GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Extensive Study of Malware Detection and Prevention system

A Project Report of Capstone Project – 2

Submitted by

Kushal Bhargava
(1613101352 / 16SCSE101747)

in partial fulfillment for the award of the degree

of

Bachelor of Technology

IN

Computer Science and Engineering

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the supervision of
Dr. M. THIRUNAVUKKARASAN.

Professor

APRIL/MAY-2020

DECLARATION

Project Title: **Extensive Study of Malware, Detection and prevention System**

Degree for which the project work is submitted: **Bachelor of Technology in Computer Science and Engineering**

I declare that the presented project represents largely my own ideas and work in my own words. Where others ideas or words have been included, I have adequately cited and listed in the reference materials. The report has been prepared without resorting to plagiarism. I have adhered to all principles of academic honesty and integrity. No falsified or fabricated data have been presented in the report. I understand that any violation of the above will cause for disciplinary action by the Institute, including revoking the conferred degree, if conferred, and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken.

Signature( )

Kushal Bhargava

Enrolment No. 1613101352

Date: 08/05/20

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

SCHOOL OF COMPUTING AND SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "**Extensive study of malware, detection and prevention system**" is the bonafide work of "**Kushal Bhargava (1613101352)**" who carried out the project work under my supervision.

SIGNATURE OF HEAD **SIGNATURE OF SUPERVISOR**

Dr. A.Satheesh, Dr. M. Thirunavukkarasan

Professor & Dean, **Professor,**

School of Computing Science &Engineering **School of Computing Science Engineering**

ABSTRACT

Malware is one of the most serious security threats on the Internet today. The threat is increasing in a greater pace with the intensive use of networks and Internet in our day-to-day activities. The most recent reports emphasize that the invention of malicious software is rapidly increasing. Over the last decade, a number of studies have been made on malware and their countermeasures. Researchers and manufacturers are making great efforts to invent effective malware detection methods to produce anti-malware systems for better protection of computers and networks. In this Project, a detailed study has been conducted on malware taxonomy and the approaches made by the researchers to improve antimalware or malware detection systems, and a machine learning based malware detection system has been proposed .Thus, it provides an up-to-date review to the researchers and developers of malware detection systems and proposes a model for detection of malware in executable files.

ACKNOWLEDGEMENT

The contributions of many different people, in their different ways, have made this possible. We would like to extend our gratitude to our project guide (Dr. M.THIRUNAVUKKARASAN) Who gave us the opportunity to make this project on this topic(**"Extensive study of malware, detection and prevention system"**), which helped us in doing a lot of research and we came to know about many new things so we are thankful to them.

Secondly, we would like to thank our parents and friends who help me in making this project with a limited frame of time.

TABLE OF CONTENTS

List of figures

Chapter.1
Introduction

1.1 Purpose

As our society is moving toward a completely technology dependent time we need to increase our security for malicious files that a attacker uses for different purposes such and to get our private information, files, to do money related fraud with us and these malicious files can also be used on a huge scale such as effecting a country's defense system , our the essential system that the government need in such cases these malicious files can have a very bad effect on our society. This project provides a detailed review of the current malware detection systems so that the researchers and developers can get a brief idea about the existing systems used and proposes a machine learning based malware detection system for windows executable files a step toward making our cyber world safer.

1.2 Types of malwares: The term malware includes viruses, worms, Trojan Horses, rootkits, spyware, adware, keyloggers, botnet and more. To get an overview of the difference between all these types of threats and the way they work, it makes sense to divide them into groups.

A. Viruses and worms - the contagious threat Viruses and worms are defined by their behaviour – malicious software designed to spread without the user's knowledge. A virus infects legitimate software and when this software is used by the computer owner it spreads the virus – so viruses need you to act before they can spread. Computer worms, on the other hand, spread without user action. Both viruses and worms can carry a so-called "payload" – malicious code designed to do damage. A virus is a type of malware that propagates by inserting a copy of itself into and becoming part of another program. It spreads from one computer to another, leaving infections as it travels. Almost all viruses are attached to an executable file, when the file is executed; the viral code is executed as well. Viruses spread when the software or document they are attached to is transferred from one computer to another using the network, a disk, file sharing, or infected e-mail attachments. Unlike viruses, worms are standalone softwares and do not require a host program or human help to propagate. To spread, worms either exploit vulnerability on the target system or use some kind of social engineering to trick users into executing them.

B. Trojans, rootkits and adware – the masked threat Trojans and rootkits are grouped together as they both seek to conceal attacks on computers. Trojan Horses are malignant pieces of software pretending to be benign applications. Users therefore download them thinking they will get a useful piece of software and instead end up with a malware infected computer. Rootkits are a masking technique for malware, but do not contain damaging software.

Rootkit techniques were invented by virus writers to conceal malware, so it could go unnoticed by antivirus detection and removal programs. Trojan is named after the wooden horse the Greeks used to infiltrate Troy. It is a harmful piece of software that looks legitimate. Users are typically tricked into loading and executing it on their systems. After it is activated, it can achieve any number of attacks on the host, from irritating the user (popping up windows or changing desktops) to damaging the host (deleting files, stealing data, or activating and spreading other malware, such as viruses). Trojans are also known to create back doors to give malicious users access to the system. Unlike viruses and worms, Trojans do not reproduce by infecting other files nor do they self-replicate. Trojans must spread through user interaction such as opening an e-mail attachment or downloading and running a file from the Internet. Adware or Advertisingsupported software automatically plays, displays or downloads advertisements to a computer after malicious software is installed or application is used. This kind of code is also embedded into free software. The most common source of adware programs are free games, Peer to peer clients like Kazaa, Bearshare etc.

C. Spyware and keyloggers – the financial threat Spyware and keyloggers are malware used in malicious attacks like identity theft, phishing and social engineering - threats designed to steal money from unknowing computer users, businesses and banks. Spyware is a collective term used for software which monitors or gathers personal information about the user like ,the pages frequently visited, email address, credit card no, key pressed by user etc. It enters a system when free or trial software is downloaded and installed without the user's knowledge. It changes the settings of yours browser and adds abdominal browser toolbars.

1.3 MOTIVATION AND SCOPE: In todays worlds the threat of malicious files is increasing day by day as our society is completely moving towards a technologically dependent world. In this situation it is very necessary for our world to secure our system from malwares. This project also provides a study about the malware and its types so that the researchers and developers so that they can also get an idea about whats going on. We also propose a system based on machine learning that detects malwares on executable files which can be a future product.

Chapter. 2 Literature Survey

Existing systems:

Various Malware detection techniques are used to detect the malware and prevent the computer system from being infected, protecting it from potential information loss and system compromise. They can be categorized into signaturebased detection, heuristic-based detection, specificationbased and data mining based detection as shown in figure 1.

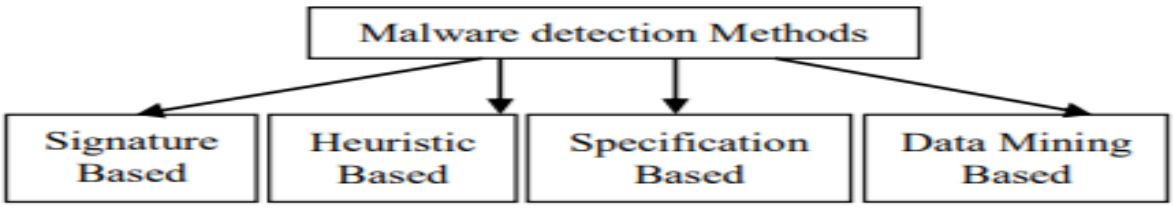


Figure 1: Types of malware detection Methods

Figure 2.1

A. Signature-Based Detection It is also called as Misuse detection. It maintains the database of signature and detects malware by comparing pattern against the database. The signatures are created by examining the disassembled code of malware binary. Disassembled code is analyzed and features are extracted. These features are used in constructing the signature of particular malware family. A library of known code signatures is updated and refreshed constantly by the antivirus software vendor so this technique can detect the known instances of malware accurately. The main advantages of this technique is that it can detect known instances of malware accurately, less amount of resources are required to detect the malware and it mainly focus on signature of attack. The major drawback is that it can't detect the new, unknown instances of malware as no signature is available for such type of malware.

B. Heuristic-Based Detection It is also called as behaviour or anomaly-based detection. The main purpose is to analyze the behaviour of known or unknown malwares. Behavioral parameter includes various factors such as source or destination address of malware, types of attachments, and other countable statistical features. It usually occurs in two phases: Training phase and detection phase. During training phase the behaviour of system is observed in the absence of attack and machine learning technique is used to create a profile of such normal behaviour. In detection phase this profile is compared against the current behaviour and differences are considered as potential attacks [6]. The advantage of this technique is that it can detect known as well as new, unknown instances of malware and it focuses on the behaviour of system to detect unknown attack. The disadvantage of this technique is that it needs to update the data describing the system behaviour and the statistics in normal profile but it tends to be large. It need more resources like CPU time, memory and disk space and level of false positive is high.

C. Specification-Based Detection It is derivative of behaviour-based detection that tries to overcome the typical high false alarm rate associated with it. Specification based detection relies on program specifications that describe the intended behaviour of security critical programs [6]. It involves monitoring program executions and detecting deviation of their behaviour from the specification, rather than detecting the occurrence of specific attack patterns. This technique is similar to anomaly detection but the difference is that instead of relying on machine learning techniques, it will be based on manually developed specifications that capture legitimate system behaviour [6]. The advantage of this technique is that it can detect known and unknown instances of malware and level of false positive is low but level of false negative is high and not as effective as behaviour based detection in detecting new attacks; especially in network probing and denial of service attacks. Development of detailed specification is time consuming.

D. Data mining based detection From last decade data mining has been the main focus of many malware researcher for detecting the new, unknown malwares; they have added data mining as a fourth proposed malware detection technique. In 2001 Schultz [7] first introduced the idea of applying the data mining and machine learning method for the detection of new, unknown malware based on their respective binary codes. Then different studies have been conducted for detection of different malwares. Data mining

helps in analyzing the data, with automated statistical analysis techniques, by identifying meaningful patterns or correlations. The results from this analysis can be summarized into useful information and can be used for prediction. Machine learning algorithms are used for detecting patterns or relations in data, which are further used to develop a classifier [8]. The common method of applying the data mining technique for malware detection is to start with generating a feature sets. These feature sets include instruction sequence, API/System call sequence, hexadecimal byte code sequence (n-gram) etc. The numbers of extracted features are very high so various text categorization techniques are applied to select consistent features and generate the training and test feature sets. Then classification algorithms are applied on the consistent training feature set to generate and train the classifier and test feature set is examined by using these trained classifiers. The performance of each classifier is evaluated by identifying the rate of False Positive, False Negative, True Positive, True Negative and calculate the TPR, FPR, Recall, precision and F1-measure.The advantage of data mining based detection is that detection rate is high as compared to signature based detection method . It detects the known as well as unknown, new instances of malware.

E. Network-based IDS A network-based IDS (NIDS) differs from an HIDS in that it is usually placed along a LAN wire. It attempts to discover unauthorized and malicious access to a LAN by analyzing traffic that traverses the wire to multiple hosts. There are many algorithms for detecting malicious traffic, but they generally read inbound and outgoing packets and searches for any suspicious patterns. Any alert generated by an NIDS allows it to notify administrators or take active actions such as blocking the source IP address. Three of the most common placements of NIDS are directly connecting it to a switch spanning port, using a network tap, and connected inline.

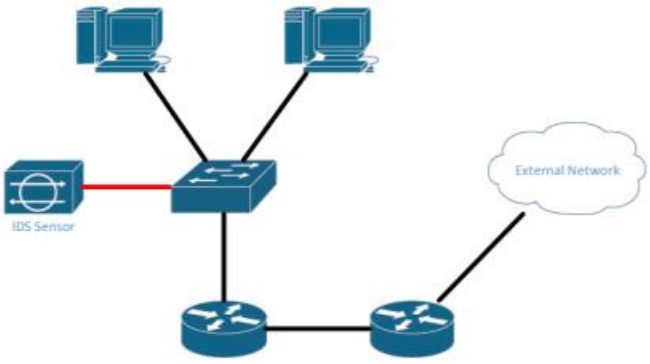


Figure 2.

Figure shows the IDS connected to a switch that has SPAN port configuration capability. On some managed switches, a SPAN port can be configured to send all packets on the network to that port as well as their ultimate destination (Baker, Beale, Caswell & Poor, 2004). In this configuration, the switch copies all traffic it receives to the IDS interface being used to monitor traffic. The major downside of this method is increased bandwidth and resource usage, since the switch must work twice as hard to deliver traffic.

Intrusion Prevention Systems (IPS)

The main function of an IPS is to intervene in cases of suspected attacks. Generally, an IPS is essentially a combination of access control devices – such as firewalls and routers – and IDSs. In other words, an IPS is an IDS with access control capabilities or active response methods. Like IDSs, an IPS can be host-based or network-based, and uses anomaly-based detection (prevention) or a signature- or ruleset-based approach. The following are common countermeasures implemented by IPSs:

A. Denying the traffic. This is the simplest method, where the intrusion system blocks the IP addresses and ports involved – both source and destination. The downside to this method is that many devices on the global network are hidden behind a global address. Blocking that address will also block other legitimate traffic that may be located behind that address.

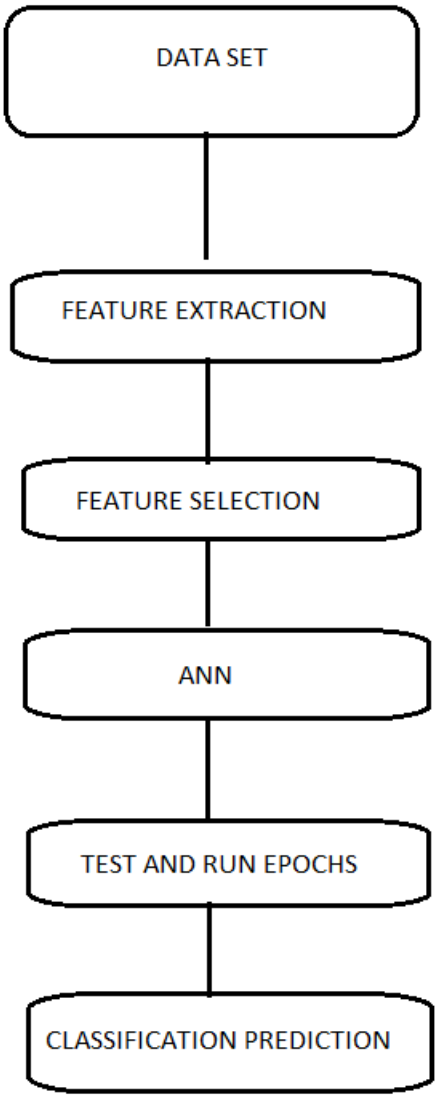
B. Active logging. Although logging is a feature shared by IDSs, an IPS can increase the usability of a log by, for example, automatically exporting traffic logs that meet certain criteria to external network analysis software such as Wireshark.

C. Communicating with a separate device with access control capabilities. Many modern IDSs and IPSs also complement the operations of a LAN by communicating with an external, or separate, firewall or router, which have access control capabilities. In the event of an intrusion, an IDS/IPS can send an alert or request to a firewall or router. The firewall or router will then take the necessary actions to deal with the intrusion, such as dropping the packets or blocking further traffic from that source.

D. Sending a TCP reset (Carter, 2005). If an attack is a TCP-based attack, an IPS can send a reset signal back to the attacker's protocol stack, which would close the current session, and can be repeated as frequently as needed.

E. Setting an SNMP trap (Burns, Adesina & Barker, 2012). When an alarm is triggered, the intrusion system will send an SNMP trap to indicate to an SNMP management system that a network or device is under attack. The management system can choose to take an action based on the event, such as polling the agent directly, or polling other associated device agent to get a better understanding of the event

Chapter 3 Proposed model



Figure(3.1)

Now, since the parts of the project are presented by the above steps, we would like to explain a variety of things on these parts. But before just moving on to these parts directly we will explain the background behind all of them –

- **Data set** – a large data set which is gathered from a open source is used as training data set.
- **Feature Extraction** – The features of the data i.e, the size of the file, DLL signature, hash code, utf8 are extracted.
- **Feature Selection** – This is done on the basis of the impact factor of the feature in this case we used the DLL signature.
- **Artificial Neural Network** – a artificial neural network is used to train the machine having 2 layers of 70 neurons.
- **Test and run Epochs** – After the training we will test the machine and run epochs to test the accuracy of the machine.

Artificial Intelligence, Machine learning and Neural networks

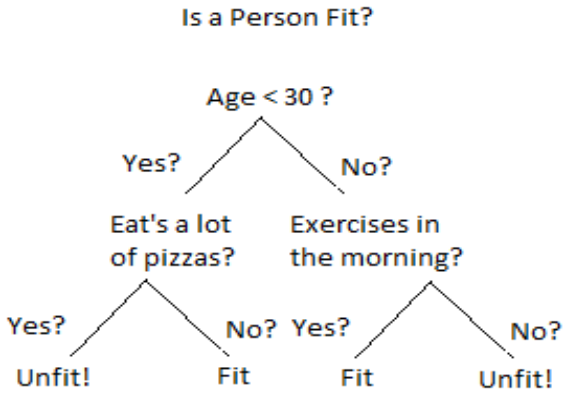
AI is a term that was coined by scientists in Dartmouth conference in 1956.Over the past years the term has become more popular and has been accepted as a key to a bright future for our civilization. The main aim of the researchers in the initial years was to construct complex machines that could think like humans. It should have had all the senses that we humans have and maybe even more, Like the ones we have seen in sci-fi movies "The Terminator". The term AI is a very broad term in itself, which includes several other terms that we are going to explain further.

The complex machines that the scientists and researchers wanted to build at that time, that could think like humans had to have some intelligence like humans also. Naturally this intelligence can't be put into them magically. We have to create some approaches that could enable the machines to think. **This "intelligence" is what is called "Machine Learning".Machine learning** is the process in which we study the data, learn from it and then make some future predictions. The machine learns the ability to do a task by learning from its experiences again and again until it gets the idea. The most basic example is a new born baby, who doesn't know anything. But the child starts to learn from his/her experiences what is good and what is bad. If the child accidently puts his finger on fire, he feels pain and then immediately removes his finger from fire. Now the next time he knows not to do something like that. Machine learning uses a lot of algorithms that helps it to learn from data and predict the outcomes. We have listed some algorithms below-

- Decision Tree
- Inductive logic programming

- Clustering
- Reinforcement learning
- Bayesian networks

Machine learning can be explained using a simple example which uses Decision Tree to predict the outcome.Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.An example of a decision tree can be explained using below binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves, which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes, no type problem)



Figure(3.2).

So now using the decision tree we can tell if a person is fit or not. This gives us a basic idea of machine learning and how it can be used.

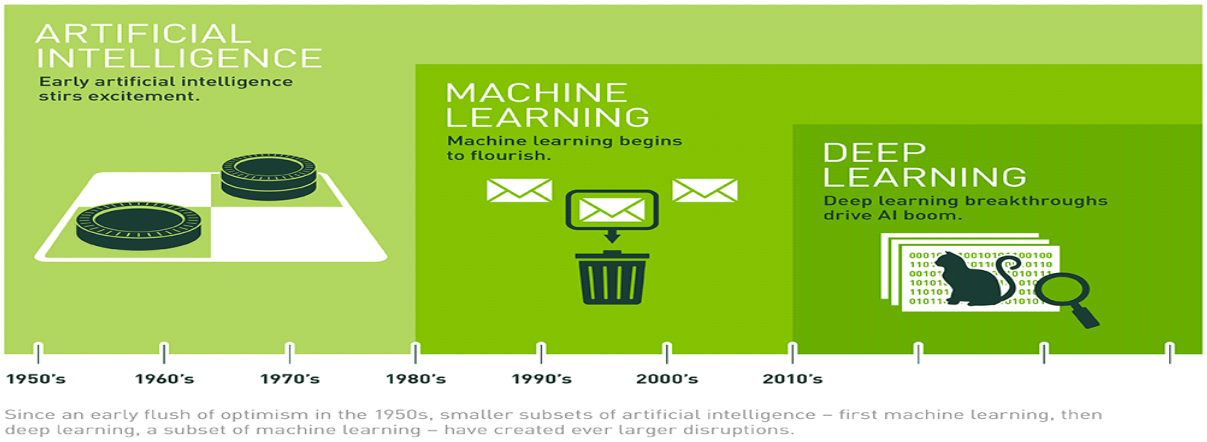
Well, Machine learning was a good technique but ultimately it did not solve the basic purpose of Artificial Intelligence. Since then a new approach was being developed by the AI researchers, this new technique was based on the biological brain. Our brain contains neurons, and each neuron is connected to some other neuron. The name of this approach **was Neural Networks.**

Artificial Neural Network - ANN is an efficient computing system whose central theme is borrowed from the analogy of biological neural networks. ANNs are also named as "artificial neural systems," or "parallel distributed processing systems," or "connectionist systems." ANN acquires a large collection of units that are interconnected in some pattern to allow

communication between the units. These units, also referred to as nodes or neurons, are simple processors which operate in parallel.

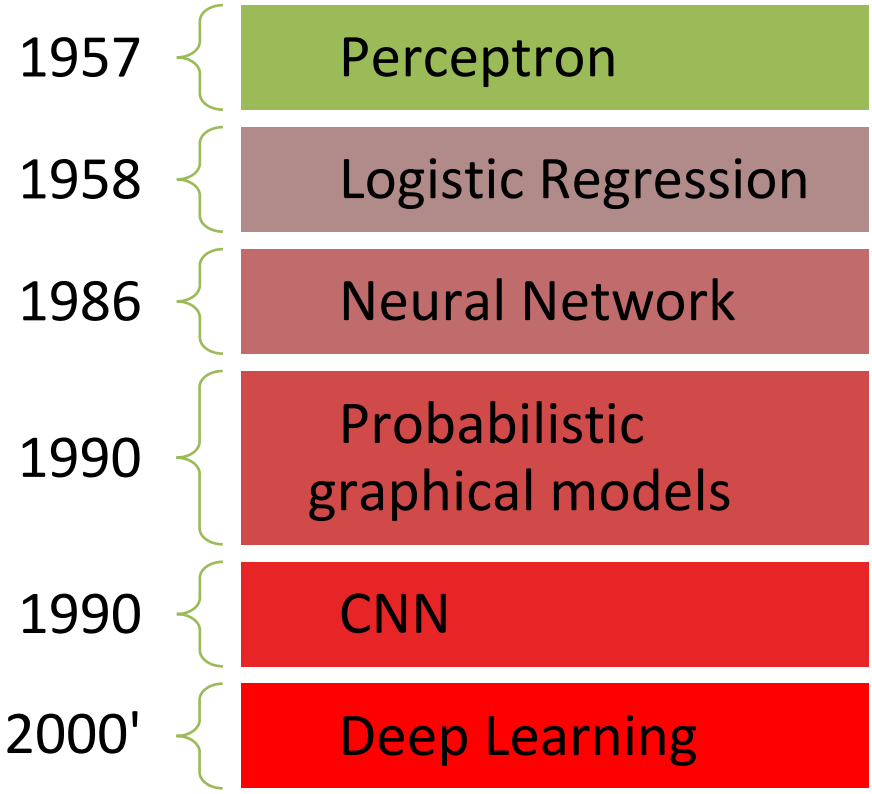
Every neuron is connected with other neuron through a connection link. Each connection link is associated with a weight that has information about the input signal. This is the most useful information for neurons to solve a particular problem because the weight usually excites or inhibits the signal that is being communicated. Each neuron has an internal state, which is called an activation signal. Output signals, which are produced after combining the input signals and activation rule, may be sent to other units.

The diagram below will explain the relation b/w all the ideologies.



Figure(3.3)

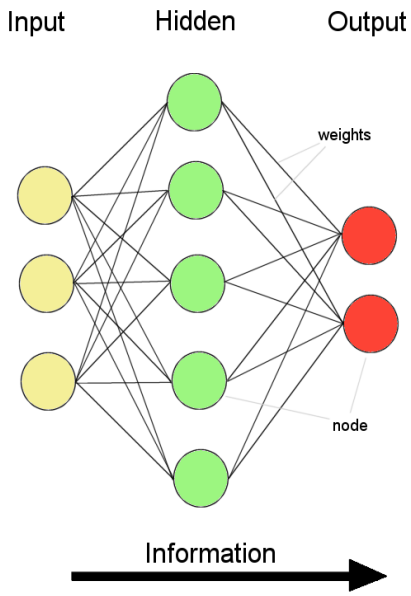
A brief timeline of Machine learning



Figure(3.4**)**

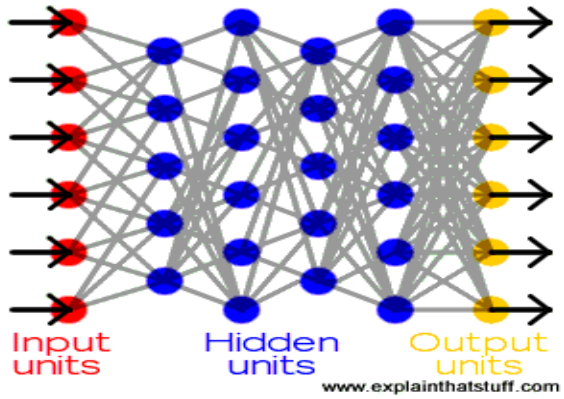
Neural Networks The Artificial Neural Nets are based on the biological neurons as mentioned earlier, so there two basic components of a Neural net

- Neurons(nodes)
- Synapses(weights)



(Figure 3.5)

A typical neural network consists of a lot of neurons (units). There are input units as well as output units, the former are designed to receive various types of information from the outside world and then it attempts to learn from that information. The output units respond to the information it has learned. There are one more type of units in between these units, called the hidden units which make up the majority of the neural nets. The connection between these units are called weights. The neural nets are fully connected.



Figure(3.6)

Learning in Neural Networks

The information in the network flows in two ways:

- **Learning phase**: Patterns of information is fed into the network via the input units, which activates the hidden layers and then the result arrives at the output layer. This design is called The **Feedforward network.**Each unit receives inputs from the units to its left, and the inputs are multiplied by the weights of the connections they travel along. Every unit adds up all the inputs it receives in this way and (in the simplest type of network) if the sum is more than a certain threshold value, the unit "fires" and triggers the units it's connected to (those on its right).
- **Back Propagation:** A feedback process is very important in a network similarly as we humans take feedback about our progress all the time. Hence a neural network also tends to do the same. Once the result is reached at the output layer, it is compared with the result it was supposed to produce. Then the difference between the two is used to adjust the weights of the connections between the units in the network that is going backwards. Hence it is called **Back Propagation.**

Once the network has been trained with enough learning examples, it reaches a point where you can present it with an entirely new set of inputs it's never seen before and see how it responds.

.

Chapter4
Implementation

As Neural network is a class of machine learning algorithms, there are different variations of neural networks. The class of Neural networks contains various architectures like **Convolutional neural networks (CNN), Recurrent neural networks (RNN)** and **Deep belief networks**.The number of (layers of) units, their types, and the way they are connected to each other is called the **network architecture.**

Now, for our project we have used **ANN** Architecture. we will explain this in detail. A ANN consists of a **A input layer, hidden layers(may or may not be)** and **output layer.**

- **The Input Layer**: The **input layer** of a **neural network** is composed of artificial **input** neurons, and brings the initial data into the system for further processing by subsequent **layers** of artificial neurons. The **input layer** is the very beginning of the workflow for the artificial **neural network**.

- **The Hidden Layer**: The **hidden layer** is a **layer** which is **hidden** in between input and output **layers** since the output of one **layer** is the input of another **layer**. The **hidden layers** perform computations on the weighted inputs and produce net input which is then applied with activation functions to produce the actual output.

- **<u>The Output Layer:</u>** The **output layer** in an artificial **neural network** is the last **layer** of neurons that produces given **outputs** for the program.

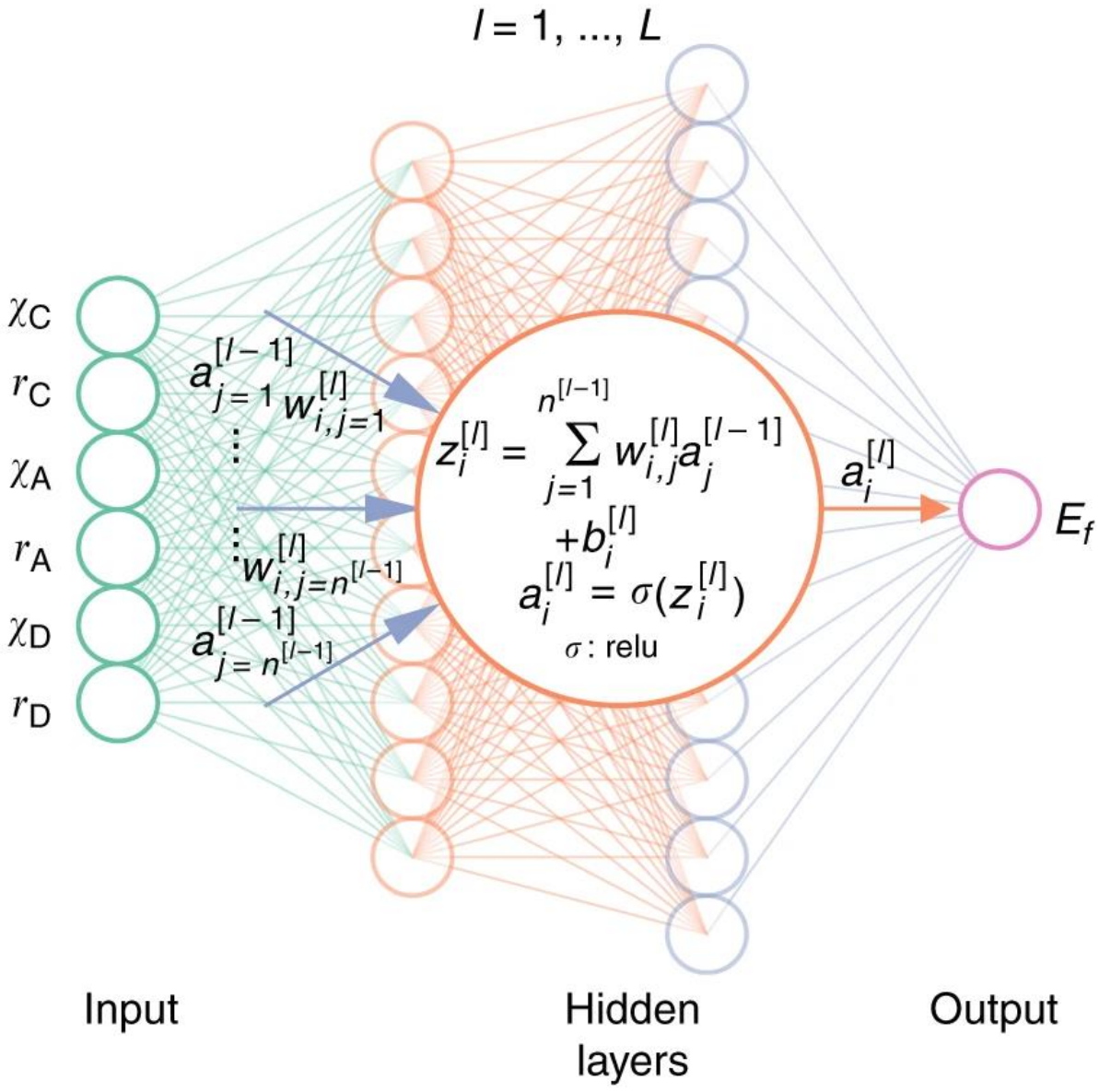


Figure (4.1)

This Image shows us the architecture of the Artificial Neural network that we are using.

- **THE DATA SET :** The training data set has been collected form a open sources which contains both malicious and clean executable files. This is a large data set as the higher the entropy(degree of randomness) the more accurate out model will be. The total size of the dataset in about 5 gigabytes which is evenly spread between malicious and clean files.

- **FEATURE EXTRACTION :** Feature extraction is done to extract the different features present in a data set in this case as the data set contains executable files the features of the data set can be file size, file extension, utf8, hash code, DLL signature,ansi etc. These extracted features can also be called as the attributes of the data, these attributes of the data can be used to define each data element.

- **FEATURE SELECTION :** This is the process in which a certain Feature is selected as the feature on the basis of which the files will be judged. In this case we have used the Feature DLL signature and done DLL enumaration as the optimal result can be found through it the problem with hash code is that it can be easily manufactured and can be easily manipulated. The size of a file can never be treated as a attributed that can be used to determine weather the file is malicious or not and the same goes for the extension of the file as a file of any size can be malicious and a file can have any extension.

- **NORMALIZATION :** Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. For machine learning, every dataset does not require normalization. It is required only when features have different ranges. In this case the normalization is done by taking the sum of every element of the matrix as the denominator and performing division with every element of the dataset matrix.

- **ENTROPY CALCULATION** : Entropy, as it relates to machine learning, is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information. The higher the entropy the more accurate our model will be.

Note :- The above steps are done on the dataset divided in small n no. of small datasets and then the whole data is concatenated and converted into a matrix which is going to be the input for our artificial neural network.

- **COSINE SIMILARITY :** Then we find the degree of similarity between the elements of the data and set a threshold to decide weather they are similar enough so that we don't need to apply classifier between them.

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 4.2

These steps were taken to prepare the data for the artificial neural network.

The artificial neural network which we are using is downloaded from a open source . The activation function used in this artificial neural network are Relu and softmat as these to functions work best on the signature based learning.

RELU FUNCTION :- The Rectified linear unit (ReLu) [3] activation function has been the most widely used activation function for deep learning applications with state-of-the-art results. It usually achieves better performance and generalization in deep learning compared to the sigmoid activation function.

- **Properties of the ReLu Function**: - The main idea behind the ReLu activation function is to perform a threshold operation to each input element where values less than zero are set to zero (figure 2).

Mathematically it is defined by:

$$f(x) = \max(0, x) = \begin{cases} x_i \; if \; x_i > 0 \\ 0 \; if \; x_i < 0 \end{cases}$$

Figure 4.3

And is shown as :

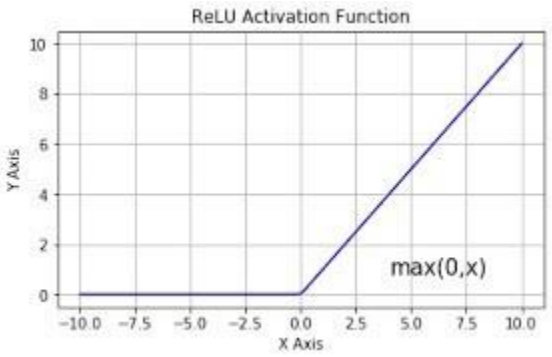


Figure 4.4

The Softmax Function

The Softmax function is used for prediction in multi-class models where it returns probabilities of each class in a group of different classes, with the target class having the highest probability. The calculated probabilities are then helpful in determining the target class for the given inputs.

Properties of the Softmax Function

- The Softmax function produces an output which is a range of values between 0 and 1, with the sum of the probabilities been equal to 1.
- The Softmax function is computed using the relationship:

$$f(x) = \frac{\exp(x_i)}{\sum_j \exp(x_i)}$$

Figure 4.5

- The main difference between the Sigmoid and Softmax functions is that Sigmoid is used in binary classification while the Softmax is used for multi-class tasks

Chapter 5 Results and Screenshot

```
ergo-pe-av (master) » ergo serve . --classes "clean, malicious"
[2019-05-22 23:27:47,310] (INFO) loading project /home/evilsocket/lab/ergo-pe-av ...
 * Serving Flask app "ergo.actions.serve" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
[2019-05-22 23:27:47,819] (INFO)  * Running on http://127.0.0.1:8080/ (Press CTRL+C to quit)
[2019-05-22 23:27:54,187] (INFO) 127.0.0.1 - -              23:27:54  "POST / HTTP/1.1" 200 -
```

```
classes » curl -F "x=@pe-malicious/af66d5db635537de043facf1580f9655fe441f03f82a7503272e32e3d8473af5.exe" "http://localhost:8080/"
{"clean":7.321406059190849e-08,"malicious":0.9999998807907104}
```

Figure 5.1

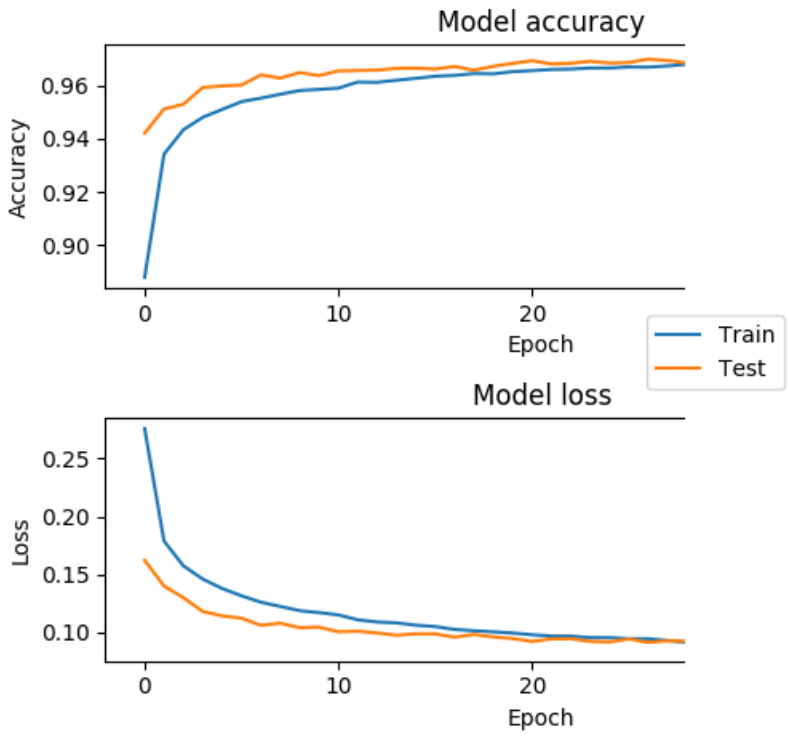


Figure 5.2

Chapter 6 Conclusions and Future works

There are various small tasks that can improve the accuracy of the existing malware detection and prevention systems and reduce the damage done by the malicious software to our system. As we move towards a totally digital dependent society our intrusion detection and prevention detection systems should become more and more accurate. To increase the accuracy of the detection and prevention systems we should try and gather a lot more information about the malwares like how they work, what are vulnerabilities that the exploit and make good use of the technological advances that we have today such as artificial intelligence . The method used in this system can be used to design a similar system which has to find malware iin even more complicated files.

To improve the proposed model for malware detection in executable files using machine learning we can add cross validation as it decreases the false positive and ture negative values so that our system becomes even more accurate.

Chapter 7 References

[1] Kirti Mathur, Saroj Hiranwal, A Survey on Techniques in Detection and Analyzing Malware Executables, International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 3, Issue 4, April 2013

[2] Vinod P. V.Laxmi,M.S.Gaur: Survey on Malware Detection Methods, 3rd Hackers‟ Workshop on Computer and Internet Security, Department of Computer Science and Engineering, Prabhu Goel Research Centre for Computer & Internet security, IIT, Kanpur, pp-74-79, March,2009.

[3] Pham Van Hung, An approach to fast malware classification with machine learning technique, Keio University, 5322 Endo Fujisawa Kanagawa 252-0882 JAPAN, 2011

[4] Ammar Ahmed E. Elhadi, Mohd Aizaini Maarof and Ahmed Hamza Osman, Malware detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph, American Journal of Applied Sciences 9 (3): 283-288, 2012, ISSN 1546-9239, 2012, Science Publications

[5] Ravindar Reddy Ravula. Classification of Malware using Reverse Engineering and Data mining Techniques, Thesis-Master of Science, University of Akron, August 2011.

[6] Robiah Y, Siti Rahayu S., Mohd Zaki M, Shahrin S., Faizal M. A., Marliza R.,A New Generic Taxonomy on Hybrid Malware Detection Technique, (IJCSIS)International Journal of Computer Science and Information Security, Vol. 5, No. 1, 2009