A Realistic Approach to Detect Malware using Binary Analysis and Machine Learning

Amit Sahu[1], Prachi Parwar[2], Deepak Agrawal[3]

[1]*Research Scholar, Takshshila Institute of Engineering and Technology, MadhyaPradesh India*
[2]*Professor, Takshshila Institute of Engineering and Technology, MadhyaPradesh India*
[3]*Hod, Takshshila Institute of Engineering and Technology, MadhyaPradesh India*

Abstract- *Malware is a computer program that is aimed to breach computers without owner's authorization. Bulk of malwares are increasing every year. Antivirus providing companies are receiving millions of malwares daily, so discovery of malwares is difficult and time consuming task. There are many malwares detection techniques like signature based detection, behavior based detection and machine learning based techniques, etc. In this paper .N-gram approach is used in opcode and random forest approach is used for classification of the malware. Initially opcodes are taken out from the software in the tri-gram form. Features extraction is used to find many features along with the PE file attributes and features selection is performed on the obtained features to eliminate the redundant data .Appropriate features are selected by using our feature selection algorithm which works over the threshold value. . The investigational outcomes shows that our method can attain higher classification accuracy, fast discovery, little power consumption and flexibility for easy functionality improvement to adapt to fresh malware samples.*
we suggest a novel method to discover malwares based on the frequency of opcodes in the portable executable file. This research applied machine learning algorithm to find for malwares and got highest success rate.

Keywords- Malware Detection, Opcode, N-gram, Machine Learning, Malware Analysis, Data Mining, Random Forest.

I. INTRODUCTION

The number of malware is increasing day by day, and it is creating many victims.

Behaviour-based detection can easily disclose the given variants because the malware samples in the similar family are born to complete the same attack goals. It means malware in the identical family is probable to have similar function, call sequences. Many researches that try to detect malware on the basis of similarity, only use some simple features such as permission [9] and call graph [10]. As it is tough to discovery out a relation between features, it is not well fit to recent malware clustering. Especially, permission information is a no longer useful feature because every app require lots of permissions. If detection systems only use one or small sort of features, then it will give yields biased results since it observes only single side of malware. To study whole characteristics of malware, a hybrid method that combines various features from the signature analysis and behaviour analysis system.

II. MALWARE

Malware is dangerous malicious computer program or software that performs dangerous actions on computer [11]. Many malware attacks occur that pose thoughtful security threats to business organisations and everyday users.

III. MALWARE DETECTION TECHNIQUES

Malware detection techniques employed in systems as ways of finding malware in the system. The purpose is to expose the malware to lessen their effects or to enable the system recovery processes. There are divided into three groups; Anomaly based detection, Signature based detection and Specification based detection. All the three can be built from using results achieved from either static, dynamic or hybrid (combination of both static and dynamic) analysis.

A. Anomaly-based Detection

Anomaly-based is a malware mitigation tactic which aims to identify existence of malware within a system before humans [27]. With anomaly-based detection, baseline metrics of the systems are measured over a given period and then saved in a database. After that the system is then continuously observed for any momentous deviation from the baseline.

B. Specification based detection

With specification based detection, expected system behaviour and properties is catalogued, any deviation from the specifications results in an alert about possibility of system compromise [29]. This approach in theory can be used to detect unknown attacks since any deviation from the known behaviour triggers an alert. However, achieving a perfect specification based detection system is an impossible task.

C. Signature based detection

Malware signatures are utilised by malware detection engines to detect occurrence of malware on a system, by comparison of the suspected malware and the malware signature database. Limitations in their inability to detect new patterns of attacks, and challenges when dealing with obfuscated malware, signatures decrease work to be done by malware analyst by removing already know samples of malware.

1) Hash Signatures
With hash signatures, hash values are generated for selected features on malware samples. Once the hashes are generated they are uploaded into a database for malware detection systems to check against future infections [31]. A weakness of hash signature is that each slight variation in the file structure or content can lead to a completely different hash which results in bloated signature database [32]. Hash signatures are however used in triaging and clustering malware as discussed later in this paper.

2) Byte Signatures
Another option for signature generation involves analysis of sequence of bytes of the malware. This analysis operates on the assumption that malware from the same family, will possess similar byte sequences enabling malware detection programs to detect them [31]. Evaluation of byte sequences can potentially enable scanning systems to detect new classes of malware and zero-days [32]. This is possible since malware from the same author can, potentially, possess traits common to both making it easier to detect. Such features can range from same encryption code for polymorphic malware, same destination IP address and utilization of same software resources.

3) Binary Diffing
Binary diffing involves analysis of code to identify sections similar to known malware code [33]. If there is greater likeness the code is marked as malware and then used as a digital signature. It would appear that the possibility for false positives and false negatives since a malware sample not matching with any known sample would be flagged as negative. Such a scenario would require addition manual analysis and generation of a new signature to address such a shortfall.

D. New approaches to malware detection

Despite relentless efforts to improve malware detection, the number of malware keeps rising exponentially as mentioned earlier. Research for innovative methods to detect malware . One such approach involves use of machine learning methods for malware detection [34]. Machine learning involves dividing data into sets, with one set being the training set and having the detection system learning from this training set in order to detect malware. Other approached include using data mining to detect patterns of malware in pieces of code [35]. Data mining utilizes statistical analysis for malware detection. This is could be useful given the large sample sizes presented by thousands of malware variants

IV. TECHNIQUES FOR MALWARE ANALYSIS

The presence of malware in the target system can be detected and the malware can be analyzed through several techniques and methods. Malware analysis can be grouped roughly into two categories:

• Static analysis.
Static analysis is a simple and quick analysis technique. In this analysis, the malware is decompiled and its source code is examined using several tools like process explorer, process monitor, and handler and so on. The source code of the malware can provide important information such as DLLs called by the malware, URLs accessed etc., since the source code of the malware can explicitly reveal its malicious intent. Hence malware writers try to hide the malicious code of the malware by using packers mentioned earlier.

• Dynamic analysis
Simple malware can be detected by static analysis. There are more sophisticated and complex malwares which cannot be detected by simple static analysis. These malware obfuscates their code to bypass static detection techniques. Such advanced malware typically appears as harmless when inspected, but when it is executed, it calls malicious code kept in some other place. These malwares can be identified only after their execution. Dynamic analysis involves the execution of malware so that the behavior of the malware [40] can be studied.

Dynamic analysis can be categorized into two main methods:
(i) in-the-box
(ii) out-the-box [41].
In in-the-box approach, all the tools for anti-malware and debugging are pre-installed in the same operating system (OS). This approach is self-surgical and more efficient but highly vulnerable.

V. MACHINE LEARNING ALGORITHMS

There is a huge diversity of classifiers that may be used for Machine Learning techniques. Once done with intensive study of existing Machine Learning approaches, it will highlight drawbacks and benefits of the subsequent algorithms

accordingly that I deliberate fancy to appear to be the foremost acceptable for the task of malware detection:

1) K-Nearest Neighbour (KNN): Though it's an awfully easy algorithmic program supposed (lazy algorithms) and exhibits quick performance, it becomes inappropriate or not immensely fruitful once the coaching set undergoes from blare or outliers.
2) Support Vector Machine (SVM): algorithm includes a sturdy and complex theoretical and abstract background, due to that its performance regarding classification results is usually higher than that achieved by alternative algorithms. However, it's conceptually advanced, arduous to interpret, Computational and memory intensive. It performs well on linearly-separable data; otherwise, it needs concerned nonlinear transformations by means that of kernels. Within the gift work, the linear kernel has been chosen.

3) Decision Tree (J48): could be a classifying tree looking forward to desiring the feature values to categorize instances accordingly. A decision tree comprises of nodes and leaves for distribution. Nodes perform estimations and the leaves contain the extended conclusion (namely, whether the app underneath classification is classed as malware, adware or benign).

4) Random Forest (RF): syndicates decision trees made up of liberated arbitrary features to draw an end and attains a relatively high detection rate. Random Forest is a Machine Learning classifier often used in malware detection studies having Android background.

VI. LITERATURE REVIEW

Amr I. Elkhawas, Nashwa Abdelbaki,[2018],In this paper we introduced our novel approach in using trigrams and PE file attributes as features for malware detection. We took a text mining approach to make our detection method more robust to polymorphism and metamorphism. We used opcodes trigram sequences as the main feature for our machine learning algorithm. We used Support Vector Machine (SVM) as our classifying algorithm which is a discriminative classifier model that gives a definite decision whether the predicted outcome belongs to the learned class or not.

Limitations can be eradicated by removing trigram, sequences of three-gram consecutive opcodes are passed in the code of malware which is time consuming. Bi-grams are most preferred usage in n-gram opcodes. PE header attributes have limited information in database and they cannot predict any new malware family. Feature extraction was done with limited steps.

Muhammad Murtaz, Hassan Azwar, Syed Baqir Ali, Dr. Saad Rehman,[2018] ,This study condenses the progression of malware detection techniques supported machine learning algorithms centred on the Android Operating systems. The model uses grouping strategies including stream based, bundle based and time-based highlights to describe malware families. During this analysis, a brand-new detection and characterization system for investigation significant deviations within the network behaviour of a smart-phone application is planned. The most goal of the planned system is to guard mobile device users and cellular infrastructure corporations from malicious applications simply nine traffic feature measurements.

Limitations are used of limited datasets. This study condenses the progression of malware detection techniques supported machine learning algorithms centred on the Android Operating systems.

Udayakumar N, Vatsal J. Saglani, Aayush V. Gupta, Subbulakshmi T[2018], There are various entry points for these programs and scripts in the user environment, but only one way to remove them is to find them and remove them out of the system , small piece of script or code can be anywhere in the user system. This contains the information of different types of malware and how we will use Machine Learning to detect these malwares. For the classification of data, the approach we are using is support vector machine. Unlike other classifiers, the support vector machine is explicitly told to find the best separating line. The support vector machine searches for the closest points which it calls the "support vectors".

Limitation The dataset used in this paper will not give good accuracy with SVM.Classification of the malware we finally conclude that neural networks is most appropriate model used for this dataset for classification

VII. PROPOSED MODEL

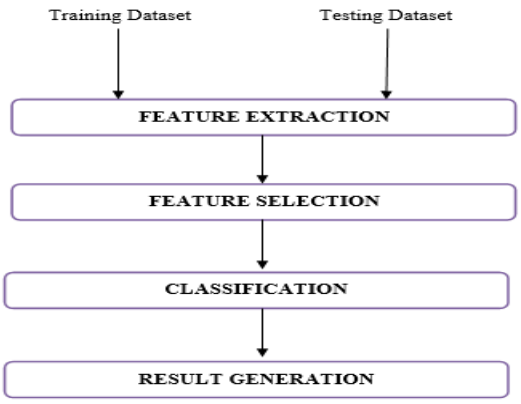

Figure-1: Structure of the Proposed System.

Firstly the dataset is prepared which consists of malware and benign executables. These files are pre-processed depending on the feature extraction method and next feature selection is done to quantify the correlation of feature for improving performance and reducing number of computations to attain the learning speed. Further after generalizing the feature capability, classifier is trained on the basis of the filtered results of feature selection. The dataset is tested corresponding to the trained classifier and results are generated as malicious or benign softwares. The obtained outcomes are evaluated with consequent performance metrics.

VIII. PROPOSED WORK

A high-level description for the proposed work includes extraction of features from our data-set, selection of the most befitting features which would then be employed for the construction of the classifier. The classifier will then be evaluated using the test data.

Feature Extraction

The process of extracting data from the files is called feature extraction. The goal of feature extraction is to obtain a set of informative and non-redundant data. Another important requirement for a decent feature set is non-redundancy. Having redundant features i.e. features that outline the same information, as well as redundant information attributes, that are closely dependent on each other, can make the algorithm biased and, therefore, provide an inaccurate result.

The current dataset consists of malware binaries and OPCODE. We plan to use the technique of frequent item set mining in order to identify frequently occurring patterns and sequences in the files. Since malware belonging to the same families have similar behavior, it is expected that they have similar patterns in their binaries and asm. Apart from the commonly occurring patterns, we also aim to use properties like file size, line count, their distribution etc. We hope to find interesting patterns by using the frequent item-set technique and move forward from there.

Following work has been done for feature extraction:

1. Counting of all the commonly occurring assembly language instruction.
2. Applied feature item set technique to select some dynamic features from files. These include:

- Removal of ID names from files.
- Removal of Feature names.
- Counting of Number of lines in a file.
- Add unique words to number of counts.
- Counting of unique section in a file.
- Count the number of assembly instruction.
- Finally normalize the entire datasets and create a CSV file.

Feature Selection

The process of reducing the vector dimensions is referred to as feature selection. At the end of this process, we expect the selected features to outline the relevant information from the initial set so that it can be used instead of initial data without any accuracy loss.

The training vector consists of following asm instruction set to train the classifier. This instruction sets are stored in comma separated file.

POPA JNP ROL POPF SAL INT JNZ PUSHF IMUL POP PUSHA JNB RCR IN SAR ROR JNO OUT JG XOR SUB CLD NEG JGE IDIV CLC RCL ADD CALL ADC XCHG MUL INC CWD RET LEA JZ JMP JP MOV JS SBB JO NOT JE JLE CWDE NOP JA JB CLI STD JL SHR STC CMC STI JECXZ JBE PUSH DIV DEC SHL OR CMP

Threshold Limit:

IF threshold/9 >= 40

Then threshold= threshold-20 (For reduction)

ELSE

threshold= threshold – (threshold/ 9*2);

Classification Algorithms

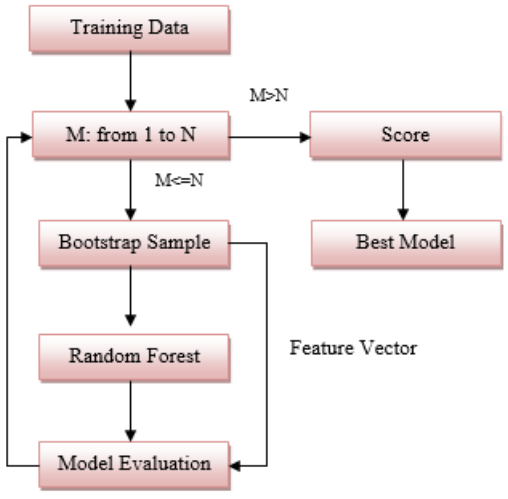

Figure-2: Working of Proposed Classifier.

Proposed Algorithm:

Algorithm of proposed classifier is as below:
Algorithm
Input: N: number of nodes or files.
M: number of features.
D: number of trees to be constructed for classification.
Output: V: the class with the highest Score.
While stopping criteria is false do
Randomly draw a file sample from the training data D

Use the steps below to construct tree Ti from the drawn bootstrapped sample A:

I. randomly select m features from$M$; where$m \ll M$
II. For node d, calculate the best split point among the m features.
III. Split the node into two daughter nodes using the best split parameter.
IV. Repeat I, II and III until n number of nodes has been reached.

Build your forest by repeating steps I–IV for D number of times.
End While
Output all the constructed trees $\{Ti\}$

Apply a new sample to each of the constructed trees starting from the root node
Assign the sample to the class corresponding to the leaf node.
Combine the decisions (or Score) of all the trees
Output V, that is, the class with the highest score.
End Algorithm.

IX. RESULTS AND EVALUATION

The classification performance can be evaluated in terms of classification accuracy as defined below. Accuracy explains correctly classified malware instances.

Classifier	Accuracy (in %)
Gaussian Naive Bayes	35.63
Decision Tree	56.89
Logistic Regression	66.09
Ridge Regression	40.57
Linear SVM	72.98
Proposed	93.10

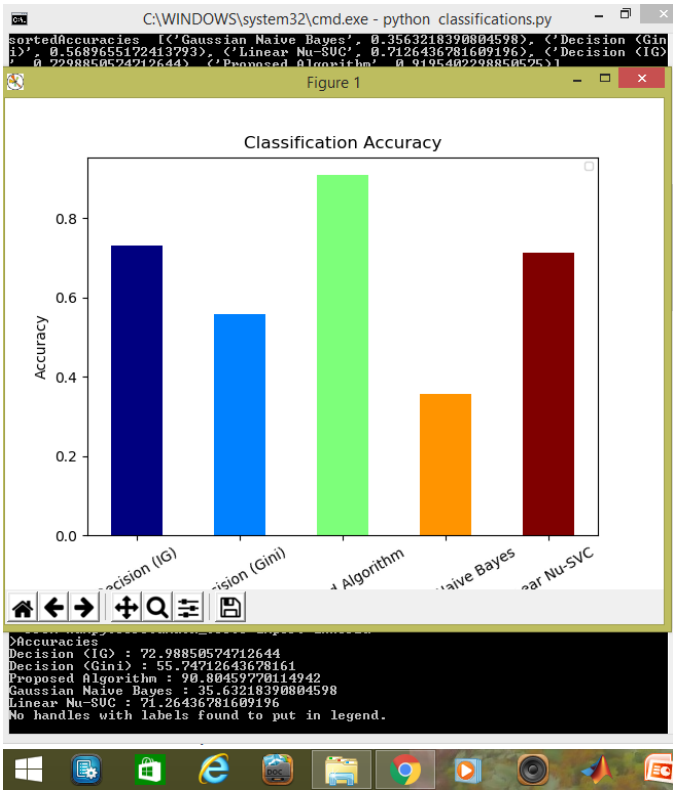

Figure-3: Working of Proposed Classifier.

X. CONCLUSION

In this thesis, Gaussian Naive Bayes Algorithm, DECISION TREE algorithm, Logistic Regression algorithm, Linear SVC algorithm and suggested algorithm for malware detection and classification have been proposed for increasing the overall accuracy of the classifier in the classification of malwares. For the same we apply feature extraction techniques so that accurate data is fed as an input to the training process, our proposed approach classify the malwares as malicious or benign which further helps in malware analysis and uses that analysis for further decision making. The work of proposed model has gone through multiple stages and classifiers learning stage. For analytical evaluation of the proposed classifier accuracy are used. The comparative results prove that proposed model improved the overall classification accuracy malware classification as compared to traditional existing techniques for classification.

REFERENCES

[1] Firdausi, Ivan, et al. "Analysis of machine learning techniques used in behavior-based malware detection." Advances in Computing, Control and Telecommunication Technologies (ACT), 2010 Second International Conference on. IEEE, 2010.

[2] Santos, Igor, et al. "Opcodes-sequence-based semi supervised unknown malware detection." Computational Intelligence in Security for Information Systems. Springer Berlin Heidelberg, 2011. 50-57.

[3] Gavriluţ, D. and Ciortuz, L., 2011, September. Dealing with Class Noise in Large Training Datasets for Malware Detection. In Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2011 13th International Symposium on (pp. 401-407). IEEE.

[4] Zhao Z. A virus detection scheme based on features of Control Flow Graph. In Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on 2011 Aug 8 (pp. 943-947). IEEE.

[5] Gavrilut, D., Benchea, R., & Vatamanu, C, Optimized zero false positives perceptron training for malware detection. In Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2012 14th International Symposium on (pp. 247-253). IEEE.

[6] Raman, K., 2012. Selecting features to classify malware. InfoSec Southwest, 2012.

[7] Shahzad, F., Shahzad, M. and Farooq, M., 2013. In execution dynamic malware analysis and detection by mining information in process control blocks of Linux OS. Information Sciences, 231, pp.45-63.

[8] Santos, I., Brezo, F., Ugarte-Pedrero, X., & Bringas, P. G. (2013). Opcodes sequences as representation of executables for data-mining-based unknown malware detection. Information Sciences, 231, 64-82.

[9] Jiang, Q., Liu, N. and Zhang, W., 2013, December. A Feature Representation Method of Social Graph for Malware Detection. In Intelligent Systems (GCIS), 2013 Fourth Global Congress on (pp. 139-143). IEEE.

[10] Khammas, Ban Mohammed, et al. "FEATURE SELECTION AND MACHINE LEARNING CLASSIFICATION FOR MALWARE DETECTION." Jurnal Technology 77.1 (2015).

[11] Ranveer, S., & Hiray, S. SVM Based Effective Malware Detection System. In: 2015 International Journal of Computer Science and Information Technologies, Vol. 6 (4) , 2015.

[12] Mauricio Macías, et. al. Proposed "SGSI Support Through Malware´s Classification Using a Pattern Analysis" 9781509011476/16/$31.00©2016IEEE.

[13] Jhu-Sin Luo, Dan Chia-Tien Lo," Binary Malware image Classification using Machine Learning with Local Binary Pattern", 978-1-5386-2715-0/17/$31.00 ©2017 IEEE.

[14] Ken F. Yu, et. al. Proposed "Machine Learning in Malware Traffic Classifications", 978-1-5386-0595-0/17/$31.00 ©2017 IEEE.

[15] Amr I. Elkhawas, Nashwa Abdelbaki," Malware Detection using Opcode Trigram Sequence with SVM" Information Security- School of Communication and Information Technology Nile University Giza, Cairo, IEEE-2018.

[16] Muhammad Murtaz, Hassan Azwar, Syed Baqir Ali, Dr. Saad Rehman," A framework for Android Malware detection and classification" NUST, College of Electrical and Mechanical Engineering Islamabad, Pakistan, IEEE-2018.

[17] Udayakumar N, Vatsal J. Saglani, Aayush V. Gupta, Subbulakshmi T, "Malware Classification Using Machine Learning Algorithms"Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018) IEEE Xplore ISBN:978-1-5386-3570-4.