استخدام الشبكات العصبونية للكشف عن التطبيقات الخبيثة في نظام أندرويد

طالب الدراسات العليا: روعة طويلة

كلية: الهندسة المعلوماتية – جامعة: البعث

الدكتورة المشرفة: رانيا لطفي

ملخص البحث

يعتبر نظام الاندرويد من أكثر نظم التشغيل المنتشرة على الهواتف المحمولة ، لذلك فإن عدد التطبيقات الضارة تتزايد مع ازدياد عدد التطبيقات المنتشرة على المتاجر. هنالك العديد من الأدوات الموقعة إلكترونياً موجودة على المتاجر والتي تحد من انتشار وتوزيع التطبيقات الضارة ، إلا أن هناك العديد من الدراسات والأبحاث التي وجدت أن نظام الكشف التقليدي المعتمد على التوقيع يعمل بشكل جيد إلى حد ما لأن مطوري البرمجيات الضارة يستخدمون تقنيات عديدة للاحتيال على تلك الأدوات. ومن هنا أتت الحاجة لإيجاد نظام بديل للكشف عن البرمجيات الضارة وذلك لاستكمال وتصحيح النظام المعتمد على التوقيع الالكتروني.

ركزت الأبحاث الحديثة على خوارزميات التعليم الآلي والتعلم العميق التي تحلل الميزات المستخرجة من التطبيقات الضارة ، كما وتستخدم هذه الميزات لتصنيف واكتشاف التطبيقات الجديدة والغير معروفة.

تلخص هذه الدراسة كيفية استخدام الشبكات العصبونية للكشف عن البرمجيات الضارة في نظام أندرويد .وتبين النتائج التجريبية أن استخدام الشبكات العصبونية كان له أثر إيجابي في تحسين الكشف عن البرمجيات الضارة وذلك يتوقف على عدد الطبقات وعدد العصبونات المستخدمة لبناء الشبكة.

كلمات مفتاحية : تحليل البرمجيات الخبيثة ، نظام أندرويد ، حماية الأجهزة الذكية ، الشبكات العصبونية.

Using Artificial Neural Networks to Detect Malicious Applications in Android System

Eng. Rawaa Taweela Dr. Rania Lutfi

Abstract

Android OS is one of the widely used mobile Operating Systems. The number of malicious applications and adwares are increasing constantly on par with the number of mobile devices. A great number of commercial signature based tools are available on the market which

prevent to an extent the penetration and distribution of malicious applications. Numerous researches have been conducted which claims that traditional signature based detection system work well up to certain level and malware authors use numerous techniques to evade these tools. So given this state of affairs, there is an increasing need for an alternative, really tough malware detection system to complement and rectify the signature based system. Recent substantial research focused on machine learning algorithms that analyze features from malicious application and use those features to classify and detect unknown malicious applications. This study summarizes the evolution of malware detection techniques based on machine learning algorithms focused on the Android OS.

Keywords: malware analysis, android, Smartphone security , Neural Networks.

المقدمة :

وفقاً لبحث جرى في عام 2014(RiskIQ 2014) ازدادت التطبيقات الضارة في المتاجر بنسبة 388% بين عامي 2011 و 2013 .

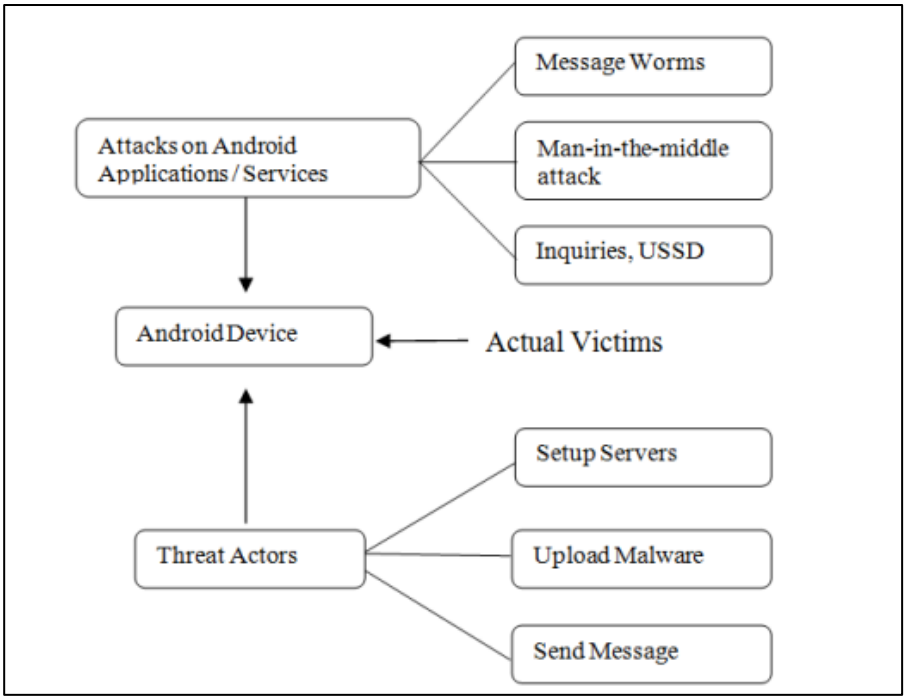
كجزء أساسي من بحثنا ، أجرينا دراسة شاملة حيث قمنا بتحليل المنهجيات الحالية المستخدمة في الكشف عن البرمجيات الضارة على نظام أندرويد باستخدام تقنيات التعلم الآلي.

الهدف العام من هذه الدراسة هو تحديد الأبحاث التي عملت على اكتشاف البرمجيات الضارة على نظام أندرويد باستخدام خوارزميات التعلم الآلي.

فمن خلال هذا التحليل يمكننا صياغة آلية للتصدي للهجمات المعدلة update attack التي يصعب الكشف عنها والقضاء عليها.

Update attack: يتم تعريفه على أنه تطبيق حميد مثبت على النظام يقوم بتحميل ملفات خبيثة أثناء تحديثه أو يحاول تنزيل تطبيقات ضارة وتثبيتها على النظام . من الصعب جداً تحديد هذا النوع من الهجوم لأن التطبيق الأساسي حميد، ولا يمكننا اكتشاف النشاط الضار ما لم نتتبع الإصدارات السابقة للتطبيق ومراقبة التطبيق بعد التحديث [2].

نهدف الى تقديم منهج مضاد لهذا الهجوم من خلال الاطلاع على الاتجاهات الحديثة في الكشف عن البرامج الضارة.



الشكل 1. لمحة عن الهجمات في نظام أندرويد [1]

هدف البحث :

تعتمد معظم أساليب الكشف عن التطبيقات الضارة على طرق تقليدية مثل دراسة التوقيع الالكتروني ، مراقبة استهلاك الطاقة في الجهاز ... الخ ، ولكن كل من الطرق السابقة لها العديد من السيئات ولم تعطِ النتائج المرغوبة.

هدفنا ايجاد حل لمعالجة التطبيق واستخراج الميزات ومحاولة كشف فيما اذا كان هذا التطبيق خبيثاً أو سليماً.

الحل البديل هو التحليل الستاتيكي أي دراسة الميزات واستخدام طرق الاستدلال التي تحاول الكشف عن البرمجيات الخبيثة من خلال مراقبة الميزات والخصائص

الاحصائية للأجهزة النقالة ، وأشهر هذه الطرق هي تحليل السماحيات التي يطلبها التطبيق عند تنصيبه مثل طلب الوصول إلى الشبكة – طلب تحديد موقع المستخدم .... الخ.

هذه الطريقة وحدها لم تكن كافية سليمة ، لذلك تم العمل على الناحية الديناميكية للتطبيقات مثل دراسة الاستدعاءات للتطبيق.

وتعتبر هذه الطريقة أكثر دقة من دراسة السماحيات فقط لأنها تلتقط التنفيذ الحالي (runtime)للتطبيق .

ومن هذا المنطلق كان الاقتراح لتصميم إطار لتحليل وتصنيف التطبيقات الخبيثة بالاعتماد على خوارزميات التنقيب في البيانات بما فيها تقنيات تعليم الآلة Machine Learning و تقنيات التعلم العميق Deep Learning .

وفيما يلي سنوضح احدى الطرق المستخدمة في هذه الدراسة ألا وهي الشبكات العصبونية الصنعية [4] :

الشبكة العصبونية – أو اختصاراً "الشبكات العصبونية " – هي تمثيل صنعي للدماغ البشري تحاول محاكاة عمليته الطبيعية للتعلم.

إن الشبكة العصبونية هي عبارة عن مجموعة من الأعصاب الصنعية المترابطة فيما بينها ، تستخدم نموذجاً رياضياً أو حسابياً لمعالجة المعلومات بالاعتماد على منهج ارتباط للحساب.

تحاكي الحواسيب العصبونية قدرات معالجة معينة في الدماغ البشري.

الحوسبة العصبونية هي نموذج لمعالجة المعلومات مستوحى من النظام البيولوجي ومركب من عدد كبير من عناصر المعالجة المترابطة فيما بينها بقوة (العصبونات) والتي تعمل بانسجام لحل مشكلة معينة.

على غرار البشر، فإن الشبكات العصبونية تتعلم من خلال الأمثلة.

تم تكوين الشبكات العصبونية بهدف تنفيذ تطبيقات معينة مثل تطبيقات التعرف على الأنماط ، تصنيف البيانات ، وكل ذلك يتم من خلال عملية التعلم.

تساعد الشبكات العصبونية في الحالات التي لا يمكن فيها صياغة خوارزمية معينة للحل ، أو عندما يمكننا الحصول على العديد من الأمثلة للسلوك الذي نطلبه.

إن الحواسيب المستخدمة تعتمد على بنية أساسية وهي "von neumann" تعتمد على عمليات المعالجة والتخزين بالذاكرة ، أما الشبكات العصبونية فتعتمد على البنية التفرعية للدماغ البيولوجي.

الشبكات العصبونية هي عبارة عن نظام حاسوبي متعدد المعالجات يتضمن:

عناصر معالجة بسيطة

درجة عالية من الارتباطات الداخلية

رسائل عددية بسيطة

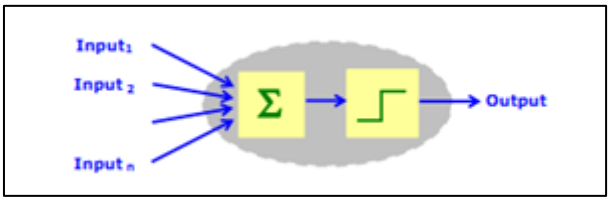
تفاعلات تكيفية بين العناصر

1.1 نموذج العصبون الصنعي:

العصبون الصنعي هو عبارة عن تابع رياضي يمثل نموذجاً مبسطاً من العصبون الحيوي الحقيقي.

- **نموذج McCulloch– pitts:**

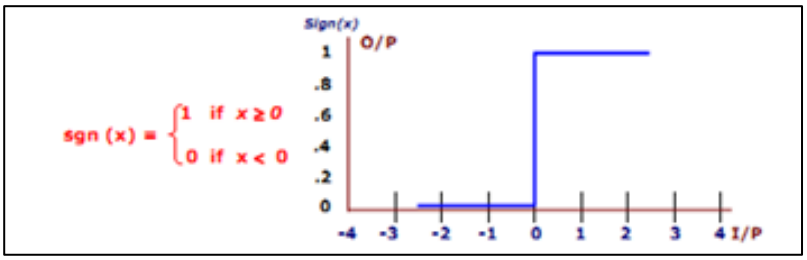
يدعى هذا النموذج وحدة منطق العتبة (threshold logic model)



- مجموعة من اتصالات الدخل مسؤولة عن إيصال أوامر التنشيط من العصبونات الأخرى.
- وحدة معالجة تقوم بجمع الدخل ثم تقوم بتطبيق تابع تنشيط غير خطي (مثل توابع squashing / transfer/threshold)
- خط خرج يوصل النتيجة إلى العصبونات الأخرى.

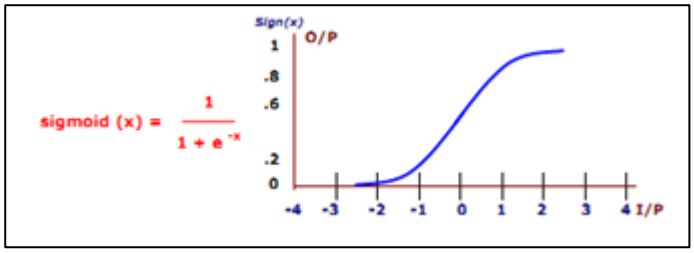
1.2 التوابع:

- Threshold or sign function: sgn (x) يعرف كالتالي



يأخذ التابع القيمة (1) إذا كانت المتحول موجبة او تساوي الصفر ويأخذ القيمة (0) إذا كانت سالبة

- Threshold or sign function: sigmoid(x) وهو يختلف عن السابق بأنه يضم تدرجاً وأنه قابل للاشتقاق



1. نماذج العصبون الصنعي :

a. نموذج McCulloch– Pitts:

والذي تم شرحه سابقاً

1– تكتب معادلة الخرج لعصبون McCulloch–Pitts كتابع لمدخلات عددها n –1 بالشكل التالي:

$$Output = sgn\left(\sum_{i=1}^{n} Input_i - \Phi\right)$$

عتبة تنشيط العصبون

وتطبق الشروط التالية:

$$If \sum_{i=1}^{n} Input_i \geq \Phi \quad then \quad Output = 1$$

$$If \sum_{i=1}^{n} Input_i < \Phi \quad then \quad Output = 0$$

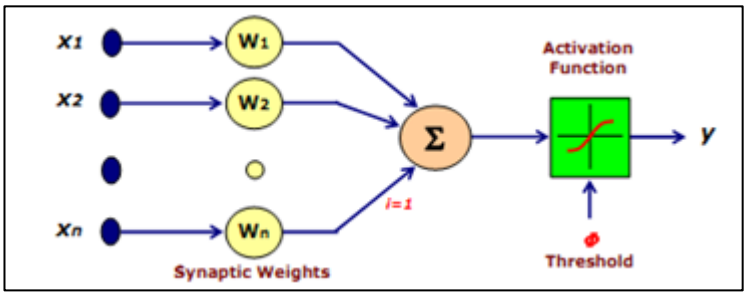
هذا النموذج يفتقر إلى الخصائص التالية:

- الدخل والخرج الغير ثنائي

- الجمع غير الخطي
- تنعيم العتبة
- العشوائية
- المعالجة الزمنية للبيانات

b. نموذج العصبون الصنعي (العناصر الأساسية):

يتكون العصبون من ثلاثة عناصر أساسية : الأوزان – العتبات – وتابع التنشيط



الشكل 2. العناصر الاساسية للعصبون الخطي الصنعي

- الأوزان W

لدينا شعاع الدخل(X) وشعاع من الأوزان (W)

تستخدم الأوزان (w1,w2,….,wn) لتحدد مدى أهمية كل دخل بالنسبة

للعصبون أو مدى قوة شعاع الدخل $X= [x1,x2,…..xn]^T$

كل دخل يضرب بالوزن الموافق له XT W

إذاً الدخل هو عبارة عن مجموع الجداء السلمي لعناصر الدخل بأوزانها

$$I = X^T.W = x_1 w_1 + x_2 w_2 + \ldots + x_n w_n = \sum_{i=1}^{n} x_i w_i$$

بعد ذلك يتم تطبيق أحد توابع التنشيط .

- العتبة Threshold:

العتبة الداخلية للعقدة هي قيمة حدية تؤثر على تنشيط الخرج (Y) لهذه العقدة

$$Y = f(I) = f\{\sum_{i=1}^{n} x_i w_i - \Phi\}$$

في النهاية يتم توليد الخرج (Y) من خلال تمرير المجموع السابق إلى تابع فلترة f يدعى تابع التنشيط أو تابع النقل الذي يعطي الخرج.

- تابع التنشيط (activation function):

يقوم التابع f بعمليات رياضية على إشارة الخرج

التوابع الأكثر شيوعاً:

- Linear function
- Threshold function
- Piecewise linear function
- Sigmoidal (S shaped) function
- Tangent hyperbolic function

يتم اختيار توابع التنشيط اعتماداً على نوع المسألة المطلوب حلها من قبل الشبكة العصبونية.

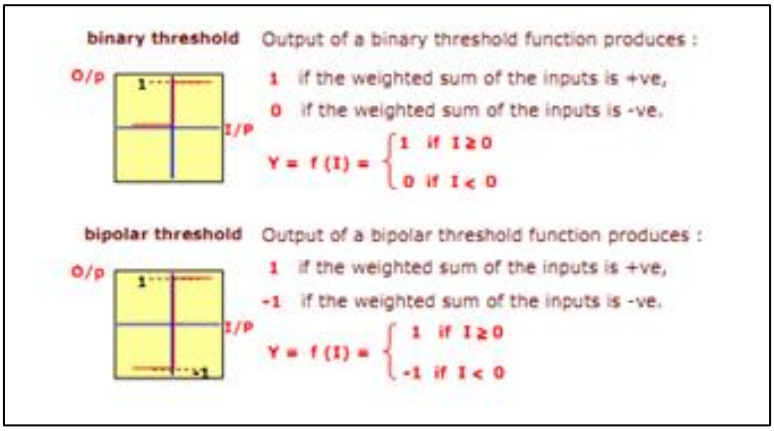
c. أنواع توابع التنشيط:

حاول الباحثون عبر السنوات إيجاد توابع لتحويل الدخل إلى خرج وفيما يلي أكثر هذه التوابع شيوعاً.

- I/P المحور الأفقي يمثل مجموع الدخل.
- O/P المحور العمودي يمثل القيمة التي يعيدها التابع (الخرج).

- **Threshold Function:**

يستخدم هذا التابع مبدأ القيمة الحدية ويكون إما من نوع ثنائي أو ثنائي القطب

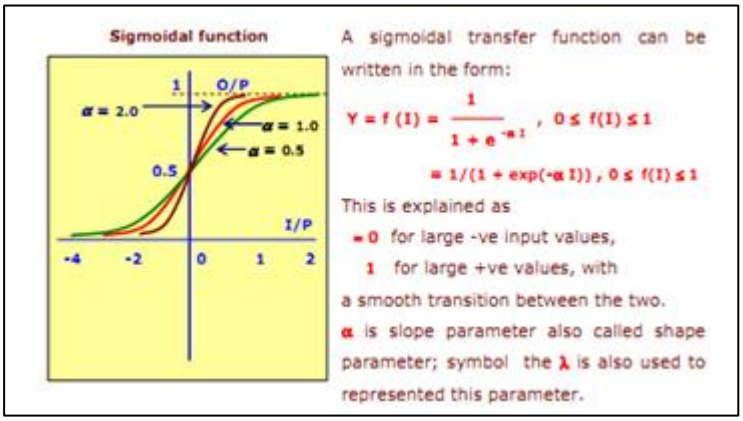


الشكل 3. تابع العتبة

- **Sigmoidal Function (S shaped)** :

تابع غير خطي وهو نوع شائع من توابع التنشيط يستخدم لبناء الشبكات.

وهو جيد رياضياً وقابل للتزايد بشكل كبير و differentiable



الشكل 4. تابع Sigmoid

2. دراسات مرجعية :

هناك منهجيتان أساسيتان لتحديد حالة البرنامج فيما اذا كان سليماً أم خبيثاً ، وهي التحليل الستاتيكي و التحليل الديناميكي .

في منهجيات التحليل الستاتيكي لا يتم تنفيذ التطبيق وإنما تعتمد على بعض الميزات التي نستخرجها من الملفات مثل (opcode frequency , strings , byte sequence , function length , API calls ....)

ومن ثم نطبق إحدى خوارزميات التصنيف لتحديد حالة البرنامج .

بينما في التحليل الديناميكي يتم تنفيذ البرنامج وخلال عمله نراقب سلوكه ونلتقط بعض البارمترات الهامة وتمثيلها كميزات تستخدم في خوارزمية التصنيف .

بالاضافة لذلك ، يوجد خوارزميات تستخدم ميزات مختلفة معتمدة على نوعي التحليل الستاتيكي والديناميكي وتسمى المنهجيات الهجينة.

Bilar[10] استخدم خاصية opcode frequency distribution لتحديد وجود تطبيقات خبيثة ، بينما Karim et al.[11] استخدم مسلسلات وتبديلات opcode كخاصية اساسية لخوارزمية التصنيف .

Ashu et al.[12] استخدم خاصيتين opcode frequency و file size وذلك بهدف زيادة دقة التصنيف مع أكثر من خوارزمية ، حيث وصلت الدقة لأكثر من 98% .

3. توصيف البيانات:

نظراً لأهمية النتائج التي أعطتها الشبكات العصبونية في المجالات العديدة التي طبقت فيها ، لذلك سنقوم ببناء شبكة وندربها لتكون بمثابة مصنف عملي يحدد حالة التطبيق سليم أم خبيث .

أولى خطواتنا تجهيز بيانات الدخل ، وهي عبارة عن تمثيل لتطبيقات الأندرويد المراد تحديد صنفها ، تتألف القاعدة من 100 سجل : 70 سجلاً لبيانات التدريب و 30 بيانات التجريب.

لذلك بعد درسة وبحث حددنا مجموعة من الخاصيات التي تميز تطبيق الأندرويد ، ومن خلالها سوف نقوم ببناء قاعدة البيانات وندرب الشبكة .

اسم الخاصية	قيمة الخاصية	توصيف
Category	Education Games MultiMedia And Video …	فئة التطبيق (فنية – رياضية – تعليمية ..)
Number of sensitive permissions	[1–10]	عدد السماحيات الحساسة في التطبيق
ask for sensitive data	Yes – no	هل يطلب التطبيق بيانات خاصة حتى يعمل
mature content	Yes – no	هل يتضمن التطبيق وجود محتوى جنسي
rating	1– عدم وجود التطبيق على متجر غوغل بلي [1–5]	تقييم التطبيق ضمن المتاجر الالكترونية
API Calls	[1–30]	عدد استدعاءات النظام التي يطلبها التطبيق
Signed Or Not	Yes – no	هل التطبيق موقع الكترونيا من هيئة معروفة
google play	Yes – no	هل التطبيق موجود على متجل google play الرسمي

الخرج هو نتيجة التصنيف هل التطبيق سليم أم خبيث .

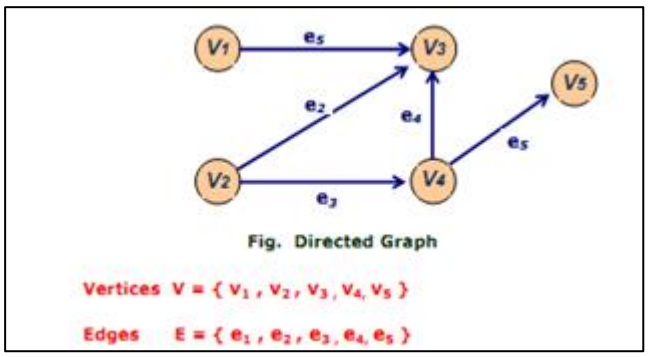
4. بنية الشبكات العصبونية:

الشبكة العصبونية هي نظام معالجة بيانات يتألف من عدد هائل من عناصر المعالجة المتصلة فيما بينها بشكل قوي ومتشابك.

يمكن تمثيل الشبكة العصبونية من خلال بيان موجه G يتضمن عدداً من الرؤوس V والوصلات E

كل رأس يمثل دخل أو خرج العصبونات والوصلات تمثل روابط المشابك

مثال:



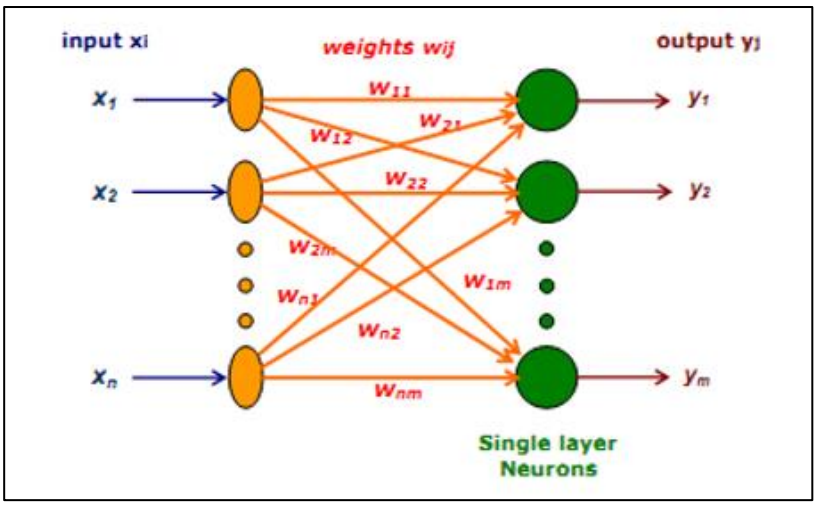
الشكل 5. بيان موجه

a. Single Layer Feed-Forward Network

تتكون هذه الهيكلية من طبقة واحدة فقط من العصبونات ، وتتصل نقاط الدخل مباشرة مع عصبونات الخرج حيث أن كل دخل يتصل مع جميع العصبونات في الخرج ، وكل اتصال يتمثل بوزن w_{ij} (وزن الدخل i مع العصبون j)

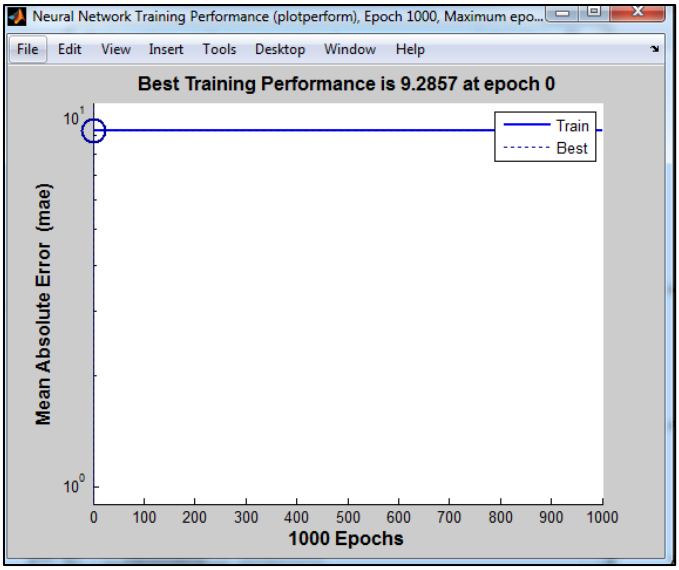
وكما تعتبر هذه الشبكة ذات تغذية متقدمة حيث كل الأسهم تتجه من الدخل نحو عصبونات الخرج ولا يوجد أسهم عائدة.

كل عصبون في طبقة الخرج يقوم بحساب مجموع أوزان الدخل القادمة إليه ، من ثم يقارن المجموع مع حد معين (عتبة) وعلى أساس ذلك يعطى الخرج المناسب 0 أو 1.



الشكل 6. شبكة من طبقة واحدة تغذية متقدمة

بالاختبار الأول سنقوم ببناء شبكة من النوع Perceptron تتألف من طبقة واحدة، وتدريب هذه الشبكة بالبيانات التي ذكرناها سابقاً ، ومن ثم استخراج ومعالجة النتائج ، فيكون الأداء كما يلي :



الشكل 7. أداء الشبكة

ونتيجة محاكاة الشبكة ظهرت كما يلي :

```
>> sim(network1,indata)

ans =

  Columns 1 through 9

    1     1     1     1     1     1     1     1     1

  Columns 10 through 18
```

الشكل 8. محاكاة الشبكة

مما سبق نلاحظ أن النتائج سيئة للغاية وأن الشبكة لا تتدرب والنتائج التي حصلنا عليها بعيدة كل البعد عن النتائج المتوقعة والمرغوبة ، لذلك نجد أن الشبكة المؤلفة من طبقة واحدة فقط لا تعطي خرجاً جيداً.

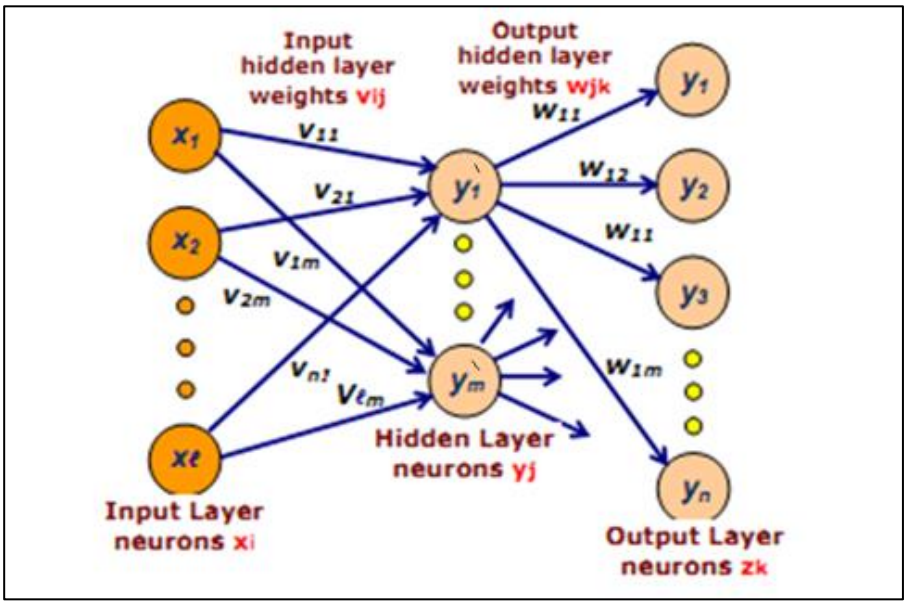
b. Multi Layer Feed-Forward Network

تتكون هذه الهيكلية من عدة طبقات ، طبقة الخرج ومجموعة من الطبقات المخفية (طبقة مخفية واحدة أو أكثر) [3]

ترتبط عناصر الدخل مع جميع عصبونات الطبقة المخفية ، وعصبونات الطبقة المخفية ترتبط مع جميع عصبونات طبقة الخرج .

أما بالنسبة للأوزان : الأوزان بين عناصر الدخل والطبقة المخفية vij (وزن الدخل i مع عصبون الطبقة المخفية j) ، و wjk تمثل الأوزان بين عصبونات الطبقة المخفية وعصبونات طبقة الدخل (وزن العصبون j من الطبقة المخفية مع العصبون k من طبقة الخرج)

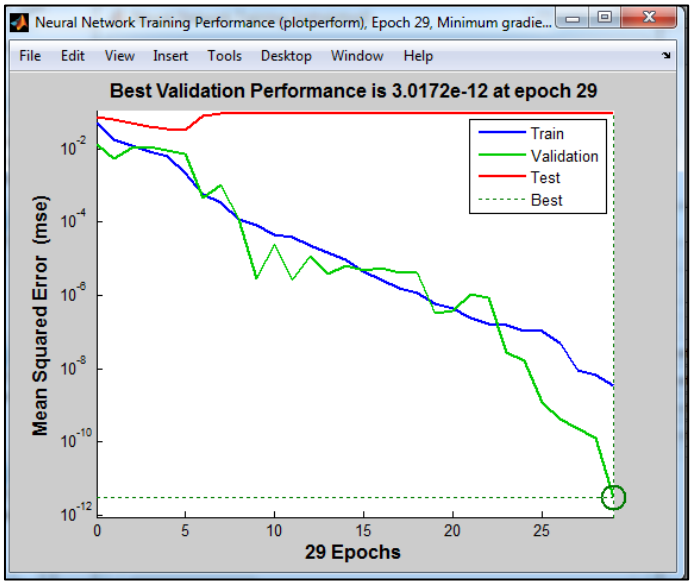
تتم في الطبقة المخفية العمليات الحسابية والمعالجات المرحلية (التكميم) اللازمة لإشارات الدخل قبل إرسالها إلى طبقة الخرج



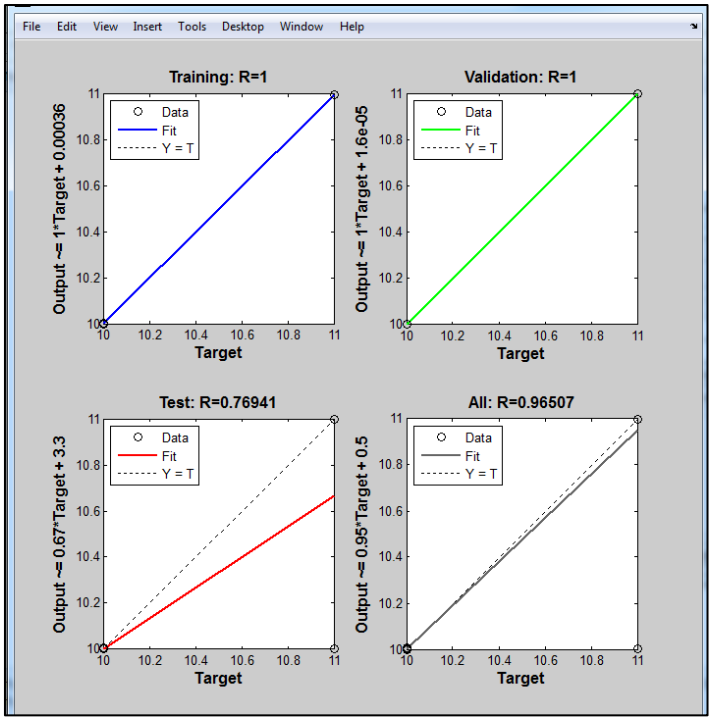
الشكل 9. شبكة من عدة طبقات بتغذية متقدمة

حاولنا أن نجري بعض التحسينات على الشبكة السابقة من خلال إنشاء شبكة جديدة من النوع feed–forward back prop مؤلفة من طبقتين مخفيتين تتألف كل منها من 10 عصبونات ونقوم بتدريب الشبكة على 1000 epochs

ويكون أداؤها:



الشكل 10. أداء الشبكة 2

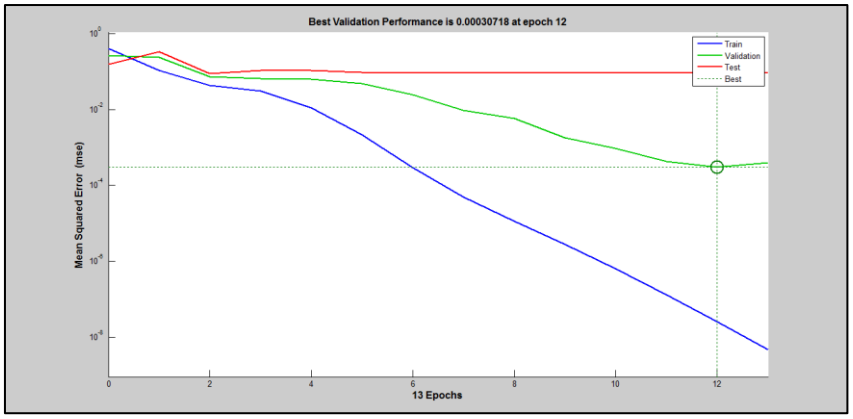


الشكل 11. Network2 Regression

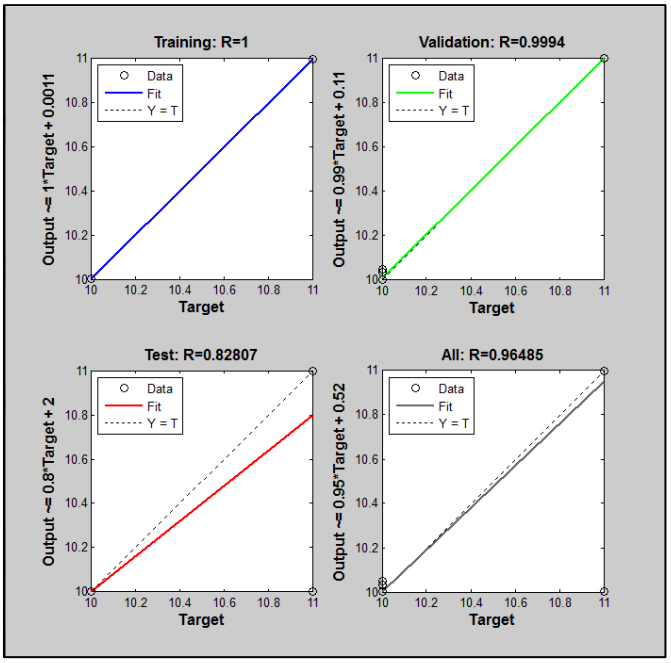
استغرق تعليم هذه epochs 29 بمدة 3 ثانية

بعد ذلك نقارن بين الخرج الذي أعطته الشبكة والخرج المطلوب لنلاحظ أن النتائج قريبة من النتائج الفعلية مع وجود خطأ مرتكب قيمته العظمى 0.0286

النتيجة جيدة ومقبولة ولكن سنحاول تحسين النتائج من خلال زيادة عدد العصبونات في الطبقات المخفية الى 30 عصبوناً في كل منها، لتكون النتائج كما يلي:



الشكل 12. أداء الشبكة 3



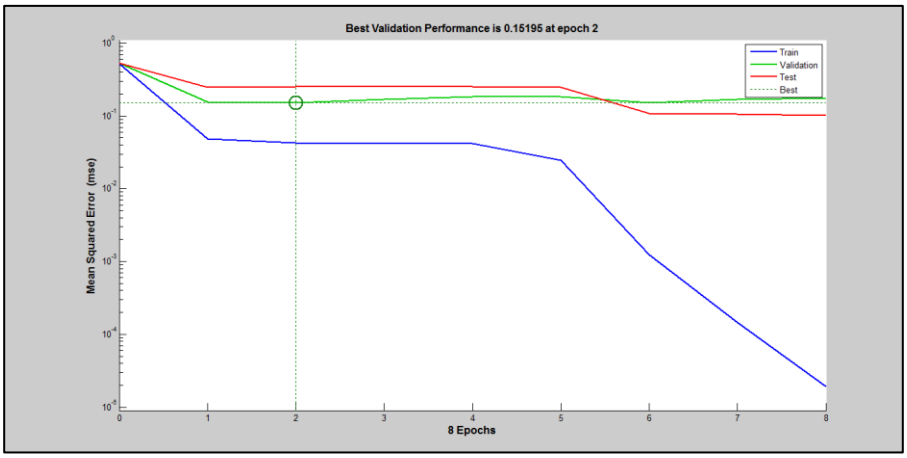
الشكل 13. Network3 Regression

انتهى تدريب شبكة بعد 13 epochs بمدة 6 ثواني ، وأعطت خطأ مرتكباً قيمته

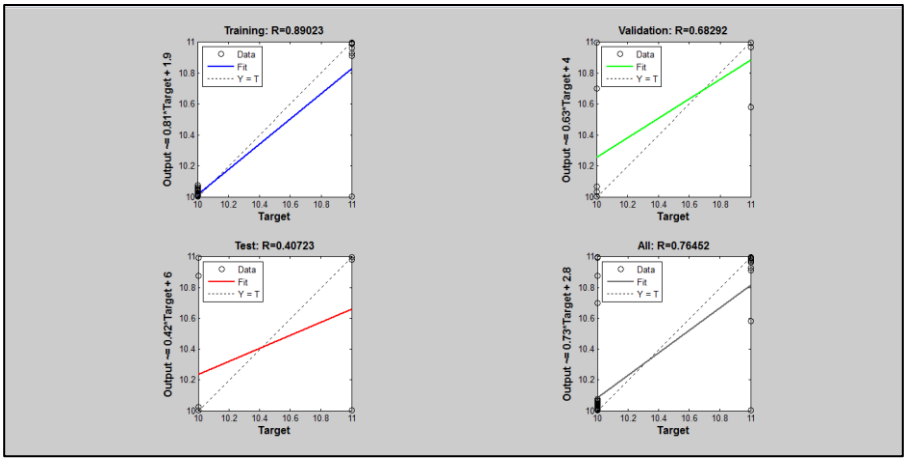
0.0130

النتائج في هذه التجربة جيدة جداً فقد تم تعليم الشبكة خلال زمن قياسي ، كما استطعنا أن نخفض قيمة الخطأ المرتكب وحصلنا على نتائج تصنيف مطابقة تقريباً للخرج المطلوب.

وفي المرحلة الأخيرة من التجريب سنعدل عدد العصبونات لنزيدها إلى 100 عصبون في كل طبقة ليكون الخرج كما يلي :



الشكل 14. أداء الشبكة 4



الشكل 15. Network4 Regression

نتيجة محاكاة تلك الشبكة ظهرت كما يلي :

```
sim(network4,testdata)
ans =
  Columns 1 through 5
   10.7601   11.0000   10.0010   10.0413   10.2834
  Columns 6 through 10
   10.0036   10.0077   10.0219   11.0000   10.0017
  Columns 11 through 15
   10.0011   10.0006   11.0000   10.0067   10.2780
  Columns 16 through 20
   10.0004   11.0000   11.0000   11.0000   11.0000
  Columns 21 through 25
   10.9994   11.0000   10.9999   10.9213   10.5260
  Columns 26 through 30
   10.9999   10.9976   11.0000   10.8404   10.9631
```

الشكل 16. محاكاة الشبكة 4

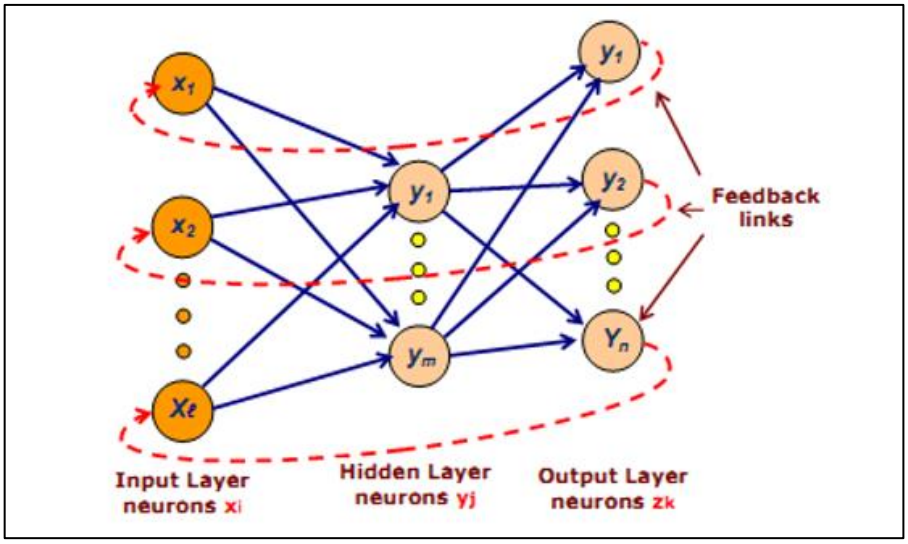
تمت عملية التعليم خلال ثانيتين و 8 epochs

الا أن قيمة الخطأ ازدادت في هذه التجربة فقد بلغت 0.482 ،لذلك نلاحظ أن خرج الشبكة قد ساء بالمقارنة مع الحالة السابقة، وبالتالي يمكننا أن نستنتج من ذلك أن زيادة عدد عصبونات الطبقة المخفية يمكن أن يحسن من النتائج ولكن لا يجب زيادة عددها إلى حد كبير لأنه قد ينعكس سلبيا على أداء الشبكة، لذلك حاولنا تحسين الشبكة من خلال زيادة عدد الطبقات المخفية إلى طبقتين كما وضحنا سابقاً وذلك لأن زيادة عدد الطبقات المخفية يساعد في زيادة كفاءة خاصية التحويل غير الخطي بين الدخل والخرج.

c. Recurrent Network

تتميز هذه الهيكلية عن سابقاتها بوجود اتصال من عصبونات طبقة الخرج إلى عناصر الدخل ما يسمى بالتغذية الراجعة أو العودية

يصعب تعليم الشبكة عند استخدام هذه الهيكلية ، لذا لها تطبيقات خاصة لاستخدامها.



الشكل 17. شبكة ذات تغذية راجعة

5. تعليم الشبكة :

يتمحور المفهوم الاساسي للشبكات العصبونية حول ألية تعليم الشبكة سواء بنماذج جديدة أو معروفة سابقاً.

تصنف طرق التعليم وفق 3 أنواع :

- التعلم بإشراف
- التعلم من غير إشراف
- التعلم بالتعزيز

5.1 التعلم بإشراف:

يعتمد هذا النمط على وجود معلم للشبكة ، يقدم نماذج الدخل للشبكة وكما يعطي نموذج الخرج المتوقع[3].

تستخدم نماذج الدخل لتعليم الشبكة ، وعملية التعليم تعتمد على المقارنة بين الخرج المتوقع والخرج الذي اعطته الشبكة وفقاً لنموذج الدخل ،ويتم حساب قيمة الخطأ المرتكب.

تتعدل قيم الأوزان وفقا للخطأ المرتكب الذي أعطته الشبكة وذلك بهدف تحسين النتائج.

هناك العديد من الدراسات التي اعتمدت هذا النهج منها : دراسة أعدت عام 2004 اعتمدت على استخدام خوارزمية Naïve Bayes[13] لاجراء التصنيف ، ودراسة أخرى تمت في عام 2013 اعتمدت على خورزمية Support Vector Machine[14] . إن الدراستين السابقتين اعتمدتا التحليل الديناميكي ، أما التحليل الستاتيكي اجريت دراسة عام 2016 اعتمدته وقام الباحثون باستخدام خورزمية Random Forest[15] لبناء المصنف .

عادة نستخدم التعلم بإشراف لتسريع أداء الشبكة لذا اعتمدناه في هذه الدراسة ، حيث قمنا بتجميع الميزات واستخدام خوارزمية اشجار القرار لتعليم الشبكة وإجراء التصنيف.

5.2 التعلم بدون إشراف:

في هذا النمط من التعلم لا يوجد معلم للشبكة ولا تعطى نماذج متوقعة للخرج ، وإنما تأخذ الشبكة الدخل المطلوب وتقوم بإيجاد علاقة بين العناصر المدخلة ومن خلالها تعطي الخرج الصحيح.

غالبية الطرق الشهيرة في استخراج الميزات تستخدم خوارزميات التعلم بدون إشراف ، وذلك لتمثيل الميزات أفضل ما يمكن.

ومن أهم طرق استخراج الميزات :

hierarchical, unary variable removal، Goodness evaluator, and Weighted Term Frequency.

5.3 التعلم بالتعزيز:

تعتمد هذه الطريقة على وجود معلم يعطي الشبكة نماذج الدخل ، ولكن دون إعطاء الخرج المتوقع ، وإنما يشير فقط للعصبون الذي أعطى الخرج الصحيح .

فتتم مكافأة العصبونات التي أعطت النتائج الأقرب للخرج الصحيح من خلال تعديل قيم أوزانها.

6. الخاتمة والتوصيات

لقد ناقشنا دراسة منهجية تستند إلى الدراسات الحديثة في الكشف عن التطبيقات الضارة لنظام أندرويد في المتاجر الرسمية والغير رسمية . لا تسلط هذه الدراسة الضوء على طرق و تقنيات الكشف المتنوعة سواء كانت ستاتيكية أو ديناميكية ، وإنما تناقش أيضاً التقنيات المختلفة التي لها القدرة على مواجهة الهجوم والأذى الذي

يتسبب التطبيق المشبوه من خلال الاعتماد على مجموعة من الميزات لبناء شبكة عصبونية تساعدنا في تحديد نوع وحالة التطبيق .

7. المراجع

[1] RIASAT, R2016-"A Survey on Android Malware Detection Techniques" .Institute of Software, Chinese Academy of Sciences, Beijing, China , 9p.

[2] Yajin ZHOU and Xuxian JIANG , 2012- "Dissecting android malware: Characterization and evolution In Security and Privacy (SP)" . 2012 IEEE Symposium on، pages 95-109.

[3] BENGIO, Y2009- "Learning deep architectures for ai. Foundations and trends in Machine Learning". 2(1):1-127p.

[4] SAXE and BERLIN , JK2015 – "Deep neural network based malware detection using two dimensional binary program features". In International Conference on Malicious and Unwanted Software (MALWARE), pages 11-20, Oct 2015.

[5] A. Sharma and S. K. Dash- "Mining api calls and permissions for android malware detection". In Cryptology and Network Security, pages 191-205. 2014.

[6] Kim, TaeGuen, et a- "A multimodal deep learning method for android malware detection using various features." IEEE Transactions on Information Forensics and Security 14.3 (2018): 773-788.

[7] Tobiyama, Shun, et al- "Malware detection with deep neural network using process behavior." 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC). Vol. 2. IEEE, 2016.

[8] Khan, Haider Adnan, et al- "Malware detection in embedded systems using neural network model for electromagnetic side-channel signals." Journal of Hardware and Systems Security 3.4 (2019): 305-318.

[9] HaddadPajouh, Hamed, et al- "A deep recurrent neural network based approach for internet of things malware threat hunting." Future Generation Computer Systems 85 (2018): 88-96.

[10] Daniel Bilar. ""Opcodes as predictor for malware". Int. J. Electron. Secur. Digit Forensic, 1(2):156–168, January 2007.

[11] Md. Enamul Karim, Andrew Walenstein, Arun Lakhotia, and Laxmi Parida. "Malware phylogeny generation using

permutations of code ". Journal in Computer Virology, 1:13–23, 2005

[12] Malicia project. http://malicia-project.com, 2012, Date last accessed 15-July-2014.

[13] Jeremy Z. Kolter and Marcus A. Maloof. "Learning to detect malicious executables in the wild". In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04، pages 470–478, New York, NY, USA, 2004. ACM.

[14] Igor Santos, Felix Brezo, Xabier Ugarte-Pedrero, and Pablo G. Bringas." Opcode sequences as representation of executables for data-mining-based unknown malware detection ". Information Sciences, 231:64 – 82, 2013. Data Mining for Information Security.

[15] Ashu Sharma and Sanjay Kumar Sahay." An effective approach for classification of advanced malware with high accuracy". International Journal of Security and Its Applications, 10(4), 2016.