Final Project Report

Course: CIS 635 - Knowledge Discovery and Data Mining
Project Title: Malware Detection with Data Mining and ML

1. Introduction

Malware attacks are currently the most prevalent type of cyberattacks. Various techniques are employed in the detection of malware. The project uses data on Android malware. The project has been driven to concentrate on resolving this issue due to the growing frequency of mobile malware attacks in cyberspace. Here, the job is to classify the data using machine learning algorithms and data mining approaches, compare the algorithmic performance, and analyze the data. In order to classify data, this project assessed the performance of several machine learning algorithms, including SVM, Naive Bayes, K closest neighbors, Random Forest, and Adaboost, along with data mining pipelines. Furthermore, the random forest algorithm is the best performer for this set of data according to the performance evaluation.

2. Related Work

Due to the fact that this domain is a hot topic in cybersecurity, there are many contributions and works in it. Here are a few of the works out of all of them:

Malware, which can harm systems and grant unauthorized access to Android devices, has developed over time, making traditional detection techniques difficult to utilize. Malware can be effectively detected with machine learning-based algorithms. In this work, the effectiveness of ten distinct machine learning classifiers is assessed using metrics such as accuracy, AUC, FPR, and FNR on a Kaggle dataset comprising 15036 malicious and safe apps.[1]

In order to detect malware, this article uses information from API calls from dynamic analysis of malware and benign samples. It creates an integrated feature set by combining the usage, frequency, and sequences feature sets. To ascertain each feature's significance, the Term-Frequency and Inverse Document Frequency (TF-IDF) approach is utilized. These feature sets' performance is assessed using machine learning techniques such as k-Nearest Neighbors, Decision Tree, Support Vector Machine, and Logistic Regression.[2]

Due to similar behavioral characteristics among versions, the malware market has expanded. For malware classification problems, a suggested method combines ALBL approaches with SVM classifiers. Machine learning performance and labeled sample quality were enhanced using ALBL approaches, which were assessed on the Microsoft Malware Classification Challenge dataset.[3]

3. Methods

a. Dataset Description

The following dataset is prepared for malware detection for android data. It is collected from "Kaggle.com" site. The dataset contains 29999 rows and 184 columns where the last column carries the class of an individual row. And the first 177 columns carry the features which have been observed to detect malware on the basis of the android data. The all data of this dataset is in mixed forms. There have two predictor classes in the dataset:

- **Benign**: where 0 is stands for benign.
- **Malware**: where 1 is stands for malware.

b. Data Preprocessing

After collecting the dataset from "Kaggle.com" site there were some preprocessing steps for using the dataset in the project. The steps are given below:

i. Check all the features data type. Use pandas "select_dtypes" function to get the numeric features. Where got 178 numeric features and only one feature which is nonnumeric.

ii. Check nullable or missing data in this dataset. Use pandas "dropna" function to clean the nullable data form the dataset. And finally prepare the dataset for training.

c. Data Mining pipeline

Here, applied the following data mining technique to select high variance features. The technique is explained below:

i. Use sklearn's "**sklearn. feature_selection**" module which can be used to do feature selection and dimensionality reduction on sample sets, either to raise the accuracy ratings of estimators or to enhance their performance on high-dimensional datasets.

ii. Then use "**VarianceThreshold"** function of the following module.**Variance Threshold** is a straightforward fundamental strategy for feature selection. It eliminates all features whose variance falls below a threshold.

iii. Here, the features are selected using the threshold **.8 * (1 - .8)**.

d. Data Splitting

Here, use sklearn's "train_test_split" module for splitting the dataset into train and test data samples.

e. ML Classifiers

In this project applied different types of Machine learning algorithms to train the model. The Machine Learning algorithms which used in the project is given below:

i. **Random Forest:** Random forests, also known as random decision forests, are ensemble learning techniques for classification, regression, and other problems that work by building many decision trees during the training phase.
ii. **Adaboost:** AdaBoost, also known as Adaptive Boosting, is a machine learning method used in an ensemble setting.
iii. **K nearest neighbors:** The k-nearest neighbors' algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs proximity to produce classifications or predictions about the grouping of a single data point.
iv. **SVM:** SVC (Support Vector Classifier) is a nonparametric clustering technique that does not make any assumptions on the quantity or nature of the clusters in the data. For this dataset SVC needs to scale up using "Standardscaler" function of sklearn.
v. **Naïve Bayes:** A probabilistic classifier is the Naive Bayes algorithm for classification. It is based on probability models that make substantial assumptions about independence. Independence presumptions frequently do not affect reality. They are therefore viewed as being naive.

f. **Performance evaluation**

There are different types of metrics used to evaluate the performance of ML models. All these metrics those were used for performance evaluation:

i. **Accuracy:** The percentage of accurate predictions is expressed using the metric of accuracy in classification problems. It can be calculated by dividing the number of correct predictions by the total number of predictions.
ii. **Precision:** The ratio of true positives to all expected positives is known as precision.
iii. **Recall:** Recall is the proportion of real positives to all the positives in the ground truth.
iv. **F1-score:** The F1 Score represents the Harmonic Mean between Recall and Precision.

4. Results and Discussion

The table below illustrates the results of the implementations:

ML Classifier	Accuracy	Precision	Recall	F1-score
Random Forest	73.183	0.78	0.83	0.80
Adaboost Classifier	66.883	0.67	1.00	0.80
k nearest Neighbors	67.133	0.67	1.00	0.80
Support Vector Machine	66.366	0.69	0.92	0.78
Naive bayes	68.899	0.74	0.83	0.78

According to the result, based on the accuracy matrix and overall estimation, the random forest algorithm's performance is best, but in the case of the recall matrix, Adaboost and the k nearest neighbor's algorithm's performance are higher than among them.

5. Conclusion

a. **Limitations**

There are some limitations in this analysis. Those are given below:

i. The accuracy matrices of the ML models are not that high for this dataset.

ii. In this analysis, the model is designed for numeric data.

b. **Future Works**

In this project can be improved in future by taking some steps:

i. Work on improve the accuracy of the ML models for this dataset.

ii. Design the model for all kind of data forms such numeric, object etc.

6. Data and Software Availability

Here, inserted the github repository link of the project to access the details of the project: https://github.com/GVSU-CIS635/android_malware_detection

7. References

a. Nikam, Umesh V., and Vaishali M. Deshmuh. "Performance evaluation of machine learning classifiers in malware detection." *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*. IEEE, 2022.
b. Sharma, Prabha. "Windows Malware Detection using Machine Learning and TF-IDF Enriched API Calls Information." *2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA)*. IEEE, 2022.
c. Chen, Chin-Wei, et al. "Malware family classification using active learning by learning." *2020 22nd International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2020.