مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (7) العدد(2) 2023
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (7) No. (2) 2023

تحسين آلية الكشف المسبق عن تطبيقات الأندرويد الخبيثة المعتمدة على السمات الستاتيكية

د.م لبنى علي *
م. علي ربيع **

(تاريخ الإيداع 2022/11/15 . قُبِل للنشر في 2023/2/13)

□ ملخّص□

تعد الهواتف الذكية من أكثر الالكترونيات انتشاراً في أيامنا هذه ، ولم يعد يقتصر استخدامها على إجراء المكالمات فقط وإنما أصبحت تستخدم في حفظ المعلومات الشخصية ، والصحية ، كذلك المعلومات البنكية للمستخدم (التي تستخدم في عمليات الدفع الالكتروني) ، ومن أكثر أنظمة تشغيل الهواتف الذكية انتشاراً نظام التشغيل Android والذي يسيطر على أكثر من ثلثي السوق العالمية، ومع انتشار البرمجيات الخبيثة – التي تختلف أنشطتها وأهدافها – وظهور أشكال جديدة منها باستمرار ظهرت الحاجة إلى إيجاد طرق لكشف هذه البرمجيات والحد من تأثيرها . وتتنوع الطرق المستخدمة في عملية الكشف هذه فمنها يقوم بتحليل التطبيقات دون تشغيلها أي من خلال تحليل الأسطر البرمجية وتسلسلها مثلاً ، ومنها ما يعتمد على تحليل البيانات المجمعة من مراقبة سلوك هذه التطبيقات في بيئة معزولة . وفي هذه الورقة نقدم نموذجاً يستخدم مجموعة بيانات صادرة عن المعهد الكندي للأمن السيبراني وهي **CIC-InvesAndMal2019** كمجموعة بيانات للتدريب والاختبار محققاً صحة accuracy وصلت إلى (96.53%) أما الضبط precision فقد وصل إلى (96.35%) والاستدعاء recall فقد كان (94.37%) مع تخفيض عدد السمات المستخدمة من (8111) إلى (210) سمة .

الكلمات المفتاحية: أندرويد، كشف التطبيقات الخبيثة، الصلاحيات، اختيار السمات، الأشجار الإضافية، تعلم الآلة.

* أستاذ مساعد في قسم هندسة تكنولوجيا المعلومات من كلية هندسة تكنولوجيا المعلومات والاتصالات بجامعة طرطوس
** طالب الدراسات العليا (ماجستير) في قسم هندسة تكنولوجيا المعلومات من كلية هندسة تكنولوجيا المعلومات والاتصالات بجامعة طرطوس .

مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (7) العدد(2) 2023
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (7) No. (2) 2023

Enhancing the android malware pre-detection mechanism based on static features

Loubna Ali *
Ali Rabia **

(Received 15/11/2022 . Accepted 13/2/2023)

□ ABSTRACT

Smartphones are one of the most prevalent electronics nowadays, and their use is no longer limited to making calls, but to save personal and health information too, as well as the user's banking data (which been used in electronic payment operations) . The Android operating system, which controls more than two-thirds of the global market , and with the spread of malware - whose activities and objectives vary - and the emergence of new forms of it constantly , the need to find ways to detect this malwares and limit its effect has emerged. The methods used in this detection process vary, some of them analyze applications without running them , by analyzing applications and sequenced code lines , for example , and some of them depend on the analysis of collected data from observing the behavior of these applications in an isolated environment. In this paper, we present a model that uses a data set made by the Canadian Institute of Cyber Security, named **CIC-InvesAndMal2019**, as a data set for training and testing, achieving (96.53%) for accuracy, (96.35%) for precision and for recall (94.37%) with reducing the number of used features from (8111) to (210) features.

Key words: Android, Malware detection, Permissions, Feature selection, Extra trees, Machine learning.

1- مقدمة:

شهدت السنوات الأخيرة تطوراً تكنولوجياً سريعاً و متسارعاً برزت إحدى أهم جوانبه في تطور و انتشار الهواتف الذكية التي تجاوز استخدامها مهام الاتصال و أصبحت تستخدم في شتى جوانب الحياة من تعلم ، مراقبة الصحة ، التجارة الالكترونية ، حفظ البيانات الشخصية ...

ومع هذا التطور الكبير تنوعت أنظمة التشغيل ولكن برز من بينها نظام تشغيل Android كأحد أهم الأنظمة وأكثرها انتشاراً وذلك في أكثر من ثلثي الهواتف الذكية في العالم [1] مشكلاً بيئة واسعة لظهور تطبيقات مختلفة بأغراض متنوعة و مع تزايد هذه التطبيقات وتنوع أهدافها ظهرت الحاجة لمنع أي استخدام غير مشروع لهذه الأجهزة وحماية المعلومات المخزنة عليها من السرقة او الإتلاف .

ومن هنا ازدادت أهمية وجود طرق فعالة لحماية هذه المعلومات من أي سلوك غير سليم قد يصدر عن تلك التطبيقات لذلك أصبح هذا المجال حقلاً للبحث تنوعت فيه مسارات العمل والطرق المستخدمة لكشف التهديد او الاحتيال ولكن بمعظمها كانت وفق منهجية محددة بأربع مراحل أساسية تبدأ بتجميع البيانات واستخراج السمات ، معالجة هذه البيانات وتنظيفها وتنتهي باستخدام خوارزميات للتصنيف وتقييم النتائج .

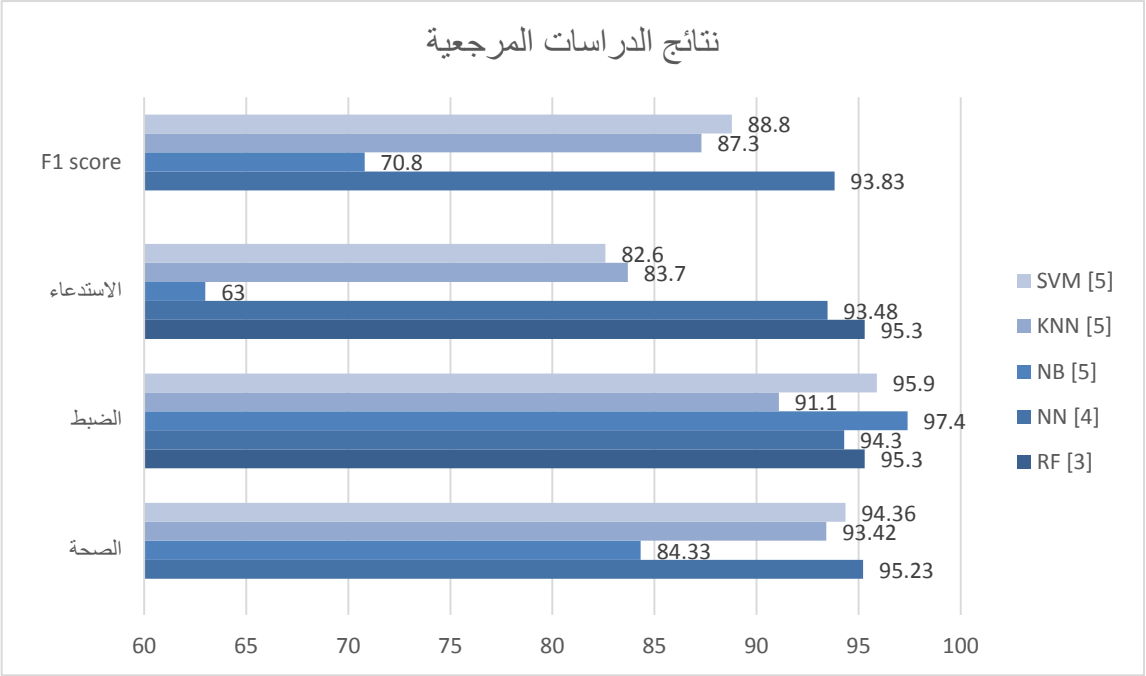
تعددت أنواع السمات وطرق استخراجها فمنها ما اعتمد على ما يسمى بالطرق الساكنة (الستاتيكية) للتحليل Static-based analysis ومنها ما اعتمد الطرق الديناميكية Dynamic-based analysis ففي الأولى يتم تحليل التطبيق واستخراج السمات دون تشغيله أما الثانية فيتم فيها تشغيل التطبيق ومراقبة نشاطه ، كذلك الأمر بالنسبة لخوارزميات التصنيف التي بدورها تنوعت وتعددت طرق تقييمها فمنها ما اعتمد الشبكات العصبونية بأنواعها [4] ومنها ما اعتمد المصنفات ذات البنية الشجرية [3] وغيرها ...

شكلت أهمية هذه البيانات وحمايتها حافزاً للعديد من الباحثين والرواد في هذا المجال لدراسة طرق ونماذج مختلفة للكشف عن الفيروسات في تطبيقات الأندرويد وتحديد نوعها .

الدراسات المرجعية :

بهدف الكشف المسبق عن التهديدات (الفيروسات) في تطبيقات الأندرويد قام الباحثون في [3] بتحليل ملفات التطبيقات واستخراج (8115) سمة من ملف يسمى Manifest.xml ، هذه السمات تمثل عدد مرات ظهور كل واحدة من الصلاحيات permissions وال intents في هذا الملف لكل تطبيق ، ثم إدخال هذه البيانات إلى مصنف الغابة العشوائية (RF : Random forest classifier) وقد تم تقسيم البيانات إلى (60%) بيانات تدريب و (40%) بيانات اختبار وقد حققوا قيم للضبط والاستدعاء وصلت إلى (95.3%) لكل منهما ، أما في [4] فقد قام الباحثون – وباستخدام البيانات المستخرجة في الدراسة السابقة – بتطبيق خوارزمية الأشجار شديدة العشوائية ERT : Extremely Randomized Trees لاختيار السمات ثم إدخال البيانات إلى شبكة عصبونية (NN : Neural Network) لكشف الفيروسات وقد حققت هذه الطريقة قيماً للصحة والضبط والاستدعاء وF1 score هي (95.23%) و (94.3%) و (93.48%) و (93.83%) على التوالي في حين قام الباحثون في [5] بتقسيم البيانات إلى مجموعتين : الأولى تحوي الصلاحيات ، وما تبقى في الثانية ثم استخدمت خوارزمية (RFE : Recursive Feature Elimination) على نموذج الانحدار اللوجستي (Logistic Regression) لاختيار السمات واستخدمت عدة

خوارزميات للتصنيف كان أفضلها (SVM : Support Vector Machine) بصحة وصلت إلى (94.36%) حيث يظهر في الشكل (1) نتائج الدراسات السابقة .



شكل (1): نتائج الدراسات المرجعية

2- أهمية البحث وأهدافه:

يقدم هذا البحث نموذجاً فعالاً للكشف المسبق عن وجود تهديدات في تطبيقات الأندرويد قبل تثبيتها على الهواتف الذكية بهدف حماية هذه الأجهزة من الاختراق وتلافي تعرض بياناتها للسرقة أو التخريب .

تكمن أهمية هذا البحث في استخدامه لأقل قدر من بيانات التطبيقات في كشف أمنها وسلامتها حيث يهدف إلى تحقيق ما يلي :

- ✓ تخفيض عدد السمات المستخدمة في التصنيف .
- ✓ المحافظة على صحة تصنيف جيدة .

ويهدف تخفيض عدد السمات إلى إزالة السمات التي قد تؤثر سلباً على تدريب وتقييم المصنف من جهة وتزيد الزمن اللازم للتدريب والوصول إلى الجودة المطلوبة من جهة أخرى .

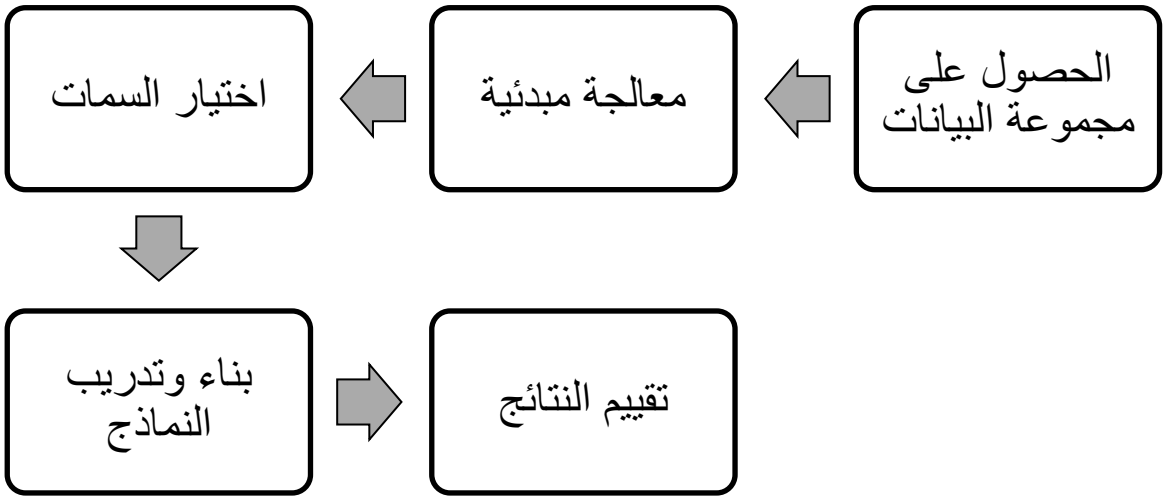
3- طرق البحث ومواده:

تم تنفيذ هذا البحث باستخدام لغة البرمجة python والتي تعد من أسهل اللغات استعمالاً ومن أشهر اللغات الرائدة في مجال الذكاء الاصطناعي وتعلم الآلة وتحتوي على العديد من المكتبات البرمجية التي تسهل العمل في هذا المجال .

تم اختيار Google Colab كبيئة عمل – والتي تعد من أهم خدمات السحابة التي تنقل العمليات الحسابية والمساحة التخزينية إلى السحابة [8] التي تقدمها شركة Google – والتي تتيح العمل ضمن Jupyter notebook دون أي عملية تثبيت للمكتبات البرمجية ومتطلبات اللغة على الجهاز المحلي.

تمت الاستعانة ببعض مكتبات اللغة مثل numpy في التعامل مع المصفوفات و Matplotlib لرسم المخططات البيانية و pandas للتعامل مع الجداول في مجموعة البيانات و sklearn لما تحويه من خوارزميات تساعد في مجال تعلم الآلة .

تم العمل على عدة مراحل تبدأ من الحصول على البيانات ثم إلى معالجتها وتحضيرها و تخفيض عدد السمات ثم استخدام عدة خوارزميات تصنيف في التدريب والاختبار ومقارنة النتائج وتقييمها وذلك كما في الشكل (2).



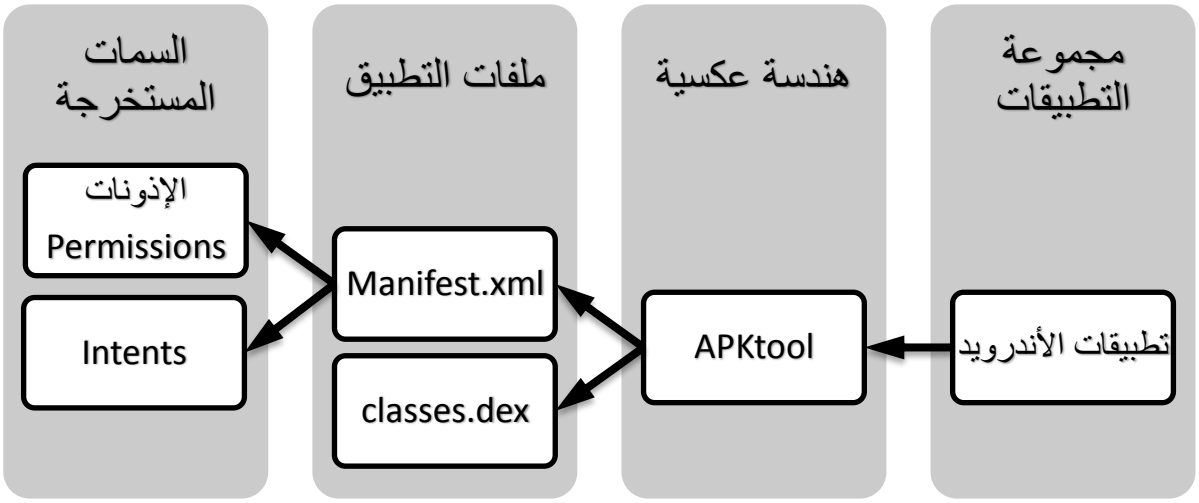
شكل (2): مراحل العمل

4- مراحل البحث:

4-1- مرحلة الحصول على البيانات :

تعتبر تطبيقات الأندرويد كملفات مضغوطة وباستخدام بعض تقنيات وأدوات الهندسة العكسية مثل Apktool التي تمكن من الوصول إلى ملفات التطبيق ، يمكن استخراج هذه الملفات وتحليلها.

يوضح الشكل التالي طريقة لاستخراج أذونات (صلاحيات) التطبيق :



شكل (1) استخراج أذونات التطبيق

في هذا البحث كما في الدراسات المرجعية تم استخدام مجموعة بيانات صادرة عن المعهد الكندي للأمن السيبراني وهي (**CIC-InvesAndMal2019**) والتي تم فيها استخراج نوعين من السمات من ملفات Manifest.xml وهي الصلاحيات (الأذونات) Permissions بالإضافة إلى Intents التي تصف وتحدد البيانات التي يتم تبادلها بين مكونات التطبيق أو بين التطبيق والتطبيقات الأخرى ، حيث تم استخراج أكثر من (8100) سمة من أكثر من (1500) تطبيق [3][7] .

4-2- المعالجة المبدئية:

- تتم في هذه المرحلة بداية التحقق من سلامة البيانات المستخرجة وخلوها من الأخطاء أو الضياع أو القيم المتطرفة وقد تبين في حالتنا هذه عند التحقق من مجموعة البيانات السابقة سلامة البيانات .
- بعدها يتم التخلص من السمات غير المرتبطة بالتصنيف والتي في حالتنا هذه تتمثل بثلاث سمات (لا تتعلق بالتصنيف الثنائي سليم – مصاب) وهي :
  - ✓ MD5 : كمعرف خاص بالتطبيق .
  - ✓ Category : نوع (صنف) الفيروس .
  - ✓ Family : العائلة التي ينتمي إليها هذا النوع .
- ثم يتم تحويل القيم المحرفية إلى قيم عددية وهنا لدينا عمود الهدف Binary_Type الذي يحتوي على القيمتين (Malware – Benign) أي (سليم – مصاب) ويتم هذا التحويل باستبدال كل كلمة برقم (1, 0) مثلاً .
- يتم ضبط وتوحيد مجالات القيم للسمات والتي في حالتنا هذه تعد مجالات موحدة وتعبر عن تكرار ظهور السمة في التطبيق وهي في معظمها أعداد صحيحة تنتمي للمجال [0, 8] ، و يبين الشكل التالي عينة من الأسطر الخمسة الأولى من مجموعة البيانات (بيانات التدريب) :

```
train_data_x_y.head()
```

	.ACCESS_NETWORK_STATE\n	.GET_ACCOUNTS\n	.INTERNET\n	.READ_CONTACTS\n	.WAKE_LOCK\n	.WRITE_EXTERNAL_STORAGE\n	.driver.permission.MAPS_RECEIVE\n	.gcm.permission
0	6	0	6	0	4	6	0	
1	1	0	1	0	1	1	0	
2	2	0	2	0	1	2	0	
3	2	0	1	0	1	2	0	
4	1	0	1	0	1	1	0	

5 rows × 8112 columns

شكل (3) : عينة من مجموعة بيانات التدريب

حيث يمثل كل عمود في الشكل (3) سمة من السمات مثل (الوصول إلى حالة الشبكة ACCESS_NETWORK_STATE) أما الأسطر فتمثل التطبيقات وكل قيمة من القيم السابقة تمثل عدد مرات ظهور السمة الموافقة في التطبيق الموافق .

4-3- اختيار السمات :

البيانات الآن جاهزة لتخفيض عدد السمات وذلك من خلال استخدام Chi-square test لحساب ارتباط السمات بعمود الهدف Binary_Type واختيار أفضل K سمة باستخدام خوارزمية SelectKBest .

يستخدم Chi-square test في حساب احتمال ارتباط حدثين أو استقلالهما حيث أن قيمة score المرتفعة تعني أن عمود الهدف يعتمد على العمود المدروس (أو يوجد ارتباط بين الهدف والسمة المدروسة) وانخفاض القيمة يعني استقلال العمودين عن بعضهما .

يتم حساب هذه القيمة باستخدام العلاقة :

$$x^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \qquad (1)$$

حيث :

O_i : القيمة الحقيقية .

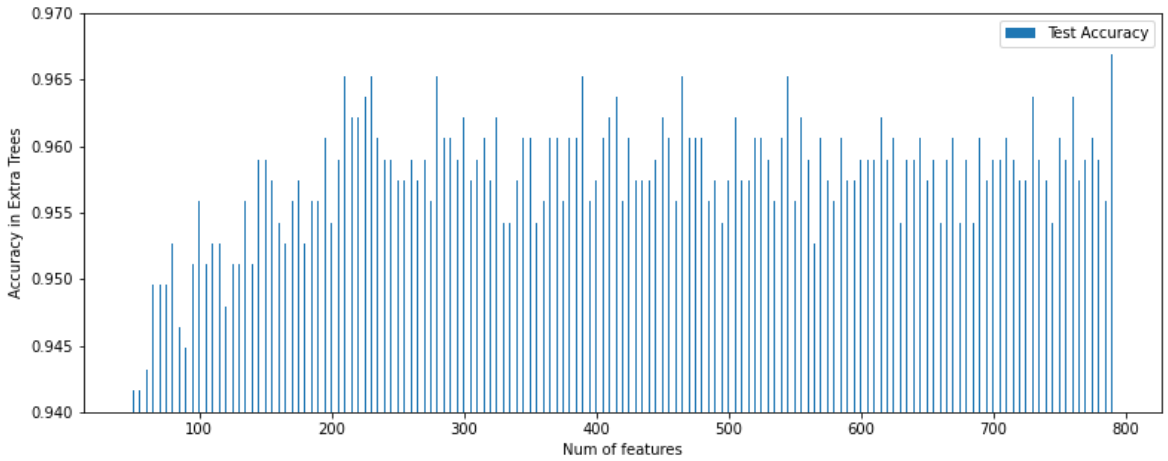
E_i : القيمة المتوقعة .

x^2 : قيمة ال score .

حيث يتم حساب القيم الحقيقية والمتوقعة وفق ما هو مبين في [15] .

باستخدام SelectKBest يتم ترتيب السمات تنازلياً وفق قيم score المحسوبة سابقاً واختيار أكبر K قيمة (تعبر عن أكثر السمات ارتباطاً بالهدف) ، ثم إعادة ضبط البيانات المستخدمة لتطابق السمات المختارة .

تم تحديد قيمة K تجريبياً بالاعتماد على نتائج تقييم النموذج المدرب لكل قيمة K وقد كانت أفضل النتائج مترافقة مع القيمة (K = 210) أي تم اختيار 210 من السمات ال 8111 وذلك كما هو مبين في الشكل (4).



شكل (4) : مخطط يبين قيم الصحة بتغير عدد السمات في مصنف Extra Trees

حيث يعبر المحور الشاقولي عن قيم الصحة في حين أن المحور الأفقي يمثل عدد السمات وقد تم اختيار أكبر قيمة للصحة (96.53%) المترافقة مع أقل عدد للسمات (210) .

و بعد الانتهاء من المعالجة المبدئية واختيار السمات أصبحت البيانات جاهزة للمرحلة التالية والتي تكون إعداد نموذج التصنيف وتدريبه .

4-4- بناء وتدريب النموذج :

البيانات مقسمة إلى (60%) بيانات تدريب تستخدم في إعداد وبناء النموذج و (40%) بيانات اختبار وتقييم للنموذج .

تم في هذا البحث استخدام عدة نماذج للتصنيف :

✓ Support Vector Machine (SVM) :

والذي يعد من نماذج التصنيف الخاضعة للإشراف ، ويستخدم غالباً مع بيانات مقسمة إلى صنفين يمكن الفصل بينها بشكل خطي [9] .

✓ K-Nearest Neighbor (KNN) :

كذلك يعد من نماذج التصنيف الخاضعة للإشراف ،و يقوم بتقسيم البيانات إلى عناقيد بالاعتماد على أقرب K جار وتحسب هذه المسافة باستخدام علاقة رياضية كالمسافة الإقليدية مثلاً [10] .

✓ Random Forest (RF) :

هذا النموذج هو عبارة عن مجموعة من أشجار القرار ، ويعد من طرق التعلم التجميعي للتصنيف ، حيث يتم تقسيم البيانات وتوزيعها على الأشجار ويكون القرار النهائي (الصنف) للغابة هو القرار المعتمد من غالبية الأشجار [11] .

✓ Extra Trees (ET) :

ويسمى أيضاً بالأشجار شديدة العشوائية (Extremely Randomized Trees) يشبه إلى حد كبير مصنف الغابة العشوائية ، إلا أن الاختلاف الأساسي بينهما يكمن في آلية تقسيم البيانات للتدريب والتحقق والتي تتم هنا بطريقة عشوائية وهذا ما يقلل من العمليات الحسابية اللازمة للتدريب [12][13] .

✓ XGBoost :

أو (extreme gradient boosting) والذي يعد تطويراً لتقنية (gradient descent boosting) ، وهو أيضا من نماذج التعلم التجميعي والذي يهدف إلى أمثلة أداء النموذج وزمن التنفيذ في عمليتي التدريب والتحقق [14] .

و باستخدام بيانات التدريب تم بناء النماذج وتم تقييمها باستخدام بيانات الاختبار .

4-5- معايير تقييم النماذج :

يقوم كل نموذج من النماذج المذكورة سابقاً بتصنيف بيانات كل تطبيق إلى صنف من أحد الصنفين : سليم (Benign or 0) أو خبيث (Malware or 1) ثم يتم تقييم كل منها بالاعتماد على مصفوفة الارتباك (Confusion Matrix) [2] من خلال بعض المقاييس .

جدول (2) : مصفوفة الارتباك

	Predicted (0)	Predicted (1)
Actual (0)	TP	FN
Actual (1)	FP	TN

حيث :

- TP : True Positive
- FP : False Positive
- FN : False Negative
- TN : True Negative

في هذا البحث كانت المقاييس المستخدمة كالتالي :

✓ الصحة Accuracy : وتمثل عدد الحالات المصنفة بشكل صحيح بالنسبة لجميع الحالات وذلك وفق:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Number\ of\ all\ predictions} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

✓ الضبط Precision : ويمثل عدد الحالات الإيجابية المصنفة بشكل صحيح بالنسبة لجميع الحالات الإيجابية وفق :

$$Precision = \frac{Number\ of\ true\ positives}{Number\ of\ all\ positive\ predictions} = \frac{TP}{TP+FP} \quad (3)$$

✓ الاستدعاء Recall : عدد الحالات الإيجابية المصنفة بشكل صحيح بالنسبة لجميع الحالات الإيجابية وفق :

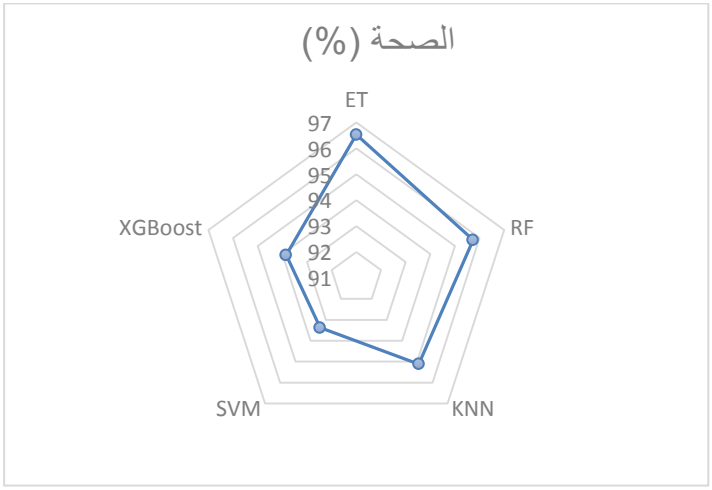
$$Recall = \frac{Number\ of\ true\ positives}{Number\ of\ all\ positives} = \frac{TP}{TP+FN} \quad (4)$$

✓ F1 score : يجمع الضبط والاستدعاء في قياس واحد وفق العلاقة :

$$F1\ score = \frac{2*Precision*Recall}{Precision+Recall} \quad (5)$$

4-6- النتائج والمناقشة :

بعد تدريب النماذج واختبارها كانت قيم الصحة وفق الشكل (5) والتي نلاحظ فيها تفوق مصنف ET على باقي المصنفات المستخدمة .



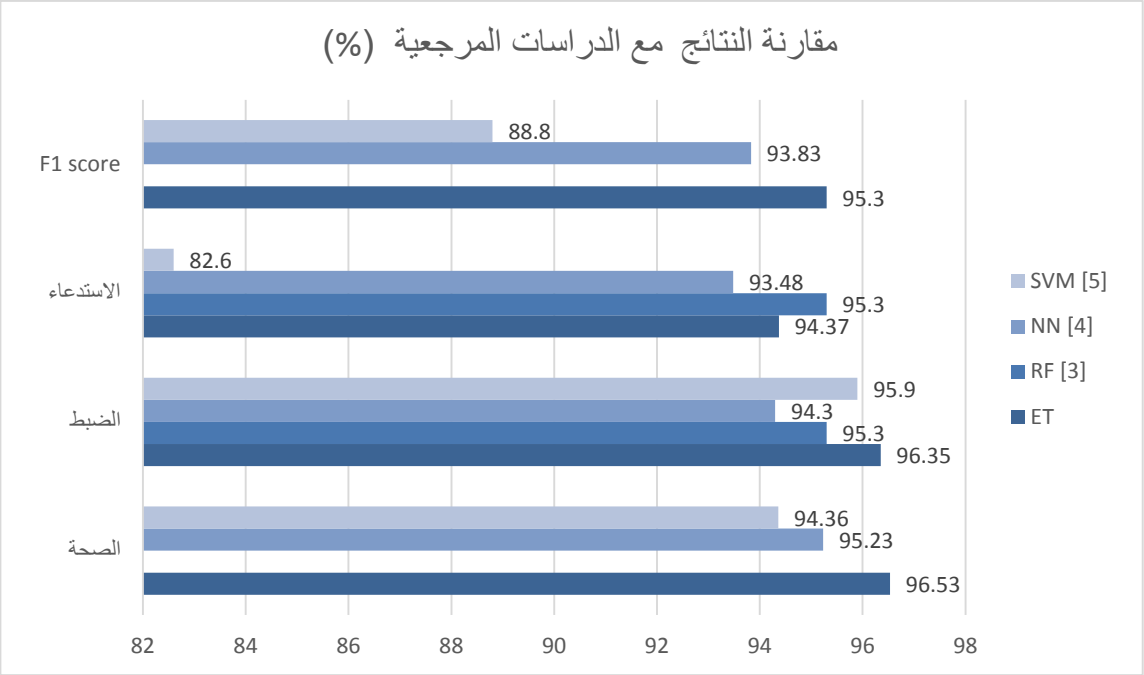
شكل (5) : قيم الصحة للمصنفات المستخدمة

في حين أن النتائج الكلية وفق معايير التقييم المذكورة سابقاً كانت كما هو موضح في الجدول (3) :

جدول (3) : نتائج اختبار المصنفات

خوارزميات التصنيف	RF	**ET**	KNN	SVM	XGBoost
الصحة (%)	95.74	**96.53**	95.11	93.38	93.85
الضبط (%)	95.57	**96.35**	95.11	91.66	92.55
الاستدعاء (%)	93.01	**94.37**	91.76	90.6	90.92
F1-score (%)	94.2	**95.3**	93.28	91.11	91.69

ونرى من الجدول السابق أن أفضل النتائج حققها مصنف الأشجار الإضافية Extra Trees classifier والذي يسمى أيضاً بمصنف الأشجار شديدة العشوائية Extremely Randomized Trees classifier والذي تفوق على المصنفات الخمسة المستخدمة في معايير التقييم الأربعة (الصحة، الضبط، الاستدعاء، F1-score) وبالمقارنة مع نتائج الدراسات المرجعية السابقة نجد :



شكل (6) : مقارنة نتائج تقييم النموذج المقترح مع نماذج الدراسات المرجعية

نلاحظ أن النموذج المقترح قد حقق أفضل النتائج بين الدراسات المذكورة وذلك بالنسبة للصحة والضبط و F1_score .

ويبين الجدول التالي عدد السمات المستخدمة ونسبة بيانات التدريب إلى البيانات الكلية (بيانات التدريب والاختبار) :

جدول (4) : مقارنة النتائج مع الدراسات السابقة

	نماذج سابقة			النموذج المقترح
خوارزمية التصنيف	RF [3]	NN [4]	SVM [5]	**ET**
خوارزمية اختيار السمات	–	ERT	RFE	Chi-square
عدد السمات المستخدم	8111	-	**100**	210
نسبة بيانات التدريب (%)	**60**	**60**	80	**60**

عدد السمات المستخدمة يمثل أقل من (2.5%) من العدد الأصلي وهو تحسين جيد ، ونلاحظ في الدراسة المعتمدة على مصنف SVM أن عدد السمات أقل مما لدينا ولكن في تلك الدراسة تم تقسيم البيانات إلى مجموعتين لكل منها مصنف (أي تم استخدام مصنفين : مصنف للصلاحيات ومصنف لما تبقى من السمات) كما أن بيانات الاختبار كانت قليلة نسبياً (20% من إجمالي البيانات) لذلك لا يمكن القول أن نتائجها أكثر تعميماً (أو أكثر قدرة

على التعامل مع بيانات جديدة لم يتم تدريب النموذج عليها) مع العلم أن مجموعة البيانات المستخدمة للتدريب والاختبار هي نفسها في جميع الدراسات (CIC-InvesAndMal2019) .

5- الاستنتاجات والتوصيات :

تمت دراسة وتقييم ارتباط السمات بالعمود الهدف وفق Chi-square test ثم اختيار أفضل) (K=210 سمة باستخدام SelectKBest حيث استخدمت البيانات المختارة في بناء وتدريب عدة نماذج و تقييم هذه النتائج ومقارنتها مع دراسات سابقة.

لقد أظهرت النتائج أن اختيار السمات الأكثر ارتباطاً بالهدف باستخدام Chi-square test كان له تأثير جيد في عملية التدريب والتصنيف حيث أن هذه الطريقة خفضت حجم البيانات المستخدمة بشكل كبير مع المحافظة على البيانات المهمة واللازمة للتصنيف كما أن الاختيار المناسب للسمات يزيد من سرعة عماية التدريب والوصول إلى النموذج الأمثل كذلك وجدنا أن خوارزميات التعلم الآلي الخاضع للإشراف وخصوصاً الخوارزميات التجميعية كالغابة العشوائية والأشجار الإضافية (RF, ET) أثبتت فعاليتها في التصنيف في هذا النوع من المسائل وقد تم اختيار هذا النوع من السمات لما له من أهمية في الكشف المسبق عن وجود برمجية خبيثة في التطبيق قبل تثبيته وبالتالي حماية الهاتف الذكي من الاختراق.

وكتوصيات مستقبلية يمكن استخدام أنواع مختلفة من السمات كتحليل وجود تسلسلات برمجية معينة تدل على برمجية خبيثة كذلك استخدام سمات سلوكية للتطبيقات تصف سلوك التطبيق بعد تثبيته كما يمكن تحديد نوع البرمجية الخبيثة والذي يمكن أن يفيد في تحديد الهدف منها والتصدي له كذلك يمكن استخدام مجموعات بيانات مختلفة لتقييم النموذج المقترح وبالتالي قياس أفضل لمدى فعاليته .

6- المصادر والمراجع:

[1] Mobile Operating System Market Share Worldwide. Accessed: Nov. 18, 2022. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide .

[2] Salman, Jaafar; Saad, Ghada; Suliman, Marie. 2021, *Using deep learning algorithms and computer vision in detecting human brain tumor,* Tartous University Journal, Vol. 5, No. 10.

[3] Taheri, Laya, Andi Fitriah Abdul Kadir, and Arash Habibi Lashkari. "Extensible android malware detection and family classification using network-flows and API-calls." *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2019.

[4] Feng, Jiayin, et al. "A two-layer deep learning method for android malware detection using network traffic." *IEEE Access* 8 (2020): 125786-125796.

[5] Shatnawi, Ahmed S., Qussai Yassen, and Abdulrahman Yateem. "An Android Malware Detection Approach Based on Static Feature Analysis Using Machine Learning Algorithms." *Procedia Computer Science* 201 (2022): 653-658.

[6] CIC-InvesAndMal2019 dataset. Accessed: Nov. 18, 2022 .[Online]. Available: http://205.174.165.80/CICDataset/CICInvesAndMal2019/Dataset/ .

[7] Investigation of the Android Malware CIC-InvesAndMal2019 dataset. Accessed:Sep.20,2021.[Online].Available:https://www.unb.ca/cic/datasets/invesandmal2019.html .

[8] Ghosna, Fadi; Ali, Loubna; Mfleh, Marah Naji. 2019, *Analysis of the Performance of a Strategy to Select the Best Data Centers to Handle User Requests in Cloud Computing Network,* Tartous University Journal, Vol. 3, No. 3.

[9] Support vector machine (SVM). Accessed: Nov. 20, 2022. [Online]. Available : https://en.wikipedia.org/wiki/Support_vector_machine .

[10] K-Nearest Neighbors (KNN). Accessed: Nov. 21, 2022.[Online]. Available : https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm .

[11] Random Forest (RF). Accessed: Nov. 21, 2022. [Online]. Available : https://en.wikipedia.org/wiki/Random_forest .

[12] Difference between Random Forest and Extremely Randomized Trees. Accessed:Nov.21,2022.[Online].Available:https://stats.stackexchange.com/questions/175523/difference-between-random-forest-and-extremely-randomized-trees .

[13] Random Forest vs Extra Trees. Accessed: Nov. 21, 2022. [Online]. Available:https://www.kaggle.com/code/hkapoor/random-forest-vs-extra-trees/notebook .

[14] How to perform xgboost algorithm with sklearn. Accessed: Nov. 22, 2022. [Online]. Available : https://www.projectpro.io/recipes/perform-xgboost-algorithm-with-sklearn .

[15] How SelectKBest (chi2) calculates score. Accessed: Jan. 12, 2022. [Online].Available: https://stackoverflow.com/questions/57273694/how-selectkbest-chi2-calculates-score .