Spyware Detection and Prevention using Deep Learning AI for user applications

Mahesh V, Sumithra Devi K A

Abstract: ***A user application (Smartphone or personal computer's) play's an essential role in our daily life. As usage of smartphones and PC's keeps on increases, every day in one life, each and everyone uses to do every task in their daily life using smartphone or PCs to access, develop, store data's. By using this everyone access the data over the internet, the user's sensitive data's were shared during the payment process, private messages online, and using their personal data to access the resource to study etc. There is the possibility of occurring attacks to this user sensitive data's. Weakness on the construction of user application will allow the hacker or attacker to steal user information. Spyware is one type of attack that steel user sensitive information without user knowledge. The proposal states that the method and technique used to detect and prevent the user application from malicious attack using deep learning AI (Artificial intelligence).***

Keywords: ***AI (Artificial intelligence), internet, malicious code, Malware, Spyware, user application.***

I. INTRODUCTION

A user application contains a set of program that is designed and developed to do the particular task of the user. The rapid growth of computer application and use of the internet are increasing and thus increasing the chance of occurring malicious attacks. Weak structure of the user applications allows the hacker to steal user information. Hacker's an attacker uses a malicious code like a virus, spyware, malware, Trojan horse, ransom ware etc. to steal user sensitive information. And this is hard to find for a normal user where these malicious codes present in the system. This malicious code like spyware is a software code that was installed without user knowledge; it can be present in anywhere like on the login page, mail message, offer's link, win prize link etc. and some occurs due to slow internet connection, compromising on the network. There is much software to detect malicious code for free download to scan the code in your system.

a. DEEP LEARNING

Deep learning is one of the subfields of machine learning concerns with algorithm contend with a learning approach that use to gain some knowledge and the structure and function of the deep learning are artificial neural networks. A large neural network is known as deep learning, it is the notion of artificial neural networks.

By using this one can make the following,

- Make a learning algorithm easy and simple.
- Make an innovative approach in machine learning and Artificial intelligence.

II. ARTIFICIAL INTELLIGENCE

An artificial intelligence is a part of computer science that allows the computer and computer software to act like a human. In today's world most of the hacker and attacker uses artificial intelligence techniques and methods to attack user system and thus most of the security products like antivirus software use advanced learning technologies and artificial intelligence to detect the virus in the system. Artificial intelligence will provide an optimistic way to identify the intrusion pattern.

Combining Artificial intelligence and a part of machine learning (Deep Learning) will provide a new secure way to detect malicious code in the system.

a. Malicious code

A set of codes or a set of programs were developed in order to get user information without their knowledge then this is said to be an intrusion code or malicious code. There were different malicious code in the market namely malware, spyware, virus, ransomware etc.

b. Virus

A virus is a software code (malware), once it executed then it will automatically replicate its file into other file and insert its own file into other. There are different types of virus in the computer each acts according to its own and different from each other but the common damage virus does in user system is it will slow down the system and network by eating up its bandwidth.

c. Worms

Worms is a self-replicating code that is standalone software that automatically replicates the infected code to another system. Mostly it uses the operating system to damage the system as it is invisible to the user to find what happened. To spread infected code to another uninfected system it is a network channel medium to spread malicious code.

d. Wabbits

Unlike virus and worm, it does not affect the host system or documents instead it will replicate its malicious code on the local system. It will not pass or spread on the network. It is also called a computer bacterium and an example of this wabbits is a fork bomb.

e. Trojan Horses

It is normally called as "Trojan" that acts as a normal file to steal the user information, modify the files and observes the user actives.

Revised Manuscript Received on 30 January 2019.
***** Correspondence Author

Mahesh V, JAIN (Deemed-to- be-University), Dayananda Sagar Academy of Technology and Management,Bengaluru-560069, Karnataka, India

Dr. Sumithra Devi K A, Dayananda Sagar Academy of Technology and Management,Bengaluru-560069, Karnataka, India

f. Bug

Bugs are the set of code and commands that affect the source code or compiler.

A slender bug affects the behaviour of the program or an application and foremost chief bug's results in gathering sensitive data that are related to security issues like authentication, access privileges and steal user information.

g. Exploits

It took something to one's advantages i.e. it takes the advantages of the weakness and lack of security in some software. It is a chunk of data and command that will take the advantage of a bug in software to affect the behaviour of the software and further infect the whole system.

h. Backdoors

It is an illegal way of accessing user data that were bypassing the system on a network channel. A programmer develops backdoors for their purpose like troubleshooting attackers uses this to exploits the system using a virus, worm etc.

i. Spyware

Spyware is a software code that uses to steal the user data by recording secretly, this is to steal personal data's like password, bank details, card details etc.

j. cumware

Scumware is a code that affects the system from the internet by clicking some links without user knowledge.

k. Stealware

It is a malicious code that will gather user activities using HTTP cookies to redirect user site to the third-party side to steal money while transferring over the internet.

l. Parasiteware

This is a malicious code that will often display on the screen on the browser or the desktop by infecting the system by displaying advertisement.

m. Adware

It is an automatic delivery system that delivery's advertisement as a pop-up ad on websites and it will appear pop-up advertisement whenever the user clicks for some links to download songs, movies etc.

n. Rootkits

This software designed to access the system from anywhere without user involvement and security software. If a rootkit installed the system then this can access the system that is remotely present by changing the structure of a system, execute files to access information.

o. Keyloggers

It is code that designed to monitor the user activity what keys were pressed by the user on their keyboard. It will keep track of all the user activity without their knowledge.

p. Dialers

It is code that designed to steal phone numbers and attempt to dial calls from the user telephone lines at other location thus leads to increase of the user phone bill without their knowledge. The main aim is to steal the number on the modem so that they can use this telephone line to call from any other location.

q. Hoaxes

It is common problem that often experienced by the entire smartphone user as it is code that will display on the user screen that they were infected with some virus as a warning message.

III. DETECTION METHODS

Artificial Intelligence techniques used in anti-virus detection Detection methods are divided into signature-based and behaviour-based. A basic malware analysis type namely static and dynamic analysis, these types helps to understand the malicious code.

Static: Perform statically without the file execution.

Dynamic: Shown on files when it is executing example virtual machine.

A static analysis on detection will rely on certain tools to analyse and provide some information on how to protect from them. The main goal of this is to discover all the possible malicious threat. Repeatedly searching for the code allows the programmer to see all possible way of malicious code. Static analysis is more save and less time consuming than dynamic as in static there will no file execution, so there will be no serious damage to the system.

Do to this static analysis is not used on the real world as real-world data are dynamic.

a. Dynamic analysis

In dynamic analysis file are executed as its behaviour of the file is closely monitored and once file executed its properties and purposes to infer from that executed file. It is run on the virtual environment and this will allows finding the all possible behaviour and it is fast compared to static analysis.

Most popular method or approach to detect malware's in AI is signature-based and heuristic-based analysis.

The signature-based analysis is a static approach that uses a predefined signature to detect malicious code. This signature can be a file fingerprint. These fingerprints are metadata about the file, a static string and an encrypted data like MD5 or SHA 1 hashes. If the file arrives at the system then this file was analyzed for the signature match by anti-virus software application and if once the match matched then an alert message will display as state the file is doubtful. Based on the hash values this will help to detect the malicious code on the file. Hacker and attacker are developing a malicious code in such a way that will change its code so that it will not match with the signature. Malicious code features were referred to polymorphism as this will not allow detecting the malicious code using signature-based detection. The main disadvantage of the signature-based techniques is that if a new code of malicious code occurs then the system will not recognize until a new signature to this is created.

This limitation motivates the vendor to create another type that will able to create to detect behaviour-based called heuristics-based analysis. Unlike signature-based analysis, in heuristic-based analysis once the file execution then it will observe the actual behaviour of the malware. Once the file tries to modify the file so that signature-based analysis will not detect it tries to change the file, changes the host file, keys and some connection. Even this will be detected by the system but this combination of events will create the suspiciousness that the file contains malicious code. Some threshold level on this will provides the level of risk on the system and produces the alert according to the level of risk.

The best accuracy level depends on the implementation on the virtual environment.

This approach is more time to consume and faster. It will detect a full family of the malicious code as well as zero-day attacks.

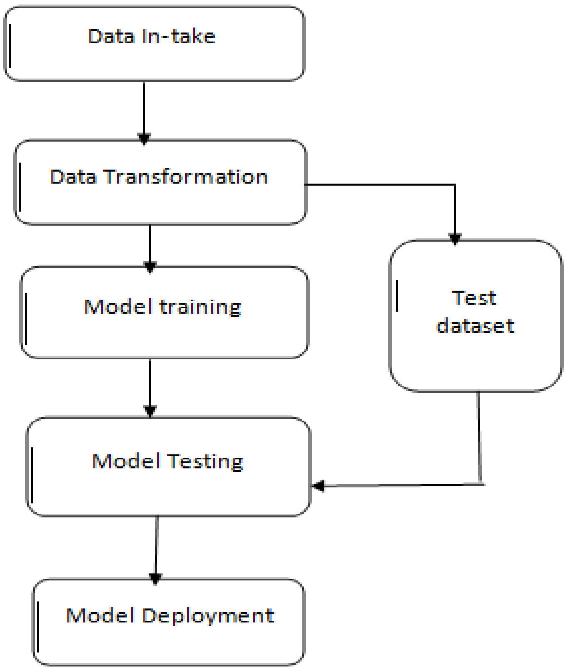


Figure 1. Dataflow of malware detection

Figure 1. shows the dataflow of malware detection and it is described as follows:

Data Intake: A dataset is loaded and stored in the memory.

Data transformation: the foremost of the model is to clear the missing data on the dataset. Dataset was cleaned and normalized using some methods or algorithms. The data arranged on some order so that it is used to predict accurately. For this, features are extracted, these were divided into train set and test dataset.

Model Training: This has an algorithm so that model is created.

Model Testing: the model is tested using test data, training data and test dataset will use to build a model and the result of this step will be used to create the module.

Model deployment: Best model is developed for accurate prediction.

Deep learning

The main concept behind deep learning is a neural network and this is considered as a brain of neural networks. The concept of neural network has a simple process with multiple parallel processes to execute output. This is compared with the human brain as it works as them like non-linear parallel information-processing. This helps non-linear parallel information processing system to rapidly compute the result.

- Single-layer perceptron (SLP)
- Multiple perceptions (MLP)

Single layer perception (SLP)

This is simple perceptron type with a single layer of weights that connecting both inputs and outputs in the free forward network (One direction network).

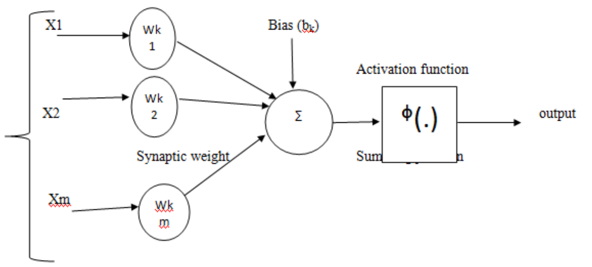


Figure 2. Single layer perceptron

Figure 2. show that m weights are a set of synapses and connecting links on a single layer and one more layer in the network. Adder function accepts the input and multiplied with the synaptic connection.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

Bias bkis an affine transformation from adder function to output.

$$v_k = u_k + b_k$$

Multiple layer perceptron (MLP)

Unlike single layer perceptron, it has several layers of connected function to predict the result. Figure 3. Shows the multiple layer perceptron. It is known as a feed-forward neural network. Same as a single layer but it has one or more hidden layer with input and output. Each layer has several neurons that will intersect with each other with weights links. The number of input and output depends on the number of features in the dataset and number of classes in the dataset.

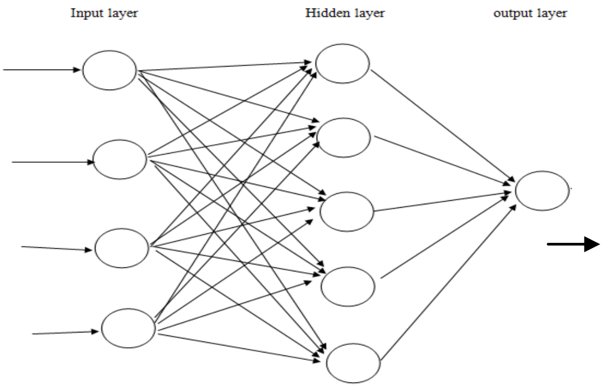


Figure 3. Multiple layer perceptron

A separate input is given on the input layer and the hidden layer will do some synaptic function to compute the result and final layer output will produce the result.

XG Boost Algorithm

It is a gradient boosting algorithm using in decision tree classification to get and enhance the performance and speed. As data's were huge and difficult to manage on decision tree XGBoost will overcome the managing of huge data in decision tree.

So using decision tree classification along with XGBoost will give a high accuracy. XGBoost is an ideal concept that are used in many field to find accuracy of the system. It works on linear and tree algorithm and applicable only when a dataset contains value i.e. it works on numeric vector, it makes the model faster and does a parallel work in a single machine.

IV. SYSTEM ARCHITECTURE

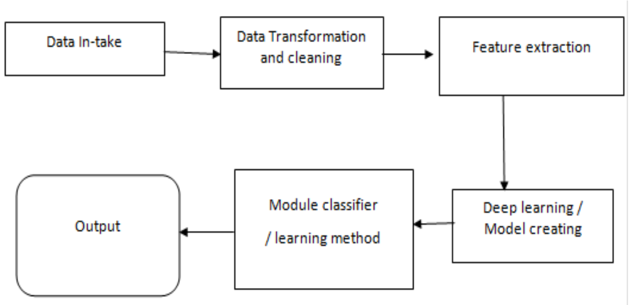

Data In-take
Data Transformation and cleaning
Feature extraction
Deep learning / Model creating
Module classifier / learning method
Output

Figure 4. System architecture

Figure 4. shows the system architecture of the malware detection. The dataset is taken to clean as the missing features will affect the result so it has to be cleaned before the actual process starts. Once the data is cleaned then the data is prepared to store for feature selection this selection will separate the main and important feature to detect the malicious code here the features were applied to neural network concept deep learning to get an accurate result that to find the suspicious activities on the system. Learning method will also find the malicious code as a secondary layer to identify the malicious code by the heuristic method. The heuristic method will find the behaviour as well as the malicious family and changes in their activities that indicate the malicious code. Finally, the result of the module will give the message that the system is infected with virus and risk of the malicious code.

V. RESULT

The major challenges in detecting malware is it has a huge number of data's and files to evaluate in order to find malicious intent in short period. The datasets are available in https://zeltser.com/malware-sample-sources/. These data sets are available for free and only registration is required. Evaluating this using a polymorphism to malicious components i.e. all the malware belongs to same family as most will give same malicious behavior. For effective and easy calculation to analysis malicious code all these data were grouped according to the family and these are criteria applied to detect the file that contain malicious code.

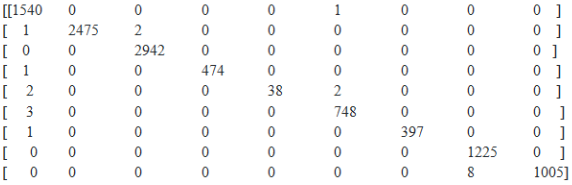
Data

All the data's were separated into different families according to its behavior. A known malware files are separated as 9 families.

Ramnit
Lollipop
Kelihos_ver3
Vundo
Simda
Tracur
Kelihos_ver1
Obfuscator.ACY
Gatak.

All these files were encrypted and features were extracted using chi-square tests to find independent class labels. Best features were selected between 10 to 50% of its original data.Tested using extra Trees Classifier and cross validation Original keyword count for 1006 features = 0.034, 10% of keyword count for 202 features contains image features = 0.0174, 20% of keyword count for 402 features has .0164, 30% of keyword count has a feature of 623 that has a multiclass value of 0.0133. An overall accuracy value of the above features is 0.9978.

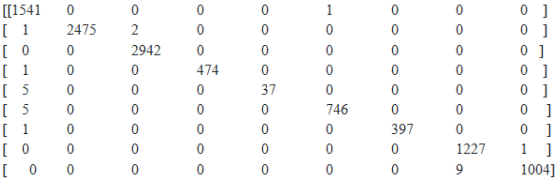
Confusion matrix



```
[[1540   0      0      0      0      1      0      0      0  ]
 [  1    2475   2      0      0      0      0      0      0  ]
 [  0    0      2942   0      0      0      0      0      0 ]
 [  1    0      0      474    0      0      0      0      0  ]
 [   2   0      0      0      38     2      0      0      0  ]
 [   3   0      0      0      0      748    0      0      0   ]
 [   1   0      0      0      0      0      397    0      0   ]
 [    0  0      0      0      0      0      0      1225   0   ]
 [    0  0      0      0      0      0      0      8      1005]
```

The next step is testing of the module using 40% of features with ExtraTreesClassifiers with more than 10 thousand samples with 823 features along with cross validation gives 0.0135 and accuracy value is 0.9976

Confusion matrix



```
[[1541   0      0      0      0      1      0      0      0  ]
 [  1    2475   2      0      0      0      0      0      0  ]
 [  0    0      2942   0      0      0      0      0      0 ]
 [  1    0      0      474    0      0      0      0      0  ]
 [  5    0      0      0      37     0      0      0      0  ]
 [  5    0      0      0      0      746    0      0      0   ]
 [  1    0      0      0      0      0      397    0      0   ]
 [  0    0      0      0      0      0      0      1227   1   ]
 [    0  0      0      0      0      0      0      9      1004]
```

This shows that ExtraTreeClasifier gives an optimal solution in 30% of data with all the image features, entropy and size. But adding further more feature needs additional improvement i.e. additional classifiers.

A new model has to be selected if the feature contains image, entropy, and size of a file and call graph files. For this aGridsearchCV is used to find optimal classifier.XGBoost is a gradient boosting method to find optimal classifier. One or more classifier can be used to find the optimal result. Simple linear regression or decision tree classifier is used.

```
Xi -→x
Yi →y
ei = 0
n→len(Yi)
pre_df = 0
fori in range(30):
tree = Decisiontree(Xi,Yi)
tree.f_split(0)
r→np.where(Xi == tree_split)[][]
Index_l = np.where(Xi <= tree.split)[]
Index_r = np.where(Xi >tree.split)[]
error→e1= y – y_predict1
new_model_error→ e1_predict
y_predict2=y_predict1+e1_predict
e2=y-y_predict2
# $i^{th}$ decision tree
predict = np.zeros(n)
```

```
np.put(predict, index_l, np.repeat(np.mean(Yi[index_l])), r)
np.put(predict, index_r, np.repeat(np.mean(Yi[index_r]), n-r))
predict = predict[:, null]
predf = predf + predict
ei = y – predf
Yi = ei  // final prediction
```

XGBoost for 30% features 0.0080 and accuracy 0.9981

Confusion matrix

```
[[1540   0     0     0     0     1     0     0     0  ]
 [  2    2475  0     1     0     0     0     0     0  ]
 [  0    0     2941  0     0     0     1     0     0  ]
 [  0    0     0     474   0     1     0     0     0  ]
 [  1    0     0     0     41    0     0     0     0  ]
 [  4    0     0     0     1     746   0     0     0  ]
 [  0    0     0     0     0     0     398   0     0  ]
 [  0    0     0     0     0     0     0     1227  1  ]
 [   0   0     0     0     0     0     0     8     1005]]
```

The accuracy occurred from XGBoost and ExtraTreeClassifier shows that XGBoost has more accuracy (0.9981 i.e. 99.81%) whereas accuracy for ExtraTreeClassifier (0.9976 i.e. 99.76%) is achieved.

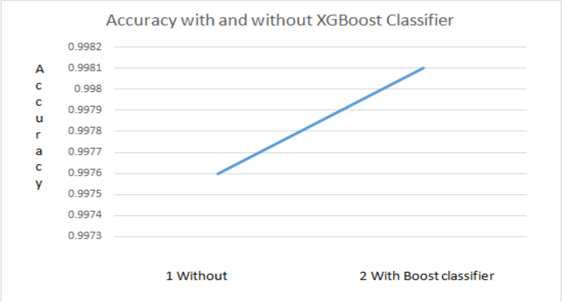


Figure 5. Accuracy with and without XGBoost Classifier

Figure 5. shows that the accuracy calculated using the Extra Tree classifier. The graph illustrates the accuracy of the data with and without applying the XGboost to show the accuracy of the classifier used. The accuracy occurred from XGBoost and ExtraTreeClassifier shows that XGBoost has more accuracy (0.9981i.e. 99.81%) whereas accuracy for ExtraTreeClassifier (0.9976 i.e. 99.76%) is achieved.

VI. CONCLUSION

Deep learning is the concept used here will improve the quality of detecting the malicious code and the threshold of the level of risk will helps to prevent the malicious activities. Using both artificial intelligence and deep learning concept might increase the efficiency of the malware detection comparing other state of art methods used in previous decades. The datasets are downloaded from the website and it is available for free. Only registration is required for downloading it. Malicious code is detected using many methods but using machine learning, it gets an accuracy of 0.98 in decision tree classifier. But an additional feature on decision tree take time and gives low accuracy. XGBoostclassifier on the model will do extra classification to find high accuracy. This classifier shows accuracy for 0.9981% accuracy.

REFERENCES

1. Sugandhasharma [2018], ”Fighting Virus and Malware with Artificial Intelligence” Available at https://www.insightssuccess.com/fighting-virus-and-malware-with-artificial-intelligence/ [Accessed on 24 July 2018]
2. Jason Brownlee [2016], “What is Deep Learning?” Available at: https://machinelearningmastery.com/what-is-deep-learning/ Accessed on 24 July 2018.
3. P. Bisht V. Venkatakrishnan "Xss-guard: precise dynamic prevention of cross-site scripting attacks" in Detection of Intrusions and Malware and Vulnerability Assessment Springer pp. 23-43 2008.
4. P. Laskov N. Šrndić "Static detection of malicious javascript-bearing pdf documents" Proceedings of the 27th Annual Computer Security Applications Conference. pp. 373-382 2011.
5. KeterynaChumachenko [2017], ”Machine Learning Methods for Malware Detection and Classification” Processig of Kaakkois-Suomenammattikorkeakoulu, University of Applied Science in 2017.
6. Jinpei Yan, Yong Qi and Qifan Rao [2018],”Detecting malware with an ensemble method based on deep neural network” Proceeding on Security and Communication Networks Volume 2018, Article ID 7247095, 16 pages https://doi.org/10.1155/2018/7247095 .
7. Karishma Pandey, Madhura Naik, Junaid Qamar ,Mahendra Patil (2015), Spyware Detection using Data Mining, International Journal of Engineering and Techniques.
8. Ms. Milan Jain, Ms. Punam Bajaj (2014), Malicious Code Detection through Data Mining Techniques, International Journal of Computer Science & Engineering Technology (IJCSET)
9. Saba Arshad, Abid Khan, Munam Ali Shah, Mansoor Ahmed (2016), Android Malware Detection & Protection: A Survey, (IJACSA) International Journal of Advanced Computer Science and Applications.
10. Z. Bakdash, Steve Hutchinson, Erin G. Zaroukian, Laura R. Marusich, Saravanan Thirumuruganathan , Charmaine Sample, Blaine Hoffman , and Gautam Das, Malware in the future? forecasting of analyst detection of cyber events Jonathan, University of Texas Dallas Dallas, TX, USA
11. Niklas Lavesson, Martin Boldt, Paul Davidsson, Andreas Jacobsson (2009), Learning to detect spyware using end user license agreements, Springer.
12. Androutsopoulos I, Paliouras G, Karkaletsis V, Sakkis G, Spyropoulos CD, Stamatopoulos P (2000), Learning to filter spam E-mail: a comparison of a naive bayesian and a memory-based approach.
13. Kirti Mathur (2013), A Survey on Techniques in Detection and Analyzing Malware Executables, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4.
14. M. G. Schultz, E. Eskin, E. Zadok and S. J. Stolfo (2001), Data Mining Methods for Detection of New Malicious Executables, Proceedings of the 2001 IEEE Symposium on Security and Privacy, IEEE Computer Society.
15. Parisa Bahraminikoo (2012),Utilization Data Mining to Detect Spyware, IOSR Journal of Computer Engineering (IOSRJCE), Volume 4, Issue 3.
16. Datasets available at: https://zeltser.com/malware-sample-sources/