Malware Detection Using Machine LearningAlgorithms

Buddhadev Pusti
Center for Cybersecurity Systems and NetworksAmrita Vishwa Vidyapeetham, Amritapuri, India

***Abstract*—Malware is a critical security risk on Internet today.Malware is a set of programs designed to damage Internet- connected devices such as servers, computer resources, networks. Criminals are using Malware to send spam and to steal personal, financial, business information. Malware detection is the primary tool to stop unauthorized access of sensitive information. These days Windows OS is the most commonly used Operating System worldwide(77% to 88.8%) also it is the most targeted OS by malware attackers. In this paper detection of malware is done by simple observation of Portable Executable(PE) headers. In this paper, I use four methodology: 1. collect the data- set from https://www.kaggle.com/c/malware-detection /data 2. usean ExtraTreesClassifier for feature importance 3. use a "most frequent" strategy for baseline model 4. use Random Forest classification algorithm as a benchmark model.**

My data-set contains 140849 benign samples and 75503 malware samples. In the data-set, the feature "legitimate" has values "0" and "1", defines valid and malware files respectively. My experiments to detect malware by Portable Executable(PE) headers have a precision score of 98% and an f1-score of 98%. My experiments indicate that it is easy to detect malware filesby observing Portable Executable(PE) headers.

***Index Terms*—Malware, Malware detection, Portable Executable(PE) headers, Internet-connected devices, security**

I.INTRODUCTION

Malicious software is an integral component of many security breaches. Malware is growing exponentially and has been causing a huge amount of financial damages. Some of the malware are Computer viruses, Worms, Ransomware, Spyware, Adware, and so on. A computer virus is a type of malicious program designed in such a way that, it can spread the whole internal system or spread from one computer to another computer by replicating the malicious codes itself when the user executes the infected program. After the successful replication of a computer virus, it can destroy data or corrupt the whole file system. The worm is a type of malware that replicates itself without human interaction and damage computer internal resource. A ransomware attack is the most serious cyber threat that encrypts the data files or locked the computer until the victim pays the demanded money within some specific time. In 2019 the damaged cost due to the Ransomware attack was 120 Billion Dollars and it is predicted thatin 2021 the damaged cost due to the Ransomware attack will be 210 Billion dollars [1]. Spywares are type malicious code that secretively installed on our computer and after activation of spyware, it can monitors user behavior on the Internet and also can steal user passwords, keystrokes. Adware is unwantedmalicious software that displays irrelevant advertisements on a computer its objective is to monitor user web surfing activities or may create remote communication for transferring useractivities or install other malicious software and make the system vulnerable. The top 10 real world malware is showing as the following:

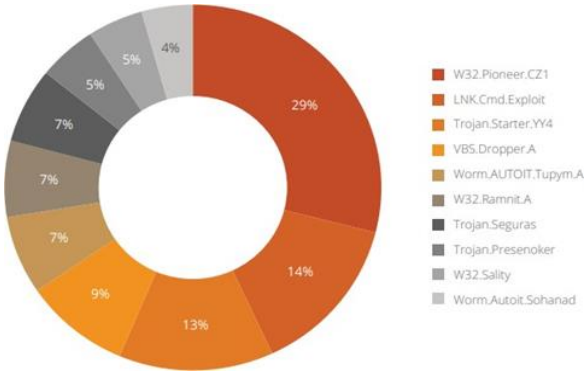

Figure 1: Top 10 Malware [2].

In this paper, I use a faster method to detect malware orbenign file by simply observing the properties of Portable Executable(PE) headers. For example, "DllCharacteristics" feature describes the characteristics of a dynamic link library and "Characteristics" features have an enumeration value that allows associated member's values to show some characteristics of the file. For example, it has some constants that define the respective file as an executable file or system file, etc. "BaseOfCode" feature characterizes the address of the beginning-of-

code area relative to the picture base when the picture is stacked into memory.

In summary, this paper has the following contributions:

Use PE-Header-Parser with the help of Extra-TreesClassifier to extract the metadata from PE headers and collects the 10 most important features that uniquely detect malware.

Conduct several classification algorithms and analysis evaluation metrics

Report that Random Forest classification is the best classification algorithm to solve detection of malware problem.

A. Overview of PE File Format

PE file header is a windows-based file format used by windows executable files(.exe files), Dynamic Link Library(DLL)file format, object code file format in 32 bit or 64 bit WindowsOS. This format is one type of data structure that wraps the relevant information required for the loader of Windows OS to address executable codes. PE header controlling exactly where to store different sections of a program what type of DLLfiles to load in memory .The basic structure of PE format as following figure [3].

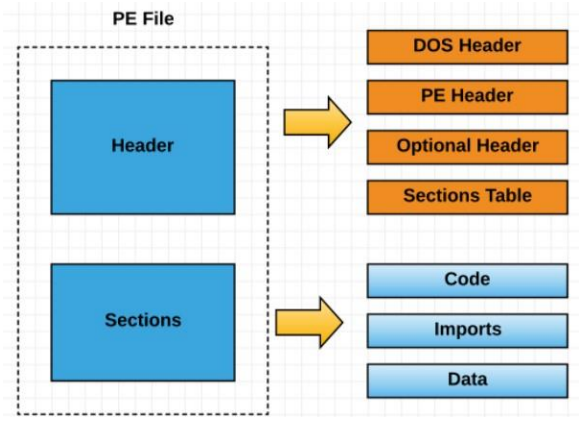

Figure 2:PE file Format structure.

PE record header format is made up of MS-DOS Header, PE Header, Discretionary Header, Segment Table. MS-Dos header involves the primary 64B of the record. All .exe files starts with the magic number 0x54AD known as DOS header. PE header contains the metadata of file some of them are Machine, Number of Sections, TimeDateStamp, PointerToSymbolTable, Number of Symbol Table, Characteristics, Size of Optional Header. The optional header contains the details of the loader. Optional headers have three parts as Standard COFF Fields, Windows Specific Fields, Data Directories. In this header contains metadata like dllCharacteristics, SizeOfImage, Checksum, SizeOfCode, BaseOfCode etc. Section Table storesimages of an executable file. Section table contains metadata like Virtual Address, SizeOfRawData, PointerToRawData.

PE file section header typically contains Code, Data, and Imports sections. The code section contains the executable code for a particular application. The data section typically stores initialized variables for an application. The imports section gives information about the actual functionality of a program or application. A combination of import sections is known as linking. There are 3 types of linking:

Dynamic linking: Where libraries are copied into the executable image when the program is executed(at therun time).

Static linking: Where all library routines that are used inthe program are copied into the executable image.

Runtime linking libraries are only copied into the executable image when they are needed.

II.RELATED WORK

Detection of malware is an ongoing challenge as day byday malware types are becoming powerful and continuously increasing. Previous works to detect malware [4] [5] are also similar in motivation and area of applications but my work is different in the analysis of validation techniques, evaluation metric between Dummy-Classifier and Benchmark model. [4] presented malware location utilizing PE header, in that work they utilize random forest classifier and accomplish 99.51% on accuracy score, they accomplish 99.24% accuracy by decision tree, by basic neural systems with a single coveredup layer accomplish 99.21% on accuracy. [5] presented a modern system for hardware-assisted malware discovery basedon persistent checking and classifying virtual memory get to designs utilizing machine learning strategies. They utilize 3 classifiers like logistic regression, SVM, and random forest algorithm. The system encompasses a discovery rate of 99.0% with less than 5% false positives. [6] introduced Hardware- Based Malware Detection, in this work On dataset 70%-30% splitting for training and testing is used. This technique givesa maximum of 90% accurately detect malware on differentML classifiers(BaysNet, J48, SGD, Jrip, MultiLperc, OneR, REPTree, SMO). [7] presented behavior-based

malware dis- covery, In that work, they collected 220 special malware and 250 unique kinds of software. In this work, they utilize five classifiers like k-Nearest Neighbors(kNN), Gullible Bayes, J48 choice tree, Bolster Vector Machine(SVM), Multilayer Perception Neural Network(MLP). After connected each dataset to these five calculations they get the generally best executionby the J48 Choice Trees calculation with the recall score of 95.9%, a FPR score of 2.4% and precision score of 97.3%,and precision of 96.8%. Paper [8] proposed a novel method to predict malware for the cloud by the concept of Semi-Supervised Transfer Learning(SSTL). They used the Microsoftmalware classification challenge dataset and XGBoost classification algorithm and reported 96.9% of detection accuracy.

III.METHODOLOGY

Dataset And Features

Dataset [9] was obtained from Meraz'18 - Annual Techno Cultural festival of IIT Bhilai association with Malware Security partner Max Secure Software. Dataset has 216352 rows and 58 columns. In the dataset, the feature "legitimate" has two values "0" and "1", represents benign and malware files. Within the dataset, no. of benign files and malware files are 140849 and 75503 respectively. Table 1 lists the features and description of features. Figure 3 shows top 20 important features by using ExtraTreesClassifier.

Table 1: Dataset feature's details [10]

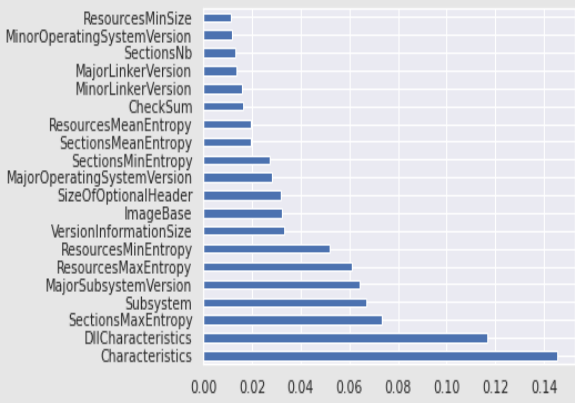

Figure 3: Top 20 Important Features

Evaluation Metrics

To detect actual Malware, we have to use some evaluation metric to know how many actual malware classes I can detect by my classification model since my target is to detect all actual malware classes. For this imbalanced dataset, Precision will be the most important evaluation metric.

Precision:

The precision metric demonstrates the proportion of expectations accurately predicted to be positive over the fullnumber of positive predictions. Precision formally calculated as follows:

$$Precision = \frac{TP}{TP + FP}$$

Precision is the proportion of malware files that are correctlyidentified. Also known as Positive Predictive Value(PPV). Some other important metrics are F-Measure, Accuracy, and Recall.

F-Measure:

This metric decides how well the prescient show can accurately anticipate positive values whereas taking into thought both FN and FP.

$$F - Measure = \frac{2TP}{2TP + FP + FN}$$

F-Measure translated as a weighted average of the precision and recall. When F1-Measure score is 1 then it reaches best value and worst score at 0

Recall:

Recall measures the percentage of the actual positive classthat is correctly classified.

$$Recall = \frac{TP}{TP + FN}$$

Accuracy:

This metric decides how well the predictive model can makecorrect predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Accuracy means the proportion of the total no of predictions that are correct over all kinds of predictions. Here FP, FN,TP, and TN speak to False Positive, False Negatives, True Positives, and True Negatives separately.

Experimental Setup

In this experiment, I used the Miniconda package, some libraries like Jupyter Notebooks, Numpy, Pandas, Scikit-learn and Matplotlib with python 3.8.5. I ran these experiments ona Dell Inspiron laptop

computer with a 2.50 GHz Intel Corei5 processor and 8GB of RAM with 1454.406 CPU MHz.

The Classifiers

In this work, I choose four classification algorithms as AdaBoost, Random Forest, Decision Tree. The parameter n estimators used as default while classification is performed.

AdaBoost:

An ensemble model that combines diverse weak classifiers(fairway better than arbitrary speculating such as small decision trees) on more than once adjusted adaptations of the training data. It has a few commonsense preferences like It is exceptionally quick, simple to actualize, and it can combine with any machine learning algorithm.

Random Forest:

Random Forest Classifier use sets of decision trees and selectsa subset of the training dataset and then chooses the most voted prediction as to the final result. Here input of eachdecision tree uses the original training dataset. I will use a Random Forest classifier for model validation mainly because it reduces the chance to overfit the model.

Advantages:

It reduces the chance of overfitting since we use many estimators(parameter n estimators), and that helps to im- prove overall better performance.

It can handle binary features, categorical features, and numerical features.

The random forest algorithm use to find the most important features from the training dataset.

Gradient Boosting:

Gradient boosting classifier could be a bunch of learning algorithms that combine numerous weak learning models and make a successful prescient model. The foremost advantage ofthis classifier that it is successfully classifies complex datasets.

Decision tree:

A tree-like structure takes a decision using relevant features byconstructing a series of if-else statements repeatedly until the tree separates data. The space complexity of this algorithmis very small and it can easily handle multi-class classification without any algorithm changes. The main disadvantage increases the change to overfitting.

IV.RESULTS AND ANALYSIS

Features	*Description*
ID	Unique identification number of data samples
md5	Defines Individual machine ID.
SizeOfOptionalHeader	Defines the size of optional header used for executable files.
Characteristics	It have a enumeration value allow associated members values show some characteristics of the file. For example it have some constants that defines the respective file is executable file or system file etc.
MajorLinkerVersion	Characterizes linker major version number
MinorLinkerVersion	Characterizes linker minor version number
SizeOfCode	Characterizes estimate of the code (content) segment, or the whole of all code segments
SizeOfInitializedData	Characterizes the estimate of the initialized information area, or the entirety of all such segments in the event that there are different information areas.
SizeOfUninitializedData	Characterizes the estimate of the uninitialized information segment (BSS), or the entirety of all such areas in the event that there are numerous BSS areas.
AddressOfEntryPoint	Characterizes the address of the passage point relative to the picture base when the PE record is stacked into memory.
BaseOfCode	Characterizes the address of the beginning of code segment relative to the picture base when the picture is stacked into memory.
ImageBase	Gets the favored address of the primary byte of the picture when it is stacked into memory.
FileAlignment	Gets the arrangement figure (in bytes) that's utilized to adjust the crude data of areas within the picture record.
MajorOperatingSystemVersion	Characterizes the major form number of the specified working framework.
MinoroperatingSystemVersion	Characterizes the minor adaptation number of the desired working frame- work.
MajorImageversion	Characterizes the major adaptation number of the header.
MinorImageVersion	Characterizes the minor form number of the header.
MajorSubsystemVersion	Gets the major form number of the subsystem.
MinorSubsystemVersion	Gets the minor version number of the subsystem
SizeOfImage	Gets the estimate (in bytes) of the image, counting all headers, as the image is stacked in memory.
SizeOfHeaders	Gets the combined measure of an MS DOS stub, PE header, and area headers adjusted up to a numerous of File Alignment.
CheckSum	Gets or sets the value of the data for the checksum calculation.
Subsystem	Describes the subsystem requirement for the image.
DllCharacteristics	Describes the characteristics of a dynamic link library.
SizeOfStackCommit	Gets the estimate of the stack to commit.

SizeOfHeapCommit	Gets the estimate of the nearby heap space to commit.
LoaderFlags	Characterizes optional header format
NumberOfRvaAndSizes	Gets the number of data-directory passages within the leftover portion of the PEHeader. Each depicts a area and measure.

Table 1 shows the results of different algorithms with different evaluation metrics, confusion matrices. In this experiment RandomForestClassifier is gives best precision score, f1-score compare to DecisionTreeClassifier, GradientBoostingClassifier, AdaBoostClassifier. The benchmark model [4] achieve 99.51% on average for accuracy but they did not discuss the evaluation metric precision or f1-scores. I used the

Table 2: Results of different classification algorithms

Classifier	TN	FP	*FN*	TP	Precision	f1-score
RandomForestClassifier	27974	189	247	14861	0.9874	0.9855
DecisionTreeClassifier	27811	352	367	14741	0.9766	0.9761
GradientBoostingClassifier	27751	412	637	14471	0.9723	0.9650
AdaBoostClassifier	27504	659	783	14325	0.9560	0.9520

most frequent strategy for the baseline model and achieved a score of 65.08%.

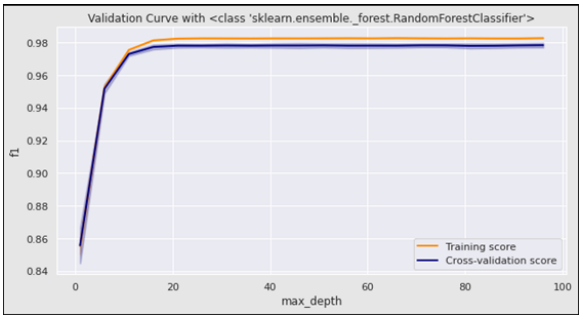

Figure 4: Validation Curve

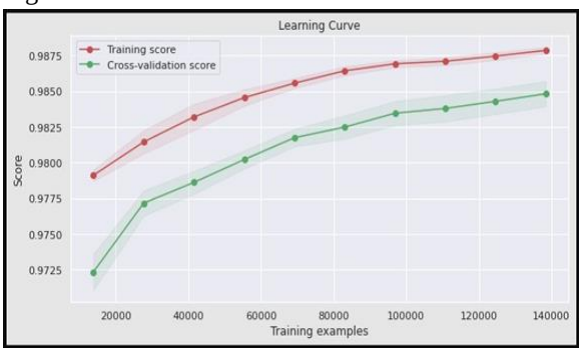

Figure 5: Learning Curve

V.CONCLUSIONS AND FUTURE WORK

ExtraTreesClassifier is used for the most important features.

Different classification algorithms like AdaBoost, Random- Forest, DecisionTree, GradientBoosting are used to classifyall the malware files and I present comparisons of different classification algorithms with evaluation metrics, confusion matrices. The leading performance was accomplished by Random Forest Classifier with a precision score 98% and f1-score98%. However, this model has not achieved a 100% precision score, these restrictions are cleared out for future work, but this work can predict a high precision score of 98% malware files .I accept that there are a few other features that can be utilized as the key highlights to identify malware, such as extricating the Flags within the Characteristics areas of file header and discretionary header. The validation curve showed that my model is not overfitting and the learning curve perfectly shows that more data will help to get better performance. Experiments show this work is reasonably effective to detect malware files.

REFERENCE

[1] M. Humayun, N. Jhanjhi, A. Alsayat, and V. Ponnusamy, “Internet of things and ransomware: evolution, mitigation and prevention,” Egyptian Informatics Journal, 2020.

[2] S. K. Sahay, A. Sharma, and H. Rathore, “Evolution of malware and its detection techniques,” in *Information and Communication Technology for Sustainable Development*, pp. 139–150, Springer, 2020.

[3] C. Chebbi, Mastering Machine Learning for Penetration Testing: De- velop an extensive skill set to break self-learning systems using Python. Packt Publishing, 2018.

[4] F. Abri, S. Siami-Namini, M. A. Khanghah, F. M. Soltani, and A. S. Namin, “Can machine/deep learning classifiers detect zero-day malware with high accuracy?,” in 2019 IEEE international conference on big data (Big Data), pp. 3252–3259, IEEE, 2019.

[5] Z. Xu, S. Ray, P. Subramanyan, and S. Malik, “Malware detection using machine learning based analysis of virtual memory access patterns,” in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, pp. 169–174, IEEE, 2017.

[6] H. Sayadi, N. Patel, S. M. PD, A. Sasan, S. Rafatirad, and H. Homayoun, “Ensemble learning for effective run-time hardware-based malware

detection: A comprehensive analysis and classification," in 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), pp. 1–6, IEEE, 2018.

[7] I. Firdausi, A. Erwin, A. S. Nugroho, et al., "Analysis of machine learning techniques used in behavior-based malware detection," in 2010 second international conference on advances in computing, control, and telecommunication technologies, pp. 201–203, IEEE, 2010.

[8] X. Gao, C. Hu, C. Shan, B. Liu, Z. Niu, and H. Xie, "Malware classification for the cloud via semi-supervised transfer learning," Journal of Information Security and Applications, vol. 55, p. 102661, 2020.

[9] "Malware detection, meraz'18 - annual techno cultural festival of iit bhilai in association with meraz'18 malware security partner max se- cure software, https://www.kaggle.com/c/malware-detection/data, online accessed may, 2021."

[10] "System.reflection.portableexecutable: Microsoft peheader properties, https://docs.microsoft. com/en-us/dotnet/api/system.reflection. Portableexecutable.peheader?view=net-5.0, online accessed June 2021."