أو نعم و لا.
- الانحدار اللوجستي متعدد الحدود: هو نوع من الانحدار اللوجستي يستخدم عندما يكون المتغير التابع (الهدف) متعدد الحدود، أي يأخذ أكثر من قيمتين مختلفتين. يستخدم الانحدار اللوجستي للتنبؤ باحتمالية وقوع حدث ما بمعرفة إضافية لقيم متغيرات يمكن أن تكون مفسرة أو مرتبطة بهذا الحدث.

9.2 طرق تقييم فعالية النماذج

هناك عدة معايير لقياس أداء خوارزميات التعلم الالى تضم دقة التصنيف Classification accuracy و مقياس F1−Score و مصفوفة الارتباك Confusion Matrix.

1.9.2 مصفوفة الارتباك (Confusion Matrix)

يبين الشكل (5.2) مصفوفة الارتباك [1][2].

		المتوقعة *Predicted*	
الحقيقية *Actual*		Yes	No
	Yes	*TP*	*FN*
	No	*FP*	*TN*

الشكل (5.2) مصفوفة الارتباك Confusion Matrix

- عدد مرات التنبؤ الصحيحة الايجابية True Positive (TP)

وهي الحالات الإيجابية في مجموعة الاختبار والتي تم توقعه بشكل صحيح. الإيجابيات الحقيقية (TP) عندما تكون القيمة الفعلية إيجابية والمتوقعة تكون إيجابية أيضًا.

- عدد مرات التنبؤ الصحيحة السلبية True Negative (TN)

وهي الحالات السلبية في مجموعة الاختبار والتي تم توقعه بشكل صحيح. السلبيات الحقيقية (TN) عندما تكون القيمة الفعلية سالبة والتنبؤ سلبي أيضًا.

- عدد مرات التنبؤ الخاطئة الايجابية False Positive (FP)

وهي حالات سلبية في مجموعة الاختبار والتي تم توقعه بشكل خطأ. الإيجابيات الخاطئة (FP) عندما يكون الفعلي سلبيًا ولكن التنبؤ إيجابيًا.

- عدد مرات التنبؤ الخاطئة السلبية FN

وهي حالات إيجابي في مجموعة الاختبار والتي تم توقعه بشكل خطأ. السلبيات الخاطئة (FN) عندما يكون الفعلي موجبًا لكن التوقع سلبي.

2.9.2 دقة التصنيف Classification Accuracy

هي قدرة أو دقة المصنف على التنبؤ الصحيح وهي عبارة عن عدد حالات التنبؤ الصحيحة علي عدد التنبؤات الكلي. وتحسب بالعلاقة

$$Accuracy=TP+TN/(TP+FP+TN+FN)$$

3.9.2 مؤشر Precision

هو نسبة الحالات الصحيحة الايجابية بالنسبة لعدد المرات الايجابية الصحيحة والخاطئة، ويعطى بالعلاقة:

$$Precision = TP/(TP+FP)$$

4.9.2 مؤشر Recall

هو نسبة الحالات الصحيحة الايجابية بالنسبة لعدد المرات الصحيحة الايجابية والخاطئة السلبية، ويعطى بالعلاقة التالية:

$$Recall = TP/(TP+FN) \quad \text{أو} \quad TN/(TN+FN)$$

5.9.2 مؤشر Fl-score

يمثل المتوسط التوافقي Harmonic Mean بين المؤشرين Precision و Recall، ويعطى بالعلاقة التالية:

$$F1-Score = 2 * ((precision * Recall) / (precision + Recall))$$

الفصل الثالث
تطبيق منهجية البحث
والنتائج والمناقشة

3. مقدمة

يتضمن هذا الفصل شرح خطوات تطبيق منهجية البحث و التجارب التي أجريت على مجموعة البيانات لتصنيف تطبيقات اندرويد الخبيثة. كذلك يوضح الأدوات البرمجية المستخدمة. يشمل كذلك هذا الفصل النتائج ومناقشتها وتحديد أفضل الخوارزميات لتصنيف تطبيقات اندرويد الخبيثة.

1.3 الأدوات البرمجية والمكتبات المستخدمة

في هذا البحث، تم استخدام عدد من المكتبات و الأدوات التي توفرها لغة Python:

- Scikit-learn تتضمن خوارزميات تعلم الآلة.
- Numpy هي مكتبة تستخدم للعمل مع المصفوفات.
- Pandas هي إحدى مكتبات بايثون مفتوحة المصدر التي توفر أدوات وتقنيات تحليل البيانات. نستخدم لتحميل و فتح ملفات مجموعات البيانات مثل CSV و XLS.
- Orange Data Mining Software

2.3 تجميع البيانات

في هذا البحث، سيتم استخدام مجموعة بيانات تسمي Dataset Derbin متاحة للبحث فى مجال كشف تطبيقات اندرويد الخبيثة مكونة من 15036 تطبيق و215 ميزة. تأخذ كل ميزة قيمة ثنائية (صفر أو واحد). القيمة صفر تعنى إن التطبيق لا يحتوي على تلك الميزة والقيمة 1 تعنى إن التطبيق يحتوي تلك الميزة. التطبيقات مقسمة إلى فئتين (حميدة وخبيثة). تنقسم مجموعة بيانات الى 5560 تطبيقًا من تطبيقات Android الضارة و9476 تطبيقًا من تطبيقات Android الحميدة. تنتمي التطبيقات الضارة الموجودة في مجموعة البيانات من 49 عائلة مختلفة من البرامج الضارة، مثل Goldmaster و DroidKungFu و GingerMaster و FakeInstaller وغيرها. بينما تم تجميع التطبيقات الحميدة من Google Play Store وبعض المصادر الأخرى [11]. يبين الشكل (1.3) مجموعة بيانات Derbin لتطبيقات Dataset Android. يبين الجدول (1.3) بعض ميزات (خصائص) مجموعة البيانات. مثل الاذونات Permissions واستدعاءات مكتبات API call

جدول(1.3) مجموعة بيانات تطبيقات Android

ACCESS_WIFI_STATE	WRITE_EXTERNAL_STORAGE	ACCESS_FINE_LOCATION	SET_WALLPAPER_HINTS	SET_PREFERRED_APPLICATIONS	WRITE_SECURE_SETTINGS	class
0	1	0	0	0	0	Malware
0	1	0	0	0	0	Malware
0	0	0	0	0	0	Malware
1	1	1	0	0	0	Malware
1	0	1	0	0	0	Malware
1	1	0	0	0	0	Malware
0	1	0	0	0	0	Malware
1	1	1	0	0	0	Malware
0	1	0	0	0	0	Malware

جدول (2.3) بعض خصائص مجموعة بياناتDerbin

Transact	API call
WRITE_SMS	Manifest Permission
INSTALL_PACKAGES	Manifest Permission
SEND_SMS	Manifest Permission
RECORD_AUDIO	Manifest Permission
READ_PROFILE	Manifest Permission
android.telephony.SmsManager	API call
READ_PHONE_STATE	Manifest Permission
READ_HISTORY_BOOKMARKS	Manifest Permission
ClassLoader	API call signature
Landroid.content.Context.registerReceiver	API call
Ljava.lang.Class.getField	API call
GET_ACCOUNTS	Manifest Permission
RECEIVE_SMS	Manifest Permission
READ_SMS	Manifest Permission
READ_EXTERNAL_STORAGE	Manifest Permission
HttpGet.init	API call
SecretKey	API call

3.3 المعالجة المسبقة لمجموعة البيانات

- القيم المفقودة: بعد استكشاف مجموعة البيانات تبين عدم وجود قيم مفقودة.
- تطبيع البيانات: تأخذ جميع قيم الخصائص لمجموعة البيانات احد القيمتين 0 أو 1 لذلك لا تحتاج قيم الخصائص إلى تطبيع.
- تقسيم البيانات: تم تقسيم البيانات إلى مجموعتين تستخدم احدها (الأكبر) في بناء نموذج التصنيف (تدريب النموذج) وتسمى Training set بينما تستخدم المجموعة الأخرى (الأصغر) لاختيار أداء نموذج التصنيف وتسمى Test set. تبلغ نسبة مجموعة التدريب 80 % من إجمالي عدد الحالات بينما تشكل مجموعة الاختبار النسبة المتبقية وهى 20 %.

4.3 التجربة الأولى

يبين الجدول (2.3) نتائج تصنيف الخوارزميات حيث تم تدريب كل مصنف على مجموعة التدريب المكونة من 12029 تطبيق (تطبيقات حميدة وخبيثة) والتي تمثل 80% من مجموعة البيانات واختبار أدائه باستخدام مجموعة الاختبار التي تضم 3007 تطبيق والتي تمثل 20% من مجموعة البيانات.حيث تم استخدم خوارزميات تعلم الآلة بمعاملاتها الافتراضية في مكتبات بايثون بدون تغيير. على سبيل المثال، القيمة الافتراضية للمعامل K لخوارزمية الجار الأقرب KNN هي 5 ودالة النواة لخوارزمية الآلة متجهة الدعم SVM هي الدالة الخطية Linear Kernel.

تبين نتائج دقة التصنيف Classification Accuracy إن خوارزمية الجار الأقرب KNN سجلت أعلى دقة تصنيف 0.951 بينما سجلت خوارزمية الآلة متجهة الدعم SVM اقل دقة تصنيف 0.61. كذلك تبين نتائج التصنيف إن خوارزمية شجرة القرار سجلت ثاني أفضل خوارزمية بعد KNN حيث بلغت دقة تصنيف خوارزمية شجرة القرار DT في تصنيف تطبيقات اندرويد إلى حميدة Benign أو خبيثة Malware 0.948. سجلت خوارزمية الانحدار اللوجستي LR دقة بلغت 0.868 تليها خوارزمية نايف بيز NB بنسبة بلغت 0.75.

يبين مؤشر Precision جزء الحالات الايجابية التي تم توقعها بشكل صحيح من العدد الكلى للحالات الايجابية المتوقعة. نلاحظ من الجدول (2.3) إن خوارزمية KNN سجلت اعلي معدل Precision يصل الى 0.951. في المقابل، سجلت خوارزمية SVM اقل نسبة وهى 0.587

وهذا يشير إلى ارتفاع عدد الحالات التي تم إسنادها إلى فئة محددة وهى فعليا لا تنتمي إلى تلك الفئة (حالات سلبية تم إسنادها إلى الفئة الايجابية) False Positive.

يبين مؤشر الاستجابة Recall أو True Positive Rate نسبة الحالات الايجابية التي تم توقعها بشكل صحيح من إجمالي عدد الحالات الايجابية في مجموعة الاختبار. نلاحظ من الجدول (2.3) تسجيل خوارزمية KNN أفضل معدل استجابة بلغ 0.951 تليها خوارزمية شجرة القرار DT بنسبة 0.948 بينما سجلت خوارزمية SVM ادني معدل استجابة يصل إلى 0.61. يشير انخفاض معدل الاستجابة إلى ارتفاع عدد الحالات التي تم إسنادها بشكل خاطئ إلى فئة غير الفئة التي تنتمي إليها فعليا تلك الحالات اى ارتفاع نسبة False Positive.

يبين مقياس F1-Score الذي يمثل المتوسط التوافقي بين مؤشر Precision و Recall تسجيل خوارزمية KNN أفضل نسبة تصل إلى 0.951 تليها خوارزمية شجرة الفرار بنسبة 0.948 بينما سجلت خوارزمية SVM ادني معدل F1-Score يصل إلى 0.59 مقارنة بجميع الخوارزميات التي تم استخدامها في تصنيف تطبيقات اندرويد الخبيثة في هذا البحث. بشكل عام، تبين نتائج التصنيف فعالية خوارزميات تعلم الآلة في مجال تصنيف تطبيقات اندرويد الخبيثة.

جدول (3.3) نتائج تصنيف مجموعة بيانات Derbin قبل اختيار الخصائص

F1-Score	Precision	Recall	Accuracy	Model
0.951	0.951	0.951	0.951	KNN
0.867	0.867	0.868	0.868	LR
0.753	0.763	0.75	0.75	NB
0.59	0.587	0.61	0.61	SVM
0.948	0.948	0.948	0.948	DT

5.3 التجربة الثانية

تهدف هذه التجربة إلى دراسة تأثير اختيار الخصائص Feature Selection على دقة تصنيف خوارزميات تعلم الآلة حيث تم في هذه التجربة تطبيق طريقة نسبة اكتساب المعلومات باستخدام طريقة(IGR) Information Gain Ration لاختيار الخصائص المهمة في التمييز بين التطبيقات الحميدة والخبيثة و استبعاد الخصائص التي تؤثر سلبا على دقة التصنيف. تم اختبار عدة قيم لنسبة اكتساب المعلومات بين 0.05 و 0.2 بهدف تحديد افصل قيمة لاختيار

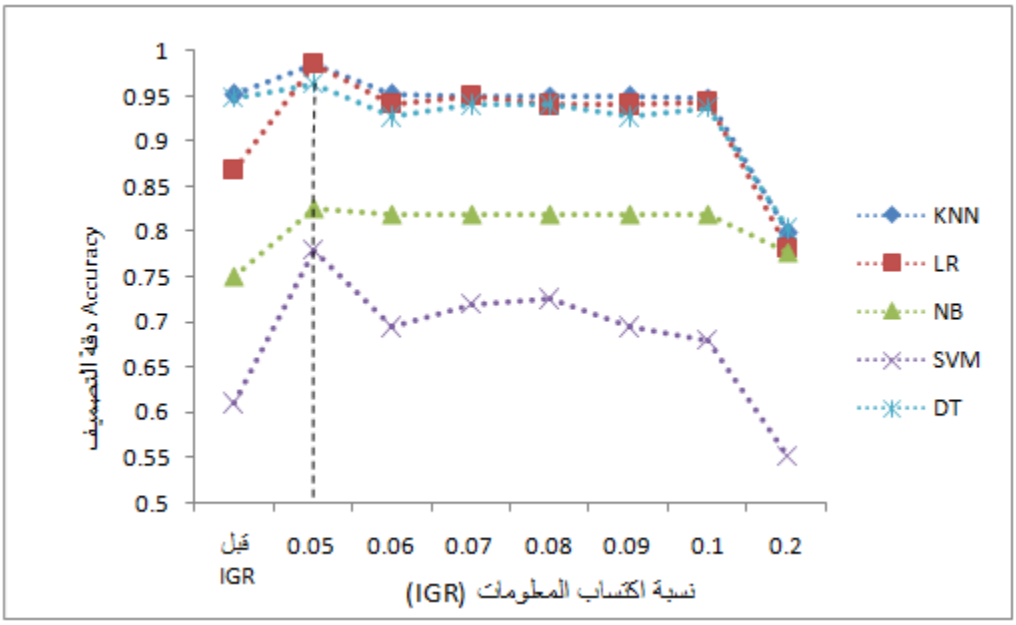
الخصائص. يبين الجدول (3.3) قيم IGR لعدد من الخصائص مرتبة ترتيبا تنازليا. يلاحظ تسجيل الخاصية transact اعلي قيمة نسبة اكتساب معلومات (0.277) تليها الميزة SEND_SMS بقيمة (0.27). في حال تحديد العتبة 0.2 لاختيار الخصائص، سيتم اختيار الميزات التسعة الأولى لتمثيل تطبيقات اندرويد في مجموعة البيانات.

الجدول (4.3) أفضل 20 خاصية بناء على قيمة نسبة اكتساب المعلومات IGR

قيمة IGR	الميزة Feature
0.277	Transact
0.275	SEND_SMS
0.262	attachInterface
0.259	\ onServiceConnected
0.258	bindService
0.256	ServiceConnection
0.242	android.os.Binder
0.208	Ljava.lang.Class.getCanonicalName
0.201	Ljava.lang.Class.getMethods
0.196	android.telephony.SmsManager
0.192	createSubprocess
0.188	Ljava.lang.Class.cast
0.182	getBinder
0.166	android.content.pm.Signature
0.164	android.telephony.gsm.SmsManager
0.161	Ljava.net.URLDecoder
0.156	getCallingUid
0.153	RECEIVE_SMS
0.148	USE_CREDENTIALS
0.143	MANAGE_ACCOUNTS

تبدأ الميزات التي لها قيمة (IGR >= 0.2) بالميزة transact التي سجلت أعلى نسبة اكتساب معلومات وتنتهي بأخر ميزة سجلت IGR اكبر من أو يساوى 0.2 وهى بالميزة Ljava.lang.Class.getMethods التي سجلت نسبة اكتساب معلومات (0.201). جميع

الميزات التي لها قيمة IGR اقل من 0.2 يتم استبعادها من مجموعة الميزات المستخدمة في تمثيل البيانات.تم اختبار عدة قيم ، لتحديد أفضل قيمة (عتبة) Threshold لاختيار الميزات لتصنيف تطبيقات اندرويد، يبين الشكل (2.3) نتائج التصنيف باستخدام عدة قيم لنسبة اكتساب المعلومات. تشير النتائج الى تسجيل القيمة (0.05) أفضل دقة تصنيف، لذلك تم اختيار الميزات التي لها نسبة اكتساب معلومات اكبر من أو تساوى (0.05). تعتبر الميزات التي سجلت قيمة IGR اقل من 0.05 لا تحمل معلومات تساعد فى الفصل بين تطبيقات اندرويد الحميدة والخبيثة وبالتالي تم استبعادها من مجموعة الميزات المستخدمة في تمثيل البيانات.



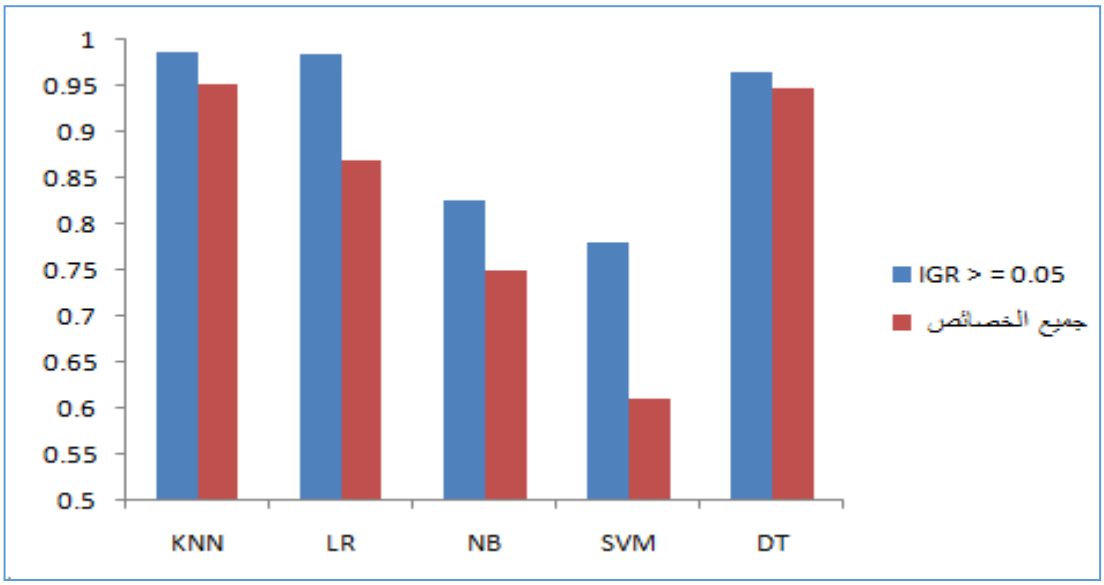
الشكل (1.3) نتائج التصنيف باستخدام قيم مختلفة لنسبة اكتساب المعلومات

يبين الجدول (5.3) نتائج التصنيف بعد اختيار أفضل الميزات باستخدام نسبة المعلومات المكتسبة IGR. مقارنة بالنتائج السابقة، نلاحظ من الجدول تحسن أداء جميع الخوارزميات بعد استبعاد الميزات التي لها IGR اقل من (0.05).

جدول (5.3) نتائج التصنيف بعد تطبيق اختيار الخصائص باستخدام IGR

F1-Score	Precision	Recall	Accuracy	Model
0.986	0.986	0.986	0.986	KNN
0.985	0.98	0.98	0.98	Logistic Regression
0.822	0.86	0.822	0.826	Naive Bayes
0.778	0.777	0.78	0.78	SVM
0.964	0.964	0.964	0.964	DT

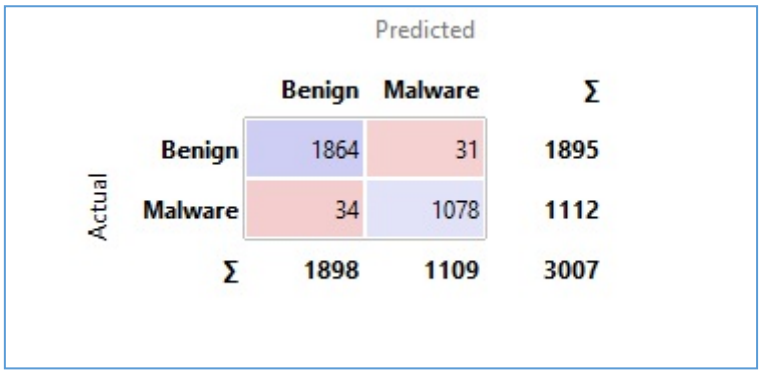
نلاحظ تسجيل خوارزمية KNN أفضل دقة 0.986 تليها خوارزمية الانحدار اللوجستى LR التي سجلت دقة تصنيف بلغت 0.98 تليها النسبة 0.964 التي سجلتها خوارزمية شجرة القرار DT, سجلت خوارزمية SVM القيمة 0.78 اقل دقة تصنيف. تبين مؤشرات Precision و Recall و F1-Score تحسن أداء جميع الخوارزميات . على سبيل المثال، ارتفع مؤشر F1-Score لخوارزمية الانحدار اللوجستى LR من 86 % إلى 98 % بنسبة 12 %. يبين الشكل (2.3) مقارنة بين نتائج دقة التصنيف قبل اختيار الميزات وبعدها لجميع المصنفات. نلاحظ تحسن أداء جميع الخوارزميات خاصة خوارزمية الآلة متجهة الدعم SVM و خوارزمية الانحدار اللوجستى LR.



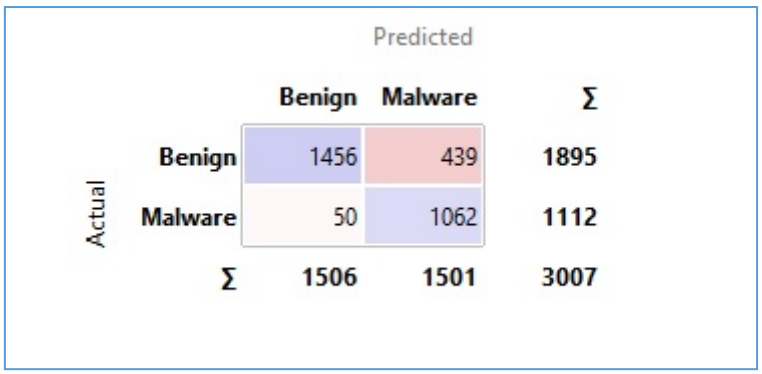
الشكل (2.3) مقارنة بين نتائج التصنيف قبل و بعد اختيار الخصائص باستخدام IGR

يبين الشكل (3.3) مصفوفة الارتباك لخوارزمية KNN بعد اختيار الميزات. تمثل القيمة 3007 عدد العينات (التطبيقات) في مجموعة الاختبار. نلاحظ أن خوارزمية KNN صنفت بشكل صحيح عدد 1864 تطبيق حميد Benign و الذي يمثل عدد الحالات الايجابية الحقيقية (True Positive) وعدد 1078 تطبيق خبيث Malware و التي تمثل عدد الحالات السلبية الحقيقة (True Negative). يمثل العدد 31 عدد حالات التي تم إسنادها إلى الفئة السلبية (Malware) وهى في الحقيقة تتبع الفئة الايجابية أي أنها تطبيقات حميدة (False Negative). نلاحظ كذلك أن خوارزمية KNN قامت بإسناد 34 تطبيق خبيث Malware إلى

الفئة الايجابية (الحميدة). يبين الشكل (4.3) مصفوفة الارتباك لمصنف نايف بيز NB. نلاحظ ارتفاع عدد حالات التصنيف السلبي الخاطئ إلى 439 حالة. هذه الحالات هي فعليا تطبيقات حميدة و لكن تم إسنادها إلى فئة التطبيقات الخبيثة (الفئة السلبية) وهذا يعكس انخفاض معدل الاستجابة Recall للمصنف NB إلى (0.822).

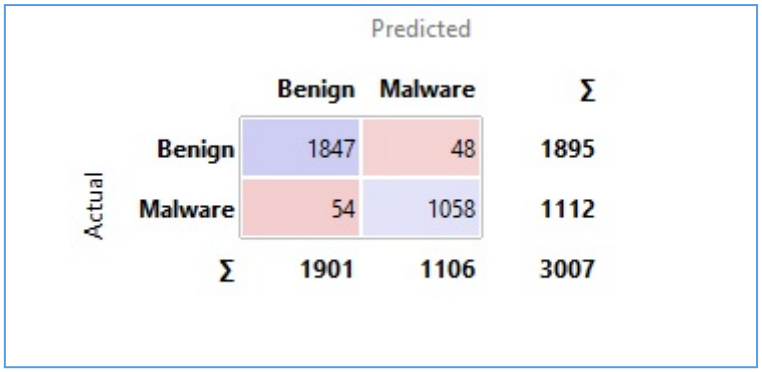


الشكل (3.3) مصفوفة الارتباك للمصنف KNN على مجموعة بيانات Derbin



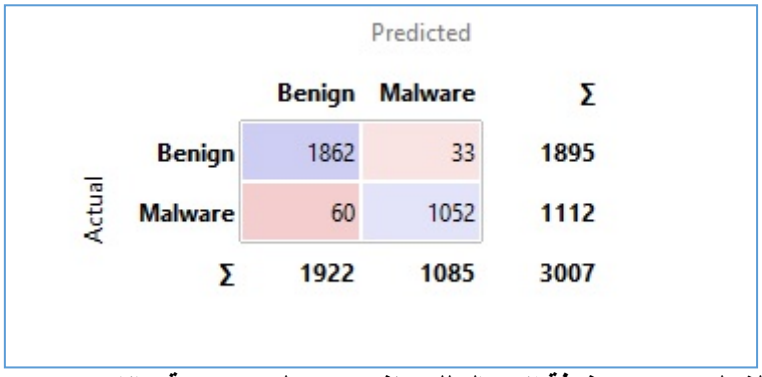
الشكل (4.3) مصفوفة الارتباك للمصنف NB على مجموعة بيانات Derbin

يبين الشكل (5.3) مصفوفة الارتباك لخوارزمية شجرة القرار DT. تمثل القيمة 48 فى مصفوفة الارتباك عدد حالات (False Negative) التي تم إسنادها بشكل خاطئ إلى الفئة السلبية (Malware) من إجمالي العدد الفعلي للتطبيقات الحميدة وهو 1895 تطبيق. نلاحظ كذلك أن خوارزمية DT قامت بإسناد 54 تطبيق خبيث Malware إلى الفئة الايجابية (الحميدة) False Positive. يعكس انخفاض عدد حالات False Negative وFalse Positive ارتفاع معدل مؤشر Precision و مؤشر الاستجابة Recall بنسبة (0.964).



الشكل (5.3) مصفوفة الارتباك للمصنف DT على مجموعة بيانات Derbin

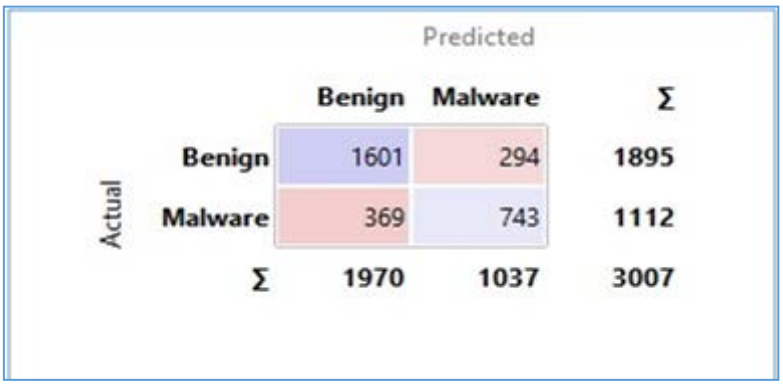
يبين الشكل (6.3) مصفوفة الارتباك لخوارزمية الانحدار اللوجستى LR. تمثل القيمة 33 فى مصفوفة الارتباك عدد حالات (False Negative) التي تم إسنادها بشكل خاطئ إلى الفئة السلبية (Malware) من إجمالي العدد الفعلي للتطبيقات الحميدة وهو 1895 تطبيق. نلاحظ كذلك أن خوارزمية LR قامت بإسناد 60 تطبيق خبيث Malware إلى الفئة الايجابية (الحميدة) False Positive. يعكس انخفاض عدد حالات False Negative و False Positive ارتفاع معدل مؤشر Precision بنسبة (0.98) و مؤشر الاستجابة Recall بنسبة (0.98) وهو اقل معدلات مقارنة بجميع الخوارزميات التي تم اختبارها.



الشكل (6.3) مصفوفة الارتباك للمصنف LR على مجموعة بيانات Derbin

يبين الشكل (7.3) مصفوفة الارتباك لخوارزمية الالة متجهة الدعم SVM. تمثل القيمة 298 فى مصفوفة الارتباك للمصنف SVM عدد حالات (False Negative) التي تم إسنادها بشكل خاطئ إلى الفئة السلبية (Malware) من إجمالي العدد الفعلي للتطبيقات الحميدة وهو 1895 تطبيق. نلاحظ كذلك أن خوارزمية SVM قامت بإسناد 369 تطبيق خبيث Malware إلى الفئة الايجابية (الحميدة) أي False Positive. يعكس ارتفاع عدد حالات False Negative وFalse Positive انخفاض معدل مؤشر Precision (0.77) ومؤشر الاستجابة Recall بنسبة(0.78) لخوارزمية SVM وهو أقل معدل مقارنة بجميع الخوارزميات التي تم اختبارها.

شكل عام، تبين النتائج فعالية خوارزميات تعلم الآلة في مجال تصنيف تطبيقات اندرويد الخبيثة. كذلك تبين النتائج فعالية اختيار الخصائص أو الميزات Feature Selection الممثلة لتطبيقات اندرويد تحسين أداء خوارزميات تعلم الآلة من خلال كشف واستبعاد الميزات التي تؤثر سلبا على دقة التصنيف.



الشكل (7.3) مصفوفة الارتباك للمصنف SVM على مجموعة بيانات Derbin

الفصل الرابع
الخلاصة

4.الخلاصة Conclusion

في هذا البحث، تم اختبار أداء خمس مصنفات للتعلم الآلي تضم كل من خوارزمية الجار الأقرب K Nearest Neighbour وخوارزمية شجرة القرار Decision Trees و خوارزمية نايف بييز Naïve Bays و خوارزمية الانحدار اللوجيستي Logistic Regression و خوارزمية الآلة المتجه الدعم Support Vector Machine لتصنيف تطبيقات اندرويد الخبيثة. كما تم اختيار تأثير اختيار الميزات Feature Selection على أداء المصنفات. أظهرت النتائج قدرة خوارزميات تعلم الآلة على تصنيف تطبيقات اندرو يد الخبيثة. كذلك اظهرت النتائج فعالية اختيار الميزات في تحسين دقة تصنيف خوارزميات تعلم الآلة من خلال كشف واستبعاد الميزات التي تؤثر سلبا على قدرة المصنفات فى الفصل بين تطبيقات اندرويد الحميدة والخبيثة.

1.4 الصعوبات والعراقيل

لا شك أن أي عمل يواجه عدد من الصعوبات والعراقيل ولعل أبرز الصعوبات التي واجهت انجاز البحث تنحصر في الآتي:

1. قلة الدراسات والمراجع العلمية باللغة العربية التي لها صلة بموضوع البحث.

2.4 التوصيات والأفاق المستقبلية

1. استخدام نماذج أخرى للتعلم الآلي في التصنيف تطبيقات اندرويد الخبيثة.

2. اختبار أداء المصنفات المستخدمة في التصنيف المتعدد لتطبيقات اندرويد الخبيثة لتحديد نوع البرنامج الخبيث.

المراجع

المراجع العربية:

[1] ميلاد وزان, د. علاء طعيمة، " كتاب تعلم الآلة وعلم البيانات : الأساسيات والمفاهيم والخوارزميات والأدوات "، كلية علوم الحاسوب وتكنولوجيا المعلومات، جامعة القادسية، العراق، 2022م

[2] مروة صالح إبراهيم دومه، استخدام تقنيات التنقيب في البيانات للتنبؤ بمراحل المبكرة لمرض الفشل الكلوي المزمن، كلية العلوم، قسم الحاسوب، جامعة اجدابيا، 2021.

المراجع الأجنبية:

[3] Statista. Available online:https://www.statista.com/statistics/1236760/worldwide-smartphone-operating-system-shipment-market-share/#statisticContainer (Accessed 15-08-2022)

[4] Liu, K., Xu, S., Xu, G., Zhang, M., Sun, D., & Liu, H. (2020). A review of android malware detection approaches based on machine learning. IEEE Access, 8, 124579-124607.

[5] Kaspersky. Available online: https://usa.kaspersky.com/about/press-releases/2022_2021-mobile-threats-report-cybercriminalspursue-banking-and-gaming-accounts (Accessed 16-8-2023)

[6] Mantoo, B.A.; Khurana, S.S. Static, dynamic and intrinsic features based Android malware detection using machine learning. In Proceedings of ICRIC; Springer: Cham, Switzerland, 2020.

[7] P. Agrawal and B. Trivedi, "Evaluating Machine Learning Classifiers to detect Android Malware," 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangluru, India, 2020, pp. 1-6.

[8] Dhalaria, Meghna & Gandotra, Ekta. (2020). A Hybrid Approach for Android Malware Detection and Family Classification. International Journal of Interactive Multimedia and Artificial Intelligence. In Press. 1. 10.9781/ijimai.2020.09.001.

[9] Feizollah, A., Anuar, N.B., Salleh, R., Amalina, F., Ma'arof, R.U.R., Shamshirband, S.: A study of machine learning classifiers for anomaly-based mobile botnet detection. Malays.J. Comput. Sci. 26(4), 251–265 (2014).

[10] Yerima, S.Y., Sezer, S., McWilliams, G., Muttik, I.: A new android malware detection approach using Bayesian classification. In: IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), pp. 121–128 (2013)

[11] Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K., & Siemens, C. E. R. .Drebin: Effective and explainable detection of android malware in your pocket. In *Ndss* (Vol. 14, pp. 23-26) (2014).

[12] Meijin, L., Zhiyang, F., Junfeng, W., Luyu, C., Qi, Z., Tao, Y.,&Jiaxuan, G. (2022). A systematic overview of android malware detection. Applied Artificial Intelligence, 36(1), 2007327.

[13] Amro, B. Malware detection techniques for mobile devices. arXivpreprint arXiv:1801.02837 (2018).

[14] https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592. Accessed 15-07-2023.

[15] Vasan K, Keerthi and B, Surendiran. (2017). Feature Subset Selection for Intrusion Detection using various Rank based Algorithms. International Journal of Computer Applications in Technology. 55. 298. 10.1504/IJCAT.2017.086017.

[16] Mantoo B. A. (2020). A hybrid approach with intrinsic feature-based android malware detection using LDA and machine learning. In The International Conference on Recent Innovations in Computing Jammu, India, 295–306. Springer.

[17] Firdaus, Anuar, A., Karim N. B., and Razak. 2018. Discovering optimal features using static analysis and a genetic search based method for android malware detection. Frontiers of Information Technology & Electronic Engineering 19 (6):712–36. doi:10.1631/ FITEE.1601491.

[18] Akhtar MS, Feng T. Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry. 2022; 14(11):2304. https://doi.org/10.3390/sym14112304

[19] https://usa.kaspersky.com/resource-center/preemptive-safety/avoid-android-malware (Last accessed 10/11/2023)

[20] https://www.kaspersky.com/resource-center/threats/mobile (Last accessed 10/11/2023)