International Journal of Computer Architecture and Mobility (ISSN 2319-9229)
Volume 7 -Issue 7, July 2019

Malware Category Prediction Using KNN And SVM Classifiers

Sanjeevani Oswal
Department of Computer Science
SVVV, Indore, M.P.
osanjeevani@gmail.com

ABSTRACT
The emergence of the vulnerability databases around the world are serving the purpose of a double edged sword. The malware researchers, industry members and end users are aware of them to initiate better prevention strategies. The dark world hackers are using them to lure into systems through the points mentioned in the vulnerability databases. Hence, it is highly necessary to predict the malware at the early stage to avoid further loss. The objective of this research work is to predict the malware using the classifiers Logistic Regression, K–Nearest Neighbors (KNN) and Support Vector Machines (SVM). We found that the appropriate use of these classifiers have resulted great improvement in prediction accuracy. Feature selection is also done to further improve the accuracy to 99% with polynomial kernel function.

Keywords: Malware, Malware prediction, K-Nearest Neighbors, Support VectorMachines.

1. INTRODUCTION

With the fast development of the internet, cyber threats also increase because of malware. Malware is defined as "a type of computer program that is made to harm the other user's computer in many ways." Nowadays different-different types of malware are present and people buy malware on the black market to increase the attacks on our system, so it is very difficult for anti-virus scanner to completely protect a computer.

Malware or malicious software is a program that affects a computer system without the user's permission and with an aim to cause harms to the system or steal private information from the system. Software that deliberately fulfils the harmful intent of an attacker is commonly referred to as malicious software or malware.

Thousands of new malwares are emerging every day and the existing malwares are evolving in their structure which is becoming difficult to detect. According to the latest internet threat from Symantec, a whopping 317 million new types of malwares were discovered.

Due to increase in new samples every day, automated malware analysis tools and methods are needed to distinguish malicious from benign code. Most of the commercial anti-virus software uses signature based malware classification method. This method compares the unknown malwares with a database of known malicious program to identify whether the file is malware or benign. The signature is a unique identification of a binary file. Signature of malware is found by using static analysis, dynamic analysis or hybrid analysis and is stored in signature database. The main disadvantage of this method is, the signature database need to be updated frequently because of the fast emerge of new malwares every day [9].

2. RELATED WORKS

Nowadays malware detection using machine learning methods is considered important for every user and network. Lots of studies have been done in this area, to come with different accuracy for different methods.

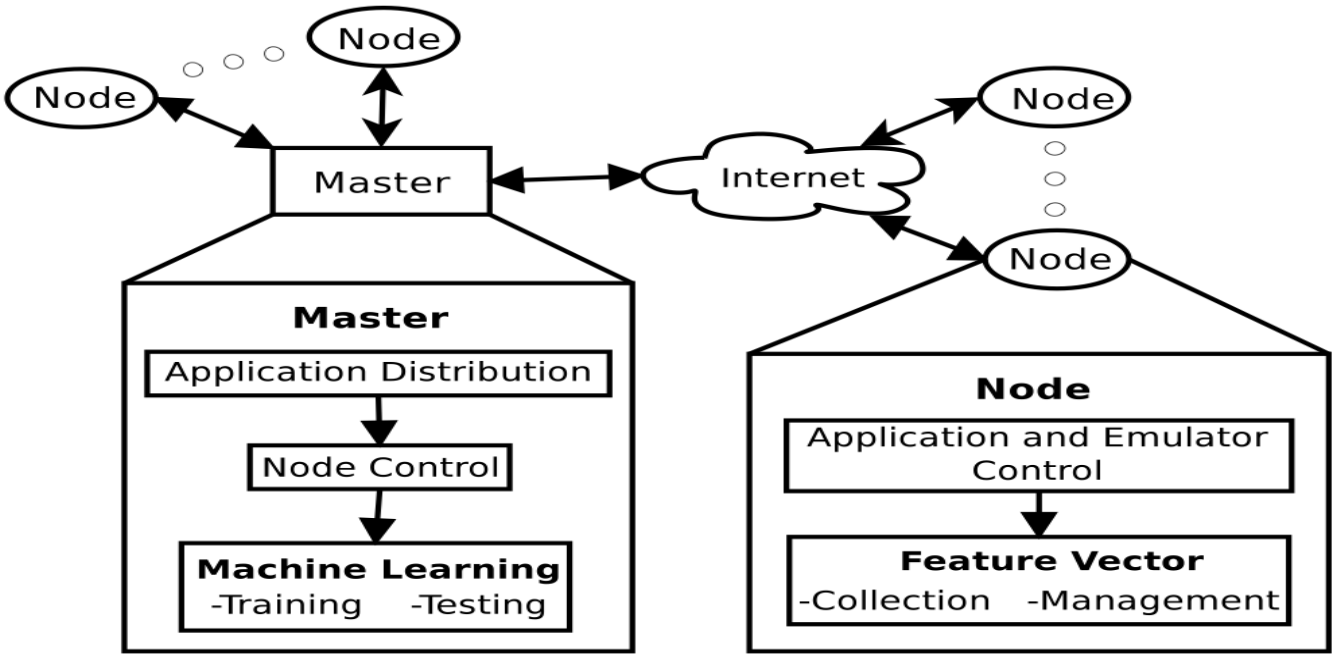
In Dragos Gavrilut paper the motive is to make a malware detection system using many modified perceptron machine learning algorithms. For different-different approaches, the accuracy of 96.18% and 69.90% is obtained. Point to be noted in this research is the higher accuracy and lesser false-positive result [1] [7].

In the research done by Tang, feature extraction is done on the malware dataset which contai
ns portable executable files. Malware analysis is done with the dataset which gave the accuracy of 97.5 and false-positive rate of 0.03. [2]

3. PROPOSED ARCHITECTURE AND IMPEMENTATION

3.1. Data Collection

For this project, we have a malware dataset that is derived using the open source software tools. The header analysis tools were used to extract the required features from the malware file which is then used for the further steps of the proposed system as in Figure 1. Ten malware categories used in this research are : virus, trojan, adware, backdoor, muldrop, sdbot, spam, rbot, ransomware and unknown.



- Virus: When a virus executes, it reproduces, itself and self-copying programs and starts inserting its own code.
- Trojan: It is a malicious program which misleads users of its true intent.
- Adware: It is software that generates money for software developers by automatically generating advertisements in the user interface at the time of installation.
- Backdoor: It is a hidden part of any computer program. It is used for securing remote access to a computer through open ports.
- Muldrop : It is a Trojan that spreads through shared networks or attaches itself to downloadable files. It can erase or modify your personal files without your awareness.
- Sdbot: It is a worm which gives remote attacker full access on the victim's computer.

From the group of ten malware samples the following features has been extracted for makingthe dataset

Debug size: The size of the debug information details

IatRVA: Relative virtual address in an image file that address of an item after it is loadedinto memory. With the base address of the image subtracted from it. The RVA of an item almost always differs from its position within the file on disk.

Export size: Export directory table information size.

Image version: The version of the image that is used for processing the operating system.

Resource size: Resource directory table has the format of offset, size and field. The fieldthat is used is resource size.

3.2. Visualization

In the visualization part, many key points have

been analyzed which helped in finding some relations between various features. The best example is IatRVA feature does not have any effect on the category of malware. This visualization work is done using a python library called seaborn which helps in plotting clean and understandable graphs.

For the visualization, each feature is removed to understand the importance that it has in the original dataset. The contribution of each feature is analyzed by using the correlation graph as mentioned in Figure 2 and Figure 3. A clear description of the dependent and independent variables such as Virtual size depends on the Number of sections as can be seen from the graph given above. The dependent features are going to be deleted from the dataset to reduce the complexity for the training purpose

3.3. Pre Processing

In this phase of proposed work, data pre-process has been done for the dataset which we have visualized in the previous section. This step is done to make data more appropriately fit into proposed model for training purpose. In this step, we closely looked on the features and tried to make possible changes in the dataset for getting more efficient results. Generally pre-processing step consist Cleaning, transformation, and Reduction of three sub steps which are to be followed for preparing the dataset to train:

In the cleaning phase, all the garbage entries present within the dataset that are not suitable for modeling are removed. Let's say our dataset doesn't contain any relevant data in some particular cell then we have to assign some integer value to these non-relevant entries. is non-integer values make it difficult for the model to train over it.

Also after dropping the dependent features, we have then analyzed our dataset and now it looks much cleaner and understandable.

Eliminated IatRVA and Resource Size from the dataset because they are not much contributing towards the prediction of the malicious content this drop function is used to delete particular row or column given the axis as 0 and 1 respectively.

This is the most crucial part in whole work since in this module only trained our model to predict the malware type as discussed in the dataset module. We have used two most famous modelling techniques for training on our model on the given dataset which contains various features to predict the type of malicious content and compared the results extracted from both of the models after training it on the particular dataset by using the two types of models Logistic Regression and KNN Classifier

KNN model implements the technique of clustering in which the whole dataset is divided into some specific number of clusters whose cluster head is the centroid of the cluster nodes. In each of the iteration, the centroid for a particular cluster updates as a new node joins the cluster. For calculating the distance between instances, various distance algorithms are used such as:

Euclidean Distance: This algorithm is mainly used for real valued input-variables and is given by the following mathematical formula:

EuclideanDistance(x, xi) = sqrt(sum((xj – xij)^2))

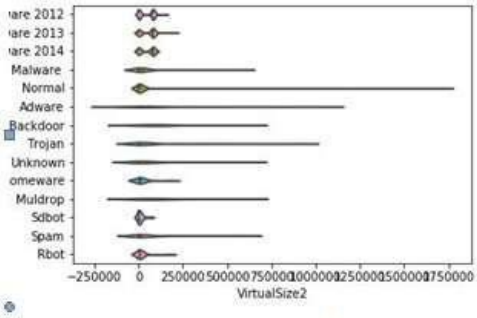
2. RELATED WORKS

Nowadays malware detection using machine learning methods is considered important for every user and network. Lots of studies have been done in this area, to come with different accuracy for different methods.

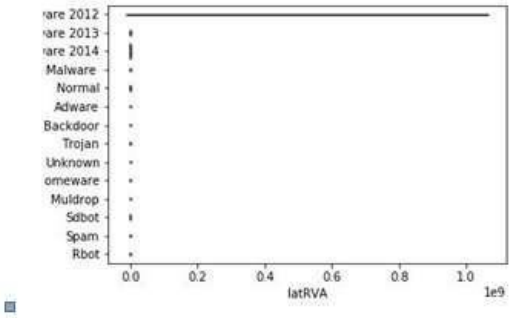
ii. Hamming Distance: It is helpful in calculating distance between binary valued input-variables.

iii. Manhattan Distance: This distance algorithm is also known as City Block Distance. In this method, the sum of the absolute differences is found for the real valued vectors.

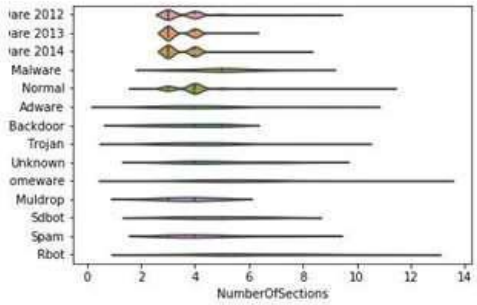
iv. Minkowski Distance: This method is the generalized way for finding the distance between two nodes using Euclidean distance and Manhattan distance.



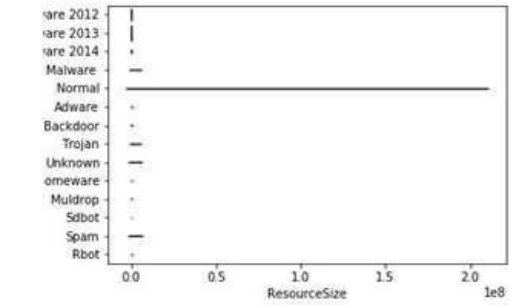
(a) VirtualSize vs Malware Type



(b) IaRVa vs Malware Type



(c) Number of Sections vs Malware



(d) Resource Size vs Malware

The testing phase involves categorization of different input variables into various categories of malwares such as Trojan, Adware, etc. on the basis of function generated in the training phase. Support Vector Machines are used to detect the classifiers with various kernel functions and learning rates.

4. IMPLEMENTATION AND RESULTS

While training our model on KNN classifier, we are giving a parameter known as ‘k’ in our model which plays a vital role in prediction. A very low value of ‘k’ will result in over-fitting of the model and prediction of new test data sets are made complex. If ‘k’ is very large then under-fitting of the model happens which will result in lesser accuracy. Therefore we first found the value of k for which our model is giving maximum accuracy.

K-values	Accuracy removing Debug Size (in %)	Accuracy removing Debug Size (in %)	Accuracy removing IatRVA(in %)	Accuracy removing Resource Size (in%)	Accuracy removing ImageVersion(in%)	Accuracy removing Virtual Size (in%)
1	92.63	92.73	92.15	92.53	90.4	91.95
2	90.69	90.5	91.66	90.11	87.4	90.21
3	90.01	90.11	91.37	89.43	86.91	88.75
4	90.11	90.21	90.98	89.34	87.01	88.95
5	90.69	90.79	91.76	90.21	88.37	89.82
6	91.18	91.27	91.95	90.21	88.35	89.53
7	90.98	90.98	91.86	90.4	88.85	89.72
8	90.98	91.08	91.76	90.79	88.66	90.21
9	91.18	91.18	91.86	90.5	88.66	90.5
10	91.18	91.18	91.86	90.69	88.66	90.69
11	90.98	90.98	91.66	90.31	88.95	89.63
12	91.08	91.08	91.37	90.4	88.75	89.63
13	91.18	91.18	91.56	90.4	88.66	89.63
14	90.98	90.98	91.66	90.5	88.56	88.95
15	91.08	91.08	91.66	90.6	88.56	89.24
16	90.79	90.79	91.27	90.98	88.37	89.34
17	91.18	91.18	91.66	90.69	88.56	89.24
18	91.18	91.18	91.66	90.69	88.27	88.95
19	90.69	90.69	91.18	90.4	88.27	88.95
20	90.89	90.89	91.27	90.4	88.17	89.53
21	90.79	90.79	91.18	90.31	87.79	89.43
22	90.79	90.79	90.69	90.11	87.4	89.63
23	90.79	90.89	90.98	89.92	87.3	89.34
24	90.79	90.79	90.79	90.79	90.79	89.05

Table 2 KNN Classifier Accuracy

Feature Removed	Accuracy in percentage(using KNN classifier)
Debug Size	91.18
IatRVA	91.59
Image Version	88.68
Resource Size	90.40
Number of Sections	91.18
Virtual Size	89.92

As we can see from the Figure 4 that it is giving its peak value at k=13(if not considering over-fitting and under-fitting values) which is giving an accuracy of 91.18%. But while training our model on logistic regression, we got accuracy of 74.61%.

5. TRAINING AND TESTING WITH SVM

An SVM is supervised learning algorithm that produces model as output It is a representation of the examples as data in path mapped so that the examples of the separate categories are divided into 2 separate categories. SVM is trained and tested using python modules.

6. UNIQUENESS OF WORK

- The dataset extraction process from the malware crossed many difficulties to bring out the useful dataset to be fed for the machine learning algorithms.
- The advanced logistic regression gave a very low accuracy of 74 % where the unsuitability of the algorithm for malware category prediction is proved.
- The KNN has shown the improved results of 92.63 % with all possible combination of the k values and feature extraction
- SVM has proved to be better classifier with 99.3% accuracy.

7. CONCLUSION AND FUTURE WORK

Malware category prediction is the core component of the research in the malware analysis world which is further tuned by the researchers and anti- virus industries. The data set extraction process has been carried out using ten PE header analysis tools with various options. The research on prediction is carried out using the machine learning algorithms. Advanced Logistic Regression, KNN and SVM are giving the accuracy of 74%, 91% and 99% accuracy respectively. The research could be further extended with various datasets and various attributes for the ensemble of the machine learning algorithms.

REFERENCES

[1] Islam, R., Tian, R., Batten, L.M., Versteeg, S.: Classification of malware based on integrated static and dynamic features. J. Netw. Comput. Appl. **36**, 646–656 (2013).

[2] Tang, K., Zhou, M.T., Zuo, Z.-H.: An enhanced automated signature generation algorithm for polymorphic malware detection. J Electron. Sci. Technol. China **8**, 114–121 (2010)