Project Report

On

MALWARE DETECTION USING MACHINE LEARNING TECHNIQUES

Submitted by

Rameez Raza(182IT018)

Under the Guidance of
Dr. Bhawana Rudra
Dept. of Information Technology, NITK, Surathkal

Date of Submission:22-03-2019

Department of Information Technology
National Institute of Technology Karnataka, Surathkal.
2018-2019

Contents

Abstract

Malware infection happens when malware, or malignant programming, penetrate your PC. Malware is a sort of programming made with the purpose of harming the injured individual's PC, taking private data or keeping an eye on a PC without the assent of the user. The most well-known classifications of malware are Trojans, infections, spyware, ransomware, adware, rootkits, worms, and so on. Each of these malware types has differing abilities. From keeping an eye on your online exercises to backing off or securing your PC, malware can unleash devastation on your PC in the event that you don't insure yourself against them. Since more individuals are associated with the web than any time in recent memory, programmers are deceiving unwary clients into downloading malware. The Windows OS is one of most ordinarily utilized OS and furthermore most guess to malware contaminations. In our undertaking, we have overviewed and actualized AI based procedure to group document as kind or malevolent with high precision and low computational power. Out of 56 crude highlights in our dataset, after the change, we have chosen just 28 highlights. We connected different Machine learning calculations and accomplished exactness up to 99%. Also, we dissected false positives and false negatives to see how to document header esteems control perniciousness of the record. Additionally, we attempted to alter PE header estimations of a couple of malignant records to check our classifier Accuracy.

Introduction

Malware is defined as software or piece of code which can infiltrate or damage a system without the owner's consent. It is becoming increasingly difficult to detect malware using old signature-based methods as most of the current malware are either polymorphic wherein each copy of virus uses a different key or are metamorphic wherein each new version uses non-cryptographic obfuscation and thereby making it more difficult to get signature. The basic idea of any machine learning task is to train the model based on some algorithm and perform classification or predict new values. Training is done on the input dataset and the model is built which is then subsequently used to make predictions. Malware detection is a classification problem. We can train a program to recognize whether a piece of software is a malware or not and thus we can then detect later that whether a given file is malware or not.

Most of the current malwares are difficult to detect using old signature-based methods as they use various polymorphic or metamorphic approaches to generate new signatures and evade detection. Signature based detection has high detection accuracy but is limited to the signature database, hence requires frequent updates. The major drawback is that it provides an attack window time (time between a vulnerability discovery to signature update or patching) equal to the sum of detection time, signature generation time and updating time.

Another approach is behavior-based also referred to as heuristics-based analysis. In this method, the actual behavior of malware is observed during its execution, looking for the signs of malicious behavior like modifying host files, registry keys, establishing suspicious connections. By itself, each of these actions cannot be a reasonable sign of malware, but their combination can raise the level of suspicion of the file. There is some threshold level of suspicion defined, and any malware exceeding this level raises an alert. Although this approach provides some level of effectiveness, it is not always accurate, since some features can have more weight than others. Non-signature based detection can raise false positive or false negative result. Non-signature based detection techniques can also detect unknown, zero-day and modern malware. To take these correlations into account and provide more accurate detection, machine learning methods can be used.

Literature Survey

A learning model to detect maliciousness of portable executable using integrated feature set. The paper explains in details about the collection, prepossessing, feature selection and analysis using six machine learning techniques and is able to achieve accuracy to the tune of 60%-98%. However, the paper didn't deliberate on reasons why few features were dropped and why classifier gave false positives or negatives on test data. The paper also does not highlight the data set which deviated from the normal course. The data set was not tweaked and tested for abnormal PE header values.

Analysis of Machine Learning Techniques used in Behavior-based Malware Detection**.** The performance comparison of 5 different classifiers was also presented. The overall best performance was achieved by J48 using the term frequency-weight without feature selection data set, with a recall (true positive rate) of 95.9%, a false positive rate of 2.4%, a precision (positive predictive value) of 97.3%, and an accuracy of 96.8%. The analysis of the tests and experimental results concluded that this proof-of-concept is quite effective and efficient in detecting malware.

Objective

The objective of this project is to use machine learning techniques to detect whether the given dataset is malware or not based on the PE file header. The aim is to implement and compare the performance of different machine learning algorithms (KNN, Naive Base, Neural Network and Decision tree).

Dataset

In our project, we have used the dataset available online. **https://www.kaggle.com/amauricio/pe-files-malwares.** The dataset set consist of 138047 records. The dataset consists of 56 raw features out of which only 28 features were selected. Out of the selected 28 features, 5 were converted to Boolean and 1 was transformed to a value between 0 to 16. The collected dataset was divided into training and testing sets, containing 90% and 10% of the samples respectively in training and test set.

Methodology

The key identifier of malware is that it either changes the control stream or do change to the information structure. These progressions impact memories get to design by the myogram. Our exploratory assessment centers around Portable executable (PE) document's header fields esteem. We removed all header information from PE informational index.

We classified the heaviness of each component in order to guarantee that just imperative highlights that guide in our investigation are at last considered. The proposed learning display for malware location has four noteworthy targets as recorded beneath:

1. Feature set extraction from PE header.
2. Select relevant headers and modify/derive few features thereby enhancing the detection capability of malware. In the paper, they have considered only one classifier to find feature importance but we used two classifiers and combined their results to find relative importance.
3. Application of machine learning techniques on the complete feature set and develop a model for classification.
4. Apply test data on the model and verify the result.

Additionally, we analyzed false positives and false negatives to understand file header values of malicious files and we also tried to modify PE header values of few malicious files to check our classifier accuracy.

Broadly, malware detection using machine learning involves the following major steps:

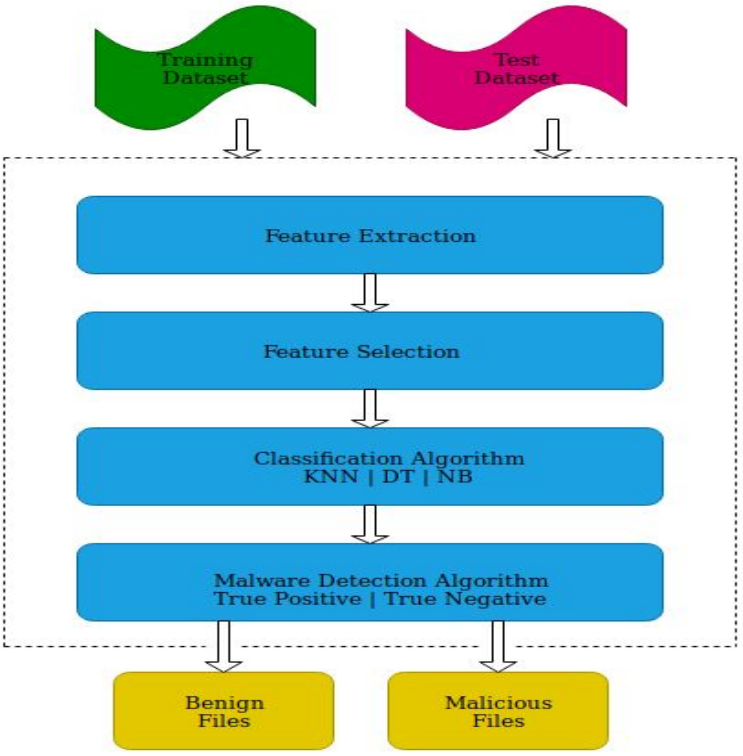

Fig: 1 Flowchart of methodology

1. The training dataset is taken and a set of features are extracted based on training scores.
2. The machine learning model is trained on this training dataset based on features selected in step 1.
3. The trained model is applied to test dataset and the accuracyof the model is calculated.

Results

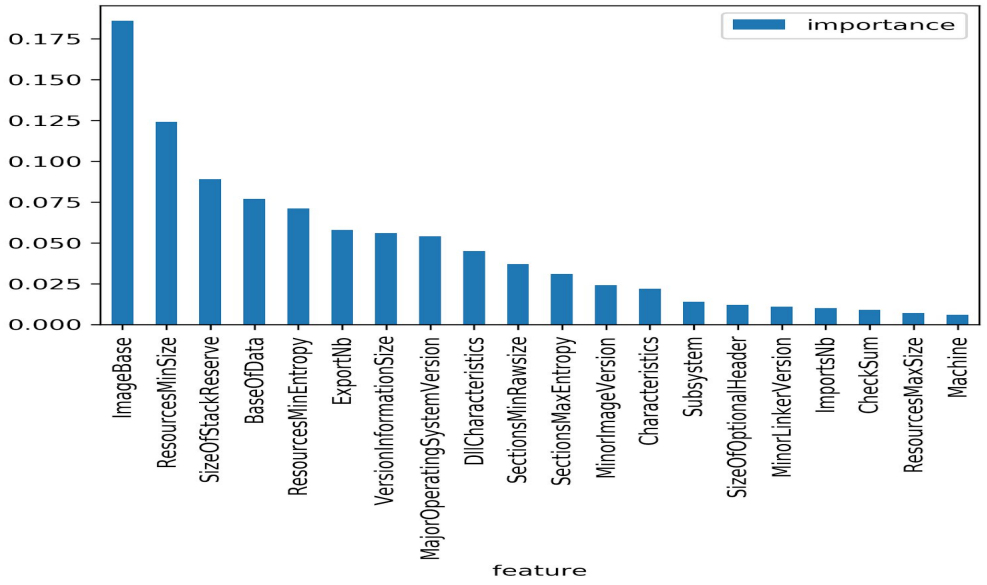

Fig: 2 Feature Importance

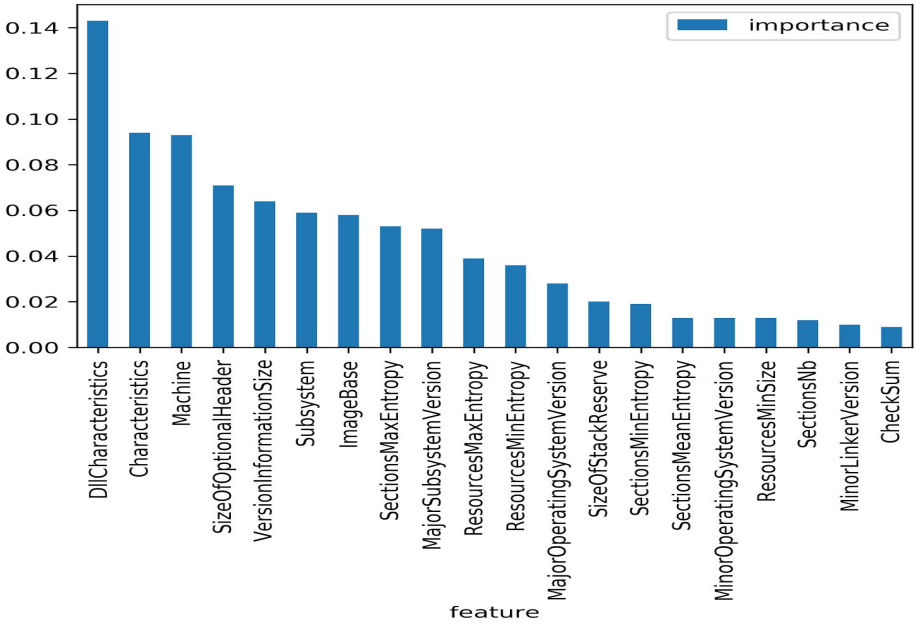

Fig: 3 Feature Importance using combined algorithm

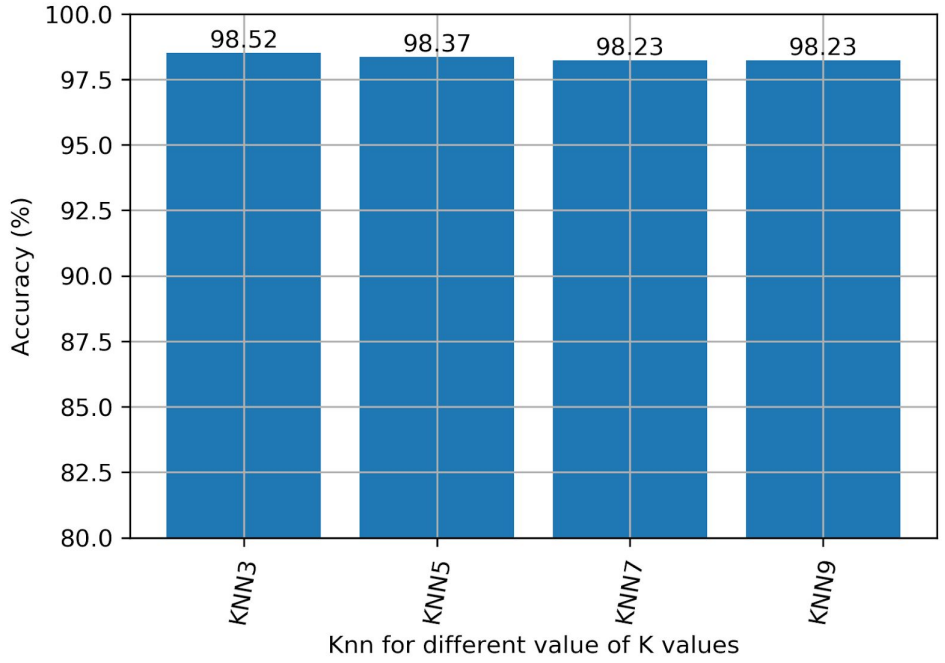

Fig: 4 KNN algorithm for different values of K

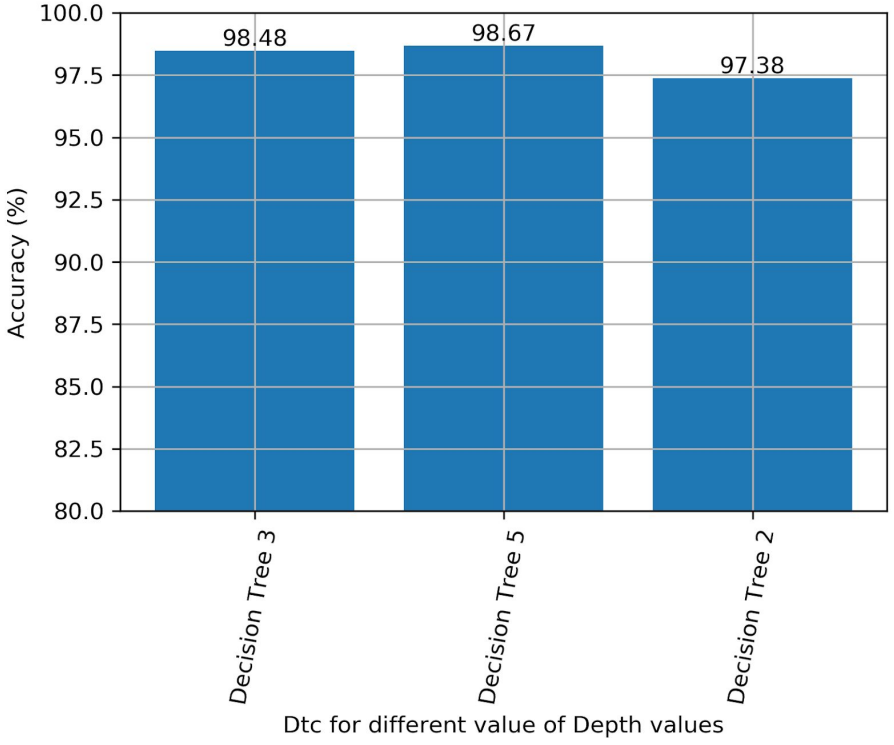

Fig: 5 Decision Tree algorithm for different values of depth

Algorithm	Accuracy	Precision	Recall	F-score	TN	TP	FP	FN
KNN	0.98	0.95	0.98	0.97	9472	4128	75	130
Decision Tree	0.98	0.95	0.98	0.97	9484	4111	92	118
Bernoulli Naive Bayes	0.95	0.88	0.93	0.90	9456	3686	517	146
Multinomial Naive Bayes	0.93	0.84	0.90	0.87	9499	3411	792	103
Neural Net	0.96	0.90	0.95	0.92	9408	3890	313	194

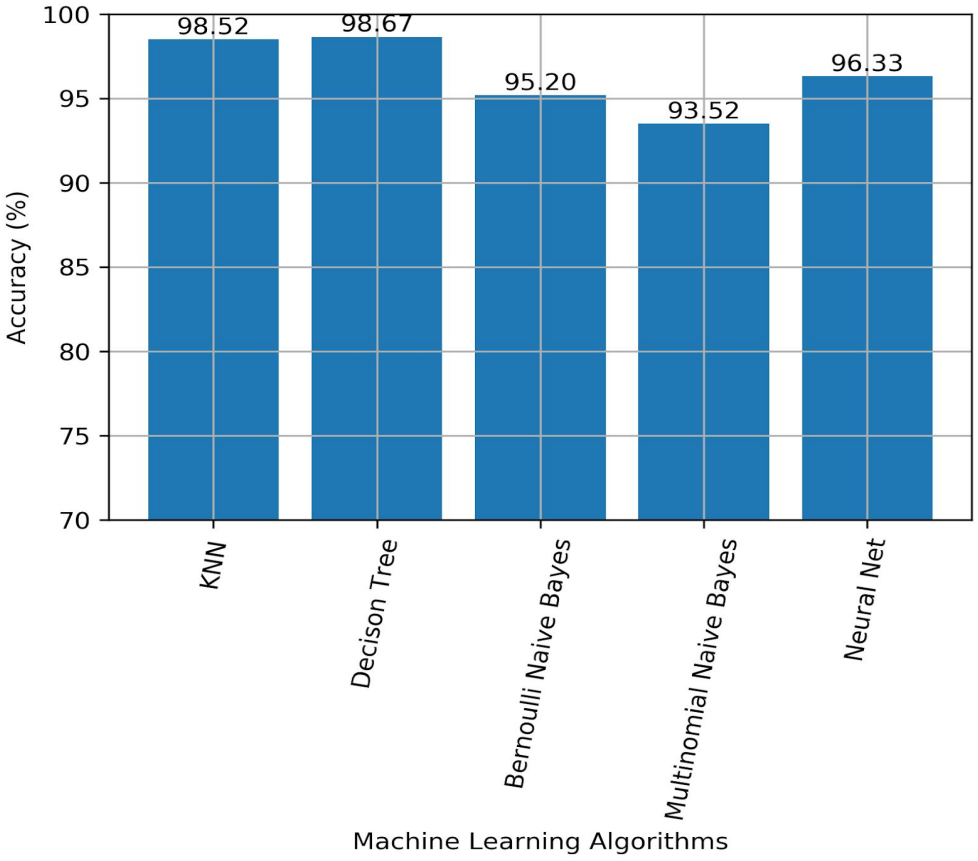

Fig: 6 Machine Learning Algorithm Result

Conclusion

1. The proposed machine learning technique used a static analysis technique to extract the features which have low time and resource requirement than dynamic analysis. Better classification accuracy can be achieved by building malware classifier using header fields' value alone as the feature.
2. The real-world scenario is unpredictable and can be different than the experimental environment so to protect a sensitive system from malware, it is not advisable to use only headers values based classifier. For example, a carefully crafted malware would have a benign header and malicious payload hidden in the body of PE file.
3. We have experimented with PE file format.

References

[1] Ajit Kumar, K S Kuppusamy, and Aghila Gnanasekaran. A learning model to detect maliciousness of portable executable using an integrated feature set. In the Journal of King Saud University - Computer and Information Sciences, 01 2017.

[2] D. Gavriluț, M. Cimpoeşu, D. Anton, and L. Ciortuz. Malware detection using machine learning. In 2009 International Multiconference on Computer Science and Information Technology, pages 735–741, Oct 2009.

[3] Swapnaja Hiray Smita Ranveer. Comparative analysis of feature extraction methods of malware detection, June 2015.

[4] Mihai Christodorescu and Somesh Jha. Static analysis of executables to detect malicious patterns. In Proceedings of the 12th Conference on USENIX Security Symposium - Volume 12, SSYM'03, pages 12–12, Berkeley, CA, USA, 2003. USENIX Association.