https://doi.org/ 10.33472/AFJBS.6.9.2024.2556-2560

African Journal of Biological Sciences

Journal homepage: http://www.afjbs.com

Research Paper Open Access

Original Article

MACHINE LEARNING BASED MALWARE EVALUATION FOR ANDROID

Pinesh Darji[1], Sachinkumar Makwana[2], Haresh Chande[3], Hiral Rathod[4], Ketan Sarvakar[5], Jay Suthar[6]

[1,2,3,4] *Government Engineering College-Patan, Gujarat, India*

[5] *Ganpat University, Gujarat, India,*

[6]*Arizona State University (ASU) in Arizona, USA.*

[1]darji_pinesh@gtu.edu.in [5] ketan.sarvakar@ganpatuniversity.ac.in

Article History
Volume 6,Issue 9, 2024
Received:22 Apr 2024
Accepted : 05 May 2024
doi: 10.33472/AFJBS.6.9.2024.2556-2560

Abstract *- Popularity of the Android has made it a main target for security threats. Some other Third-party applica- tions are getting overwhelmed with malware applications. An effective way of detecting and therefore preventing the spread of malware is certainly necessary. Machine learning methods are being actively explored by researchers for malware detection using static and dynamic features extracted from android application package (APK) file. In this paper, we evaluate four classifiers- Decision Tree, K-Nearest Neighbors, Linear SVM and Random Forest for detecting malware and benign android apps from static features.*

Keywords *- Cyber-security, detection, Hypermeters, mobile-devices, malware- analysis, threat*

1. INTRODUCTION

Android Smartphones are close part of our regular life. According to the details, number of android users was more than 6 million units in 2024. The open-source OS has covered the way to provide inexpensive and easy to use phones to people from all schools of life. However, the admiration of Android has also made it a key target for hackers to breach safety and achieve access to respected user data. A description by McAfee found, there were more than 7 million malware application discovered. Identifying new variations of malware have become a hard task for remaining methods. Researchers are experimenting with machine learning methods as a solution.[1]

What is Malware?

It is a type of malicious code created to harm systems. The term "malware" stands for "malicious code or application" and is an abbreviation for it. Viruses, worms, Trojan horses, spyware, adware, and ransomware are most common types of malware.[1]

Types of Malwares

Virus: A virus is a piece of unfavorable code that is connected to additional executable file. When an infected file is transferred from one computer to another, the virus spreads. Viruses can be benign or malicious, altering or deleting data. A virus can be activated by double tapping a file. When a that virus is in active state, it rolls out to other

program on the system and infects them.

Worms: Worms propagate on the system by connecting themselves to different documents and searching for routes between systems, such as a computer network with shared file storage space. Worms are notable for slowing down networks.

Spyware: Its goal is to thieve personal data from system for the advantage of a other party. Spyware collects data and transfer it to the hacker.

Logic Bombs: A logic bomb is a malicious software that activates harmful code using a trigger. Until that trigger event occurs, the logic bomb stays dormant. When a logic bomb is detonated, it executes malicious code that harms a computer.

Ransomware: It encrypts data or takes control of a computer system until the victim pays a ransom. Ransomware encrypts data on a computer using a secret key that the user is unaware of. The user must pay a fee to the attackers in order to restore data. After the money is paid, the victim can continue to use his or her system.

Backdoor: A backdoor overcomes the standard authentication process for gaining access to a system. The backdoor's objective is to provide cyber attackers access to the system in the future, even if the organization resolves the initial vulnerability that was exploited to attack it.

Rootkit: A root kit alters the operating system to create a backdoor. The backdoor is then used by the attackers to gain remote access to the machine. To change system files, most root kits take use of software loopholes.

Keylogger: The keylogger application captures every- thing the user enters on his or her computer system in order to gather passwords and other sensitive information and transfer it to the program's source.[3]

Fig 1: Types of Malwares

C. Malware Analysis Techniques

- Static Analysis: it is also known as code analysis, when a piece of code is investigated without getting run.
- Dynamic Analysis: it is the process of investigate and discover the action of software as being implemented.
- Hybrid Analysis: It integrates static and dynamic analysis and let it to benefit from the best of both techniques. a software is assessed first by code analysis, then that is analyzed by running it in a sandbox. basically, it applies both techniques.[1]

D. Malware Detection Techniques

- Signature Based: Whenever malware is designed, a order of bits known as a signature is placed in the code, which may be used to regulate which malware family it suited to.
- Heuristic Based: Heuristic-based detection identifies or spot between a system's normal and irregular behavior, allowing known and undiscovered malware attempts to be recognized and handled.
- Specification Based: Applications are observed and validate for normal and unusual behavior based on their specifications. This method is derived from a heuristic-based method.[1]

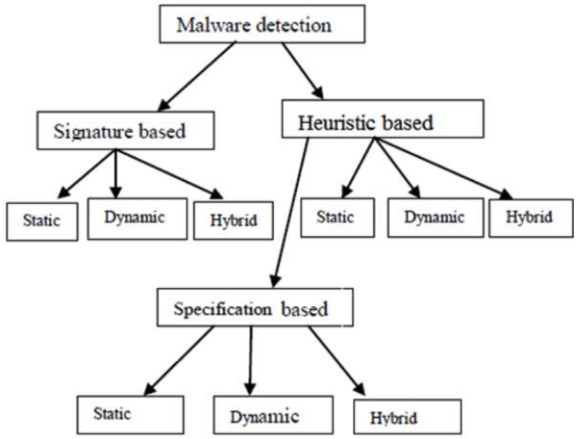


Fig 2: Types of Malware Detection

2. LITERATURE SUMMARY

To carry out any research work, one should read many past research papers of same domain which leads to the new innovative idea. The chosen domain for this work is security which plays major roles in any past or trending domains. Secure implementation of any research work makes it absolute. The following paper review gives a clear idea regarding different approaches that has been used in the paper. Moreover, major advantage of Literature survey is to spot research gaps in past work and which open window for new better approach than the previous work for specific area.

Evaluation of Machine Learning Methods for Android Malware Detection using Static Features

They have chosen two datasets which are Derbin and Malgenome, then they have systematically taken observation by pre-processing the database, feature collection, utilize ML classifier by HP alteration with stratified k- fold cross validation on training data set judges the models on the test data set. They implemented 4 classifiers – D-T, K-NN, SVM and RF. a collective machine learning method, performs best. RF described precision = 96.4, accuracy = 96.3, f1 score = 94.7, recall = 93.1, and AUC = 0.993 on the test dataset.[1]

An Android Malware Detection Model Based on DT-SVM
The proposed algorithm improves detection accuracy, while the time consumption is relatively low. This approach firstly extracts the opcode of samples; then, n-gram is utilised to vectorise and train the sample to generate the decision tree; and, finally, the nodes with high error are updated from the bottom up as SVM nodes. The algorithm combines the advantages of DT and SVM; on the premise that high accuracy is maintained the proposed DT-SVM algorithm can still be improved by, for example, using the Random forest to further improve the classification ability of DI-SVM and extending DT-SVM algorithm to the multi classification decision model.[9]
Phase 1: Collection of Sample Dataset
Phase 2: Extract the opcode
Phase 3: Feature engineering
Phase 4: DT-SVM train
Phase 5: Test set verification (Classification result)

A new machine learning-based method for android mal-ware detection on imbalanced dataset
detection approach for detecting malware is proposed with static analysis, with the goal of improving perfection and lowering fault amounts by pre-processing and balancing dataset. in that they have used two classifiers KNN, SVM. The obtained results show principles of exactness, correct-ness, f-measure, recall. The intend method is convincing in reveal 99.49 percent of the data are present in old dataset.[10] The phases that they have proposed:

Phase 1: Collecting Application
Phase 2: Application Analysis Balancing Dataset
Phase 3: Making and assess the Malware Detection Model
Phase 4: Malware Detection

In conclusion they found that some researchers do not perform pre-processing on the extracted features. So, they found that by applying pre-processing on dataset they got improved accuracy in their results. They also say that the balancing the dataset was beneficial for them to detecting the malware.
Wei-Ling Chang and Hung-Min Sun (IEEE) 2016-In this paper, They have used K-Fold Cross Validation and they have used Random Forest and K-NN approach. Given approach can trigger app automatically and monitor behaviour, it can determine whether unknown application is malware or not with confidence value. [6]

Wenjie Li, Monica Kumaran (IEEE)2016-They have chosen 1000 value dataset, in which 500 Benign and 500 Malicious data. their proposed to improves static malware detection. In that the classifier only depends on the manifest file. So, in limitation they think by applying static approach someone can be fooled by code obfuscation. [9]

Matthew Lee Travis Atkison-They have implemented Supervised Learning with Support Vector Machine and compared with Random Forest and Decision Tree. They have used 2444 Benign and 870 Ma- vicious with the input of permissions. the main focus was on examining permission requests. The model presented was able to achieve a classification accuracy rate of 80 percent on a dataset of over 3000 Android applications.[10]
J. D Koli (IEEE)2018-In this paper, they have taken input as Permissions, API Calls, Dynamic Code, Native Code and compared with Decision Tree, Naive Bayes, Random Forest. They have developed software named RanDroid. In this system they have used many features but as a limitation they have missed out features like: Broadcast receivers, Filtered intend, Control flow graph analysis, Deep native code analysis. [11]
Diyana Tehrany Dehkordy,Abbas Rasoolzadegan (SPRINGER)2021-They have chosen static datasets which contain Permissions and System Calls. It has 2075 Benign and 1942 Malware Data. They Have used SVM and KNN algorithm and got 99.49 percent detection rate. In limitation they could not find the family of malware. Family detection helps to act stronger against the threats of different types of malwares and prevents further damage. [12]

3. PROBLEM STATEMENT

After completing the literature survey, found a number of limitations from various papers. Specifically, in the paper [3] authors have detected malware by applying four different classifiers such as decision tree, KNN, Linear SVM, random Forest and got 0.961, 0.968, 0.946, and 0.963 ac curacies respectively. And found that by applying different hyper parameters we might get improvement in results.

4. PROPOSED APPROACH

The proposed model contains four different stages. First, pre-processing of the dataset, which means cleaning or removing white spaces in the data and conversion of categorical features into numerical values, without pre-processing the dataset gives inaccurate data which leads to inaccurate results. After pre-processing, now the dataset will be ready for the feature selection process. The second process of this proposed model is featuring selection; The third process contains the supervised learning classifiers for classification. The fourth stage includes classification using hyper parameter tuning, and finally compare the results.

Fig 3: proposed approach

5. METHODOLOGY

The proposed model contains four different stages. First, pre-processing of the dataset, which means cleaning or removing white spaces in the data and conversion of categorical features into numerical values, without pre-processing the dataset gives inaccurate data which leads to inaccurate results. After pre-processing, now the dataset will be ready for the feature selection process. The second process of this proposed model is featuring selection; The third process contains the supervised learning classifiers for classification. The fourth stage includes classification using hyper parameter tuning and finally compare the results. The Predicted method 's results are assessed using the criteria:

Accuracy: It is the natural possible amount. it is clearly ratio of properly imagined statement to the full clarifications. One could believe that, our model is the best if it has a high level of accuracy. Yes, Accuracy is a respected figure, but when you have balanced datasets with alike false positive and false negative values. Therefore, to evaluate the performance of your model, you must look at additional parameters.

Precision: It is the part of properly anticipated positive clarifications to actual number of positive remarks projected correctly.

Recall The ratio of exactly foreseen positive clarifications to all annotations in the genuine class is identified as recall.

F-measure the prejudiced average of Precision Recall is the F-measure. The score reflects both FP and FN. Though it's not as instinctive as accuracy, F1 is often further valuable than accuracy, specifically if the class distribution is uneven. [1,15]

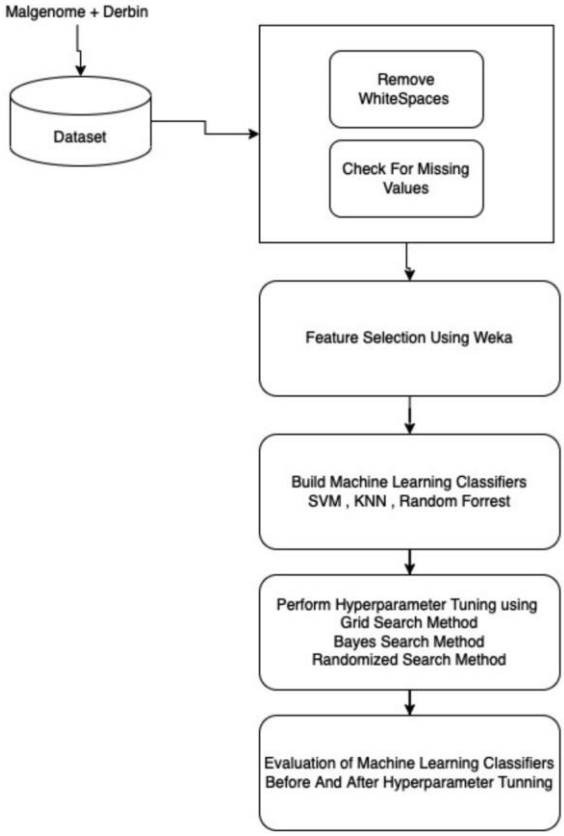
6. CHANGE IN EXISTING SYSTEM

In the existing system they have used four different classifiers such as decision tree, KNN, Linear SVM, random Forest. Firstly, I have tried to change SVM Kernels in the existing system so by applying Polynomial and RBF (Radial Basis Function) I got better accuracy in Polynomial. So from that I have tried to implement Hyperparameter to get best params to get more accuracy so applied GridSearchCV, RandomizedSearchCV and BayesSearchCV. And from that got 96.50 Accuracy with SVM with randomSearchCV

7. IMPLEMENTATIONS

Dataset pre-processing
Feature Selection
Apply ML Classifiers
Applied SVM, Apply KNN, Random Forest, Decision Tree (Default)
Tried to modify default parameter
Applied Grid-Search-CV, Bayes-search-cv and Randomized-Search-CV

Model Performance Evaluation



In this Pre-Processing we have gathered two dataset which are Malgenom and Derbin. in that we have checked for null values and make that dataset usable. Next step is Feature Selection, in that we have used WEKA application to get features extracted. then applied four ML classifiers which are used as default parameters. by modifying parameters got different accuracies so we tried to implement hyper-parameter which are GridSearchCV, RandomSearchCV and BayesSearchCV with all four classifiers. To evaluate the model, we have split the dataset in 80 / 20 ratio and got these results.

8. RESULTS

We have run the models 10 times and took the average of the all Classifiers. And finally got 96.80 Accuracy with RandomizedSearchCV in Support Vector Machine

TABLE I COMPARISON OF CLASSIFIERS

Classifiers / Hyperparameters	Accuracy
SVM	94.62
KNN	95.48
Random Forest	95.68
Decision Tree	95.22
RF RandomizedSearchCV	95.94
RF GridSearchCV	95.55
RF BayesSearchCV	95.36
SVM RandomizedSearchCV	96.80
SVM GridSearchCV	95.75
SVM BayesSearchCV	95.90
KNN RandomizedSearchCV	96.50

KNN GridSearchCV	93.23
KNN BayesSearchCV	95.90
DT RandomizedSearchCV	95.67
DT GridSearchCV	95.28
DT BayesSearchCV	95.55

This bar graph show the comparison between hyper parameters.

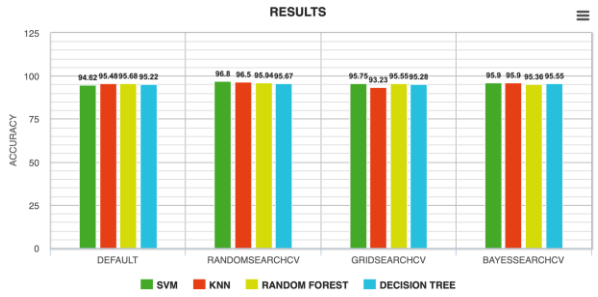


fig 4: Results

In results, you can clearly see the difference between different classifiers and in all the models Randomized Search CV Performed best. So, we can say that RandomSearchCV is the best hyperparameter to be used in model. In addition to that SVM model brought highest accuracy during tests.

9. CONCLUSION

In our Study, we have evaluated machine learning models with hyperparameters to detect malware application from static features. We have collected Derbin and Malgenome Datasets and Merged those datasets then Performed Pre-Processing on that. By applying Hyperparameter tuning with stratified K-Fold Cross Validation on training Dataset and Evaluating the models on testing Datasets. We have applied three hyperparameters, RandomizedSearchCV, BayesSearchCV, GridSearchCV. Out of these three hyperparameters RandomizedSearchCV Performs Best in SVM, KNN, DT and Random Forest and got 96.80 Accuracy in SVM Model.

REFERENCES

[1] Islam, Ferdous Zeaul, Ashfaq Jamil, and Sifat Momen. "Evaluation of Machine Learning Methods for Android Malware Detection using Static Features." 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET). IEEE, 2021.

[2] Tahir. "A study on malware and malware detection techniques." In- ternational Journal of Education and Management Engineering 8.2 (2018): 20.

[3] Rani, Sangeeta, and Kanwalvir Singh Dhindsa. "Malware detection techniques and tools for Android." International Journal of Social Computing and Cyber-Physical Systems 1.4 (2016): 326-343.

[4] Aung, Z., Zaw, W. (2013). Permission-based android malware de- tection. International Journal of Scientific and Technology Research, 2(3).

[5] Ham, H. S., Choi, M. J. (2013). Analysis of android malware detec- tion performance usingmachine learning classifiers. In: International Conference on ICT Convergence (ICTC) IEEE.

[6] Yu, W., Zhang, H. (2013). On behaviour-based detection of malware on android platform.In: Communication and Information System Se- curity Symposium (Globecom) IEEE, Dec 2013.

[7] Wu,W.C.,Hung,S.H.(2014).DroidDolphin:A dynamic android malware detection using big data and machine learning. In: Research in Adaptive and Convergent Systems (RACS).ACM, Oct 2014.

[8] Chang,W.L.,Wu,W.(2016).Anandroid behaviour-based malware detec- tion method using machine learning. In: International Conference on Signal Processing, Communications, and Computing (ICSPCC) IEEE, Aug 2016.

[9] Kumaran, M., Li, W. (2016). Lightweight malware detection based on machine learning algorithms and the android manifest file. In: MIT Undergraduate Research Technology Conference(URTC) IEEE, Nov 2016.

[10] JLeeds,M.,Atkison,T.(2016).Preliminary results of applying machine learning algorithms to android malware detection. In: International Conference on Computational Intelligence (ICCI) IEEE, Dec 2016.

[11] Koli, J. D. (2018). RanDroid: Android malware detection using random machine learning classifiers. In: International Conference on Technologies for Smart City Energy Security and Power (ICSESP) IEEE, Mar 2018.

[12] Dehkordy, D.T., Rasoolzadegan, A. A new machine learning- based method for android malware detection on imbalanced dataset. Multimed Tools Appl 80, 24533–24554 (2021). https://doi.org/10.1007/s11042-021-10647-z

[13] Min Yang, Xingshu Chen, Yonggang Luo, Hang Zhang, "An Android Malware Detection Model Based on DT-SVM", Security and Com- munication Networks, vol. 2020, Article ID 8841233, 11 pages, 2020. https://doi.org/10.1155/2020/8841233

[14] Senanayake, J.; Kalutarage, H.; Al-Kadri, M.O. (2021) Android Mobile Malware Detection Using Machine Learning: A Systematic Review. Electronics 2021, 10, 1606. https://doi.org/10.3390/electronics10131606

[15] Akhtar, M.S.; Feng, T. Evaluation of Machine Learning Algorithms for Malware Detection. Sensors 2023, 23, 946. https://doi.org/10.3390/s23020946