PE Malware Detection

Using Machine Learning

Mohamed Asem • Flothers Interview Task

Introduction

Malware analysis

- Basic static.
- Advanced static.
- Basic dynamic.
- Advanced dynamic.

Basic static

- No execution.
- Inspecting the PE File Format.
- Strings.
- DLL’s.
- Imported Functions.
- Signes of Packed Malware.
- Resources.

Objective

1. Collect Dataset
2. Extract Features
3. Train different ML models
4. Test with random samples
5. Build simple GUI

Dataset Collection

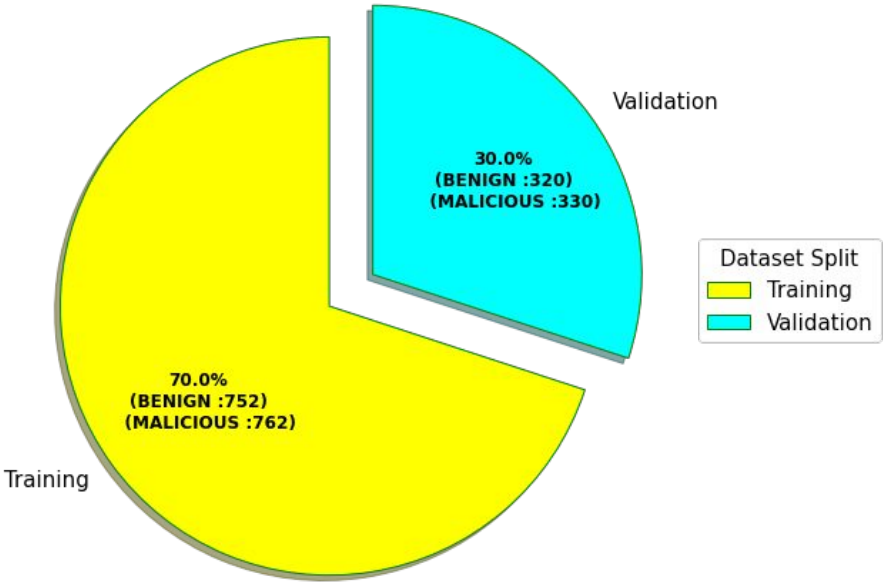
Benign Samples

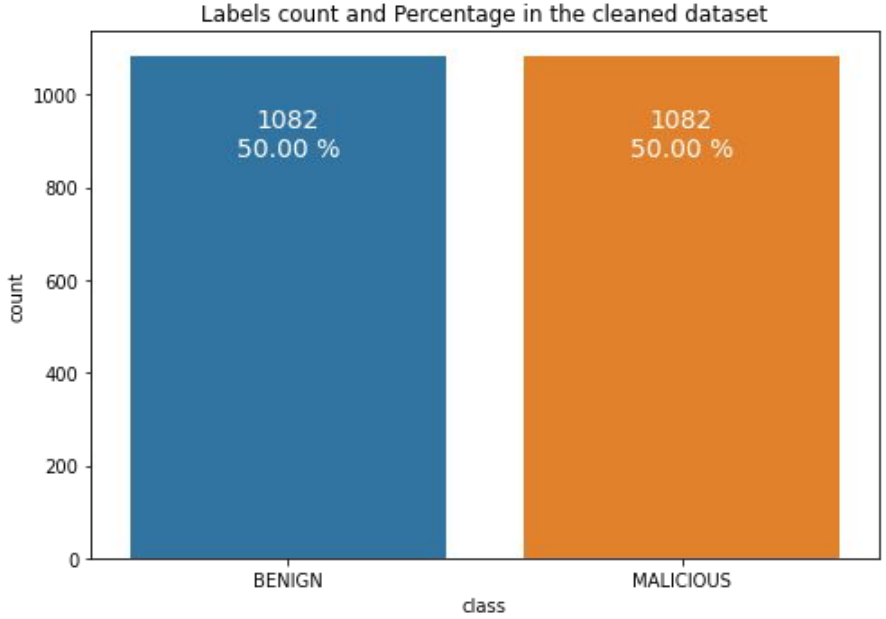
- https://github.com/bormaa/Benign-NET
- https://figshare.com/articles/dataset/Malware_Detection_PE-Based_Analysis_Using_Deep_Learning_Algorithm_Dataset/6635642

Malware Samples

- https://dasmalwerk.eu/
- https://figshare.com/articles/dataset/Malware_Detection_PE-Based_Analysis_Using_Deep_Learning_Algorithm_Dataset/6635642
- https://bazaar.abuse.ch/

Dataset Collection (statistics)





Extract Features

YARA rules

- PEID signatures
- Packer signatures
- Crypto signatures
- Anti-debug/Anti-VM
- Capabilities

PE Header

- Sections name (.rsrc, .txt, ……)
- Imported DLLs
- Imported DLLs imports (removed later for performance issues)

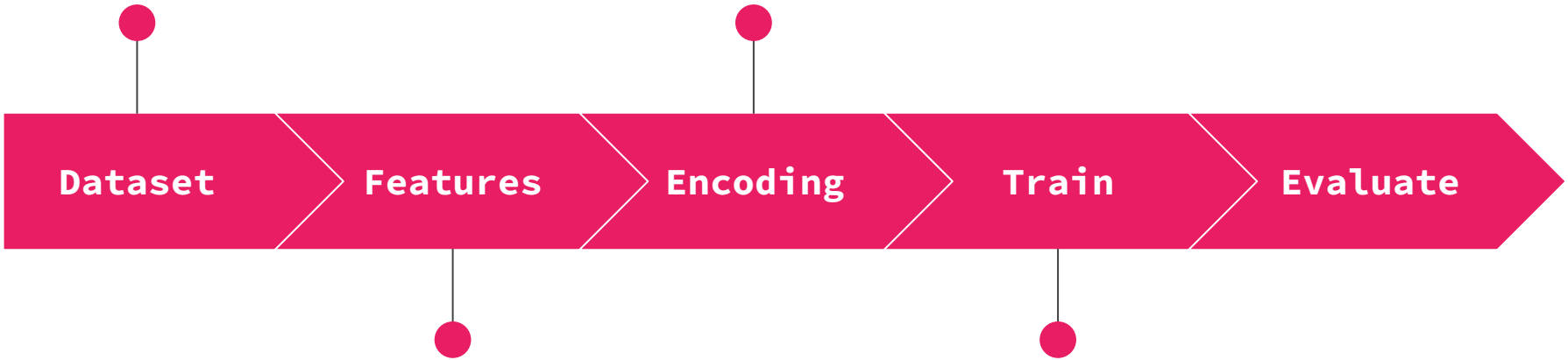
Training the Models

1082 Benign

1082 Malware

One-Hot-Encoding

859 features



5 Yara rules

2 PE format

70% of the dataset

7 ML classifiers

Dataset Split

Training

```
Number of 1s (MALICIOUS): 752
Number of 0s (BENIGN)   : 762
```

Validation

```
Number of 1s (MALICIOUS): 752
Number of 0s (BENIGN)   : 762
```

Random Malware-only for client environment testing

```
Number of 1s (MALICIOUS): 973
```

Machine learning models

XGBoost

Random Forest

Decision Tree

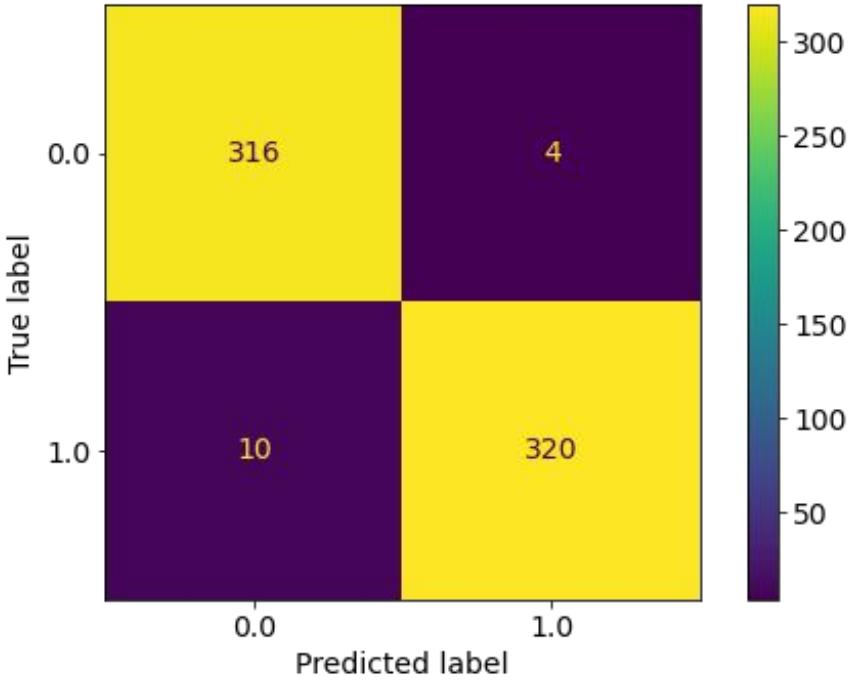
Adaptive Boosting

Naive Bayes

Stochastic Gradient Descent (SGD)

Multi-layer perceptron (MLP)

1- XGBoost

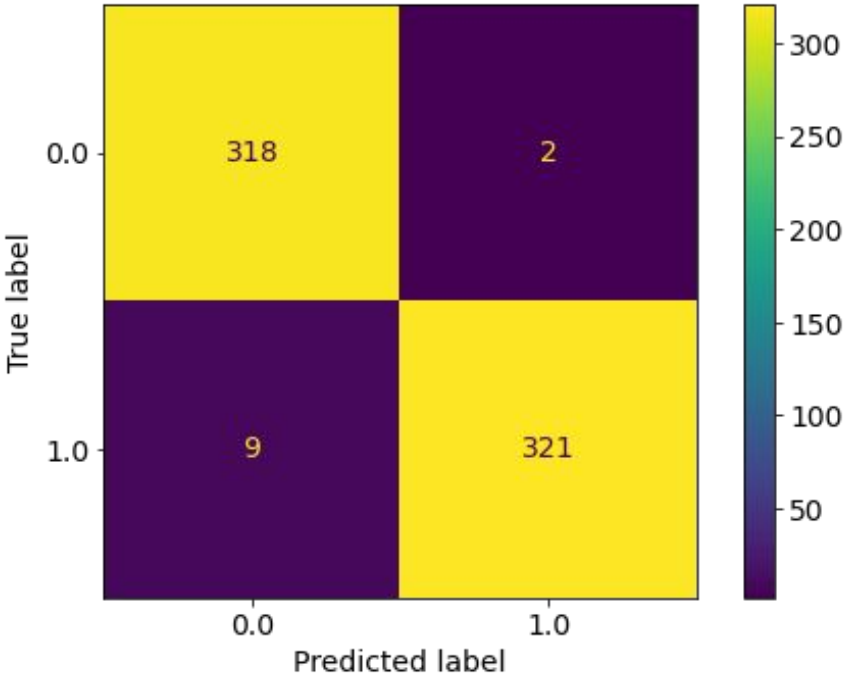


Detailed Report :

Metrics	Value
Sensitivity	0.96970
Specificity	0.98750
Precision	0.98765
Negative Predictive Value	0.96933
False Positive Rate	0.01250
False Discovery Rate	0.01235
False Negative Rate	0.03030
Accuracy	0.97846
F1-Score	0.97859
Matthews Correlation Coefficient	0.95709

Accuracy of the model: 97.85 %
F1_score of the model: 97.86 %

2- Random Forest

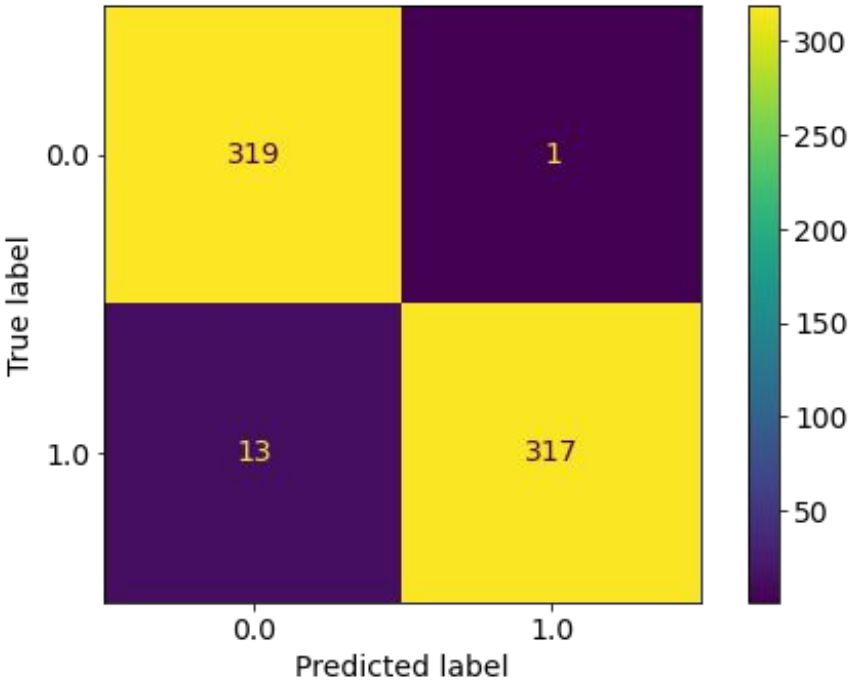


Detailed Report :

Metrics	Value
Sensitivity	0.97273
Specificity	0.99375
Precision	0.99381
Negative Predictive Value	0.97248
False Positive Rate	0.00625
False Discovery Rate	0.00619
False Negative Rate	0.02727
Accuracy	0.98308
F1-Score	0.98315
Matthews Correlation Coefficient	0.96638

Accuracy of the model: 98.31 %
F1_score of the model: 98.32 %

3- Decision Tree

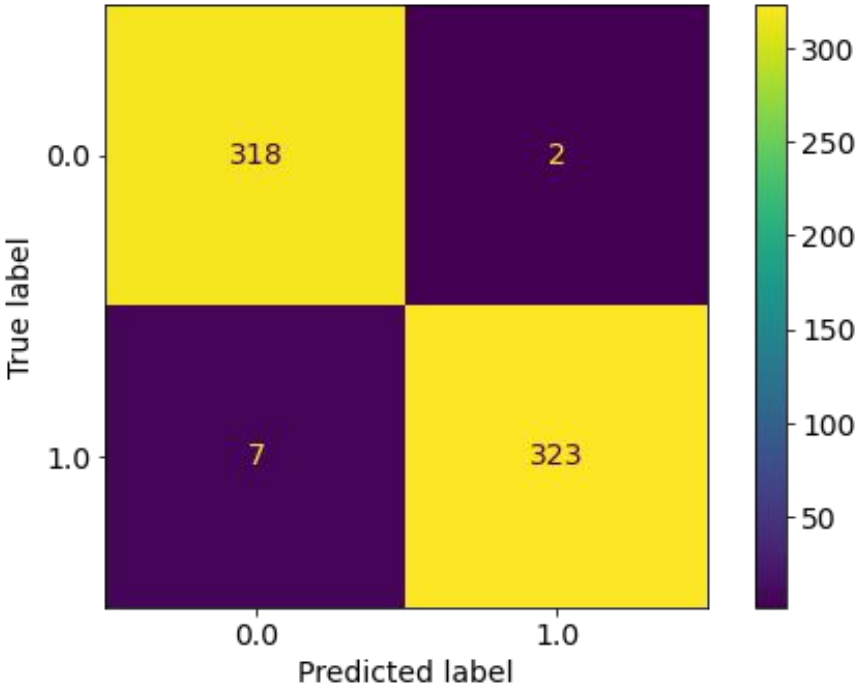


Detailed Report :

Metrics	Value
Sensitivity	0.96061
Specificity	0.99687
Precision	0.99686
Negative Predictive Value	0.96084
False Positive Rate	0.00313
False Discovery Rate	0.00314
False Negative Rate	0.03939
Accuracy	0.97846
F1-Score	0.97840
Matthews Correlation Coefficient	0.95759

Accuracy of the model: 97.85 %
F1_score of the model: 97.84 %

4- Adaptive Boosting

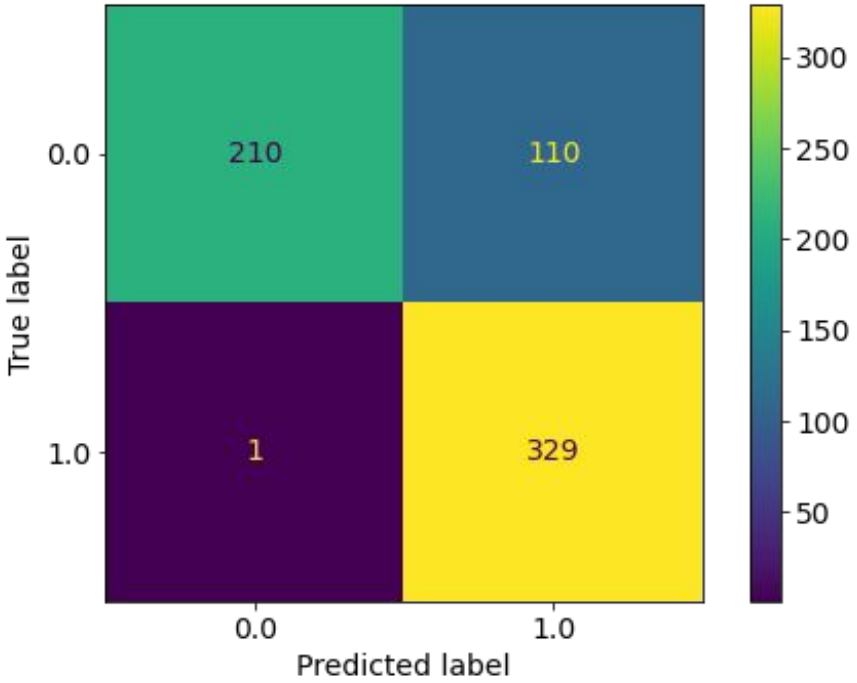


Detailed Report :

Metrics	Value
Sensitivity	0.97879
Specificity	0.99375
Precision	0.99385
Negative Predictive Value	0.97846
False Positive Rate	0.00625
False Discovery Rate	0.00615
False Negative Rate	0.02121
Accuracy	0.98615
F1-Score	0.98626
Matthews Correlation Coefficient	0.97242

Accuracy of the model: 98.62 %
F1_score of the model: 98.63 %

5- Naive Bayes

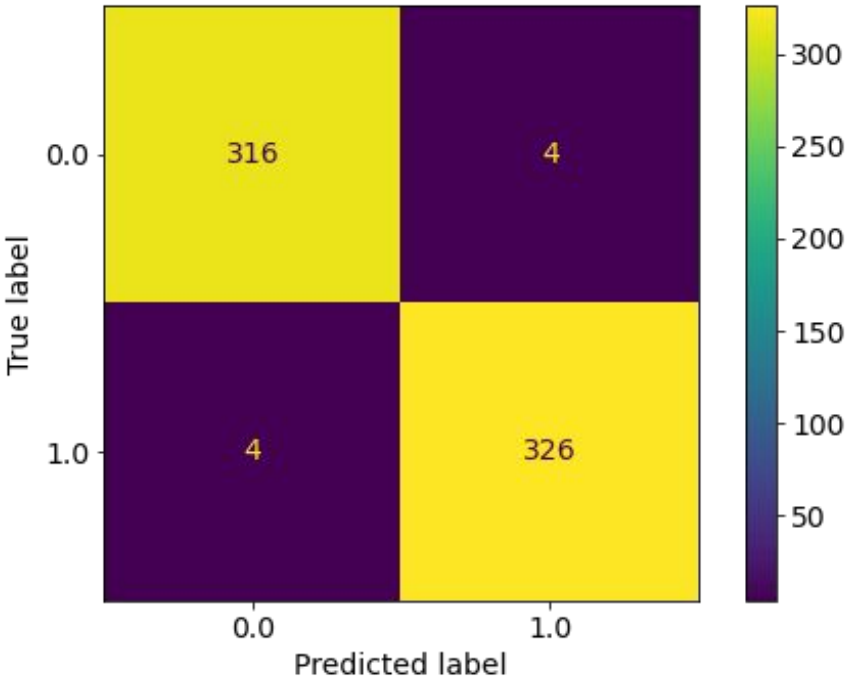


Detailed Report :

Metrics	Value
Sensitivity	0.99697
Specificity	0.65625
Precision	0.74943
Negative Predictive Value	0.99526
False Positive Rate	0.34375
False Discovery Rate	0.25057
False Negative Rate	0.00303
Accuracy	0.82923
F1-Score	0.85566
Matthews Correlation Coefficient	0.69746

Accuracy of the model: 82.92 %
F1_score of the model: 85.57 %

6- Stochastic Gradient Descent (SGD)

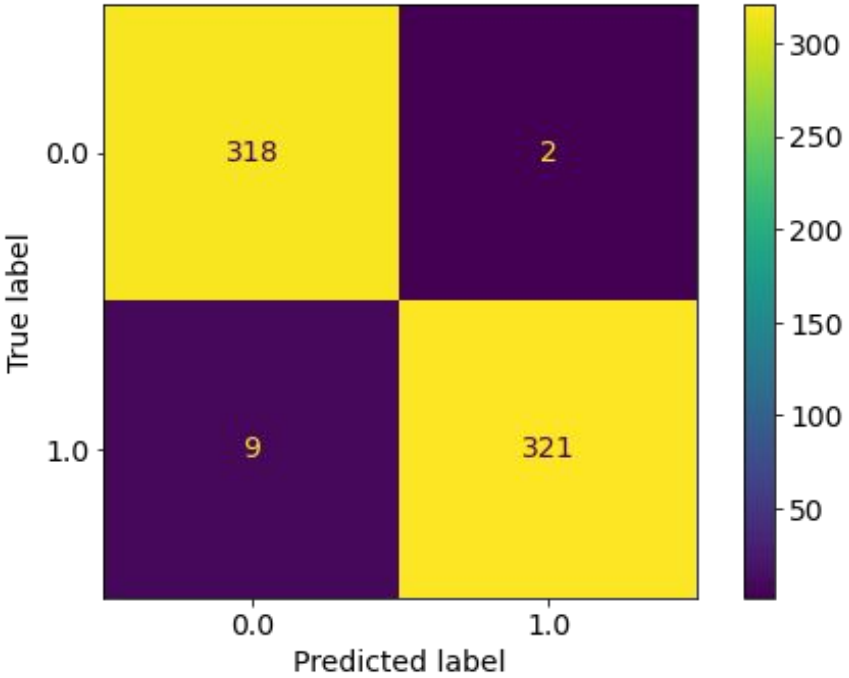


Detailed Report :

Metrics	Value
Sensitivity	0.98788
Specificity	0.98750
Precision	0.98788
Negative Predictive Value	0.98750
False Positive Rate	0.01250
False Discovery Rate	0.01212
False Negative Rate	0.01212
Accuracy	0.98769
F1-Score	0.98788
Matthews Correlation Coefficient	0.97538

Accuracy of the model: 98.77 %
F1_score of the model: 98.79 %

7- Multi-layer perceptron (MLP)

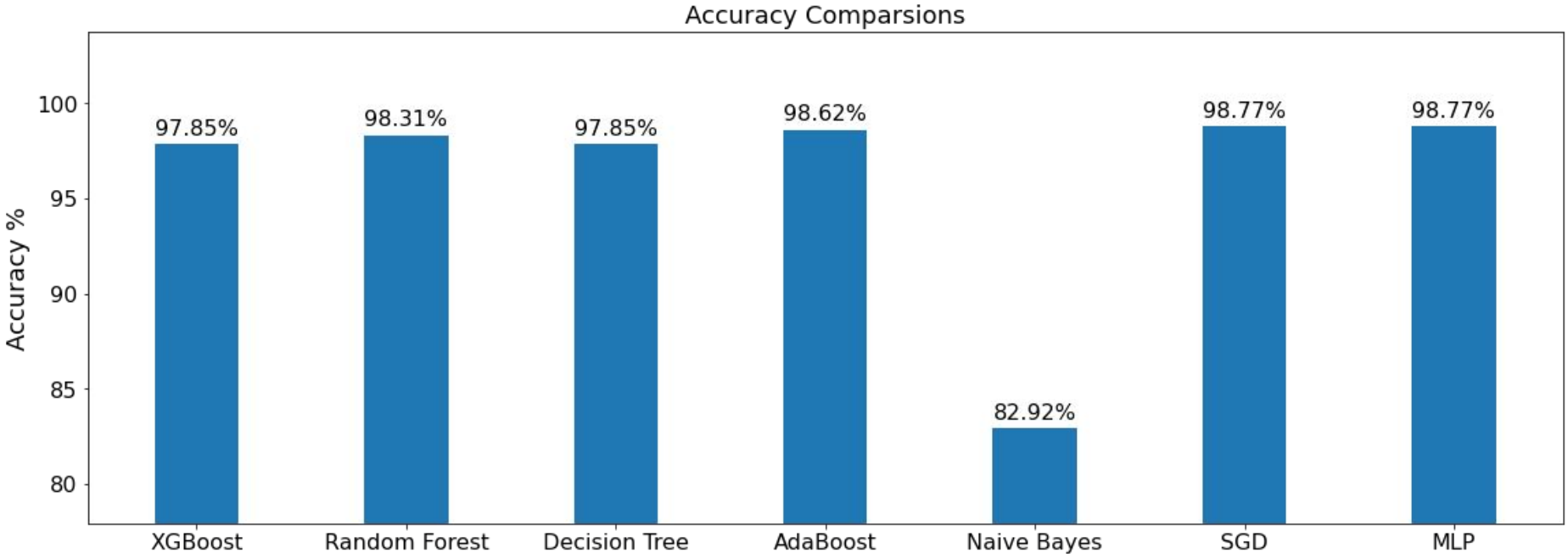


Detailed Report :

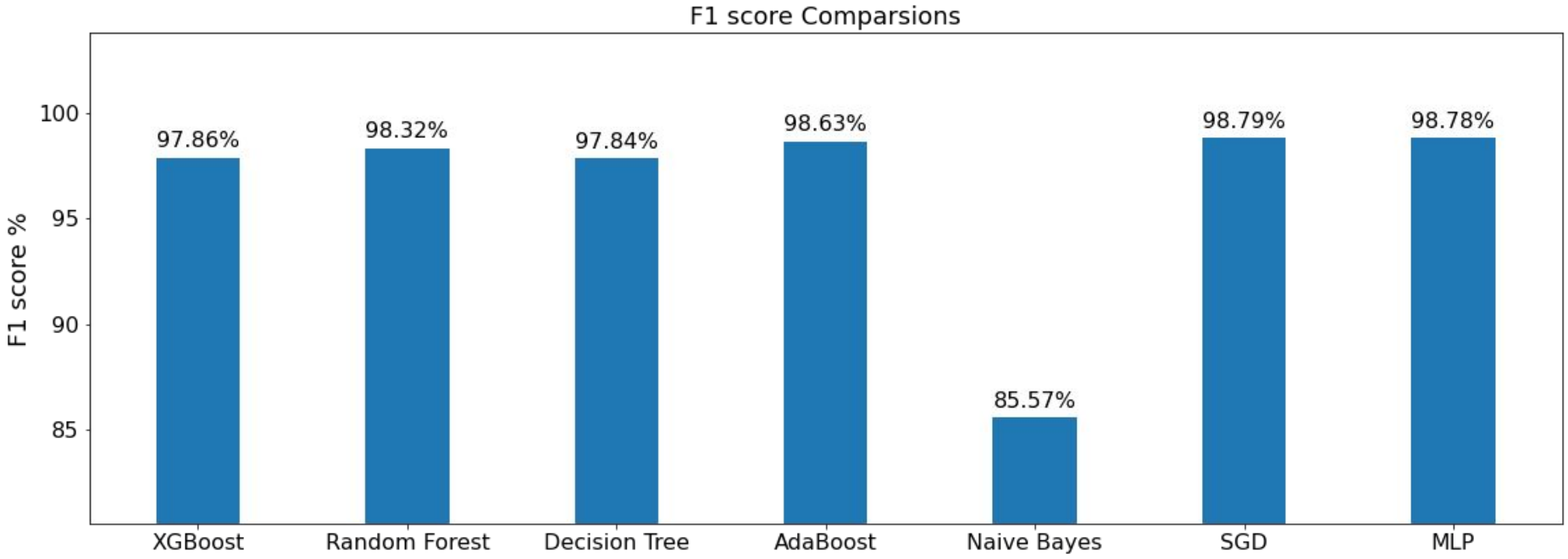
Metrics	Value
Sensitivity	0.97273
Specificity	0.99375
Precision	0.99381
Negative Predictive Value	0.97248
False Positive Rate	0.00625
False Discovery Rate	0.00619
False Negative Rate	0.02727
Accuracy	0.98308
F1-Score	0.98315
Matthews Correlation Coefficient	0.96638

Accuracy of the model: 98.31 %
F1_score of the model: 98.32 %

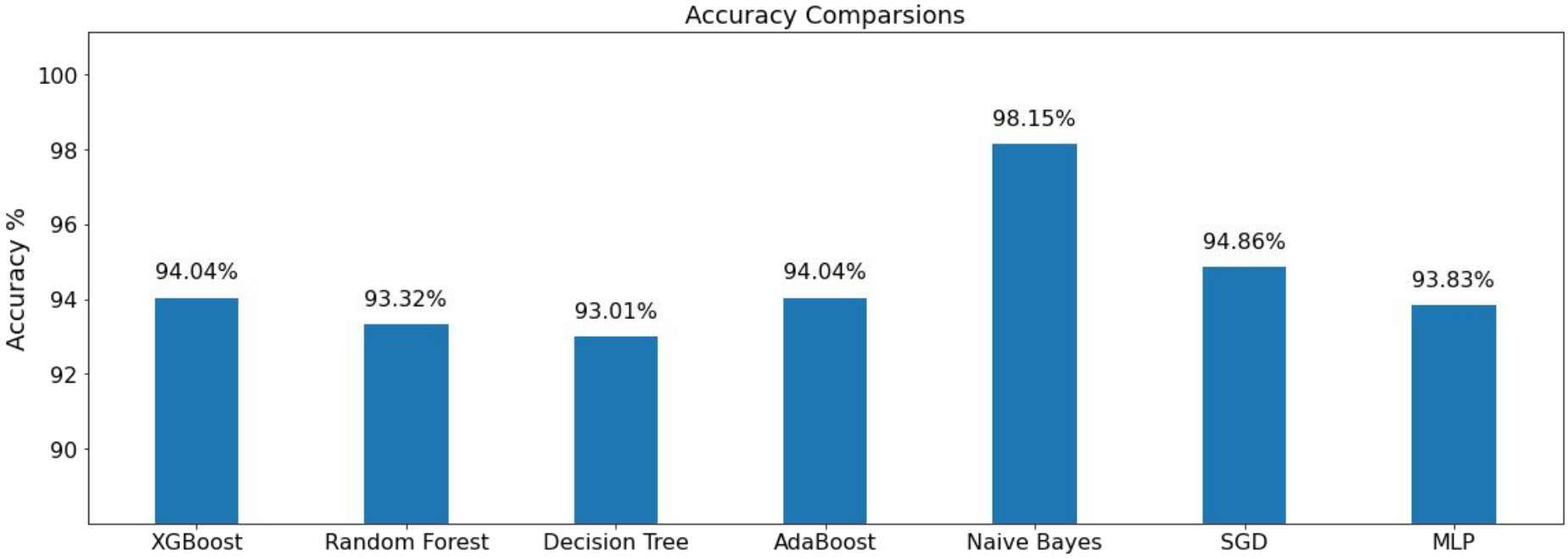
Accuracy Comparisons



F1-Score Comparisons



Production Environment Performance (973 Malware)



Simple GUI

User Scenario

---

window 1

Chose the file path for analysis

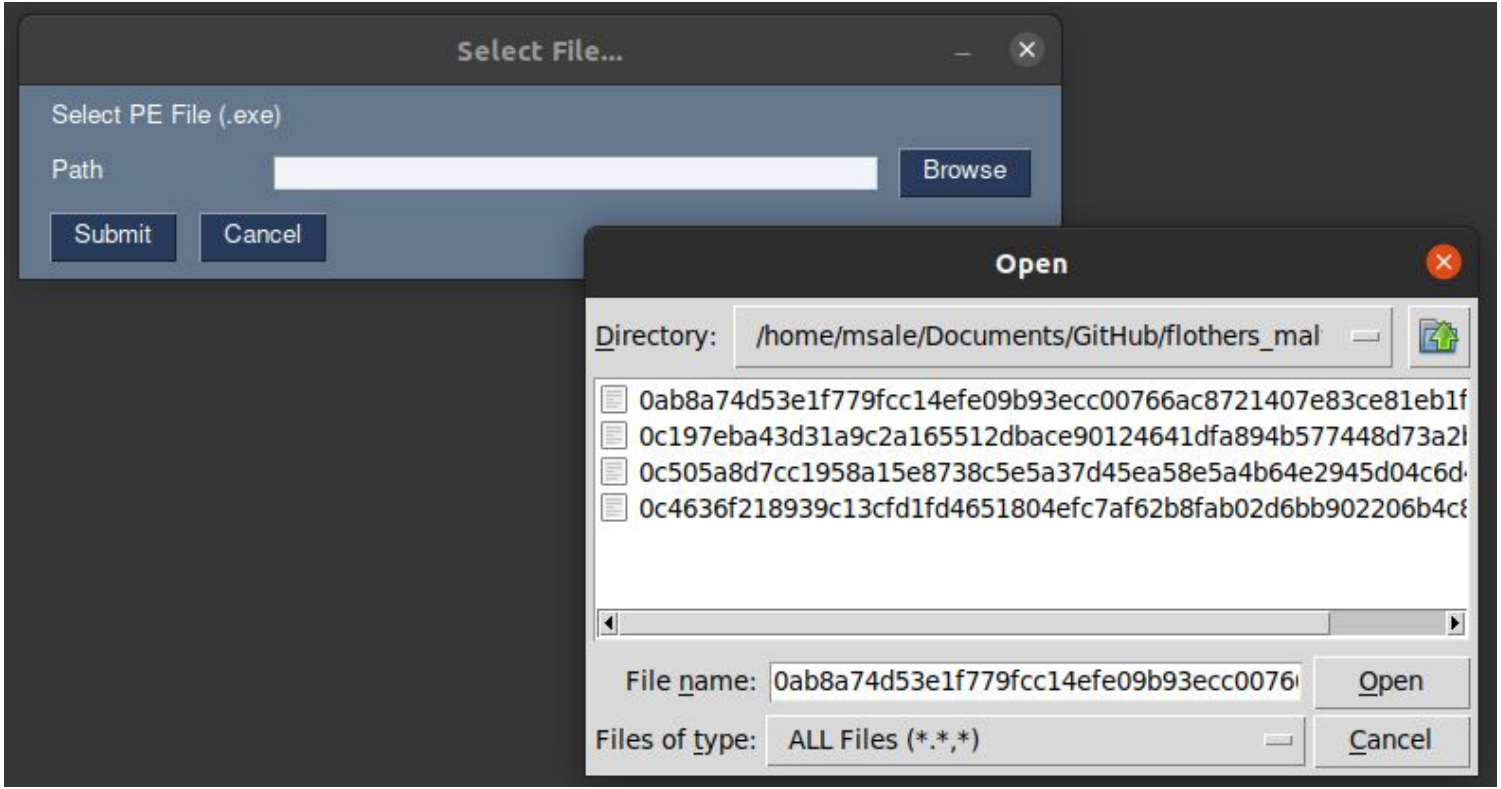
window 2

Progress bar represent the feature extraction and analysis

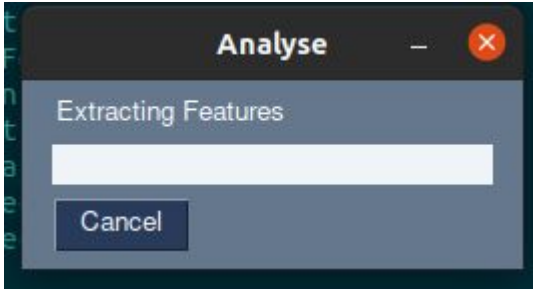
window 3

Print the final result of the models (benign or malicious)

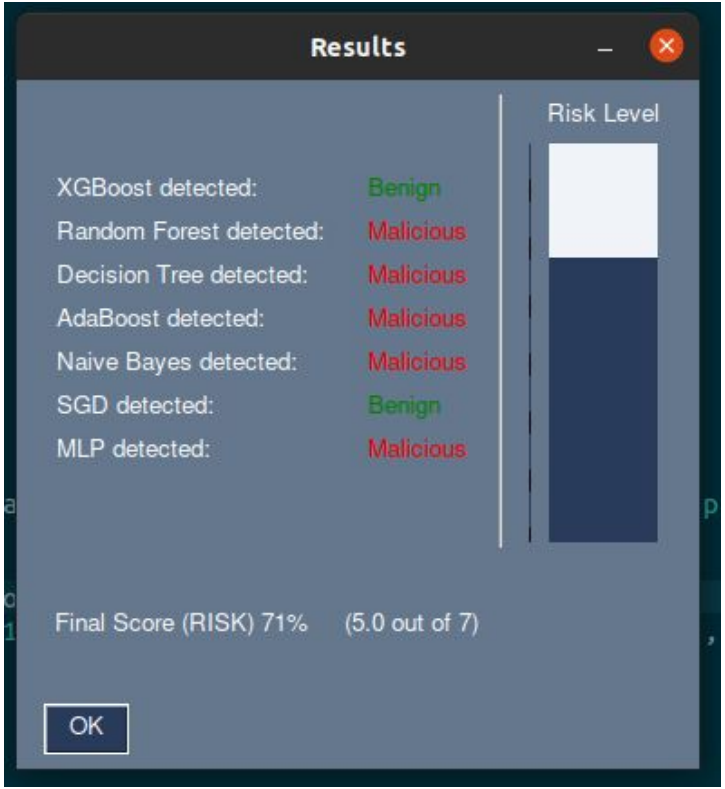
Window 1


Select File...
Select PE File (.exe)
Path
Browse
Submit
Cancel
Open
Directory:
/home/msale/Documents/GitHub/flothers_mal
0ab8a74d53e1f779fcc14efe09b93ecc00766ac8721407e83ce81eb1f
0c197eba43d31a9c2a165512dbace90124641dfa894b577448d73a2
0c505a8d7cc1958a15e8738c5e5a37d45ea58e5a4b64e2945d04c6d
0c4636f218939c13cfd1fd4651804efc7af62b8fab02d6bb902206b4c
File name:
0ab8a74d53e1f779fcc14efe09b93ecc0076
Open
Files of type:
ALL Files (*.*,*)
Cancel

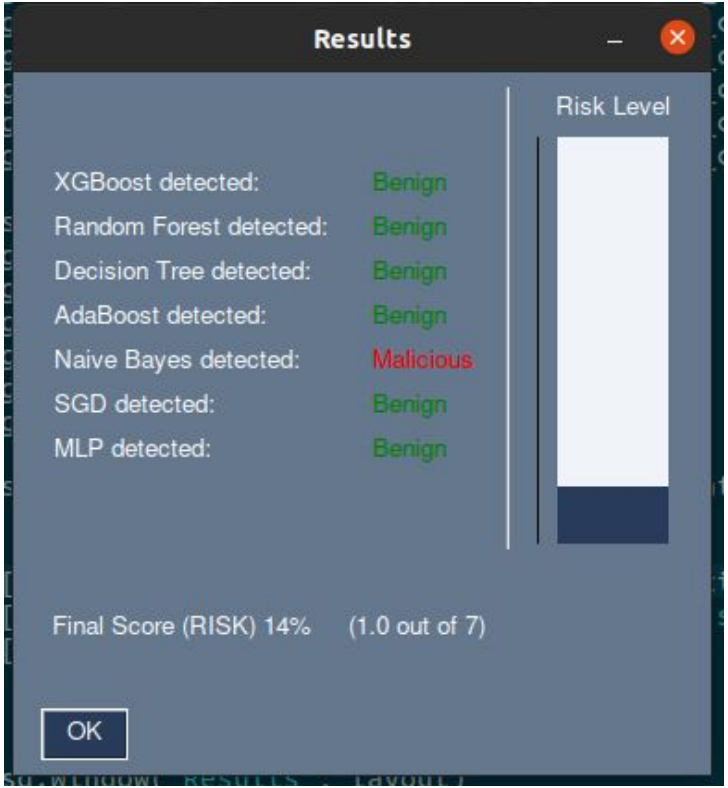
Window 2

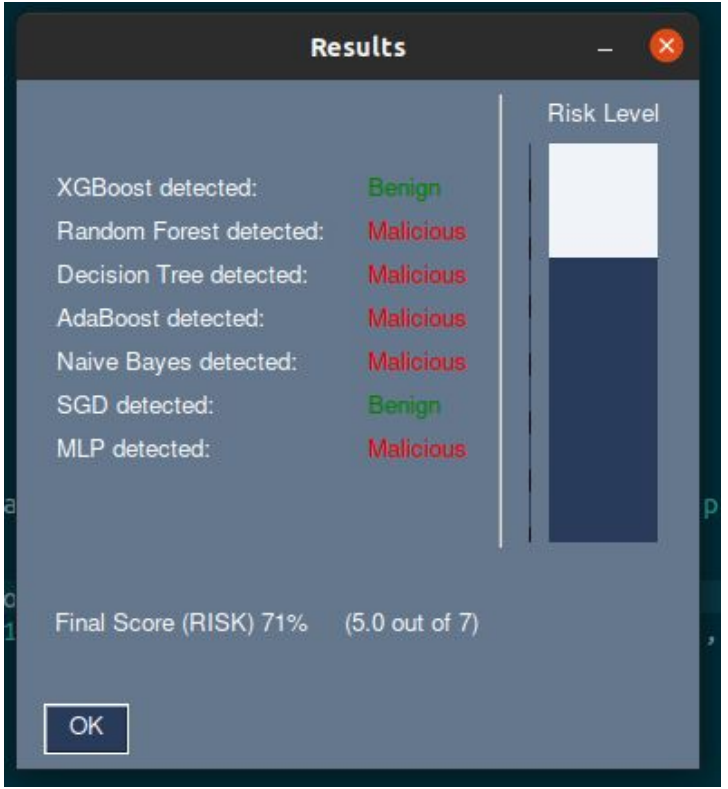


Window 3



Window 3 (Different results)





Limitation

1. Files used for training are not variant enough.
2. More features to be extracted needs more computational power.
3. More tests are required for the GUI and proper documentations.

Thanks
Q&A

Appendix

Detailed Metrics Calculations

Accuracy	$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
F1 Score	$F1 = \frac{2\times TP}{2\times TP+FP+FN}$
Matthews Correlation Coefficient	$MCC = \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Detailed Metrics Calculations

Precision	$Precision = \frac{TP}{TP+FP}$
Sensitivity	$Recall = Sensitivity = \frac{TP}{TP+FN}$
Specificity	$Specificity = \frac{TN}{FP+TN}$