ENCRYPTED MALWARE DETECTION USING MACHINE LEARNING

ADITYA HEGDE (201IT105)
RAKESH KUMAR (201IT146)
SAURAV KUMAR (201IT157)

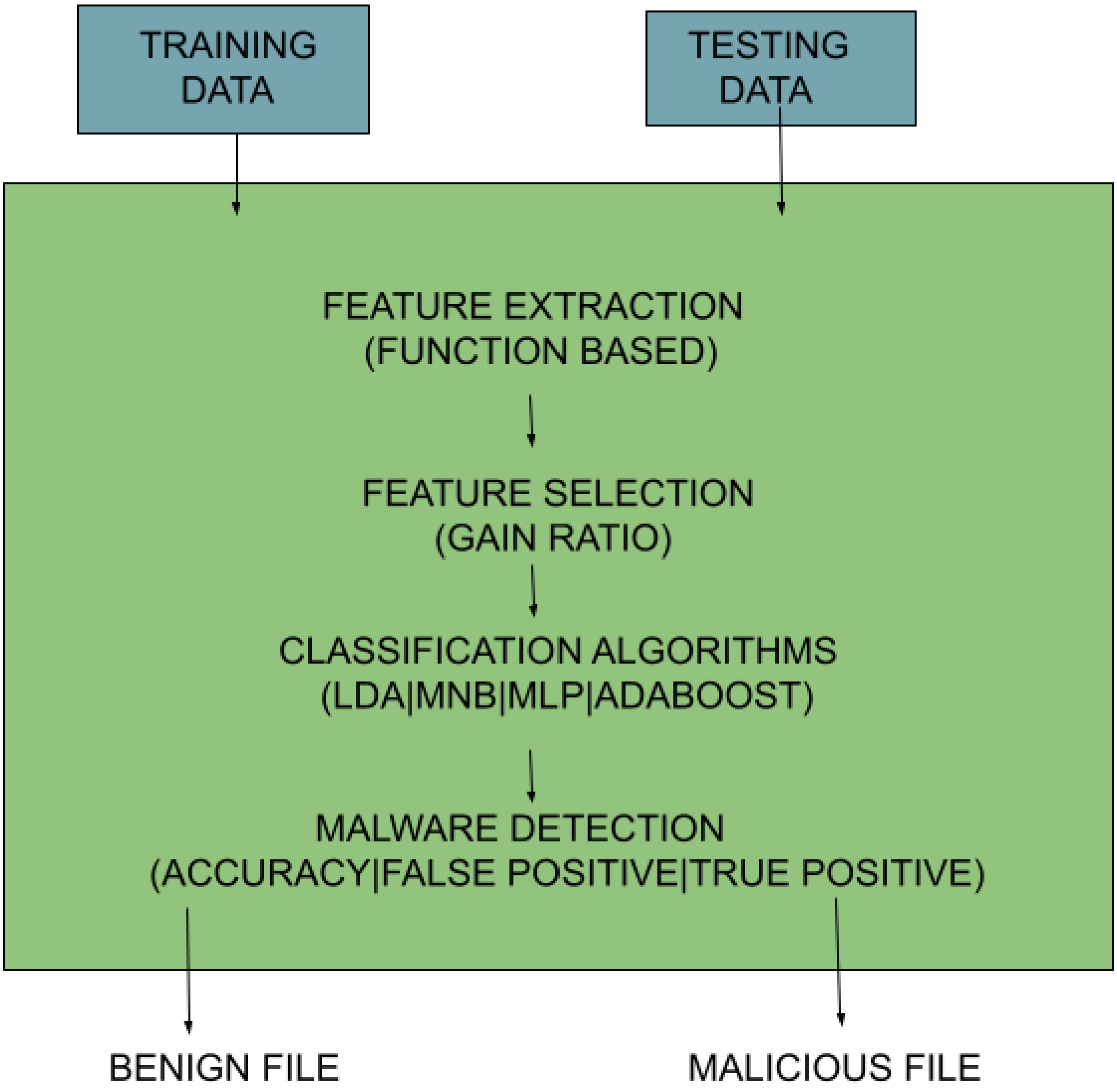
DATASET LINK:
https://drive.google.com/file/d/1vA7qAVm-8_9ppgrHuN8uAxxxLAIT1NZL/view?usp=share_link
NOTEBOOK LINK:
https://drive.google.com/file/d/1vA7qAVm-8_9ppgrHuN8uAxxxLAIT1NZL/view

1. INTRODUCTION

Malware is a software that is specifically designed to disrupt, damage, or gain unauthorised access to a computer system without its owner consent. It is one of the most commonly used vectors to propagate malicious activity and exploit system vulnerabilities. Signature-matching methods are not resilient enough to handle obfuscation techniques or to catch unseen malware types . Due to the huge amount of network traffic these days, the detection of malware needs to be automated.
The basic idea of any machine learning task is to train a model using an algorithm on a given input dataset and then use that model to make predictions or classify new data.Machine learning-based malware detection involves using algorithms to learn from past malware data and identifying new malware based on the patterns or characteristics found in the data.

Feature set extraction from PE header.

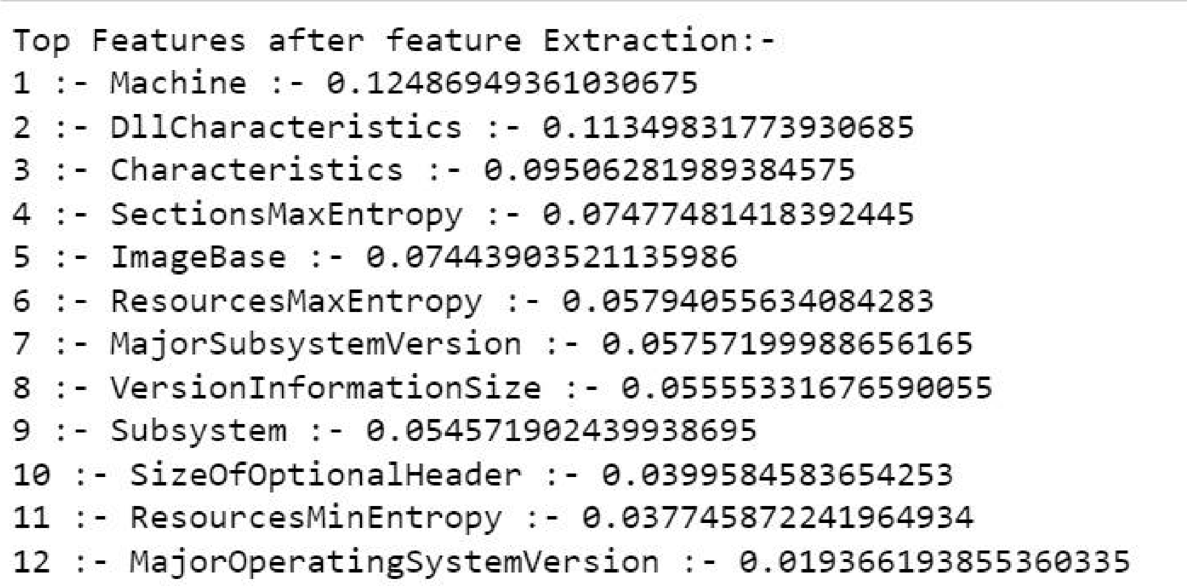
- Select relevant headers and modify /derive few features thereby enhancing the detection capability of malware.
- In paper, they have considered only one classier to and feature importance but we used two classifiers and combined their results to and relative importance.
- Application of machine learning techniques on the complete feature set and develop models for classification.
- Apply test data on the model and verify the result.



- Additionally, we analysed false positives and false negatives to understand header values of malicious les and we also tried to modify PE header values of few malicious les to check our classifier accuracy.
- Broadly, malware detection using machine learning involves following major steps:
- Training dataset is taken and a set of features are extracted based on training scores.
- The machine learning model is trained on this training dataset based on features selected in step1.
- The trained model is applied on a test dataset and accuracy of the model is calculated.

2. FEATURE EXTRACTION

There are two common approaches to extract features from malware: static analysis and dynamic analysis. Here, in our project, we are using static analysis method by examining PE header information.The dataset set consist of 138047 records.The dataset consists of 56 raw features out of which only 12 features were selected.Among model based feature selection methods (wrapper methods) tree based methods are easier to apply and without much tuning, it can also model non-linear relations. We have used Random Forest Classifier with all other default settings (scikit-learn implementation) to get feature importance for raw feature sets. It is meta es-timator that is a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-tting.By default Gini impurity is used to measure the quality of a split.



```
Top Features after feature Extraction:-
1 :- Machine :- 0.12486949361030675
2 :- DllCharacteristics :- 0.11349831773930685
3 :- Characteristics :- 0.09506281989384575
4 :- SectionsMaxEntropy :- 0.07477481418392445
5 :- ImageBase :- 0.07443903521135986
6 :- ResourcesMaxEntropy :- 0.05794055634084283
7 :- MajorSubsystemVersion :- 0.05757199988656165
8 :- VersionInformationSize :- 0.05555331676590055
9 :- Subsystem :- 0.054571902439938695
10 :- SizeOfOptionalHeader :- 0.0399584583654253
11 :- ResourcesMinEntropy :- 0.037745872241964934
12 :- MajorOperatingSystemVersion :- 0.019366193855360335
```

- We combined the results obtained from Random forest and Extra Trees Classifier to select important features.
- Selected features like Characteristics , DLL Characteristics, Resources- Mean Size, Resources MeanEnropy as they are the property of each file and describe the behaviour of the application.
- Features like SizeOfInitializedData , SizeOfHeaders, SizeOfStackReserve, SizeOf HeapReserve describe the memory role of an application.
- VersionInformationSize, Number of dll importes etc denes the others resources required

Reason for Dropping

- Features like Name, md5 , BaseOfData, checksum are dropped because these features have different values for each record in the dataset.
- Some of the features are having the values that are correlated like min, max and mean of the attributes so we drop those values which are low in relative importance as found out earlierResourcesMinSize, Sec-tionMaxRawsize, etc.

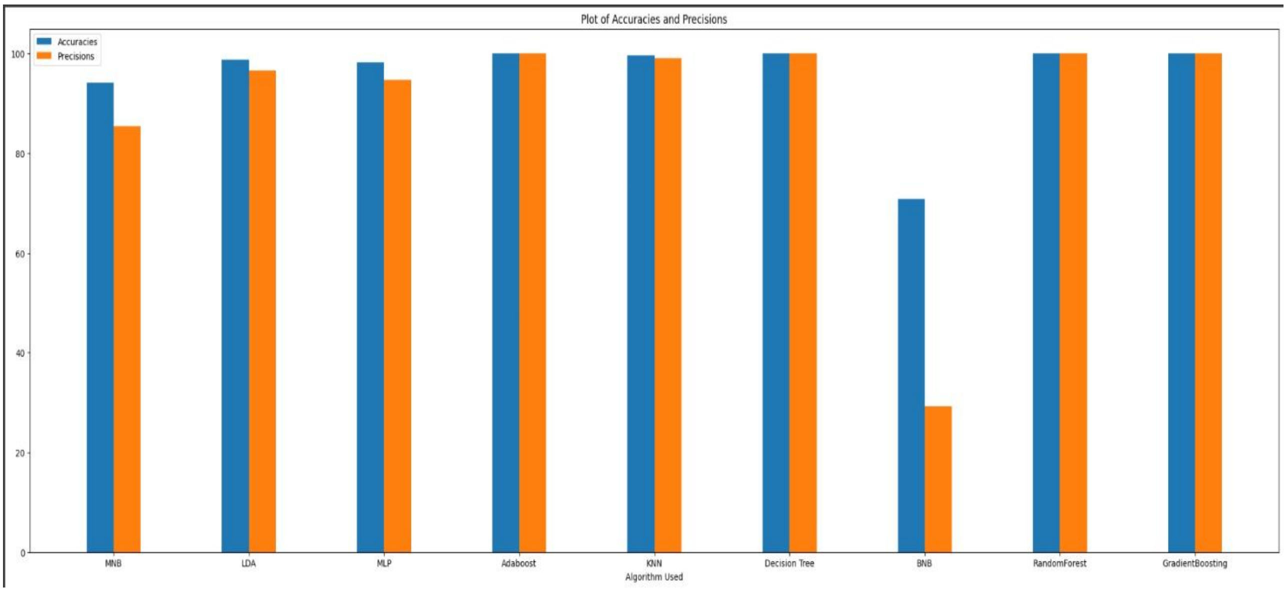
- Some features are dropped because they were uniformly distributed among malicious and benign les like OS version , linker version etc and don't deny the type of file.

3. PERFORMANCE EVALUATION

In our project, we have used the dataset available online. The collected dataset was divided into training and testing sets,containing 90% and 10% of the samples respectively in training and test set.The Training sample size is 124242 and Test samples is 13805.

Comparing results obtained using different algorithms

CLASSIFIER	ACCURACY	PRECISION	F-SCORE	RECALL
MLP	0.9288	0.8130	0.8675	0.9299
NB	0.9303	0.8350	0.8631	0.8932
LDA	0.9506	0.8783	0.9033	0.9297
ADBOOST	0-0.99 1-0.98	0-0.99 1- 0.97	0-0.99 1-0.98	0-0.99 1- 0.97
K Nearest Neighbour	0.997	0.990	0.995	0.997
Decision Tree	0.99	0.998	0.992	0.997
Naive Bayes	0.708	0.292	0.369	0.50
Gradient Boosting	1.00	1.00	1.00	1.00



4. CONCLUSION

- The proposed technique used static analysis techniques to extract the features which have lower time and resource requirements than dynamic analysis. Good classification accuracy can be achieved by building malware classifiers using header fields value alone as the feature.
- As of now ADABoost classifier has given most accurate and precise results for malware detection.