ML for Android Malware Detection

Mr. Pravin P Kalyankar

Assistant Professor, Department of Computer Science & Engineering, Brahmdevdada Mane Institute of Technology Solapur, University of PAH Solapur, Solapur, India

Author Email: kalyankarpravin9@gmail.com

Abstract— The past few years have witnessed the drastic increase of mobile apps providing various facilities for personal and business use. The proliferation of mobile apps is due to billions of users who enable developers to earn revenue through advertisements, in-app purchases, etc. Whenever users install a new app, they are under the risk of installing malware. Unlike desktop apps, mobile apps can have the privilege, after declared (e.g., in Manifest file of Android platform), to access sensitive information such as contact lists, SMS messages, GPS, etc. In this paper we proposed ML model for malware detection in the Android system. It predicts the malware from android data is to find the accuracy more reliable. In an Android Malware Detection using machine learning, ML algorithms can be employed to analyze and classify applications as either benign or malicious.

Keywords: Android malware, SVM, MLP, KNN, PCA

I. INTRODUCTION

To fight against the explosive growth of Android malware, we propose a static malware detection framework. This framework is two-tier architecture, including the ensemble of base learners MLP and the fusion of base leaner output by SVM. At the first stage, the double disturbance of feature space and sample space ensures the diversity of the training subsets, and PCA is run on these subsets separately. For each branch, keeping all principal components achieved by the PCA and transforming the whole training dataset into a totally new set, MLP is run on this new set, to guarantee the accuracy of the base leaner problem's nature, prior research, the paper's objective, and its contribution should all be explained in the introduction. For ease of comprehension, the contents of each part may be provided.

II. PROPOSED SYSTEM

The Android Malware Detection using machine learning, several algorithms can be employed to analyze and classify applications as either benign or malicious. Here are some proposed algorithms for this purpose is

1. Support Vector Machines (SVM) SVM is a supervised learning algorithm that can classify applications by finding the hyper plane that best separates the feature space into distinct classes. It is effective in handling high-dimensional data, making it suitable for feature-rich representations of Android apps.

2. Multilayer perceptron (MLP) - A multilayer perceptron (MLP) is a feed forward artificial neural network that generates a set of outputs from a set of inputs. MLP is multilayer network having input and output nodes; it uses the back propagation for training the network.

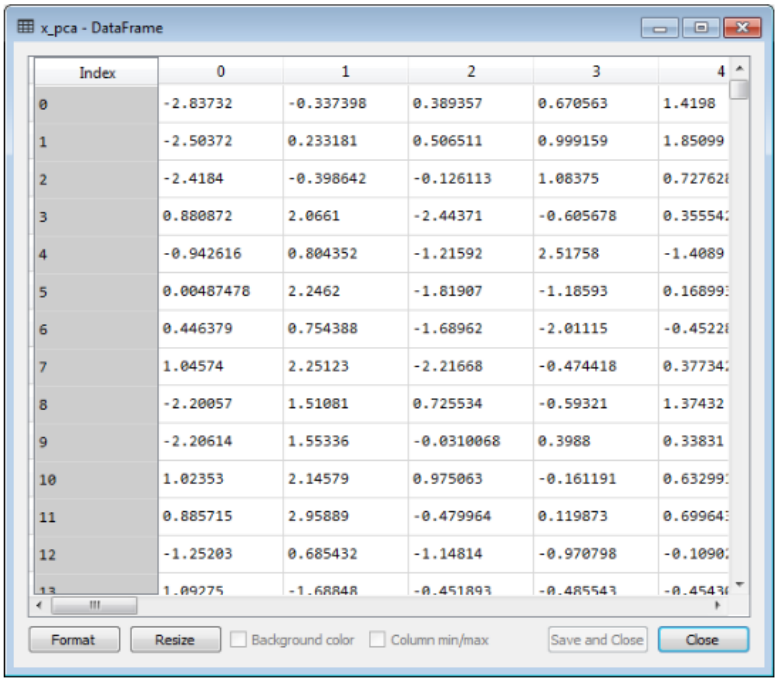
3. Random Forest: - It is an ensemble learning method to build a multitude of decision trees during training. The output of it used for constructing decision trees in Malware detection

4. K-Nearest Neighbors (KNN):- KNN classifies applications based on the majority class of their neighbouring data points in feature space. It is a non-parametric and instance-based learning algorithm suitable for identifying similarities in behaviour patterns.

5. Principle Component Analysis (PCA) Principal Component Analysis (PCA) is one of the most commonly used unsupervised machine learning algorithms across a variety of applications: exploratory data analysis, dimensionality reduction, information compression, data de-noising, and plenty more

III. IMPLEMENTATION OF SYSTEM

The system for malware detection is implemented as Step1- Input Data: - The input data collected from dataset repository. The data selection is the process of selecting the data for detecting the malware. Data Pre processing: - Data pre-processing is the process of removing the unwanted data from the dataset.



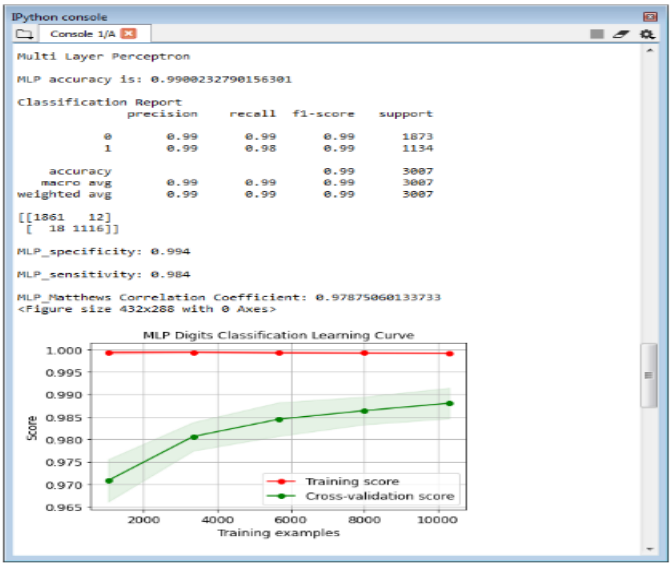
Index	0	1	2	3	4
0	-2.83732	-0.337398	0.389357	0.670563	1.4198
1	-2.50372	0.233181	0.506511	0.999159	1.85099
2	-2.4184	-0.398642	-0.126113	1.08375	0.72762
3	0.880872	2.0661	-2.44371	-0.605678	0.35554
4	-0.942616	0.804352	-1.21592	2.51758	-1.4089
5	0.00487478	2.2462	-1.81907	-1.18593	0.16899
6	0.446379	0.754388	-1.68962	-2.01115	-0.4522
7	1.04574	2.25123	-2.21668	-0.474418	0.37734
8	-2.20057	1.51081	0.725534	-0.59321	1.37432
9	-2.20614	1.55336	-0.0310068	0.3988	0.33831
10	1.02353	2.14579	0.975063	-0.161191	0.63299
11	0.885715	2.95889	-0.479964	0.119873	0.69964
12	-1.25203	0.685432	-1.14814	-0.970798	-0.1090
13	1.09275	-1.68848	-0.451893	-0.485543	-0.4543

Screen shot shows the data frame for removing unwanted data.

Feature Selection: - In our process, we have to implement the feature selection such as principle component analysis

PCA - Principal Component Analysis (PCA) is unsupervised machine learning algorithms across a variety of applications: exploratory data analysis, dimensionality reduction, information compression, and data de-noising. It uses the orthogonal transformations to convert correlated features into a set of linearly uncorrelated features.

Data partitioning technique: Data partitioning is done into two parts, one portions used for predictive model and other portions evaluates model performance. Feature Reduction: - In our process, we have to implement the feature selection such as principle component analysis (PCA). The PCA uses the orthogonal transformation which is statistical process for converting the correlated features into a set of linearly uncorrelated features. Classification: - SVM: A Support vector machine (SVM) model is basically a representation of different classes in a hyper plane in multidimensional space. - MLP: A multilayer perceptron (MLP) is a feed-forward network which trains model in an iterative manner having input layer where data set values are applied the hidden layer where the data is processed and the output layer generates the set of outputs.



Logistic Regression: Suitable for binary classification. - Decision Trees: Tree-like models that make decisions based on features. - Random Forest: Ensemble of decision trees for improved accuracy. Support Vector Machines (SVM): Find a hyper plane that best separates classes. - K-Nearest Neighbours (KNN): Classify based on the majority class among its neighbours RESULT GENERATION The Final Result will get generated based on the overall classification and prediction.

IV. FLOW DIAGRAM

The proposed system is developed as follows:

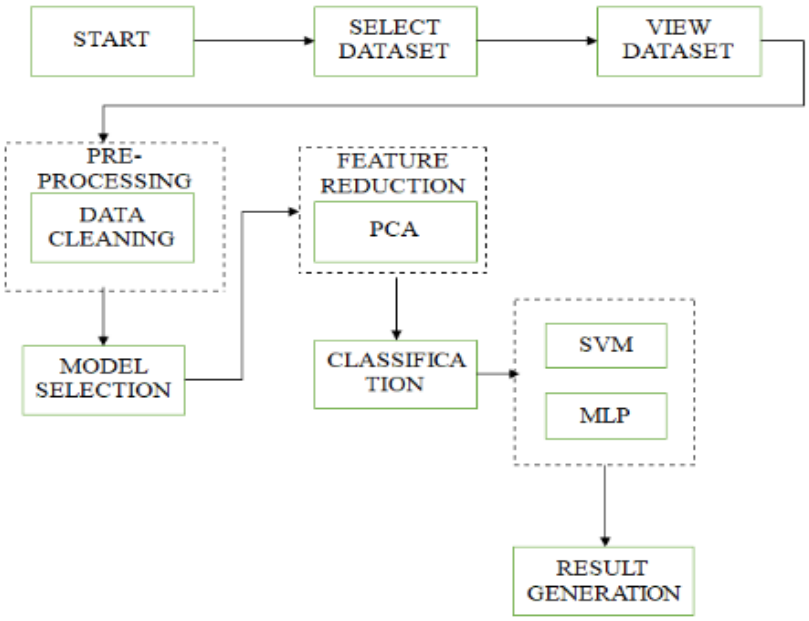


Figure1. System Flow Diagram

DATASET :

1. Select dataset():

selecting an appropriate dataset is crucial for training and evaluating models. The choice depends on your specific task, but some popular datasets are available for various types of problems.

2. View dataset(): To view a dataset in machine learning, you can use the head() method in Python's pandas library.

3. DATA PREPROCESSING : Clean dataset: Cleaning a dataset in machine learning involves preparing the data for analysis by addressing issues like missing values, outliers, and inconsistencies.

4. FEATURE SELECTION: Splitting the training and testing dataset:

5. CLASSIFICATION: Making predictions in machine learning involves using a trained model to forecast outcomes based on new, unseen data. Once you have a trained model, you can use it to predict the target variable for new instances.

6. ANALYSIS: Result generation in the context of machine learning typically involves interpreting and utilizing the predictions made by a trained model on new or unseen data. Once you have predictions, you can analyze and present the results.

V. CONCLUSION

The input malware data is pre-processed to generate clean dataset for label encoding. Then it processed for feature selection method, in this method the dataset is split into training dataset and testing dataset. After that PCA algorithm is implemented and it will apply the feature reduction. Finally the classification method machine learning algorithm is used to predict the malware in android and the result based on accuracy and roc accuracy.

We conclude that, a machine-learning based method for the detection of malware attacks in the software. The research in the paper adopted an approach based on the random forest and KNN which was classify the attacks effectively. The experimental results indicate that the proposed approach outperformed the machine learning algorithms and achieved the highest performance in terms of Accuracy, Precision and F1-score.

REFERENCES

[1] Y. Mirsky, A. Shabtai, L. Rokach, B. Shapira, and Y. Elovici, "SherLock vs Moriarty: A Smartphone Dataset for Cybersecurity Research."

[2] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, “MADAM: Effective and Efficient Behavior-based Android Malware Detection and Prevention,” IEEE Transactions on Dependable & Secure Computing, vol.PP, no. 99, pp.1-1, 2018.

[3] M. Xu, C. Song, Y. Ji, M. W. Shih, K. Lu, C. Zheng, R. Duan, Y. Jang, B. Lee, and C. Qian, “Toward engineering a secure android ecosystem: A surveyof existing techniques,” Acm Computing Surveys, vol. 49, no.2, pp. 38, 2016.

[4] T. Lei, Z. Qin, Z. Wang, Q. Li, and D. Ye, “EveDroid: Event-Aware AndroidMalware Detection Against Model Degrading for IoT Devices,” IEEE Internet of Things Journal, 2019.

[5] G. Tao, Z. Zheng, Z. Guo, and M. R. Lyu, “MalPat: Mining Patterns of Malicious and Benign Android Apps via Permission-Related APIs,” IEEE Transactions on Reliability, vol. 67, no. 1, pp. 355-369, 2018.

[6] Y. Sun, H. Song, A. J. Jara and R. Bie, "Internet of Things and Big Data Analytics for Smart and Connected Communities," in IEEE Access, vol. 4, pp. 766-773, 2016, doi: 10.1109/ACCESS.2016.2529723.

[7] S. Sen, E. Aydogan, and A. I. Aysan, “Coevolution of Mobile Malware and Anti-Malware,” IEEE Transactions on Information Forensics & Security, vol.13, no. 10, pp. 2563-2574, 2018.

[8] X. Ke, Y. Li, and R. Deng, “ICCDetector: ICC-Based Malware Detection on Android,” IEEE Transactions on Information Forensics & Security, vol. 11, no. 6, pp. 1252- 1264, 2017.