2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)
Mody University of Science and Technology, Lakshmangarh, Feb 21-22, 2020

Malware Detection & Classification using Machine Learning

Sunita Choudhary
Computer Science and Engineering
Mody University of Science and Technology
Laxmangarh, India
sunitadangi@gmail.com

Anand Sharma
Computer Science and Engineering
Mody University of Science and Technology
Laxmangarh, India
anand_glee@yahoo.co.in

***Abstract*— With fast turn of events and development of the web, malware is one of major digital dangers nowadays. Henceforth, malware detection is an important factor in the security of computer systems. Nowadays, attackers generally design polymeric malware [1], it is usually a type of malware [2] that continuously changes its recognizable feature to fool detection techniques that uses typical signature based methods [3]. That is why the need for Machine Learning based detection arises. In this work, we are going to obtain behavioral-pattern that may be achieved through static or dynamic analysis, afterward we can apply dissimilar ML techniques to identify whether it's malware or not. Behavioral based Detection methods [4] will be discussed to take advantage from ML algorithms so as to frame social-based malware recognition and classification model.**

Keywords*— *Machine Learning, Malware detection, KNN, SVM.

I. INTRODUCTION

Malware is any product tenaciously intended to make harm a PC, server, customer or PC organization. With the fast advancement and development of the web, malware has turned out to be one of the major digital dangers in nowadays. In the year 2017, a cybersecurity and anti-virus provider like Kaspersky Labs defined malware as "a kind of PC program planned to taint an authentic client's PC and perpetrate hurt on it in different manners."

As the decent variety of malware is expanding nowadays, antivirus tools are not capable of fulfilling the need of protection and results in millions of hosts being hacked. In addition to that, the skills required for malware development is decreasing due to high availability of attacking tools on the Internet. According to data of AV-TEST Institute, they reported over three lac fifty thousand novel malware (malicious projects) and PUA (Potentially Unwanted Applications) consistently. Therefore, to protect computer system from malware is one significant tasks of cybersecurity for a single user as well as for businesses because even a single attack can result in huge information and financial loss.

A. Types of Malware

Malware can be isolated into a few classes relying upon its purpose. The classes are as per the following:

- *Adware*: It is the slightest risky and the most rewarding malware, it shows advertisements on PC.
- *Spyware*: As it implies from the name, the malware that uses for spying. Some run of the mill activities of spyware incorporate following inquiry history to send customized advertisements, following exercises to offer them to the outsiders in this way.
- *Virus*: *This is the most straightforward type of the software. It is basically any bit of programming that is stacked and propelled without client's authorization while replicating itself or contaminating changing other programming. Regularly this is spread by sharing records or programming between PCs.*
- *Worm*: *It is a program that imitates itself and obliterates data and records on the PC.*
- *Trojan*: *It is a kind of malicious code and computer software to look authorized but can control the system or machine.*
- *RootKit*: *An assortment of vindictive programming created to enable access to a framework or on particular area of the system.*
- *Backdoors*: *It is a method to convert the bypassing normal authentication and encryption.*
- *Keyloggers*: *It is totally depend on Keyboard working style like the action of keyboard typically covertly unaware their actions are being method and monitored.*
- *Ransomware*: *This is malware software but extreme use of this software is for Accounts section like access to system until a sum of money is paid.*
- *Browser Hijacker*: *It is a type of undesirable programming that changes a program's setting without client authorization, to infuse the profitless promoting into program.*

B. Malware Discovery Investigation Procedures

All malware discovery procedures can be partitioned into mark based and conduct based strategies. How about we examine a few procedures for the examination.

- *Static Method*: *A static strategy for examination of malware depends on pre- characterized marks. These can be document fingerprints, e.g. file metadata, static strings, MD5 or SHA1 hashes.*
- *Dynamic Method*: *A dynamic method of analysis and for resultant of malware relies for the change according to time and classifies the malware-based approach on the hand behalf of the time approach.*

978-1-7281-1420-0/20/$31.00 ©2020 IEEE

II. METHODOLOGY

In the event that we study the writing of malware-location dependent on AI strategies, we will locate that many AI methods are discovered reasonable for recognizable proof of malware classification and identification, a portion of the systems that have been utilized are Support Vector Machines (SVM), Random Forest and unsupervised techniques like k-means has been used to cluster or group the malwares on the basis of their behavior but key challenge with clustering is to adopt the optimal number of clusters of groups. This section will cover some research by the renowned researchers those have used these AI strategies for malware identification and arrangement into malware families.

A. *Data Mining and Machine Learning for Detecting & Classifying Malwares Beats Traditional Signature-based Detection*

In year 2001, M. Schultz et al discussed about the idea of malware detection with machine learning and data mining techniques [5]. Results shown in the paper are very much relatively better than the old traditional signature-based detection methods.

a) Data acquisition: M. Schultz et al. in 2001 utilizes dataset of 4,267 programs fragmented into 1,001 clean programs and 3,266 malicious binaries.

b) Information Preprocessing and Feature choice: There are no copy programs in the dataset and each model in the set is marked either malignant or generous with the assistance of an antivirus scanner. Furthermore, it additionally has three sorts of static highlights for preparing AI models so as to recognize and group malware. These kinds of highlights are as per the following:

- *Portable Executable (PE):* A library inside Bin-Utils [6] that was libBFD [7] is used to remove data from the versatile executable header. A portion of the highlights like size of document, names of powerfully connected libraries and progressively connected libraries capacity calls were gotten from the PE header. A portion of different highlights like rundown of DLLs were utilized by double and furthermore tally framework calls inside each powerfully connected libraries.

- *String Sequences:* Those highlights using string were furthermore isolated depending upon how these strings were encoded inside the records. In view of the experimentation, M. Schultz et. al in 2001 found that cord designs in every single spotless document were comparable and this made it not quite the same as malware records as malware records had various examples. Along these lines of location isn't not quite the same as signature- based detection. In any case, M. Schultz et. al utilized it as a component for the classification model. The significant issue with these kinds of features is that they are lacking in robustness as utilizing shrewd procedures they can without much of a stretch be changed, so M. Schultz et. al likewise utilized sequence of Bytes as another component.

- *Sequence of Bytes:* Utilized n-gram set up together techniques [8] as for executable records, using n-gram and device called hexdump hexadecimal archives can be gained from the twofold records. On the off chance that we separate these features and different features, M. Schultz et. al found that Sequences of bytes were generally important as it had machine code executable when diverged from asset data, for instance, conservative executable highlights.

c) Machine learning calculations

- *Ripper algorithm [9]:* It is a standard based student that assembles a lot of guidelines that recognize the classes while limiting the measure of blunder. This calculation creates an identification model made out of asset decides that is worked to distinguish future instances of malignant executables records. This calculation utilizes libBFD data as highlights.

- *Naive Bayes [10]:* Naive Bayes is a direct framework for building categorizers: models that dispense class names to give models, addressed as vectors of highlight esteems, where the class imprints are drawn from some constrained set. In various valuable applications, parameter estimation for credulous Bayes models uses the methodology for most extraordinary probability. We can utilize every one of our highlights to group malware or not. For our needs, we can utilize it to discover the probability of being malware given every one of the highlights.

- *Multi-Naïve Bayes:* The Multinomial Naive Bayes categorizer is appropriate for order with discrete highlights e.g. Word counts for content arrangement. The multinomial appropriation ordinarily requires whole number component checks. By applying multinomial distribution on the feature set which seems to be a reasonable reason to classify malware or not.

B. *Data Mining and Machine Learning for Detecting & Classifying Malwares Beats Traditional Behavioral-based Detection*

In the year 2010, Firdausi et al. 2010[11] tries various different ML approaches for malware discovery and summarizes the experimental results in a nicely manner. Firdausi et al. 2010 uses sandbox condition so as to dissect the malware and collecting behavioral data automatically.

a) Data acquisition: A sum of 220 one of a kind malware tests are gathered. Additionally gathers clean framework documents from a perfect establishment from framework records of Windows XP Professional. What's more, a report is created by leading conduct observing as for malware documents and clean records.

b) Data Preprocessing and Feature choice: Firdausi et al. applies information preprocessing for highlight creation and choice, as pursues:

- XML report records are acquired to get the significant and significant properties.

- After choosing the important qualities, an information structure is made to store properties.

- The made information structure is utilized to diverge from XML report record and with tally the presence of each word in the information structure double weight and word recurrence weight.

- Sparse vector model and Attribute Relation File Format (ARFF) are made with each report.

c) Machine Learning algorithms: The ML algorithms are applied on each mentioned data-set. The model that is applied by the author Firdausi et al. are Multi-layer perceptron (MLP), Decision Tree and SVM.

- *SVM [12]:* SVM assembles a hyperplane or set of hyperplanes in a high or boundless dimensional space, which can be used for arrangement. A decent partition is practiced by the hyperplane that has the greatest partition to the nearest getting ready data motivation behind any class (called useful edge), since when everything is said in done greater edge, lower speculation mistake of characterizer.
- *KNN [13]:* KNN may be utilized for arrangement and relapse issues. Although in our problem set, it is used to classify malwares in view of those k preparing models or instances, which are in majority with respect to the input i.e. which class it closely associated with. There are only two classes which the input can be associated with one is malware detected or not. KNN basically using for Classification techniques but in Malware Detection many terminally says and according to our study also its evaluate depends on the "ease to interpret output, calculation time and predictive power".
- *Naïve Bayes [14]:* As we discussed earlier, for malware detection Naive Bayes categorizer may be utilized to characterize or distinguish malware dependent on the conditional probability.
- *J48 Decision Tree [15]:* Decision tree is a structure that consolidates a root hub, branches and leaf hubs. Each hub connotes a test quality, each branch implies the consequence of the test, and each leaf hub holds class name. Decision tree doesn't require any area learning yet it utilizes the idea of data entropy and it is anything but difficult to fathom, and the learning and characterization steps of choice tree are straightforward and quick.
- *Multi-layer Perceptron [16]:* For malware detection, multi-layer perceptron can be used. Basically Perceptron multiplies with weights and adds bias in one layer only. It is a linear categorizer also. A Multiple Layer Perceptron can be thought of, therefore, as deep artificial neural network.

TABLE I. PERFORMANCE METRICS RESULTS

Profile Type	TP (True Positives)	TN (True Negatives)	FP (False Positives)	FN (False Negatives)	Detection Rates	False Positive Rate	Overall Accuracy
Signature Method -Bytes	1101	10001	0	2159	33.69%	0%	49.31%
RIPPER							
-DLLs used	21	188	20	17	57.90%	9.19%	84.01%
-DLL function Calls	26	189	15	12	72.10%	8.01%	88.99%
-DLLs with counted function calls	19	194	12	19	53.11%	5.29%	88.79%
Naïve Bayes -Strings	3181	959	39	90	96.99%	3.79%	98.00%
Multi-Naïve Bayes -Bytes	3189	939	59	75	97.54%	5.99%	97.01%

III. RESULTS

A. Performance Metrics

These are the aftereffects of ordering new pernicious projects sorted out by calculation and highlight. Multi-Naïve Bayes utilizing Bytes had the most elevated Detection Rate, and Signature Method with strings had the least False Positive Rate. Highest generally precision was the Naive Bayes calculation with strings. Note that the recognition rate for the mark-based strategies are lower than the information mining techniques. As per Firdausi et. al, the experimental results are as follows:

TABLE II. PERFORMANCE METRICS RESULTS (BINARY, NO FEATURE SELECTION)

Categorizer	TPR	FPR	PPV	ACC
KNN	82.1%	8.2%	91.7%	87.6%
Naïve Bayes	59.2%	13.1%	94.3%	66.5%
SVM	90.9%	8.5%	91.3%	92.1%
J48	91.1%	3.9%	94.9%	94.7%

TABLE III. PERFORMANCE METRICS RESULTS (TERM FREQUENCY, NO FEATURE SELECTION)

Categorizer	TPR	FPR	PPV	ACC
KNN	87.1%	8.9%	90.5%	89.2%
Naïve Bayes	57.1%	23.1%	87.4%	63.1%
SVM	91.5%	7.4%	91.8%	92.0%
J48	96.1%	2.3%	96.9%	95.9%

TABLE IV. PERFORMANCE METRICS RESULTS (BINARY, FEATURE SELECTION)

Categorizer	TPR	FPR	PPV	ACC
KNN	93.4%	8.2%	91.0%	93.2%
Naïve Bayes	93.1%	9.3%	88%	92.3%
SVM	93.4%	8.2%	91.0%	93.8%
J48	93.1%	9.1%	88%	92.3%
MLP	93.9%	12.1%	87.0%	90.9%

TABLE V. PERFORMANCE METRICS RESULTS (TERM FREQUENCY, FEATURE SELECTION)

Categorizer	TPR	FPR	PPV	ACC
KNN	93.9%	8.2%	89.9%	93.1%
Naïve Bayes	59.1%	18.9%	86.9%	64.9%
SVM	93.2%	09.9%	88.6%	89.8%
J48	93.6%	5.7%	95.0%	93.8%
MLP	93.5%	5.9%	94.1%	93.9%

The exhibition correlation of five distinct characterizer was shown. The general optimum introduction was practiced by J48 utilizing the expression recurrence weight without

highlight assurance educational record, using a survey (certified positive pace) of 96.1%, a false positive pace of 24% an exactness (positive perceptive estimation) of 96.8%, and an exactness of 96.8%. The assessment of the tests and preliminary outcomes assumed that this confirmation of-thought is exceptionally fruitful and capable in recognizing malware.

IV. CONCLUSION

According to the study and observation we can see the potential of machine learning algorithm over traditional methods that are used by anti- virus tools for malware detection. And we have also discussed about different machine learning algorithms that can be of great help in detecting malware as with the quick development and advancement of web, malware is major threat.

REFERENCES

[1] K. Tang, M. T. Zhou, Z. Z -H, "An enhanced automated signature generation algorithm for polymorphic malware detection", JESTC, vol. 8, pp. 114-121, 2010.

[2] J. Landage, Prof. M. P. Wankhade, “Malware and Malware Detection Techniques: A Survey”, IJERT, Vol. 2, Issue 12, pp. 1-8, Dec. 2013.

[3] M. F. B. Abbas, T. Srikanthan, "Low-complexity signature-based Malware detection for IoT devices", Proc. Appl. Techn. Inf. Secur., pp. 181- 189, Jun. 2017.

[4] H. S. Galal, Y. B. Mahdy, M. A. Atiea, "Behavior-based features model for Malware detection", JCVHT, vol. 12, pp. 59-67, May 2016.

[5] M. Schultz, E. Eskin, F. Zadok, and S. Stolfo,” Data Mining Methods for Detection of New Malicious Executables”, Proceedings of 2001 IEEE Symposium on Security and Privacy.

[6] W.-C. Wu, S.-H. Hung, "Droid Dolphin: A dynamic android malware detection framework using big data and machine learning", Proceeding Conference Research Adaptive Convergent System, pp. 247-252, 2014.

[7] F. Wei, Y. Li, S. Roy, X. Ou, W. Zhou, "Deep ground truth analysis of current Android malware", Proceeding International Conference Detection Intrusions Malware Vulnerability Assessment, pp. 252-276, 2017.

[8] Abou-Assaleh, T., Cercone, N., Keselj, V., &Sweidan, R. (n.d.).N-gram-based detection of new malicious code. Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004.

[9] R. Jabri, B. Ibrahim, "Phishing Web. Detection Using Data Mining Classification", Society for Science and Education United Kingdom, vol. 3, no. 4, pp. 42- 51, 2015.

[10] Ömer Faruk Arar, Kürşat Ayan, "A Feature Dependent Naive Bayes Approach and Its Application to the Software Defect Prediction Problem", Applied Soft Computing, 2017.

[11] I. Firdausi, C. Lim, A. Erwin,”analysis of ML Techniques Used in Behavior Based Malware Detection", Proc. of 2nd Int. Conf. on Adv. in Computing, Control and Telecom. Tech., (ACT)2010.

[12] M. Alazab, "Profiling and classifying the behavior of malicious codes", Journal of System Software, vol. 100, pp. 91-102, Feb. 2015.

[13] R. Agrawal, "K-Nearest Neighbor for Uncertain Data", International Journal of Computer Applications (0975–8887), vol. 105, no. 11, pp. 13-16, 2014.

[14] K. Gautam, V. K. Jain, and S. S. Verma, “A Survey on Neural Network for Vehicular Communication”, Mody University International Journal of Computing and Engineering Research, vol. 3, no. 2, pp. 59-63. 2019.

[15] Quinlan J R. "Induction of decision tree", Machine Learning, 1986, 1: 81~106.

[16] Y. LeCun, Y. Bengio, G. Hinton, "Deep learning", Nature, Vol. 521, pp. 436-444, 2015.