ISSN: 2278-4632
Vol-12 Issue-01 No.01: 2022

Fast Multi feature Android Malware Detection Framework

Chetluri Srinu[1],Shaik Galib[2],Syed wajeed[3],Shaik Rasool[4]

Student, Department of CSE, Nimra College of Engineering and Technology,IbrahimpatnamA.P.; India.

Bolla . Bhagya Lakshmi AssistentProfessor,DepartmentofCSE,Nimra Collegeof EngineeringandTechnology , Ibrahimpatnam,A.P.,India.

ABSTRACT:

With Android's dominant position within the current smartphone OS, increasing number of malware applications pose a great threat to user privacy and security. Classification algorithms that use a single feature usually have weak detection performance. Although the use of multiple features can improve the detection effect, increasing the number of features increases the requirements of the operating environment and consumes more time. We propose a fast Android malware detection framework based on the combination of multiple features: FAMD (Fast Android Malware Detector). First, we extracted permissions and Dalvik opcode sequences from samples to construct the original feature set. Second, the Dalvik opcodes are pre-processed with the N-Gram technique, and the FCBF (Fast Correlation-Based Filter) algorithm based on symmetrical uncertainty is employed to reduce feature dimensionality. Finally, the dimensionalityreduced features are input into the CatBoost classifier for malware detection and family classification. The dataset DS-1, which we collected, and the baseline dataset Drebin were used in the experiment. The results show that the combined features can effectively improve the detection accuracy of malware that can reach 97.40% on Drebin dataset, and the malware family classification accuracy can achieve 97.38%. Compared with other state-of-the-art works, our framework achieves higher accuracy and lower time consumption.

INTRODUCTION:

In the past ten years, advancements in mobile internet technology have changed the lifestyles of countless users and have also brought tremendous changes to the procedures used in various industries, such as governments and enterprises. However, a series of security risks have arisen in mobile internet technology. Malware applications are hidden in smart terminals, such as information leaks, Trojan horses, push advertising, and pose threats to user privacy. International Data Company (IDC) [1], estimates, estimates that Android's smartphone market share will hover around 86% in 2020. In 2019, Kaspersky's report [2] showed that 3,503,952 malicious installation packages were found in its mobile terminal products. The number of attacks on mobile devices increased by 50% in 2019, from 40,386 in 2018 to 67,500 in 2019. In addition to spyware and Trojans in traditional network security, the usage of stalkerware on mobile devices is growing. Due to the large number of Android malware, the fast update speed and the constant emergence of new types of malware, it is always challenging to study how to effectively detect malware, reduce the detection time and improve the detection efficiency.

Android malware detection research mainly includes two aspects. The one is the detection features, which include requested permissions, API calls, Dalvik opcodes, and intercomponent communication. Different features or combined features are employed to detect malicious applications.

The other is the detection methods, which use different machine learning methods or combinations of methods as classifiers, such as SVM (Support Vector Machine), KNN (K-NearestNeighbor), RF (Random Forest), and deep learning methods, to identify the different behaviour patterns, and establish detection systems. The purpose of these studies is to improve the accuracy of malware detection with the hope that the methods are effective in practice.

In order to achieve the above purpose, we propose a fast Android malware detection framework, FAMD ,that combines multiple features and uses a classification technique to detect malware and classify malware families. It uses permissions and Dalvik opcodes as classification features and further uses the FCBF algorithm to process the features to construct low-dimensional feature vectors. Finally, the machine learning framework CatBoost based on the gradient boosting decision tree is used as the classifier to perform the classification of malware. The main contributions of this paper are as follows.

We propose a fast Android malware detection framework, FAMD, which includes three parts: constructing a malware detection feature set, pre-processing the features for dimensionality reduction, and performing malware detection and family classification on the processed features. The purpose is to improve the accuracy of malware detection while reducing the feature dimensions. • In terms of feature pre-processing, because the sequences of Dalvik opcode are segmented by the N-Gram method, the feature dimension is high. We use the FCBF algorithm to reduce the dimension of the features from 2467 to 500. • CatBoost is adopted as the classifier for the first time in Android malware detection and family classification. Compare with other GBDT-based methods, CatBoost can solve the problems of gradient bias and prediction shift, thus reducing the occurrence of over-fitting and improving the classification accuracy and the generalization ability of the model.

EXISTING SYSTEM

Theexisting system there are many methods to check and perform system's malware detection process.

We have used many processes like adding a drebin dataset to the program and tries to check it with several techiques like N-Gram and FCBF selection algorithms , KNN algorithms done its execution process and now the real process has been started to check with some of the testing algorithm techniques .

KNN algorithm we got 76% accuracy.

For executing purpose we use testing algorithms.

Testing Accuracy:

Random Forest	82%
XGBoost	76%
LGBM	79%

PROPOSE SYSTEM

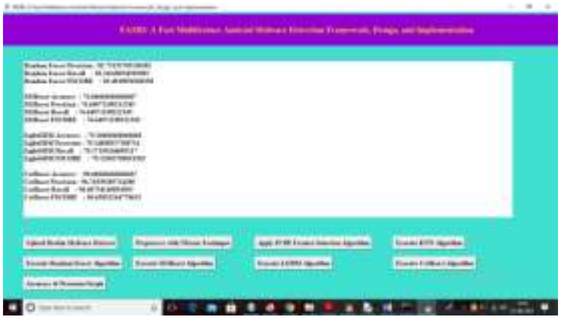
In our proposed system we have used an algorithm called CatBoost Algorithm. This algorithm has gained popularity because of its superior performance over the aforementioned malware detection techniques and is best known for its speed and accuracy. It is important because of the following reasons:

Speed:

This algorithm improves the speed of detection because it can predict all the features of a software in real time.

High accuracy:

CatBoost is a predictive technique that provides accurate(96%) results with minimal background errors.


FAMD: A Fast Multifeature Android Malware Detection Framework, Design, and Implementation
Upload Drebin Malware Dataset
Preprocess with NGram Technique
Apply FCBF Feature Selection Algorithm
Execute KNN Algorithm
Execute Random Forest Algorithm
Execute XGBoost Algorithm
Execute LGBM Algorithm
Execute CatBoost Algorithm
Accuracy & Precision Graph

IMPLEMENTATION

MODULES DESCRIPTION:

1) Upload Drebin Malware Dataset: using this module we will upload 'Drebin' dataset to application and then find out total malware labels available in dataset
2) Preprocess with NGram Technique: using this module we will read entire dataset and then convert each DALVIK OPCODE and permissions into NGRAM sequences and then apply TFIDFVECTORIZER to convert entire NGRAM sequences into vector. TFIDFVECTOR will replace each permission with its average frequency and then generate a vector. This vector will contains more number of features and to reduce this features we will use FCBF algorithm. Generated vector will be splitted into train and test part.
3) Apply FCBF Feature Selection Algorithm: using this module we will find important features from dataset by applying FCBF algorithm and then remove redundant and irrelevant features. This important features help machine learning algorithm for better prediction.
4) Execute KNN Algorithm: using this module we will train KNN algorithm with above dataset and then calculate prediction accuracy.
5) Execute Random Forest Algorithm: using this module we will train Random Forest algorithm with above dataset and then calculate prediction accuracy.
6) Execute XGBOOST Algorithm: using this module we will train XGBOOST algorithm with above dataset and then calculate prediction accuracy.
7) Execute LIGHTGBM Algorithm: using this module we will train LIGHTGBM algorithm with above dataset and then calculate prediction accuracy.
8) Execute CatBoost Algorithm: using this module we will train CatBoost algorithm with above dataset and then calculate prediction accuracy.

Accuracy & Precision Graph: using this module we will plot comparison graph between all algorithms

CONCLUSIONS

The number of applications that can be classified as malware continues to increase, new types of malware and camouflage techniques are constantly updating, effectively detecting malware in a relatively short time is of considerable significance to the third-party application markets and users. How to improve the detection accuracy and reduce the detection time are still the problems to be solved. We present a fast Android malware detection framework, FAMD, which combines permission features and Dalvik opcode features from different operation levels to construct feature vectors. To reduce the feature dimensionality and time complexity of the method, the FCBF algorithm is employed for feature selection. As a classifier proposed in recent years, CatBoost is employed in this work to conduct malware detection and family classification. In the experiments, we segment the opcodes with 4-Gram and vectorize the features combined with permissions. With the CatBoost as the classifier, the result achieves an accuracy of 97.40% in malware detection, and 97.38% in family classification. Compared with other state-of-the-art works, FAMD performs better comprehensively in accuracy and time consumption. It can be seen in the experiments that there is a clear difference in the distribution of certain key features in malicious applications and benign applications.

Since CatBoost is a supervised learning framework, this work is inadequate in detecting new emerging malicious applications, which we aim to improve in further work.

REFERENCES

[1] (2020). Smartphone Market Share. [Online]. Available: https://www.idc.com/promo/smartphone-market-share/os

[2] (2020). Mobile Malware Evolution 2019. [Online]. Available: https://securelist.com/mobile-malware-evolution-2019/96280

[3] W. Enck, M. Ongtang, and P. McDaniel, ‘‘On lightweight mobile phone application certification,’’ in Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS), 2009, pp. 235–245.

[4] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli, ‘‘MADAM: Effective and efficient behavior-based Android malware detection and prevention,’’ IEEE Trans. Dependable Secure Comput., vol. 15, no. 1, pp. 83–97, Jan. 2018.

[5] T. Kim, B. Kang, M. Rho, S. Sezer, and E. G. Im, ‘‘A multimodal deep learning method for Android malware detection using various features,’’ IEEE Trans. Inf. Forensics Security, vol. 14, no. 3, pp. 773–788, Mar. 2019.

[6] H. Zhang, S. Luo, Y. Zhang, and L. Pan, ‘‘An efficient Android malware detection system based on method-level behavioral semantic analysis,’’ IEEE Access, vol. 7, pp. 69246–69256, 2019.

[7] L. Onwuzurike, E. Mariconti, P. Andriotis, E. De Cristofaro, G. Ross, and G. Stringhini, ‘‘MaMaDroid: Detecting Android malware by building Markov chains of behavioral models (extended version),’’ ACM Trans. Privacy Secur., vol. 22, no. 2, pp. 1–34, 2019.

[8] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, ‘‘TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones,’’ ACM Trans. Comput. Syst., vol. 32, no. 2, pp. 1–29, Jun. 2014.

[9] J. Chen, C. Wang, Z. Zhao, K. Chen, R. Du, and G.-J. Ahn, ‘‘Uncovering the face of Android ransomware: Characterization and real-time detection,’’ IEEE Trans. Inf. Forensics Security, vol. 13, no. 5, pp. 1286–1300, May 2018.

[10] H. Cai, N. Meng, B. Ryder, and D. Yao, ‘‘DroidCat: Effective Android malware detection and categorization via app-level profiling,’’ IEEE Trans. Inf. Forensics Security, vol. 14, no. 6, pp. 1455–1470, Jun. 2019.

[11] S. Y. Yerima, M. K. Alzaylaee, and S. Sezer, ‘‘Machine learningbased dynamic analysis of Android apps with improved code coverage,’’ EURASIP J. Inf. Secur., vol. 2019, no. 1, p. 4, Dec. 2019.

[12] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, ‘‘Droid-sec: Deep learning in Android malware detection,’’ in Proc. ACM Conf. SIGCOMM, 2014, pp. 371–372.

[13] K. Tam, S. J. Khan, A. Fattori, and L. Cavallaro, ‘‘CopperDroid: Automatic reconstruction of Android malware behaviors,’’ in Proc. Netw. Distrib. Syst. Secur. Symp., 2015, pp. 1–15.

[14] (2020). Android Permission. [Online]. Available: https://developer. android.com/reference/android/Manifest.permission

[15] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, J. Nieves, P. G. Bringas, and G. Á. Marañón, ‘‘Mama: Manifest analysis for malware detection in Android,’’ Cybern. Syst., vol. 44, nos. 6–7, pp. 469–488, Oct. 2013.

[16] W. Wang, X. Wang, D. Feng, J. Liu, Z. Han, and X. Zhang, ‘‘Exploring permission-induced risk in Android applications for malicious application detection,’’ IEEE Trans. Inf. Forensics Security, vol. 9, no. 11, pp. 1869–1882, Nov. 2014.

[17] K. A. Talha, D. I. Alper, and C. Aydin, ‘‘APK auditor: Permission-based Android malware detection system,’’ Digit. Invest., vol. 13, pp. 1–14, Jun. 2015.

[18] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an, and H. Ye, ‘‘Significant permission identification for machine-learning-based Android malware detection,’’ IEEE Trans. Ind. Informat., vol. 14, no. 7, pp. 3216–3225, Jul. 2018.

[19] Q. Jerome, K. Allix, R. State, and T. Engel, ‘‘Using opcode-sequences to detect malicious Android applications,’’ in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2014, pp. 914–919.

[20] N. McLaughlin, J. M. D. Rincon, B. Kang, S. Yerima, P. Miller, S. Sezer, Y. Safaei, E. Trickel, Z. Zhao, A. Doupé, and G. J. Ahn, ‘‘Deep Android malware detection,’’ in Proc. 7th ACM Conf. Data Appl. Secur. Privacy, 2017, pp. 301–308.