ISSN:0975-9646

Voting Methods For Malware Detection

Pranit Gaikwad
M.E. - COMPUTER
SLRTCE-Mumbai University,
Mumbai, Maharashtra, India

Prof.Dilip Motwani
Assistant professor
VIT-Mumbai University,
Mumbai, Maharashtra, India

Prof.Vinayak D.Shinde
Assistant professor
SLRTCE-Mumbai University
Mumbai, Maharashtra, India

Abstract-The internet plays an important role in all areas of society from economy to the government. Nowadays Computer Security is pretentious by malicious data. Computer security means keeping the information on computer in a secure manner. so detection of malware is an paramount substance. In an ensemble learning algorithm, there are two pre-processing techniques, and an pragmatic evaluation of the planned algorithm. In this method Sequences of operational codes are extracted as features from malware and benevolent files. These sequences are used to produce three different data sets with different configurations. A set of learning algorithms is evaluated on the data sets and the predictions are combined by the ensemble algorithm. The predicted output is strong-willed on the basis of veto voting. The trial results show that the approach can accurately detect both novel and known malware instances with higher recall in comparison to majority voting. A veto-based classification was proposed that was able to predict about malware better than majority voting. A series of experiments with n-gram data sets, generated from different strategies, were performed. A recent threat, i.e. scareware was used as malware representation. The results indicated that the proposed model reduced the number of false negatives. The proposed model will be tested for detection of di_erent types of malware. This proposed system use Trust Base Veto Algorithm for deciding malware or benign. Also research work proposes to develop appropriate malware detection.

Keywords-Data mining, Ensemble, Feature Extraction, Feature selection, Machine learning, malware detection, Majority voting, Trust, Veto Voting.

1. INTRODUCTION

Malware is a term for any malicious software which enters system without permission of user of the system. Malware is a combination of two words 'malicious' and 'software' together called as Malware. Malware is a very big vulnerability in today's computing world. It continues to grow in size and evolve in complexity. As more and more organizations try to tackle the problem, the number of websites distributing the malware is increasing at an frightening rate and is getting out of control. Most of the malware enters the system while downloading files over Internet. It checks for vulnerabilities of operating system and perform unplanned actions on the system finally slowing down the performance of the system once the malicious software finds its way the system. Malware has ability to infect other executable code, data/system files, boot partitions of drives, and create excessive traffic on network leading to denial of service. When user executes the infected file; it becomes resident in memory and infect any other file executed afterwards. If operating system has vulnerability, malware can also take be in charge of system and infect other systems on network. Such malicious programs are also known as parasites and favorably affect the performance of machine generally resulting in slow-down.

In today's world security is major issue in every field of technology. Information security, network security, computer security all are branches of information technology which deal with protection of information on a network or standalone computer. As every organization depends on the computer and technology of security requires constant development. A more recent annual report on the Internet security threat-2013 from Symantec says "Threats to online security have grown and evolved considerably in 2012, In particular, social media and mobile devices have come under increasing attack in 2012."[1]

Malware is a general term for all types of malicious software, which in the context of computer security means: software which is used with the aim of attempting to breach the computer systems security policy with respect to confidentiality, integrity and availability [2]. The main characteristics of the malware are replication, propagation, self–execution and corruption of computer system. It spread over the connected system in the network or internet connection. It infects the system by transferring malware from a polluted device to another uninfected one using local or network file system [3]. Malwares are classified according to their propagation method and they come in the different forms like Virus, worms, Trojan horse, Spyware, scareware, adware, Backdoors, Botnets etc. Malware detection is the key to protect the system from these types of malware. There are two main traditional malware detection techniques: Signature-based detection and Heuristic-based detection. In signature-based technique specific features or unique strings are extracted from binaries, which are later used for detection of malware. However a copy of malware is required to extract and develop a signature for detection purposes. A database of known code signature is updated and refreshed constantly by anti-virus software vendor as a result it detects only known instances of malware accurately. It cannot detect the new, unknown malware as no signature is available in database for such types of malware. Heuristic-based technique detects known as well as unknown instances of malware but level of false positive is high i.e. accuracy is low and it is more time and resource consuming technique, therefore a new malware detection technique named as data mining based detection is proposed.

The aim of this study is to investigate malware detection and enhance the idea of heuristic-based detection method by using machine learning algorithm and data mining technique. The purpose is to detect both known and unknown instances of malware with high accuracy. The proposed system framework consists of data pre-processing

techniques, ensemble learning algorithm and evaluation of proposed algorithm for malware detection.

This paper is organized as follows. Section 2 briefly describes related work. section 2 explains the related work in malware detection using data mining and machine learning methods, section 3 explains the proposed system architecture & their modules and section 4 conclude the paper.

2. RELATED WORK

In 2001, Matthew G. Schultz et al.[4] Method presents a data-mining framework that detects new, previously unseen malicious executables accurately and automatically. The data-mining framework automatically found patterns in data set and used these patterns to detect a set of new malicious binaries. Comparing detection methods with a traditional signature based method, research method more than doubles the current detection rates for new malicious executables. The first contribution presented in this paper was a method for detecting previously undetectable malicious executable. Method showed this by comparing our results with traditional signature-based methods and with other learning algorithms.

In 2010, Raja Khurram Shazhad *et al.*[5] presented a Spyware detection approach by using Data Mining (DM) technologies. Work approach is inspired by DM-based malicious code detectors, which are known to work well for detecting viruses and similar software. Method extract binary features, called *n*-grams, from both spyware and legitimate software and apply five different supervised learning algorithms to train classifiers that are able to classify unknown binaries by analyzing extracted *n*-grams. The experimental results suggest that method is successful even when the training data is scarce. Data mining-based malicious code detectors have been proven to be successful in detecting clearly malicious code e.g. like viruses and worms. Results from different studies have indicated that data mining techniques perform better than traditional techniques against malicious code. The main objective of this study was therefore to determine whether spyware could be successfully detected by classifiers generated from *n*-gram based data sets, which is a common data mining-based detection method for viruses and other malicious code.

In 2010 Yi-Bin Lu et al improved the accuracy of malware detection using the classifier ensembles to replace individual classifier. The combination of multiple classifiers to reach final prediction is called ensemble. Ensemble model performs better than single classifier model; He introduced the different ensemble learning algorithm like bagging, boosting, voting, stacking and grading. The new ensemble learning method SVM-AR was proposed, it combines the SVM and association rules based on hierarchical taxonomy, also proposed the framework for malware detection using machine learning. According to review of related papers on topic of malware detection using machine learning it was found that decision tree, SVM, NB and KNN are most common classification algorithms used by researchers. The overall accuracy of each algorithm was tested using collected dataset. The result showed that NB is the worst classification algorithm. Accuracy improvement is achieved using the multi-classifier as ensemble learning method [7].

3. PROPOSED WORK

The overall process of classifying the unknown files as either benign or malicious using machine learning method has been classified into two phase: training phase and testing phase. In training phase training data set of malicious and non-malicious files are prepared. Each file is processed with feature extraction and selection techniques, which results into a desired data set. The vectors of files in the data set and their known classification are the input for learning algorithm. The learning algorithms process these vectors and generate the trained classifiers. The trained classifiers are used in the proposed classification model. During testing phase, test set collection of new, unknown benign and malicious files which did not appear in the training data set are classified by the classifier that was generated in the training phase. Each file in the test data set is pre-processed as in the training phase. Based on the vectors of files in the test data set the trained classifier will classify the file as either benign or malicious. In the testing phase the performance of the generated classifiers is evaluated by standard accuracy measures. The system architecture for the proposed work is shown in the fig. 1.

3.1 System Module

Data Set*:* For malware classification, data sets have been prepared using various representations of files. Features that are commonly extracted from executable files include byte code n-gram, printable strings, instruction sequence, system calls, opcode n-gram. N-gram is sequence of n characters. one or more operands for performing the operations. The opcodes are extracted as feature to prepare the dataset.

Feature Extraction: When the input data to an algorithm is too large to be processed and it is supposed to be disgracefully redundant then the input data will be transformed into a reduced depiction set of features. Transforming the input data into the set of features is called Feature Extraction. If the features extracted are carefully selected it is expected that the features set will remove the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input. Feature extraction is performed on raw data prior to applying *k*-NN algorithm on the transformed data in Feature space.

Three feature extraction techniques are used to extract the opcode as feature and three correspondence datasets are prepared.

a) Opcode n-gram

Data set is prepared with size of n=2 i.e. opcode bi-gram. To understand this process, assume that a disassembled binary file contains the following given data. A pair of characters represents an opcode.

pp 33 qq 55 rr 77 ss 99 tt 00

The generated bi-grams are pp33 qq55 rr77 ss99 tt00.

b) Overlapping n-gram

This technique is used to take out all possible blend of strings. Two parameters namely size and step are used. The Size parameter defines the size of n-gram to be extracted and step parameter defines the number of opcodes to be skipped before extracting the next n-gram. Following the above example,

if size=2 and step =1, the generated string will be pp33 33qq qq55 55rr rr77 77ss ss90 90tt tt00

c) Sliding Window opcode extraction

Some changes are made in overlapping n-gram i.e. size parameter is changed to the start-end size parameter. The start-end size parameter defines the number of adjacent opcodes to be extracted for start and end of n-gram. The step parameter defines the number of opcodes to be skipped for extracting a new n-gram and Gap size parameter specifies the gap between start and end opcode or number of opcodes to be skipped between start and end opcode of n-gram. If start-end size= 1, step =1 and gap=1, the generated output will be ppqq 3355 qqrr 5577 rrss 7799 sstt 9900, and so on. These three feature extraction techniques are used to extract no. of possible combination of strings and prepared three connection datasets.

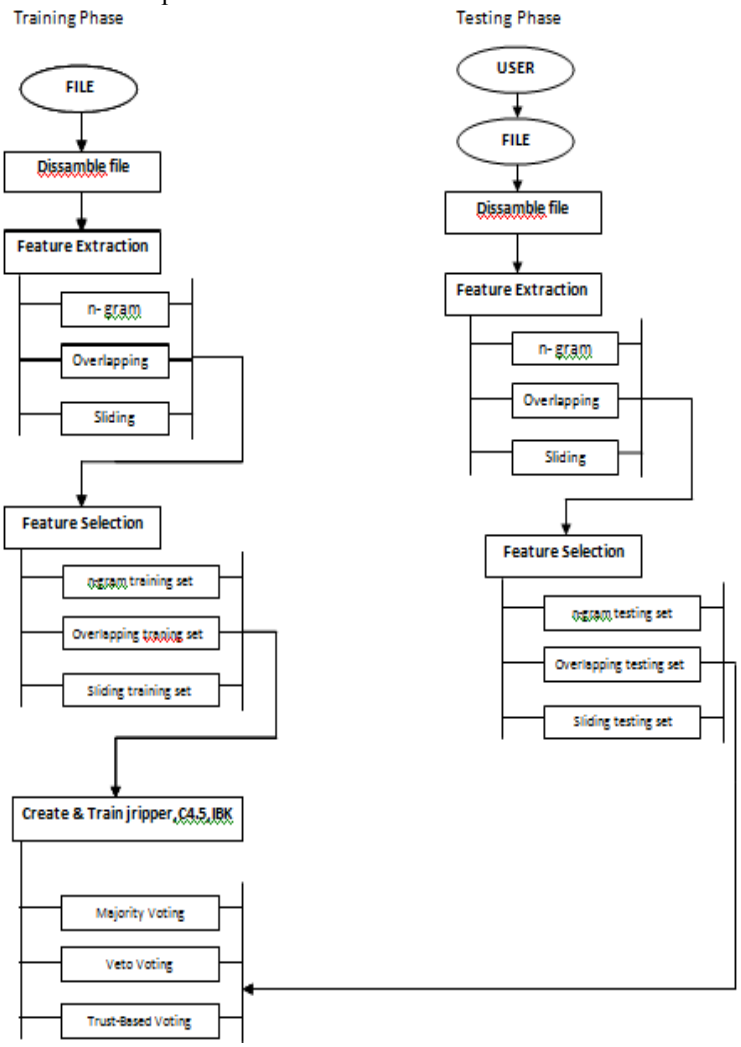

Fig 1: System block diagram

Feature Selection*:* It is necessary to remove the irrelevant, redundant, noisy data from the entire large dataset, so we need to select small, relevant, consistent features from the entire large feature set as a result reduced feature dataset will be achieved. Many techniques have been used to select best features like gain ratio, information gain, fisher score, term frequency-inverse document frequency (TF-IDF). In our work TF-IDF is used. It is a text categorization technique. N-gram is analogue to word or term in text document. A vocabulary of words or term is ext racted from so called document set. For each word or term (t) in the vocabulary, its frequency (f) in the single document (d) and in the entire set i.e. document set (D) is calculated. Weight is assigned to each word; weight is equal to its frequency (f) in d. such weights are called as term frequency (tf) i.e. frequency of term in document. The frequency (F) of each term is calculated in D, this is called Document Frequency (DF).

Classification Model*:* Every classifier has its own decision. In proposed system they are used as committee in classification model. Here we used classifier ensemble which can use method like voting to reach the final prediction. It performs better than single classifier and helps to improve the detection accuracy. Three voting

schemes are used viz. majority voting, veto voting and trust-based veto voting.

a) Majority Voting

The decision from more than one expert (Classifier) may be required in certain situation, so committee of experts (Ensemble) is formed as it is expected that a committee always performs better than a single expert. Normally committee uses majority voting for combining the decisions of the experts to reach a final conclusion. The majority voting is considered as simple and effective scheme. This scheme follows democratic rules i.e. the class with highest number of votes is the outcome.

b) Veto Voting

In veto voting the committee may give the right to veto the decision of committee to any member. It is used to give importance to a single expert (classifier) who predicts against the majority. Any vote indicating an instance as malware, alone can determine the outcome of the classification task regardless of the count of other votes.

c) Trust-based Veto Voting

Trust can be computed as +1 or -1, the increased or decreased value can help in determining the extent of the trust. The trust can

be calculated as single trust or group trust.be calculated as single trust or group trust. Mostly the group trust is calculated for different computational problems. A set of inference rules are used to value the trust i.e. and derived value is further used for the decision. In trust-based veto voting three types of trust viz. local trust, recommended trust and global trust are calculated.

Algorithm: Conviction Counting

I/P: Existed class(Ce), Class by algorithm A(Ca),
Class by algorithm B(Cb)

O/p: Conviction of algorithm A

```
Repeat
If(Ca=Cb)

   Do nothing

End if


If(Ca != Cb)

    If(Ca=Ce)

            false=false(A,B)+1

    Else(Cb=Ce)

            true=true(A,B)+1

    End if

End if
Until !EOF
```

In local trust each algorithm in the system calculates its trust level for other algorithms in the system which means how much algop trusts the algoq in term of predicting the class of an instance, called as local trust. Local trust of algop on algoq is calculated by comparing the predictions (d) of both algorithms with each other and actual class (C) of instance, so from data set of benign and malicious instances, an instance of benign class is given to both algop and algoq for predicting the class of instance. The possible predictions are, both algorithms may predict correct or both algorithms may predict incorrect or any one of the algorithm may predict the correct class. If both algorithms give the same prediction either correct or incorrect, trust is not affected. If algop predicts the incorrect class and algoq predicts the correct class then algop increases the trust level of algoq with +1. If algop predicts the correct class and algoq predicts the incorrect class then algop increases the distrust level of the algoq with +1. Likewise all the instances in the dataset are given to both algorithms sequentially for the prediction. At the end of process local trust of algop on algoq is calculated by dividing the trust (sat) with the sum of trust (sat) and distrust (unsat). The algorithm for local trust calculat ion is given below

4. CONCLUSION

Ensemble based malware detection using different voting schemes can predict known as well as unknown malwares with high accuracy because ensemble model performs better than single classifier model in term of improving the detection accuracy. In proposed system static analysis is used which is safe and fast technique as files are analyses without its execution and it detects the malware accurately. Also Heuristic based malware detection is extended by using the data mining and machine learning techniques to detect known as well as unknown malwares. Different voting schemes are used to determine which voting scheme is best to detect the known as well as unknown malware with high accuracy.

REFERENCES

[1] Symantec Corporation, Internet security threat report-2013, *Volume 18*

[2] Robin Sharp, An Introduction to Malware, Spring 2012. Retrieved on April, 10, 2013

[3] Imtithal A. Saeed, Ali Selamat, Ali M. A. Abuagoub, A Survey on Malware and Malware Detection Systems, *International Journal of Computer Applications (0975 – 8887) Volume 67– No.16*, April 2013

[4] K.Mathur,S.Hiranwal, "A Survey on Techniques in Detection and Analyzing Malware Executables", International Journal of Advanced Research in Computer Science and Software Engineering vol. 3, 422-428,2013.

[5] Raja Khurram Shazhad *et al.*, "Spyware detection approach by using Data Mining (DM) technologies.", Availability, Reliability, and Security, 2010. ARES '10 International Conference vol. 3, 295 - 302,2010.

[6] Asaf Shabtai, Robert Moskovitch, Clint Feher, Shlomi Dolev and Yuval Elovici, Detecting unknown malicious code by applying classification techniques on OpCode patterns, Security Informatics2012.

[7] Yi-Bin Lu, Shu-Chang Din, Chao-Fu Zheng, and Bai-Jian Gao, Using Multi-Feature and Classifier Ensembles to Improve Malware Detection, *JOURNAL OF C.C.I.T., VOL.39, NO.2*, NOV., 2010.

[8] M. Schultz, E. Eskin, F. Zadok, and S. Stolfo, "Data mining methods for detection of new malicious executables," in Proceedings of the Symposium on Security and Privacy, 2001, pp. 38–49.

[9] Vinod P. V.Laxmi,M.S.Gaur: Survey on Malware DetectionMethods, 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.

[10] F. Adelstein, M. Stillerman, and D. Kozen, "Malicious code detection for open firmware", In Proceedings of the 18th Annual Computer Security Applications Conference,2002.

[11] D. Uppal, V. Mehra and V. Verma, "Basic survey on Malware Analysis, Tools and Techniques " , International Journal on Computational Sciences & Applications (IJCSA)Vol.4, No.1, 103-112,February 2014

[12] Bergeron, J., Debbabi, M., Desharnais, J., M., E., M.,Lavoie, Y., &Tawbi, N. (2001). Static Detection of Malicious Code in executables programs. International Journal of Req Engineering.

[13] G. McGraw and G. Morrisett. Attacking malicious code: A report to the infosec research council. IEEE Software, 17(5):33–44, 2000.

[14] SavanGadhiya,KaushalBhavshar "Techniques for Malware Analysis" Volume 3, Issue 4, April 2013 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Softwar Engineering.

[15] Robiah Y, SitiRahayu S., MohdZaki 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.