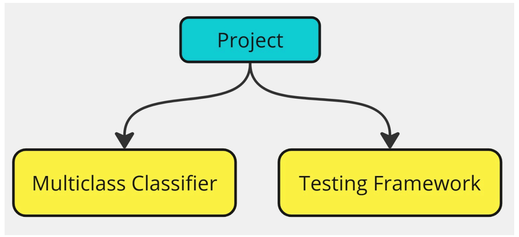
Multiclass Android Malware Detection

Ankit Gautam, Sambuddho Chakravarty
ankit21518@iiitd.ac.in, sambuddho@iiitd.ac.in

Scan QR for detailed report

Problem Statement & Motivation

The widespread adoption of Android devices has made them prime targets for malicious attacks, with Android malware posing a persistent and evolving threat to the security, privacy, and performance of these devices. As Android malware continues to grow in sophistication, there is an urgent need for effective detection systems that can identify and mitigate these threats. However, existing detection systems often face challenges, such as handling imbalanced datasets and lacking comprehensive evaluation frameworks that can assess their resilience against emerging and advanced malware techniques.
This project is motivated by the need to develop more robust and adaptable malware detection systems capable of effectively addressing these challenges. It aims to improve detection accuracy and ensure the models' ability to generalize across new, unseen malware families. The project is structured into two main directions:



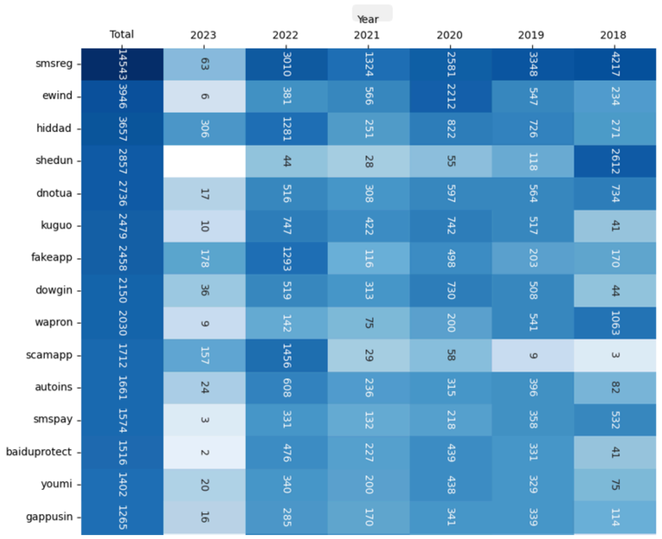
Dataset

- The Dataset is created by collection of 288.4k APKs from Androzoo and VirusShare across different years from regular time intervals

Year	Total Samples	Training:Testing (80:20)
2018	16,382	13,105:3,276
2019	29,126	23,301:5,825
2020	36,953	29,562:7,391
2021	21,886	17,509:4,377
Total	**104,347**	**83,478:20,869**

Table 4.1: Training and Testing Split (80:20) for the Dataset

- The dataset for Android malware detection by sourcing data from three prominent repositories: Androzoo, VirusShare, and RmvDroid

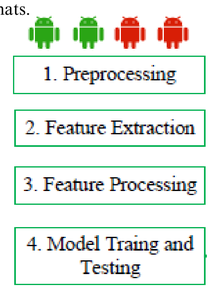


Top 15 Families with their year-wise count

Methodology

- **Feature Extraction and Processing**

Features were extracted using Androguard and Wallmauer tools, which parse malware samples into structured JSON and text formats.



- **Extraction Process**

1. Identified relevant features such as external package usage, code structure, and basic block coverage.
2. Employed regular expressions (regex) to process Wallmauer's instrumentation data, isolating external libraries and their dependencies.

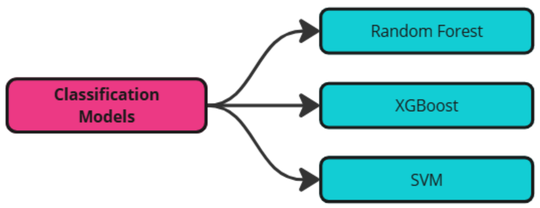
- **Extraction Process**

1. Identified relevant features such as external package usage, code structure, and basic block coverage.
2. Employed regular expressions (regex) to process Wallmauer's instrumentation data, isolating external libraries and their dependencies.
3. Integrated these features into a comprehensive dataset tailored for malware family detection.

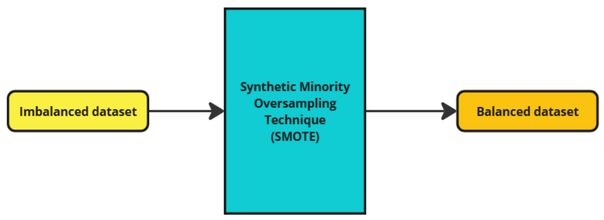
Model Development

- **Training Pipeline**

1. Utilized an 80:20 train-test split for model evaluation.
2. The training phase is optimized with supervised learning techniques and advanced hyperparameter tuning strategies such as GridSearch.
3. Incorporated robust feature engineering to enhance performance.



- **Balancing of the dataset**
- The dataset used for malware classification was highly imbalanced, with certain malware families being underrepresented. To address this, we applied SMOTE (Synthetic Minority Oversampling Technique), which generates synthetic samples for minority classes by interpolating between existing data points.



Results so far

Imbalanced Dataset

The goal was to classify malware samples into distinct families using features extracted from APKs with Androguard and Wallmauer. Machine learning algorithms, including Random Forest, SVM, and XGBoost, were applied to tackle this multiclass classification problem. Performance was evaluated using Accuracy, Precision, Recall, and F1-score, focusing on handling highly imbalanced datasets where certain malware families were underrepresented.

Algorithm	Accuracy	F1-score Macro	F1-score Macro
Random Forest	0.8706888623	0.75	0.86
XGBoost	0.806294	0.56	0.77
SVM	0.78	0.65	0.71

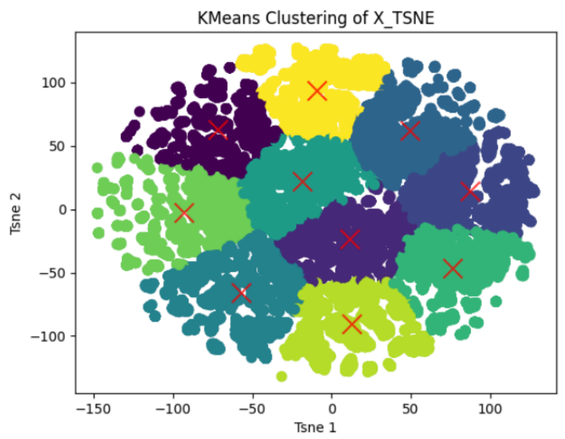
- Random Forest: Highest accuracy at 87.1%, but struggled with minority classes, leading to a macro F1-score of 0.75.
- XGBoost: Accuracy of 80.6%, with a macro F1-score of 0.56, indicating difficulty in handling imbalanced data.
- SVM: Accuracy of 78%, macro F1-score of 0.65, but limited by non-linear data relationships.

Balanced Dataset

To mitigate imbalance, SMOTE (Synthetic Minority Oversampling Technique) was applied, leading to substantial improvements:

Algorithm	Accuracy	F1-score Macro	F1-score Macro
Random Forest	0.915432	0.89	0.91
XGBoost	0.849523	0.72	0.81
SVM	0.82	0.75	0.78

- The Random Forest classifier showed significant improvement after applying SMOTE, achieving an accuracy of 0.915 and F1-scores of 0.89 (macro) and 0.91 (weighted). This indicates better handling of minority classes and overall balanced performance. XGBoost saw moderate improvement with an accuracy of 0.849 and a macro F1-score of 0.72, though it still struggles with class balance, as reflected in its lower macro F1-score. The SVM model improved to an accuracy of 0.82 and macro F1-score of 0.75, with SMOTE helping to better distinguish minority classes, though further enhancement is needed compared to Random Forest.



Kmeans clustering Plot on applying TSNE Feature reduction for visualisation

Future Work

The project is still a work in progress, with several areas identified for further exploration and development across the two directions:

1. Multiclass Classifier Development
- Feature Engineering and Selection: Future efforts will explore advanced feature selection methods, including deep learning-based techniques, to improve performance, reduce complexity, and enhance interpretability.
- Research will also focus on improving the classifier's ability to generalize across previously unseen malware families. Additionally, we will integrate explainability into the models, enabling more transparent decision-making and fostering trust in the classification results.

2. Testing Framework Development
- A key goal is to develop a testing framework to systematically benchmark classifiers based on their feature requirements and data dependencies.
- The framework will assess the ability of various classifiers to handle a wide range of malware families, highlighting limitations and strengths.
- The testing pipeline will measure classifiers' robustness in real-world scenarios, including performance on adversarial examples and obfuscated malware
- Metrics such as computational efficiency and scalability will be considered, providing insights into the trade-offs between performance and resource utilization.
- Insights from the framework will guide refinements for existing classifiers and inform the development of new, more resilient solutions.

References

1. Team, A. Androguard: A powerful reverse engineering tool for android apks. https://androguard.readthedocs.io/, 2024.
2. Team, W. Wallmaure: External library extraction tool for android apks. https://github.com/mate-android-testing/wallmauer, 2024.
3. Liu, Z., et al. Mdmc: Deep learning for malware detection in android applications. Proceedings of the 2020 IEEE International Conference on Data Mining (2020), 1012–1020.
4. Arp, D., et al. Drebin: Effective and explainable detection of android malware in your pocket. Proceedings of the 2014 Network and Distributed System Security Symposium (2014), 1–16.
5. Chen, Z., et al. Mudflow: A malware detection system using dynamic behavior flow analysis. Proceedings of the IEEE Symposium on Security and Privacy (2020), 1–10.
6. Yu, S., et al. Api graph: A graph-based android malware detection framework. Proceedings of the IEEE International Conference on Big Data (2019), 123–130.
7. Younis, M., et al. Aldroid: Lightweight android malware detection using machine learning. Proceedings of the 2017 IEEE International Conference on Mobile Software Engineering and Systems (2017), 199–206.
8. Sun, Y., et al. Imgdroid: A comprehensive image-based malware detection framework for android. Proceedings of the ACM International Conference on Mobile Systems, Applications, and Services (2019), 123–130.
9. Luo, X., et al. Revealdroid: A state-of-the-art android malware detection system using static analysis. Proceedings of the ACM on Security and Privacy 5, 3 (2017), 1–14.
10. Fernandes, E., Correia, M., Silva, J., Lopes, P., Almeida, H., Pinto, L., Rodrigues, J., Veiga, L., Ferreira, M., et al. Avclass2: Massive malware tag extraction from av labels. arXiv preprint arXiv:2006.10615 (2020).