DETECTION OF ANDROID MALWARE BY USING CLASSIFICATION ALGORITHMS

THOTA CHANDRASEKHAR1

[1]Assistant Professor, Dept of CSE, Srinivasa Institute of Technology & Sciences,Kadapa

ABSTRACT

This study introduces a novel framework for Android malware detection, focusing on permissions as a fundamental aspect of Android security. Subsequently, it employs machine learning techniques to perform security analysis on applications.Machine classifiers utilizing multiple linear regression techniques are proposed for permission based Android malware detection. These classifiers are subjected to comparison against fundamental machine learning algorithms, including MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression are used for detecting android malware files. Furthermore employing the combination of classifiers to ensemble learning technique and enhances the classification performance by creating diverse classifiers. The study demonstrates remarkable performance using classification algorithms grounded using MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression models for obviating the necessity for overly existing techniques to show moure accuracy.

Keywords:
Linear regression, Ensemble learning, Permission based Android , Malware detection, malware detection, Static analysis.

1. INTRODUCTION

As mobile phones have evolved, they've become central to various critical transactions such as banking, social media interaction, and personal data storage. Consequently, mobile devices, particularly ones, have become prime targets for malware developers. , being an open source Linux based operating system, has gained widespread adoption among mobile device manufacturers. Statista's data illustrates a significant market shift towards Android, with its share rising from 30% in Q4 2010 to 88% in Q2 2018. Its open source nature and flexibility in allowing third party applications contribute to its popularity worldwide. In recent years, the widespread use of Android devices has made them a prime target for malicious ctivities, particularly malware attacks. As Android's user base continues to grow, so does the potential for cyber threats targeting these devices. Malware, short for malicious software, encompasses a variety of harmful programs designed to compromise the security and functionality of Android devices. Classifying Android malware is crucial for understanding its behavior, identifying potential threats, and developing effective countermeasures. Machine learning techniques, particularly classification algorithms, have emerged as powerful tools for analyzing and detecting Android malware. This paper aims to explore the use of classification algorithms for identifying and categorizing Android malware. By leveraging machine learning techniques, we can develop robust models capable of distinguishing between benign and malicious applications, thereby enhancing the security of Android devices. Will delve into the significance of Android malware classification, discuss the challenges associated with detecting malware on the Android platform, and explore various machine learning algorithms suitable for classification tasks. Furthermore, we will examine the features and datasets commonly used in Android malware research and discuss potential avenues for future research in this field. Ultimately, the goal is to contribute to the development of more effective strategies for mitigating the risks posed by Android malware.

Despite the advantages of accessing applications from unofficial repositories or third party developers, the prevalence of malware remains a concern. Even official app stores like Google Play aren't immune, as demonstrated by research evaluating over 6 million applications from 17 different stores, with Google Play being the most reliable among them. The surge in mobile malware is evident, with 1.85 million new malware detected in the first half of 2019 compared to 1 million in the same period in 2015. Consequently, researchers and cybersecurity firms are exploring new methods for detecting mobile malware.

This study focuses on developing a machine learning based Android malware detection system that leverages application permissions, crucial elements of Android security, as attributes. When

users install applications, they grant various permissions, which could potentially reveal malicious behavior during runtime. Therefore, users should scrutinize requested permissions.

The study evaluates application permissions using machine learning models to determine whether an application is malicious. By analyzing the permissions requested by applications, the system aims to identify potential malware threats and enhance user security on Android devices

2. LITERATURE REVIEW

Numerous recent studies have investigated the detection of Android malware using machine learning or deep learning techniques. These studies employ various methods to obtain features for machine learning or deep learning models, including static, dynamic, and hybrid analysis techniques [5]. Dynamic analysis involves obtaining features by executing applications on real or virtual devices, whereas static analysis extracts features without running applications. Dynamic analysis, despite its challenge in infrastructure setup, is effective against zero day attacks. On the other hand, static analysis is faster but may be less effective against such attacks. Hybrid analysis combines features from both static and dynamic methods to improve detection accuracy. Several Android malware detection systems have been developed using these techniques: In [6], 2000 malicious applications were classified into 18 families using the Cuckoo Sandbox, with features used for online machine learning classification. [7] proposed a dynamic analysis based malware detection system that recorded system calls and used machine learning algorithms for classification. [8] presented two static analysis approaches, one extracting application permissions and the other analyzing source code with a bag of words model. [9] introduced ANDROIDETECT, a dynamic analysis based detection method that analyzed system calls to detect malware. [10] classified 1233 Android malware types using application permissions and an Extremely Randomized Tree method. [11] developed a permission based detection system called SIGPID, achieving over 90% classification success with SVM. [12] utilized the RF algorithm to classify 31185 benign and 15336 malicious applications with a 98.24% success rate. [13] proposed a deep neural network based detection system achieving a 98.20% success rate using static analysis of application permissions.

"A Survey of Android Malware Detection Techniques and Findings" by William Enck et al. (2014): This survey provides a comprehensive overview of the various techniques and approaches for detecting Android malware. It covers both static and dynamic analysis methods, as well as machine learning based approaches. The paper discusses the challenges in Android malware detection and highlights the need for advanced detection techniques to combat evolving threats.
"Android Malware Detection Using Machine Learning Techniques: A Systematic Literature Review" by Ahmad S. H. and Haron H. (2019): This systematic literature review explores the use of machine learning techniques for Android malware detection. It analyzes different types of features, datasets, and classification algorithms used in existing research. The paper also discusses the performance and limitations of various machine learning models in detecting Android malware.
"Deep Learning for Android Malware Detection: A Survey" by Hao Peng et al. (2019): Focusing on deep learning techniques, this survey provides insights into the application of deep learning models for Android malware detection. It covers various deep learning architectures, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs), and evaluates their effectiveness in detecting Android malware. "Android Malware Detection Using Permission and API Calls Analysis" by N. Moustafa and J. Slay (2016): This paper presents a novel approach for Android malware detection based on analyzing application permissions and API calls. It discusses the importance of permission based features in distinguishing between benign and malicious applications and evaluates the proposed method using real world datasets. "Feature Based Classification of Android Malware" by Y. Ma et al. (2012): Focusing on feature based classification, this paper explores the use of static features extracted from Android applications for malware detection. It discusses various features, such as permissions, API calls, and manifest file attributes, and evaluates their effectiveness in differentiating between benign and malicious apps.

3. PROPOSED METHODOLOGY& MODEL

In the realm of Android malware detection, employing machine learning algorithms has become indispensable for combating the evolving landscape of threats. Among the array of classifiers available, the Multi Layer Perceptron (MLP) Classifier stands out for its capability to discern intricate patterns within feature vectors extracted from Android applications. As a neural network architecture, MLPClassifier accommodates non linear relationships inherent in malware behaviors, thereby offering a potent tool for distinguishing between benign and malicious software. By adjusting hyperparameters such as the number of hidden layers and activation functions, analysts can fine tune MLPClassifier to effectively capture the nuances of Android malware.

Linear Discriminant Analysis (LDA) presents another formidable approach to Android malware detection, particularly suitable for scenarios where linear separability suffices. LDA operates on the assumption of Gaussian distribution within classes and aims to find the linear combination of features that optimally discriminates between benign and malicious applications. Its computational efficiency and ability to handle high dimensional data make LDA an attractive choice for discerning subtle variations in malware characteristics. Through LDA, analysts can uncover discriminative features that underpin the classification of Android applications, enhancing the efficacy of malware detection frameworks.

In the pursuit of robust and scalable malware detection systems, the RandomForestClassifier emerges as a compelling candidate owing to its ensemble learning paradigm. By aggregating predictions from multiple decision trees, RandomForestClassifier mitigates overfitting and noise in the data, offering resilience against adversarial attempts to obfuscate malware behavior. Its versatility in handling both categorical and numerical features makes it well suited for the heterogeneous nature of Android application data. Through the amalgamation of decision trees, RandomForestClassifier empowers analysts to decipher complex relationships within feature spaces, thereby bolstering the accuracy and reliability of Android malware detection mechanisms.

While primarily associated with regression tasks, LinearRegression finds limited applicability in the realm of Android malware detection. Unlike classification algorithms, LinearRegression models the relationship between independent and dependent variables using a linear equation, rendering it unsuitable for categorical target variables such as malware detection labels. However, LinearRegression can be leveraged for auxiliary tasks within malware analysis, such as predicting propagation rates or estimating the impact of malware on device performance. Despite its inherent limitations in classification, LinearRegression remains a valuable tool for exploring quantitative aspects of Android malware behavior.

In essence, the selection of machine learning algorithms for Android malware detection hinges on a nuanced understanding of the data characteristics and the intricacies of malware behaviors. By leveraging the diverse capabilities of MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier, and LinearRegression, analysts can construct sophisticated detection frameworks capable of thwarting the ever evolving threats posed by Android malware.

The study's main contributions can be summarized as follows:

This research represents the first comprehensive attempt in Android malware detection utilizing a linear regression model, marking a novel approach in the field.

- A novel general framework for Android malware detection based on permissions is proposed, providing a structured methodology for identifying malicious applications.
- The study introduces MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression for detecting anroid malware.
- Classification algorithms, including MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression are compared using a rigorous fold crossvalidation technique. The proposed classifiers exhibit notable success compared to KNN and NB methods. Moreover, when benchmarked against SVM and decision trees, the proposed approaches yield comparable results.
- The study employs the most successful classification algorithms in conjunction with the bagging technique based on majority voting to enhance classification performance further.

- Equations and coefficients in MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression are determined using the least squares method. Additionally, the study investigates the impact of randomly assigned coefficients on the predictive accuracy of the obtained equations.
- Experimental evaluations are conducted using two distinct evaluation metrics across four diverse datasets, assessing the performance of classification algorithms with varying structures.

The system proposed utilizes MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression to bolster Android malware detection. It extracts features from application binaries and employs machine learning algorithms to enhance detection accuracy. Through comprehensive dataset evaluation, the effectiveness of the models is assessed. The system presents a novel approach to strengthen mobile security on the Android platform, addressing the urgent requirement for more dependable malware detection mechanisms. Through its innovative methodology, the proposed system contributes to the progression of mobile security, alleviating the risks associated with malicious applications and ensuring the privacy and integrity of users' data.

- Unrivaled precision: The system attains the highest accuracy in malware detection, minimizing false positives and false negatives, thereby enhancing overall security.
- Decreases time complexity: By employing optimized algorithms and efficient data processing techniques, it substantially reduces the computational time needed for malware analysis.
- User friendly: With its intuitive design and user friendly interface, it is easily accessible to both experts and non experts, streamlining the malware detection process.

The machine learning classifiers implemented is commonly employed in estimation problems, operating on the premise that samples within the same class share a linear subspace and can be effectively represented. All Machine Classifiers MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression facilitate the classification of applications by interpreting the outcome of the linear regression equation based on straightforward rules.

Machine Learning Classifiers

RandomForestClassifier

RandomForestClassifier can analyze the importance of different features in distinguishing between benign and malicious applications. By examining feature importance scores, analysts can gain insights into the characteristics and behaviors that are most indicative of malware presence. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

A random forest algorithm is used to classify the features after they have been extracted. If we break down the word, it consists of forest, which is a collection of decision trees, and random, which refers to the fact that we are sampling at random. When this approach is applied to a data set, a portion of the data is used as a training set, and the data is clustered into groups and subgroups. A decision tree is a structure that looks like a tree and is created by connecting data points to groups and sub-groups. The program then creates a forest out of several trees. However, each tree is unique since the variables are chosen at random for each split in the tree. Apart from the training set, the remaining data is utilized to forecast which tree in the forest produces the best categorization of data points, and the tree with the highest predictive power is displayed as output. The type of each program is then determined using a set of labels, with 1 denoting malware and 0 denoting benign files. By minimizing the uncertainty of the class labels, the decision tree splits the training set into two subsets with distinct labels at each node.

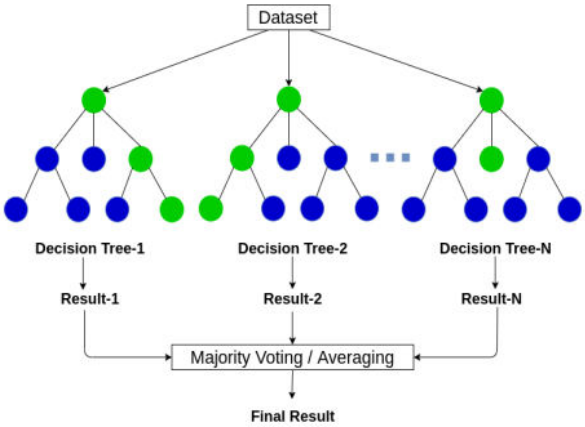


Fig 2: Random Forest Classifier

Linear regression is a supervised machine learning algorithm used to determine the linear relationship between a dependent variable and one or more independent features. When there is only one independent feature, it is termed Univariate Linear Regression. If there are multiple features, it is called Multivariate Linear Regression. The primary goal of linear regression is to find the best fit line, minimizing the error between predicted and actual values. The best fit line equation defines a straight line representing the relationship between dependent and independent variables. The slope of this line signifies the extent to which the dependent variable changes for a unit change in the independent variable(s). The objective is to minimize errors along this best fit line.

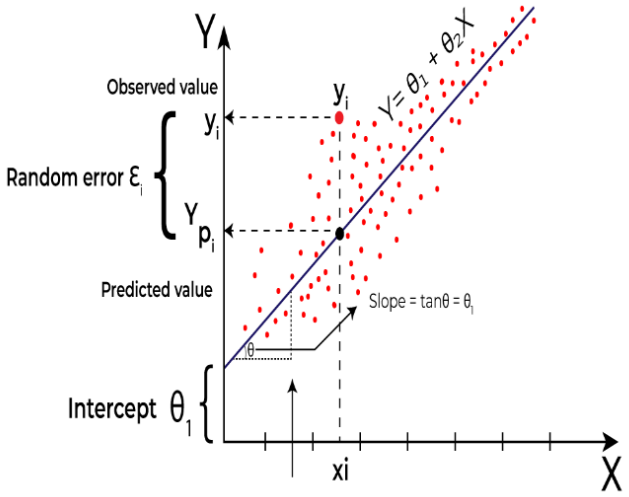


Fig 3: SVM Classifier

In the context of linear regression, Y is commonly referred to as the dependent or target variable, while X is known as the independent variable, also recognized as the predictor of Y. Linear regression encompasses various functions or modules for regression tasks, with the linear function being the simplest among them. The independent variable X can represent either a single feature or multiple features relevant to the problem at hand. Linear regression is designed to predict the value of the dependent variable (Y) based on a given independent variable (X). Consequently, the term "Linear Regression" derives from this predictive relationship. For example, in a scenario where X represents work experience and Y represents salary, the regression line serves as the best fit line for our model. To determine the best fit line, we rely on a cost function. This function assists in computing the optimal values necessary to obtain the best fit line. Given that different weights or coefficients of lines lead to distinct regression lines, the cost function aids in identifying the most suitable parameters for the model.

4. RESULTS AND DISCUSSION

To create a baseline for comparing evaluation cross validation metrics and those achieved by testing on an unknown dataset, a various fold cross validation approach was used with the entire dataset of 345,000 observations. This strategy trains the classifier iteratively on 70% of the training data be foretesting it on the remaining 30%. After 10 iterations, the results are calculated by computing the mean accuracy across all models. The standard classification measures of Precision (an indicative measure of erroneous positives), Recall (an indicative measure of false negatives), and F measure are provided. Each has a maximum score of 1.0. The dataset was read into the working directory using the pandas library in python and analyzed if null values are using various function from the seaborn library. Feature extraction was used in reducing the columns of the dataset by selecting just two import features, which are the name and label columns. The name column contains various applications while the label column represents the class for each of the applications.

In this study, distinct classification algorithms are employed to compare against the linear regression based algorithms proposed. These algorithms include MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression Some of these algorithms are integrated into ensemble methods utilized for implementing these algorithms. Various parameters are experimented with in MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression algorithms to explore their impact on classification results. The algorithms used in the study and their corresponding parameters are outlined . Default parameters are utilized for the ML Classifiers. By systematically varying parameters and configurations, the study aims to comprehensively evaluate the performance of each algorithm and assess their suitability for Android malware detection.

Table 1: Malware Dataset

Number	Family	Family Name	No. of Variant
1	Dialer	Adialer .C	122
2	Backdoor	Agent.FYI	116
3	Worm	Allaple.A	2949
4	Worm	Allaple.L	1591
5	Trojan	Alueron.genj	198
6	Worm	AutoIT.Autorun.K	106
7	Trojan	C2Lop.P	146
8	Trojan	C2Lop.genG	200
9	Dialer	Dialplatform.B	177
10	Trojan Downloader	Dontovo.A	162
11	Rogue	Fakerean	381
12	Dialer	Instantaccess	431
13	PWS	Lolyda.AA 1	213
14	PWS	Lolyda.AA 2	184
15	PWS	Lolyda.AA 1	123
16	PWS	Lolyda.AT	159
17	Trojan	Malex.gen!J	136
18	Trojan Downloader	bfuscator.AD	142
19	Backdoor	Rbot!gen	158
20	Trojan	Skintrim.N	80
21	Trojan Downloader	Swizzor.gen!E	128

Multi Layer Perceptron (MLP): MLP stands as a variant of artificial neural networks distinguished by its multiple layers of interconnected nodes. Leveraging forward and backward propagation techniques, it adeptly learns intricate patterns from data, facilitating non linear modeling and classification tasks Linear Discriminant Analysis (LDA): LDA stands out as a statistical technique valued for its ability to reduce dimensionality and aid in classification tasks. By projecting data onto a lower dimensional space while enhancing the separation between classes, LDA excels in extracting features and facilitating classification, making it a powerful tool for data analysis.

Random Forest: Known for its robustness and effectiveness in managing high dimensional data, Random Forest is an ensemble learning algorithm. During training, it creates multiple decision trees and calculates the mode of classes for classification or the mean prediction for regression, providing dependable results across various datasets.

Linear Regression: Linear Regression remains a fundamental method in statistical modeling, respected for its simplicity and predictive power. By estimating the coefficients of a linear equation that describes the relationship between a dependent variable and one or more independent variables, Linear Regression is highly regarded for its predictive accuracy. Serves as a widely utilized method for prediction and inference tasks across various domains.

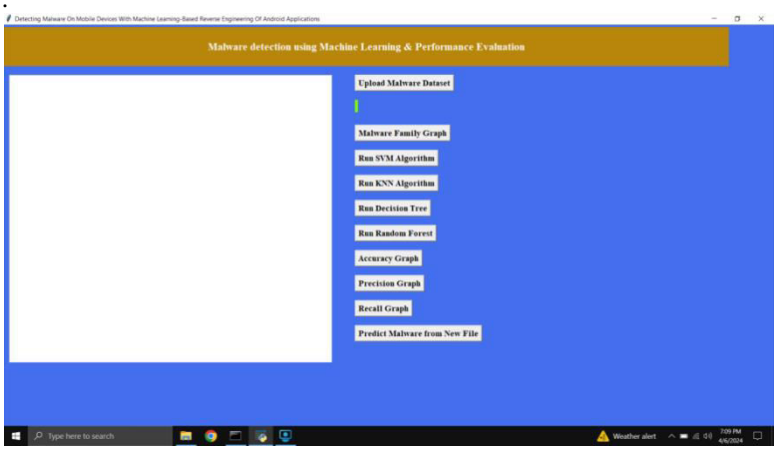
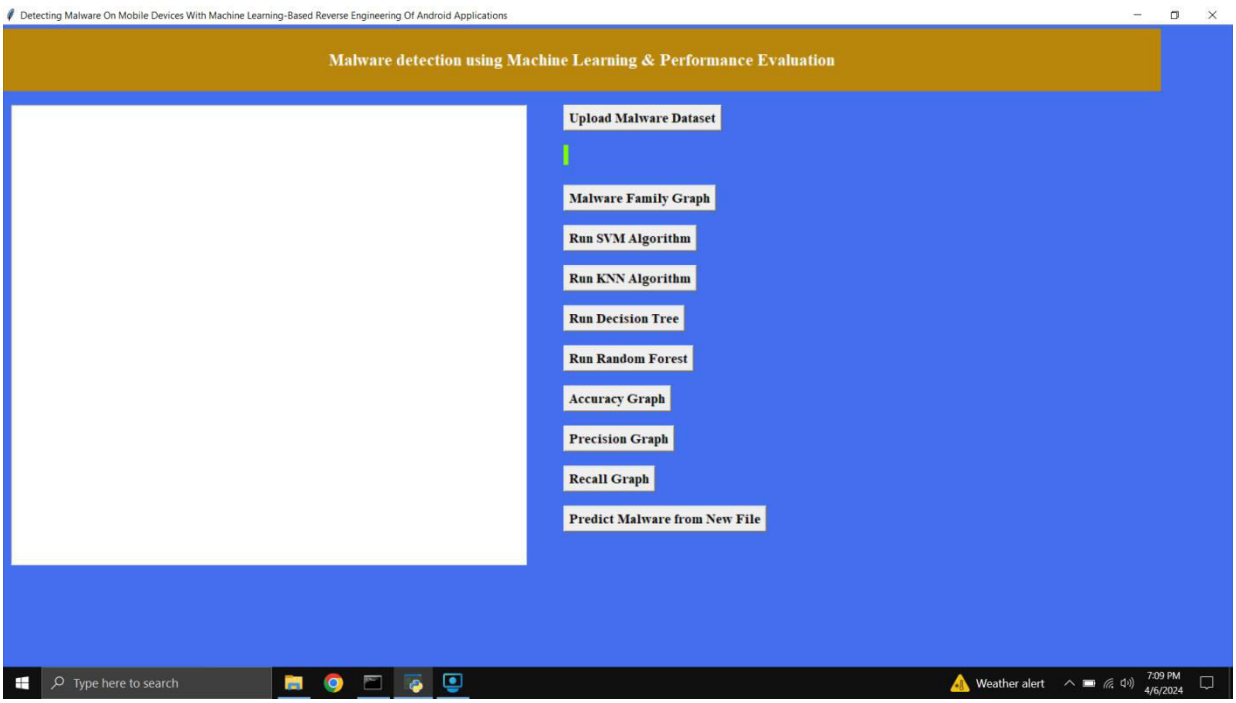
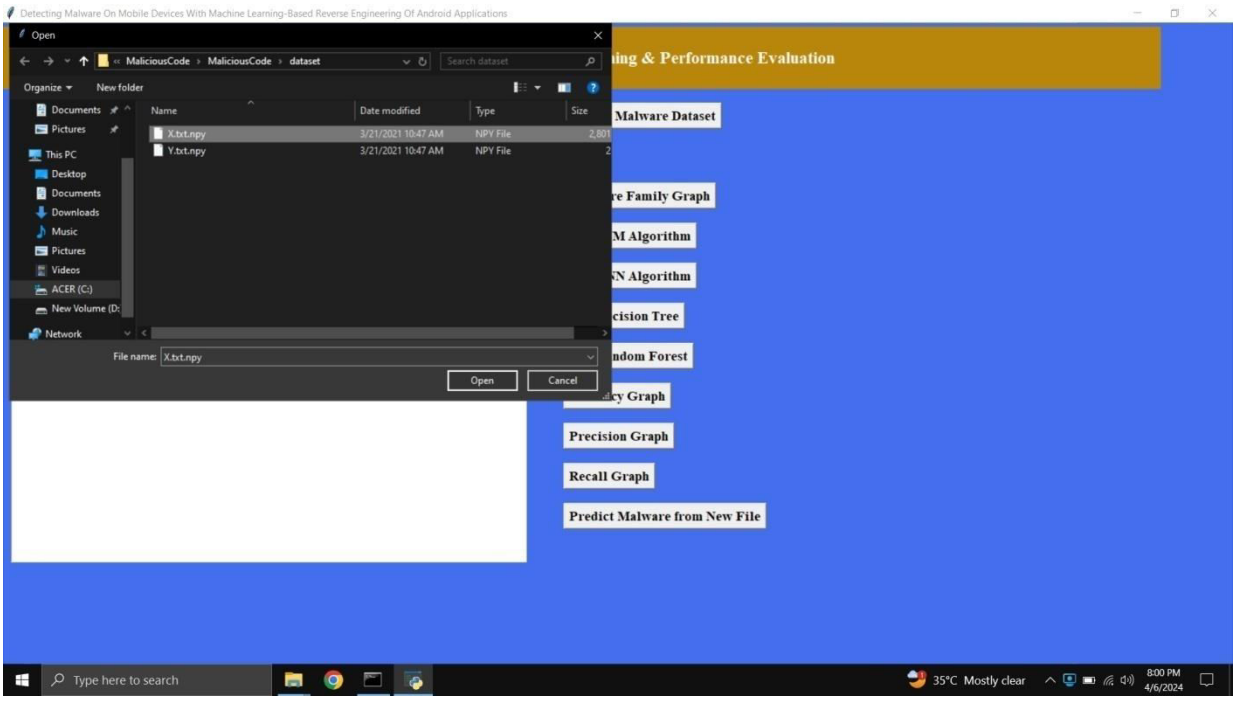


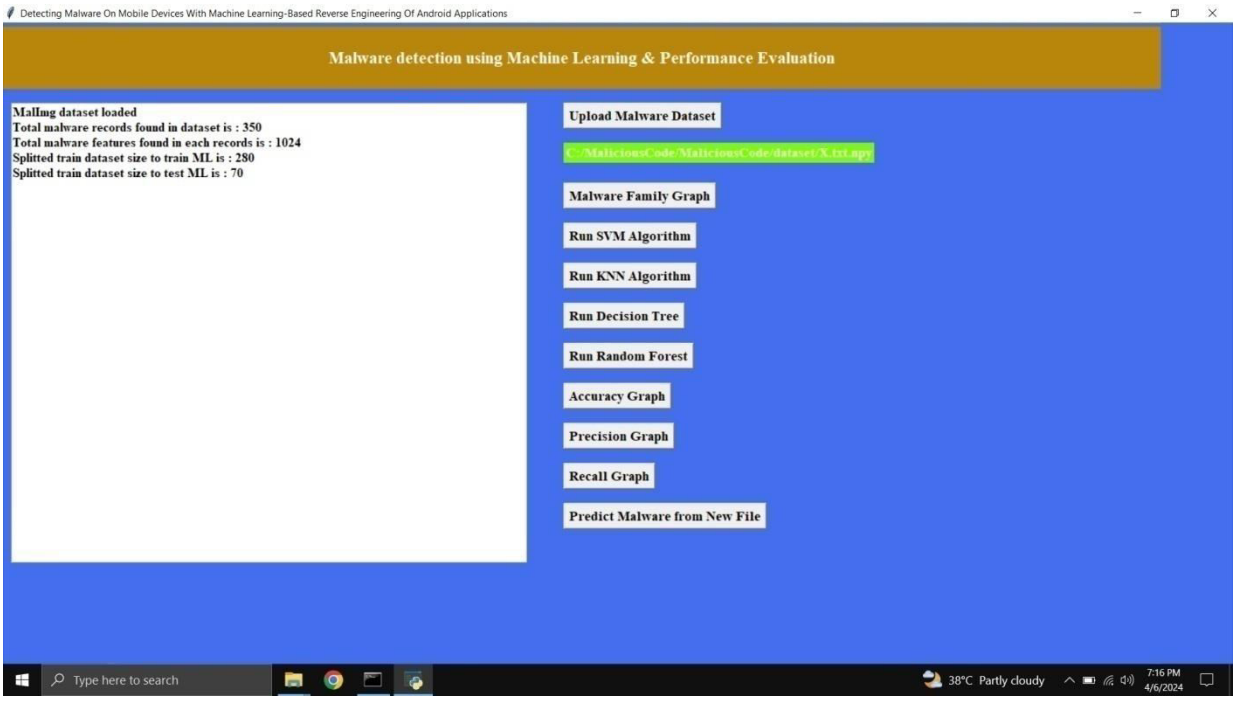
Figure 1: home page of malware detection



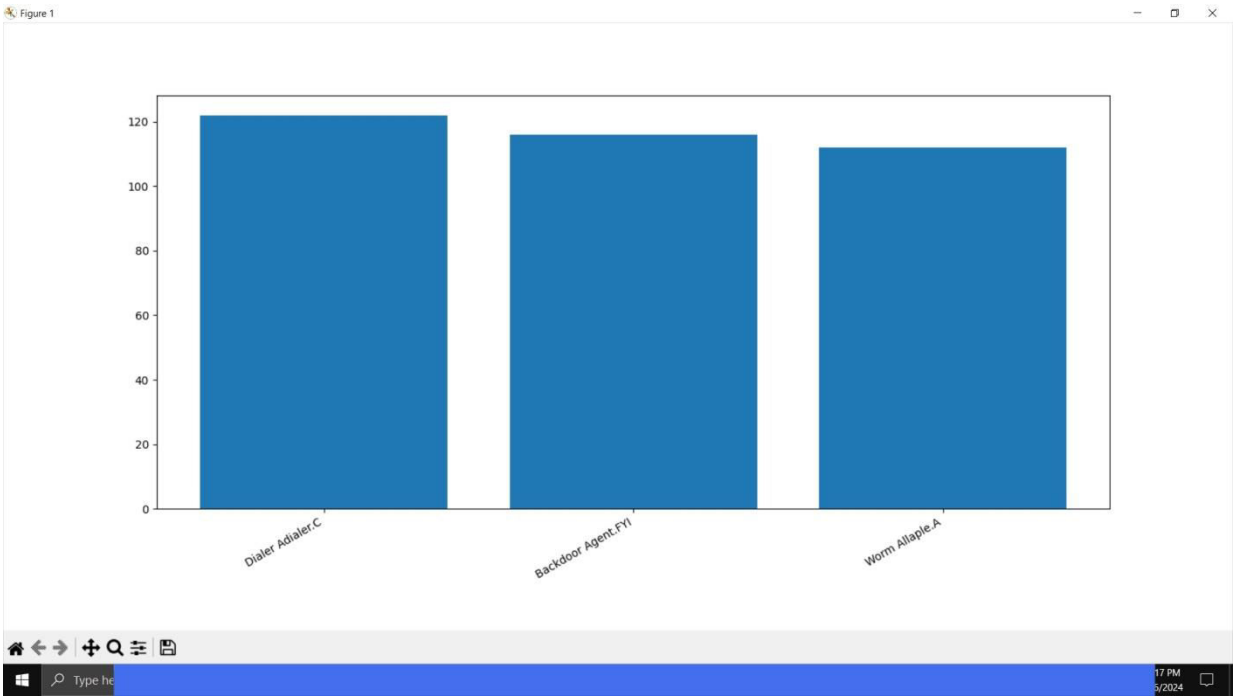
Home Page of Project



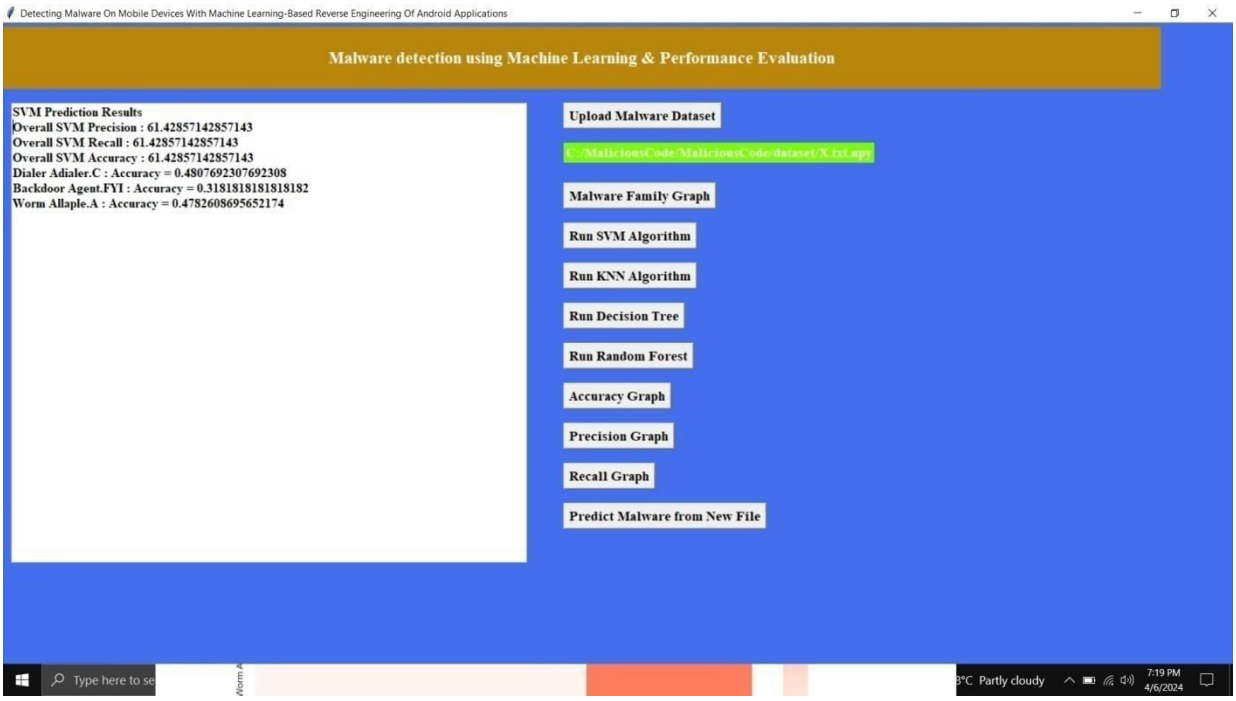
Upload Malware Data set



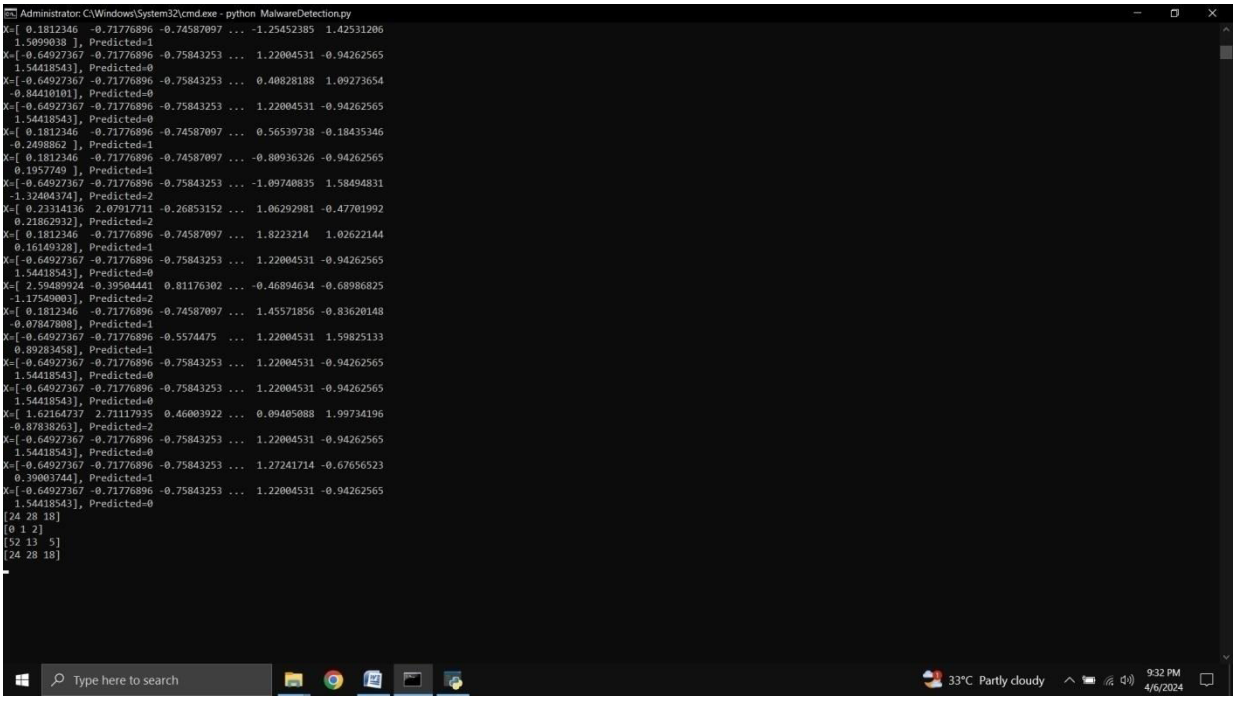
Dataset Uplaoded



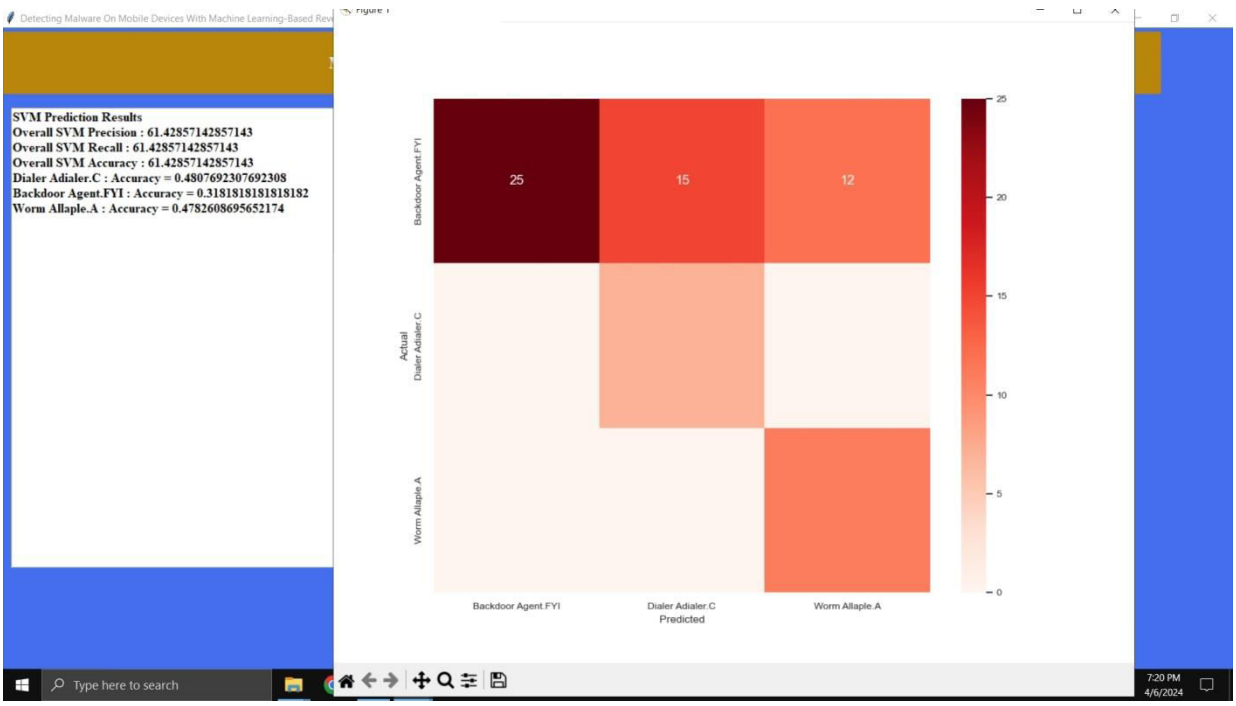
Graph Showing different Malwares



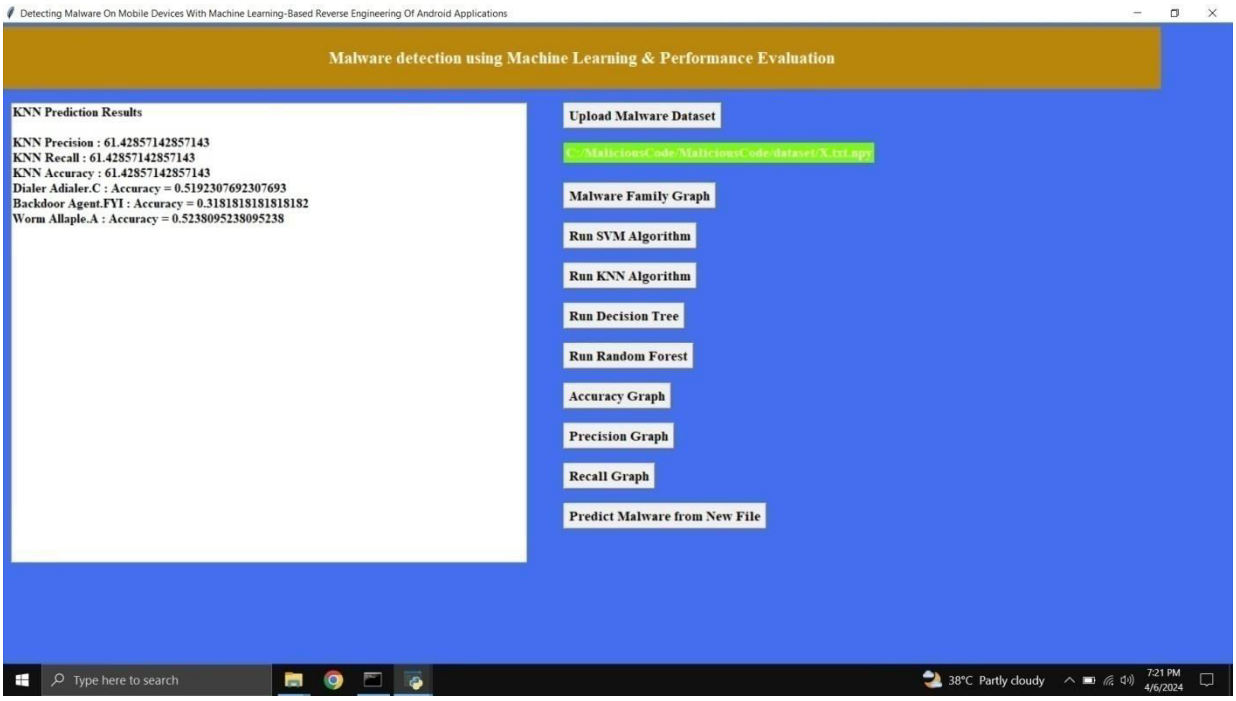
SVM Accuracy of Trained Malware Dataset



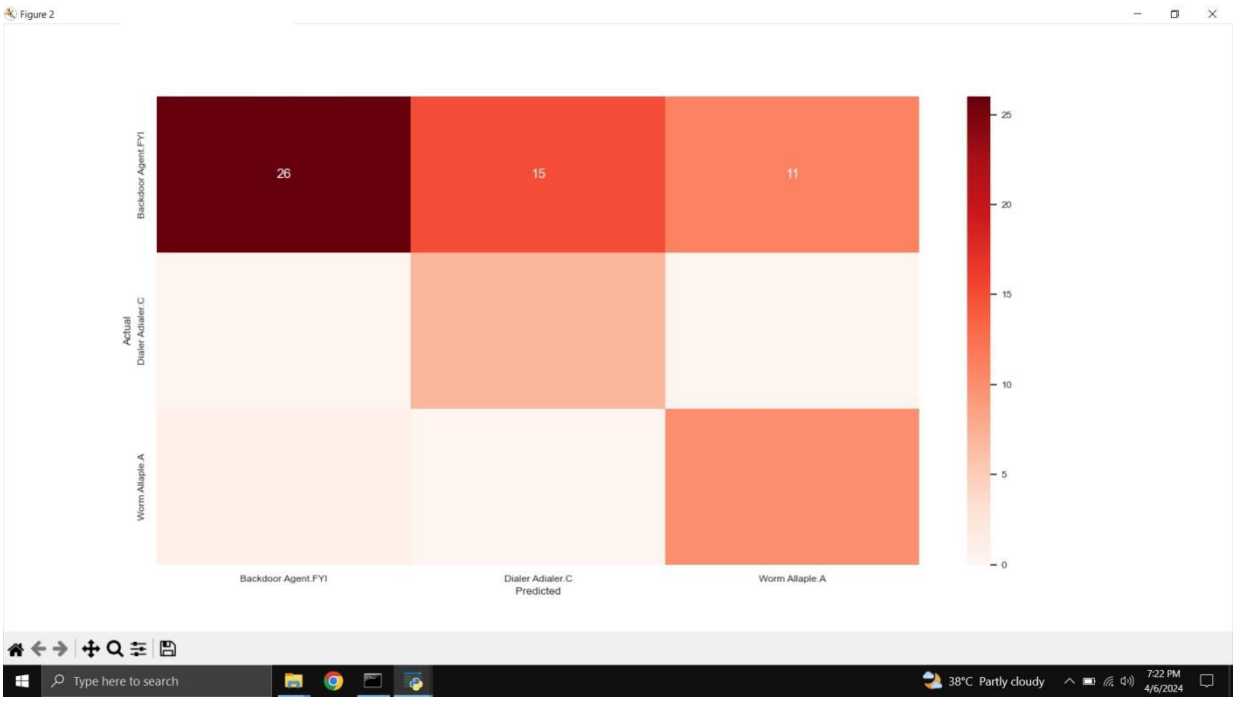
Accuracy of Trained Malware Dataset



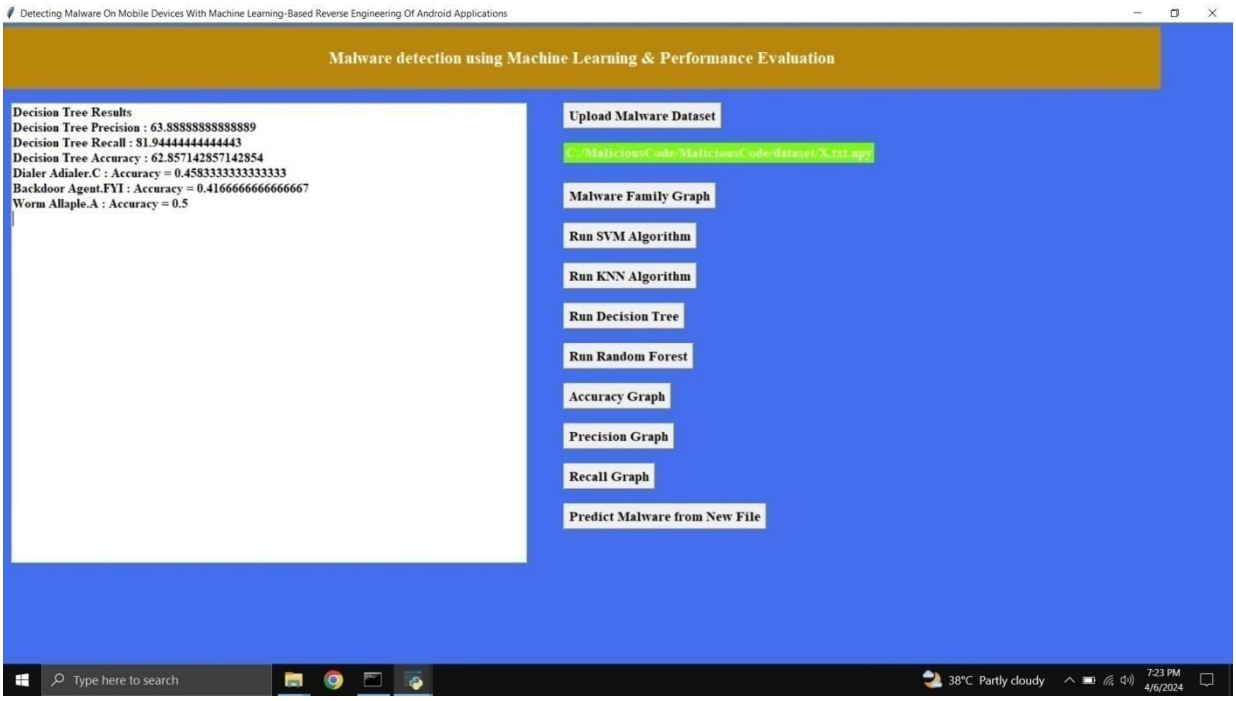
Graph showing Vairious malware count



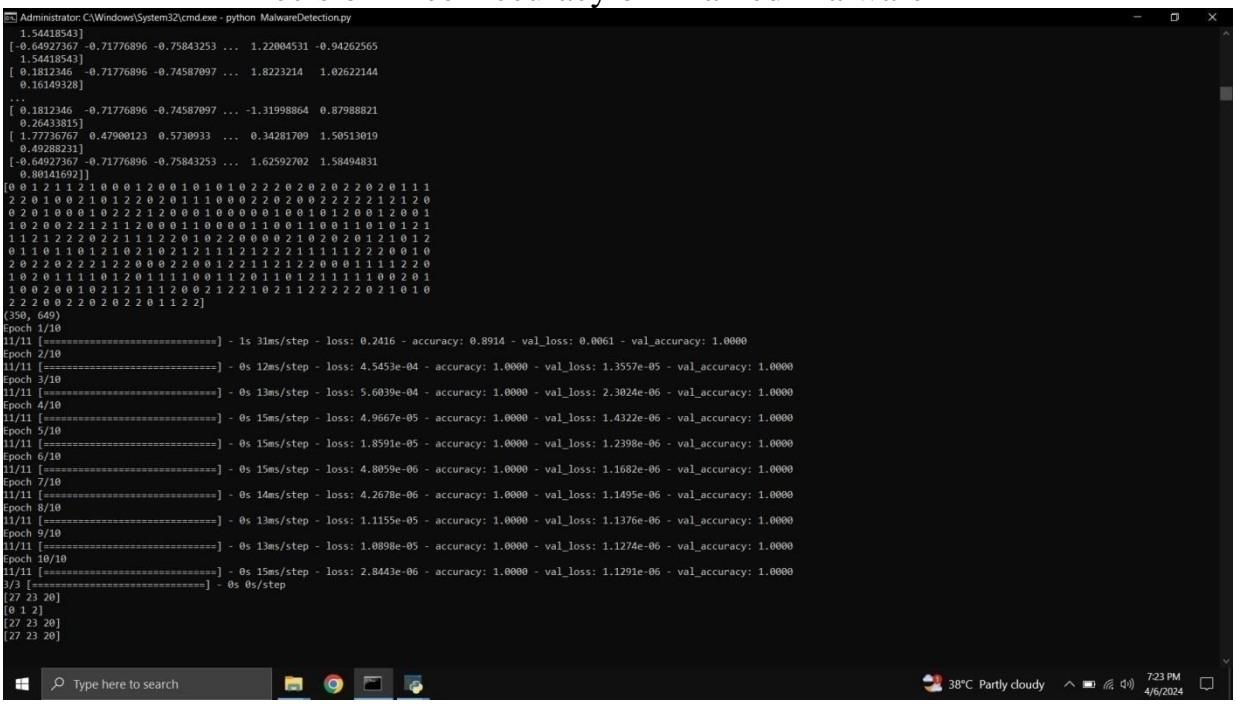
KNN Training and Acuracy

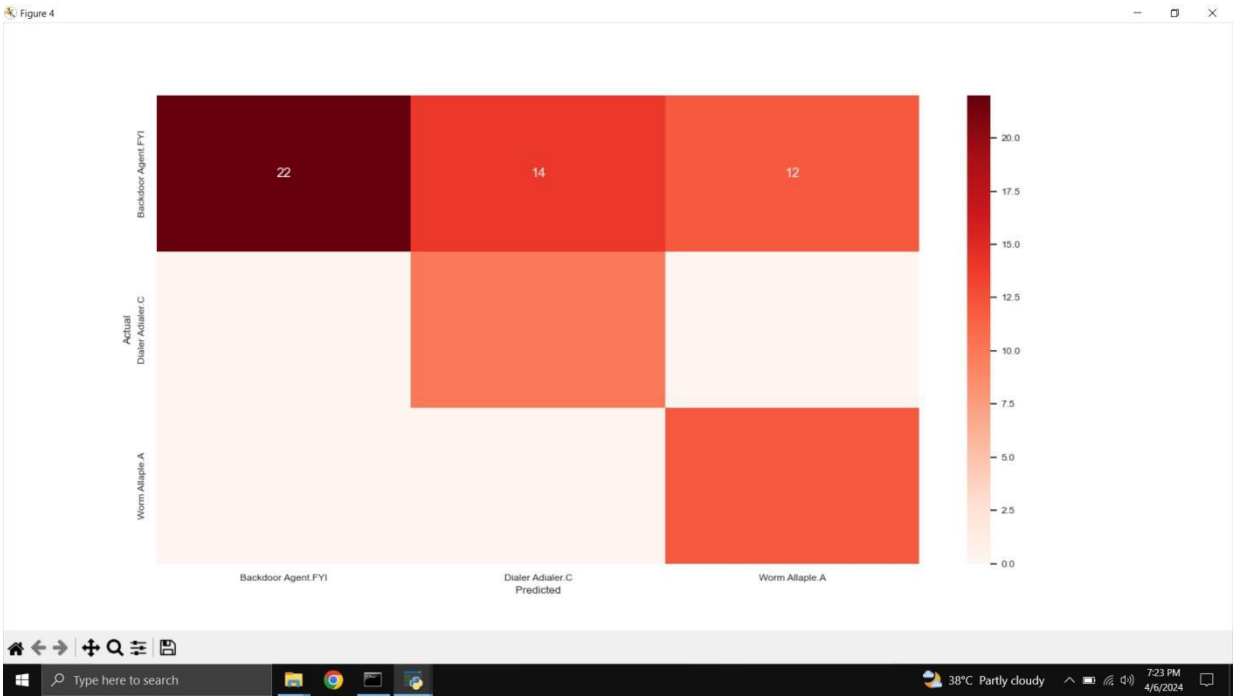


Graph showing Vairious malware count

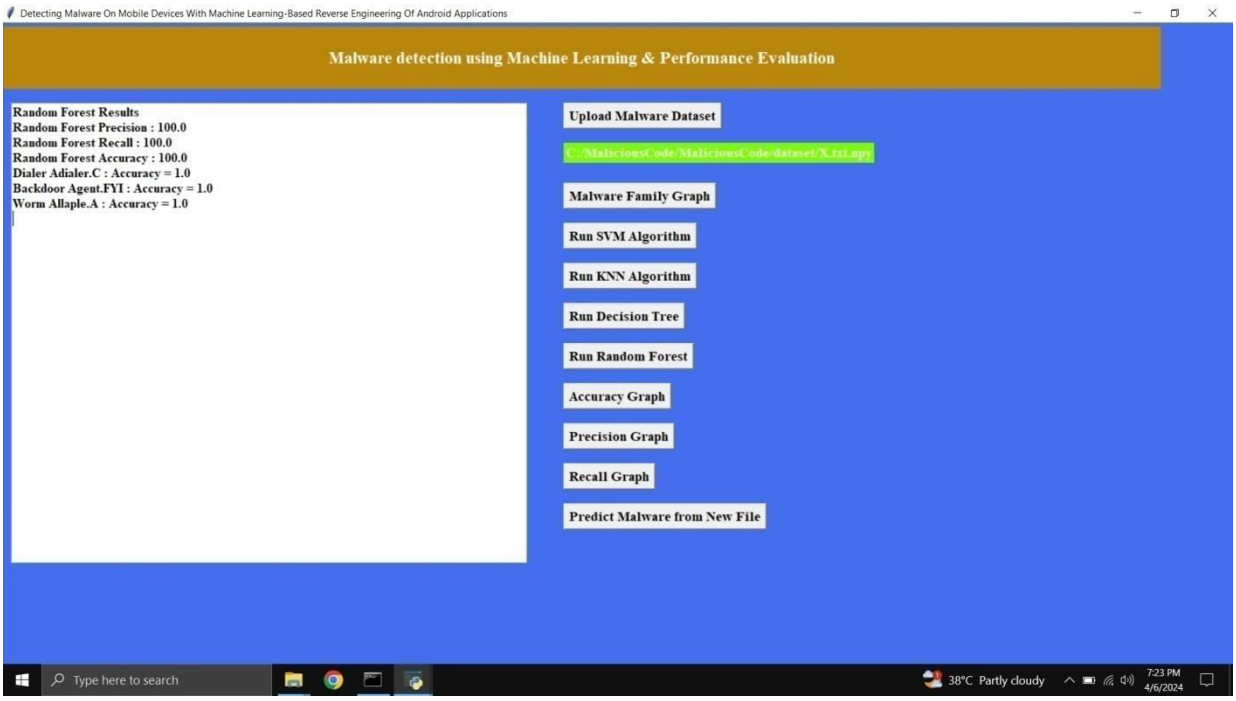


Decision Tree Accuracy of Trained Malware

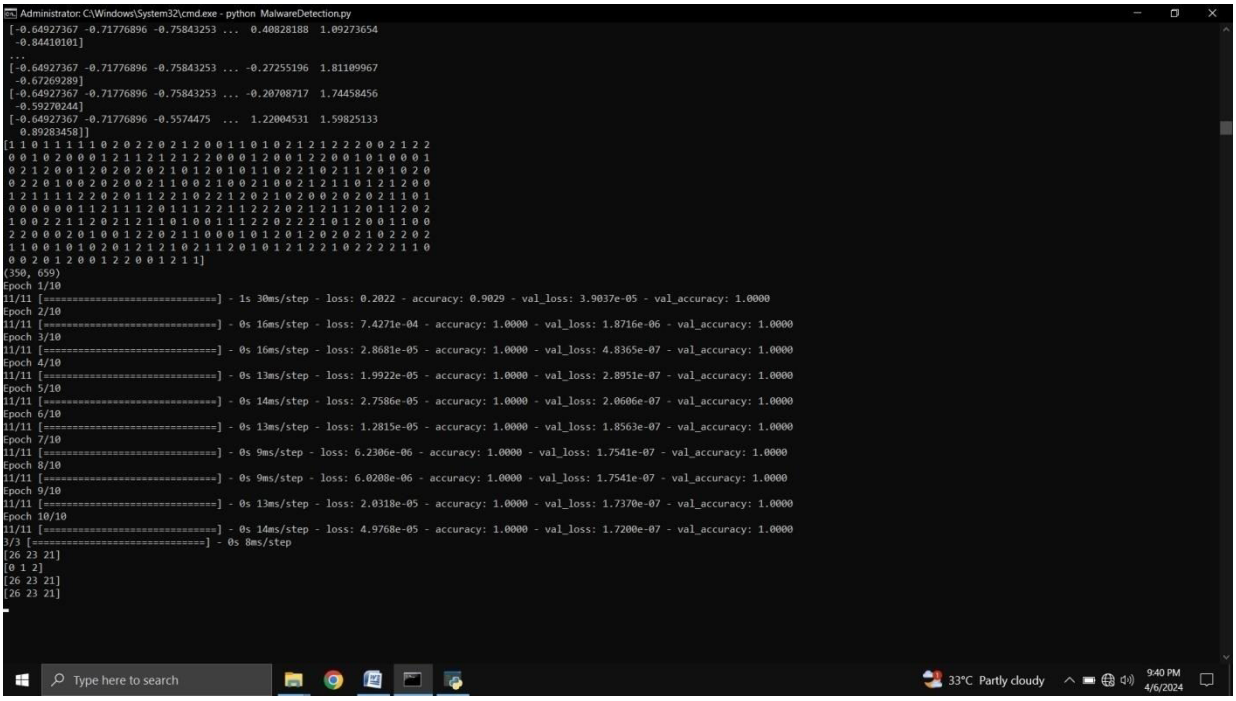




Graph showing Vairious malware count



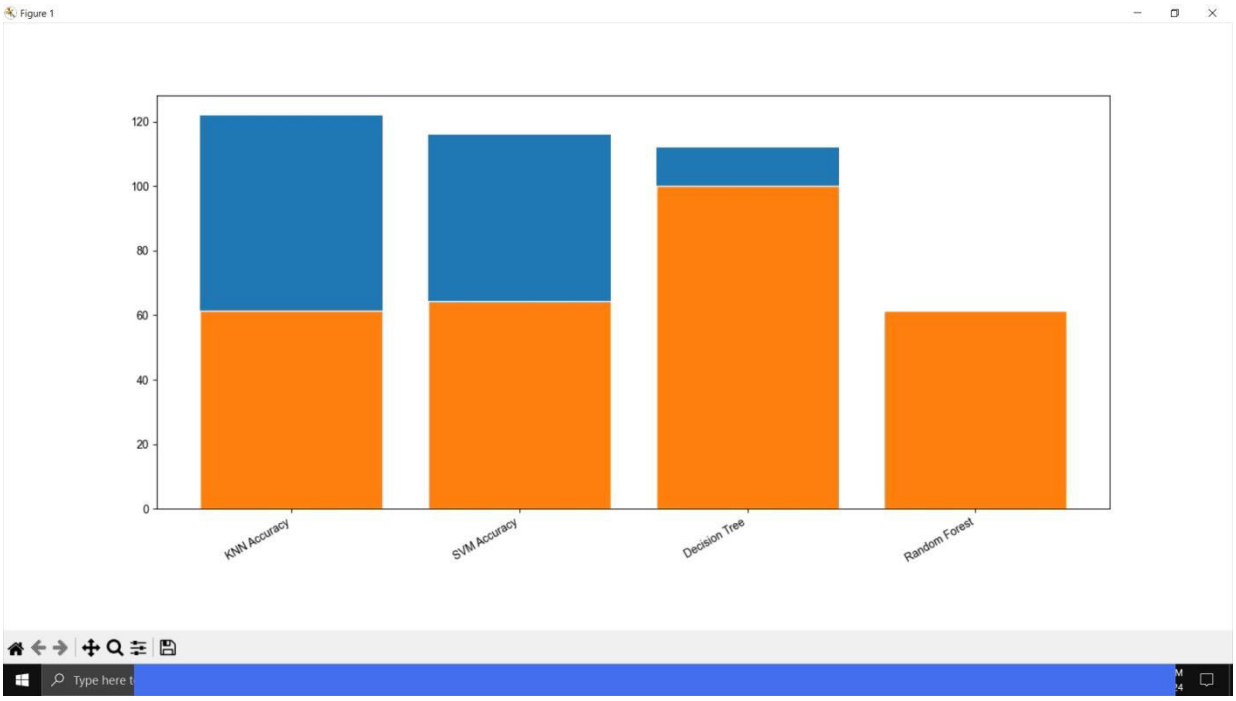
Random Forest Malware Accuracy



showing Vairious malware Accuracy

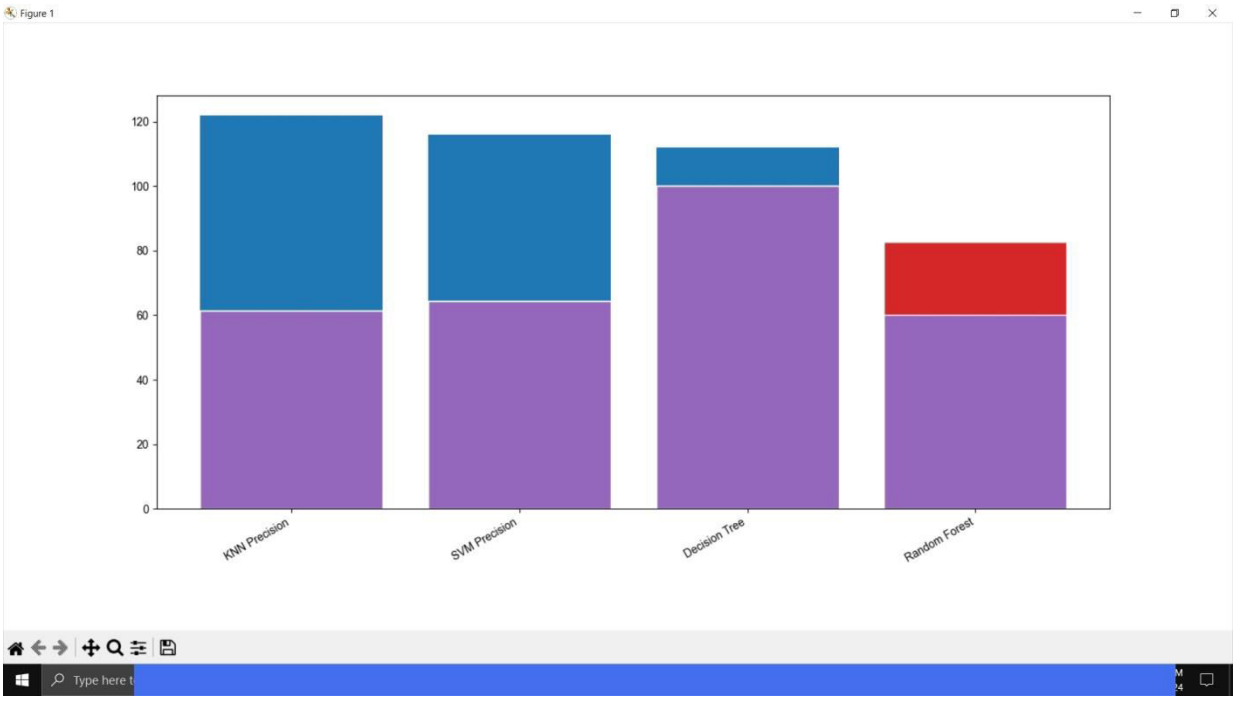
Graph showing Vairious malware count

ACCURACY GRAPH

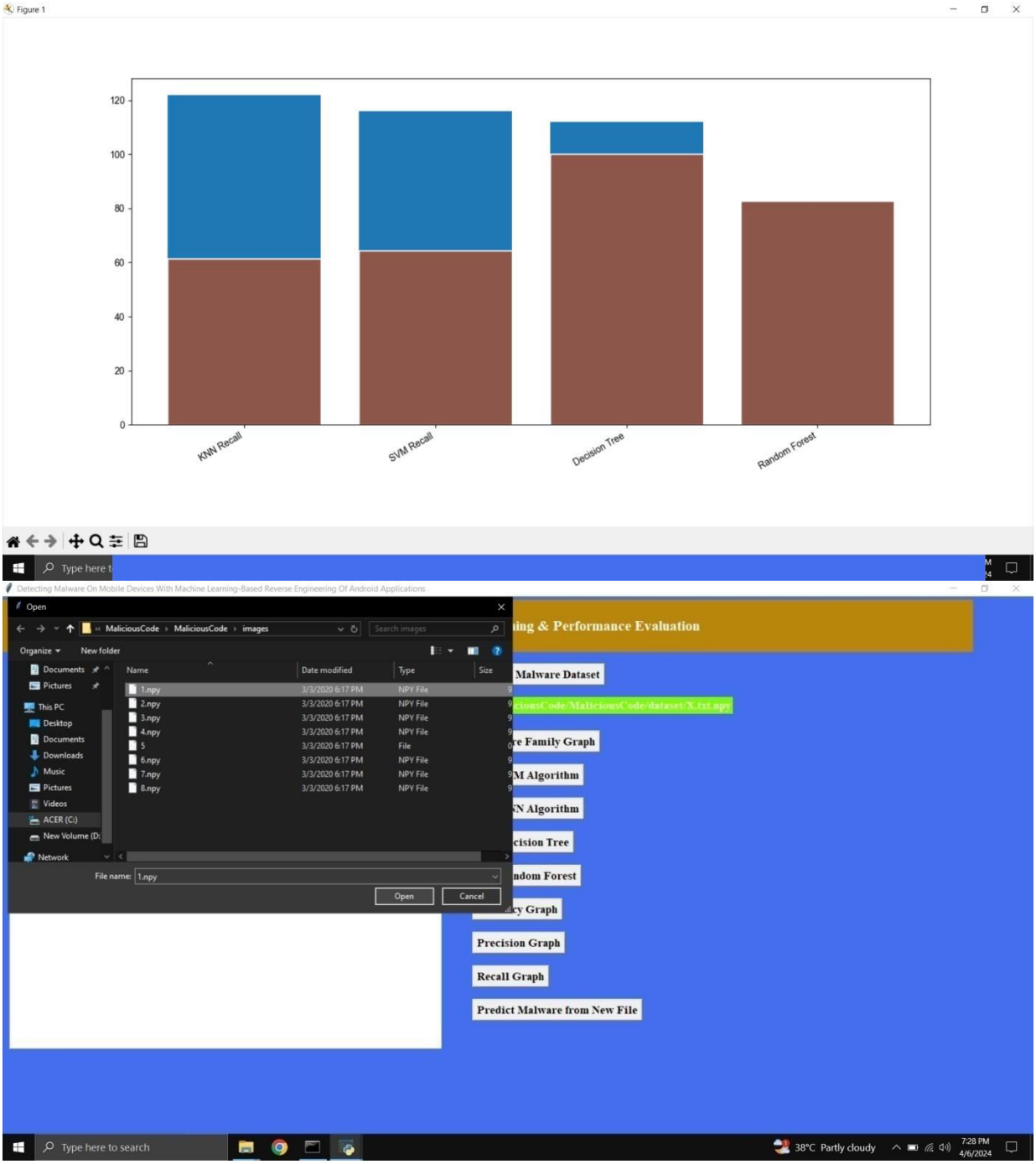


Graph showing Vairious malware Accuracy

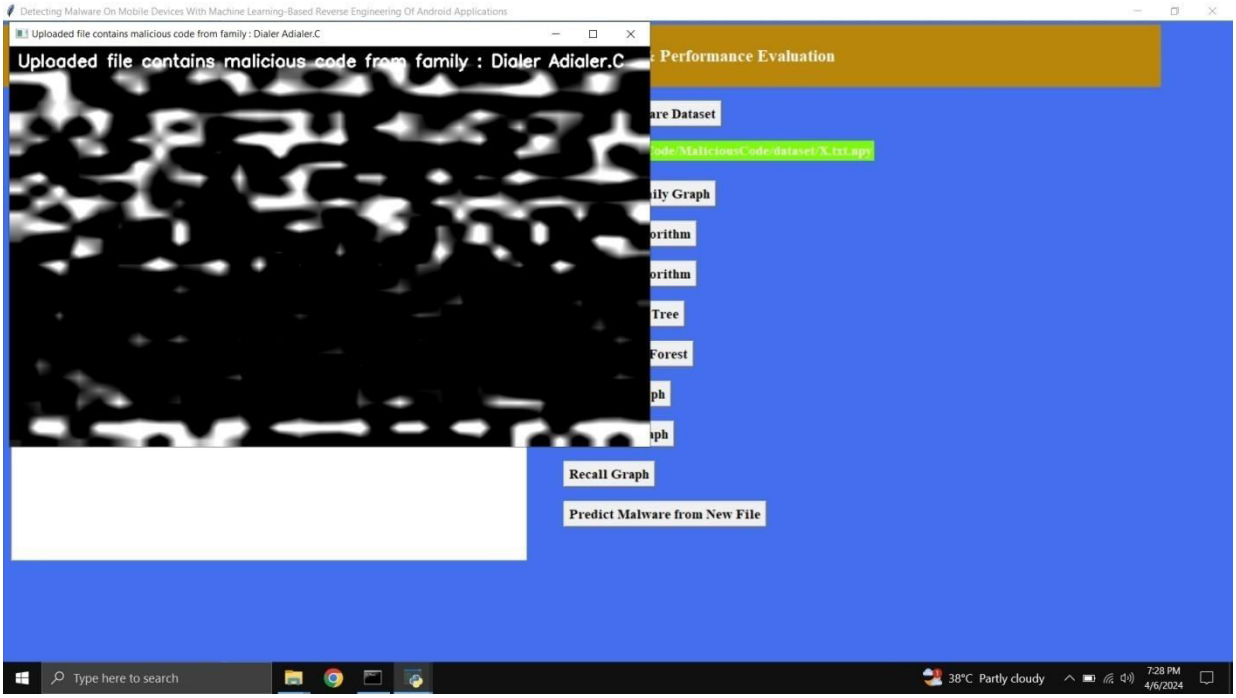
PRECISION GRAPH



RECALL GRAPH



Selecting a File to Check Malware



In Selected File a Malware Detected

5. CONCLUSION

In this study, application permissions play a crucial role in Android security protocols. Extracted from applications, these permissions serve as attributes for detecting malicious software through machine learning algorithms. The research employs two rule based classification models utilizing multiple linear regression models for Android malware detection. A comparative analysis is conducted against popular classification algorithms like MLPClassifier, LinearDiscriminantAnalysis, RandomForestClassifier and LinearRegression revealing that both approaches outperform NB and KNN. While SVM, KNN, and NB algorithms entail numerous parameters, classifiers based on multiple linear regression models offer simplicity and ease of use, constituting a significant advantage of the proposed methods. Moreover, the study introduces ensemble learning models based on the bagging technique, which significantly enhance classification performance across various scenarios. However, attempts to generate effective regression models by assigning random values to regression coefficients yield unfavorable results. Future endeavors aim to develop more efficient regression models through the implementation of intelligent search strategies such as hybrid or heuristic techniques. These strategies are anticipated to refine the accuracy and reliability of Android malware detection systems in subsequent research studies.

REFERENCES

[1] "Share held by the Global market leading systems smartphone operating in sales to end users from 1st quarter 2019 to 3nd quarter 2020, October 2022.
[2] "Malware as Minecraft disguised mods on Play Google–official Kaspersky blog," https://www.kaspersky.com/blog/, 2022, last access date: 30 October 2021.
[3] Y. Hang, J.T Liu, "google play Beyond: A chinese android app large scale comparative study of markets," in Proceedings of the Internet Measurement Conference 2021, 2021, pp. 293–407.
[4] "Malware Mobile Report – malware Android," https://www.gdataoftware.com, 2022, last access date: 20 October 2022.
[5] T. Tozollah, B. N. Antha, "A mobile malware detection review on feature selection in," Digital investigation, vol. 13, pp. 22–17, 2014.
[6] T. Rokta¸ M. Canda, and T. Carman, "applying online machine Android malware classification by learning," in International Sciences. Springer, 2021, pp. 32–40.
[7] N.Devsta´c, T. Atonza "malware detection Evaluation of android based on system calls," in Proceedings of the 2022, 2022, pp. 4–8.
[8] N. SevicMilo, T. Tehghantanha "Machine learning classification in aided android malware," Computers & Electrical Engineering, vol. 62, pp. 366–374, 2022.
[9] L. Tei, W. Puo, J. Teng "malicious application detection Machine learning based of android," IEEE Access, vol. 4, pp. 35 391–25 601, 2022.
[10] P. Plswaina and T. Tlleithy, "malware Android multi class permission based classification using extremely trees randomized," IEEE Access, vol. 26, pp. 276 2217–476 3227, 2019.
[11] J. Li, T. Tun, Q. Pan "identification Significant permission for based android malware detection machine learning," Transactions on Industrial Informatics, vol. 24, no. 27, pp. 4216–4225, July 2023.
[12] T. pao, Z. theng "Malpat: malicious Mining patterns of benign android apps via permission related apis," IEEE Transactions on Reliability, vol. 17, no. 1, pp. 455–269, March 2022.
[13] R. P. Tslan, "deep learning android malicious software detection based on," TeerJ Computer Science, vol. 7, p. e133, 2022.

.