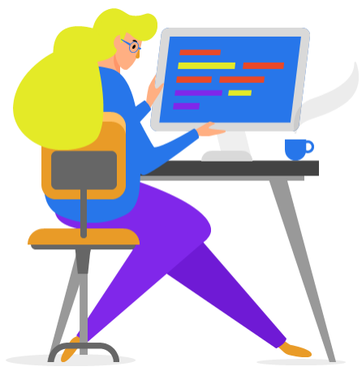
2334: MALWARE ANALYSIS WITH MACHINE LEARNING

Tan Carlven 1181200964
Dr Ng Hu, Dr Soo Wooi King

Abstract



Malware analysis is crucial in cybersecurity for identifying and eliminating malicious software. This research aimed to enhance efficiency and accuracy using machine learning techniques. A malware detection system was developed, training on a diverse dataset from GitHub. It utilized Decision Trees, Random Forests, and LightGBM algorithms. Several stages were completed, including exploratory data analysis, preprocessing, splitting, and feature selection. LightGBM consistently outperformed other models, achieving accuracy rates of 90.59% and 89.55% for 80-20 and 60-40 splits, and 90.33% for the 70-30 split using SMOTE and Boruta. The system will be deployed via a machine learning API using LightGBM.

Problem Statement & Objectives

Problem Statement

- Advanced malware (APTs)
- Evasion technique in malware
- Balancing false positive and negative rates

Objective

- To analyze the malware feature with EDA
- Developing a MLmodel to accurately classify and detect the malware
- To evaluate the performance of machine learning models

Literature Review

Malware Analysis

Static analysis examines malware files safely without execution, , while dynamic analysis runs the malware to observe its network interaction

Machine Learning Technique

Machine learning in malware analysis automates classification using data and algorithms, such as DT, RF and LightGBM.

Feature Selection

Feature selection is the process of selecting relevant attributes from a dataset. Three main approaches exist: wrapper, filter, and embedded.

REST API

A REST API is a web API using HTTP requests for resource retrieval and modification in JSON or XML format.

Methodology

01 Data Collection
The dataset used was compiled from a public repository on GitHub, provided by Marco Ramili

02 Data Preprocessing
Data preprocessing involved data integration, cleaning, transformation, and balancing.

03 Data Spliting
The dataset was split into training and testing sets for evaluation.

04 Feature Selection
Boruta algorithm, based on RF models, was used for feature selection.

05 Model Construction
The ML algorithms used for multi-class classification were DT, RF and LightGBM. Grid search was used for hyperparameter tuning.

06 Model Evaluation
The metrics used for evaluation were accuracy, precision, recall, and F1 score,

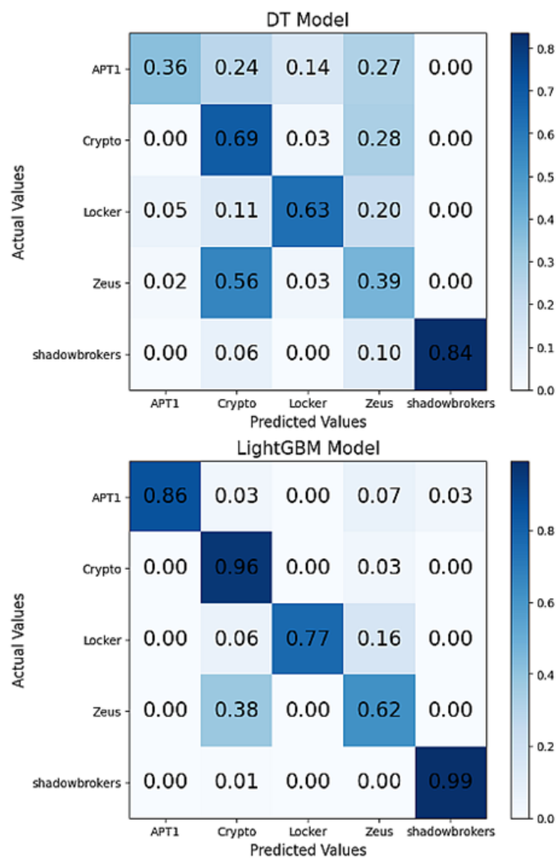
07 Application Deployment with ML API
LightGBM model was used to deploy the machine learning model as a web service with the library of FLASK and Joblib.

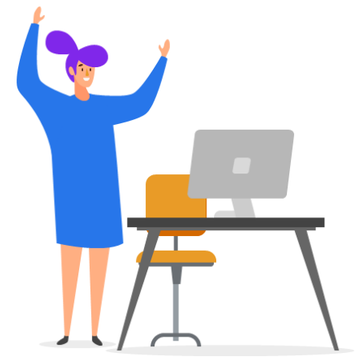
Result and Discussion

The classification results involved four experiments that assessed the performance of DT, RF, and LightGBM models under different train-test split ratios. The first experiment did not involve feature selection or oversampling. In the second experiment, Boruta was used independently of SMOTE. The third experiment utilized SMOTE without Boruta. Lastly, the fourth experiment combined both SMOTE and Boruta. The below table shown was the example of accuracy of different models.

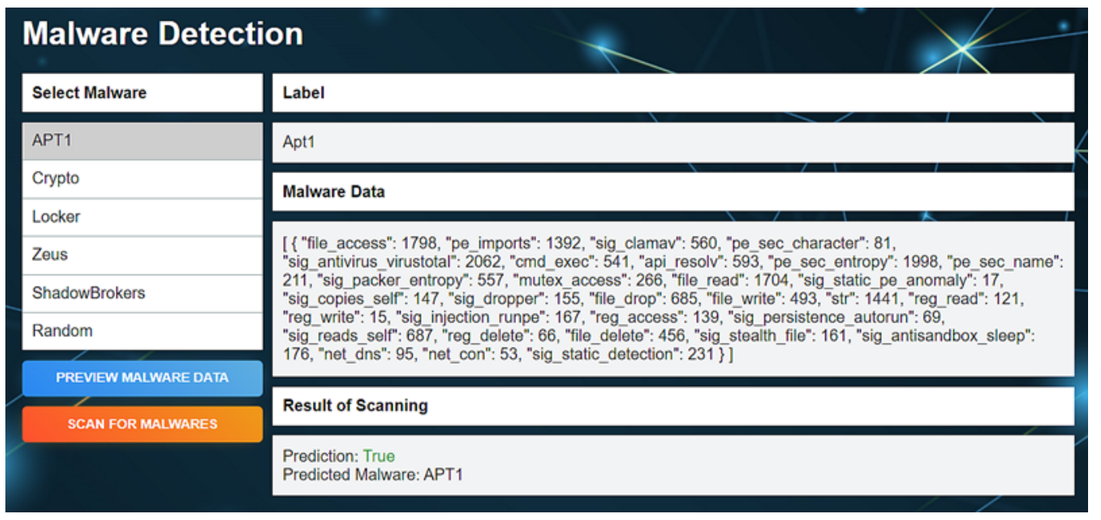
	Accuracy of Models		
Experiments	**DT**	**RF**	**LightGBM**
Without SMOTE & Boruta	0.6015	0.7854	0.8351
Using Boruta Without SMOTE	0.6744	0.7978	0.8376
Using SMOTE Without Boruta	0.7594	0.8426	0.9059
Using SMOTE & Boruta	0.7916	0.8540	0.9050

The models were compared using confusion matrices in four experiments with an 80-20 train-test split ratio. DT had the lowest TP rates for all malware families, while RF performed slightly better than DT. However, LightGBM consistently outperformed both DT and RF, achieving the highest TP rates for each malware family. This highlights LightGBM's effectiveness in accurately identifying and distinguishing various types of malwares.





The model utilizes LightGBM for malware detection through a REST API. Users can select from malware types like APT1, Crypto, Locker, Zeus, Shadowbrokers, and Random. If no type is chosen, the option to preview malware data is disabled. After selecting a type and clicking "Preview Malware Data," the system displays the label and corresponding data. Clicking "Scan for Malwares" shows the scanning results, indicating true or false for each malware. The scanning option is disabled if "Preview Malware Data" is not clicked.



Conclusion

Machine learning is crucial for malware detection. LightGBM outperformed DT and RF with 0.9059 accuracy in the experiment without Boruta but with SMOTE (80-20 split). The research achieved EDA insights, a robust model, and performance evaluation. Challenges include obtaining labeled data due to dynamic malware and limited resources for complex model training.