iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** III **Month of publication:** March 2019

DOI: http://doi.org/10.22214/ijraset.2019.3346

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue III, Mar 2019- Available at www.ijraset.com

Survey on Android Malware Detection Using Multilevel Classifier Fusion

Mahesh R. Gawale[1], Prashant M. Yawalkar[2]
[1]*Student,* [2]*Professor, M.E Computer Engineering, MET's institute of engineering, Nashik, India*

Abstract: ***Android Malware detection is a very important factor in security of the smartphones and computer systems. However, now days used signature-based technique cannot be provide accurate detection in polymorphic viruses and zero-day attacks. So there is a need to malwares detection using machine learning. The purpose of this work is the improved malware and benign detection accuracy based on machine learning algorithms. The Multilevel classification model for malwares detection by using training a model of the base classifiers machine learning algorithms at the lower level and A set of the ranking algorithms at higher level to enable fusion using pairwise combination of ranking algorithms and detect the better average accuracy of fusion.***

Keywords: Android Malware detection, Malware classification, Classifier fusion, Machine learning, Mobile security.

I. INTRODUCTION

The Internet is becoming an important part of everyone's life as the online transaction and online net banking is being popular now a day. The users of the Internet, including corporate faces security threats caused by malware. Malware is a program that affects a computer, laptops and mobile without the user's permission and with an intention to cause damages to the system or steal private information from the system. Trojans, spyware and viruses are examples of malware that have minima security defenses. These malware infected apps can then compromise security and privacy by allowing unauthorized access to privacy sensitive information.
Android Malware - The malicious software is a specifically build for attack on mobile devices. The types of malware are exploited on OS (Operating System), mobile phone and software where mobile devices are increasingly common.

A. Types of Malwares

1) *Ransom Ware***:** The ransom ware is a type of malicious which abstracts the information about infected mobile devices and asks for pay money and retrieves that files.
2) *Spyware:* The spyware is a type of malicious that continues to monitor the sensitive information and collect the user information like SMS, MMS and contact numbers.
3) *Mobile Bot Nets:* The Mobile Bot nets is type of malicious that is a network of affected mobile devices that are controlled and administered by remotely using bot-masters. D - DOS attacks, Carry out Spam, delivery of the host devices.
4) *Dynamically Downloaded Code:* It is a type of malware that is an installed application in mobile devices to download a malware source code and deploys on mobile devices.
5) *SMS Sending:* It is a type of malicious, that is used to send SMS to VIP mobile numbers so the attacker/victim theft or fraud a money.

Android malware detection is based on the static analysis and dynamic analysis. Static analysis methods are examining the contents of an APK (android application package), such as the source code file, the Dalvik byte code and AndroidManifest.xml file. Dynamic analysis is monitoring the behaviour of an application by running and its controlled environment. In this context, machine learning methods learn general rules and patterns of malware and benign samples, allowing data-driven predictions of decisions, For example, classification. Meta-learning is one approach to improve the classification through the combination of various ensemble learning algorithms.
Malware detection using classifier fusion is based on a multilevel architecture that uses machine learning algorithms. It's designed to induce a classification model for Android malware detection by training a number of base classifiers machine learning algorithms at the lower level. A set of ranking algorithms at a higher level to enable fusion using pairwise combination of ranking algorithms and find best fusion average accuracy. The classifier fusion approach is evaluated on four data sets (Malgenome 215, Drebin 215, McAfee 100 and McAfee 350). It's applied on machine learning's Algorithm and ranking based algorithms that enable classifier fusion for final improved classification of Android malware detection.

II. LITERATURE REVIEW

Android malware detection can be categorised into classification, detection, analysis & eventual containment of malware. A signature based technique is used by commercial antivirus in which the database is regularly updated so that the latest virus data detection mechanisms can be possessed. However, it is not possible to detect malware in zero-day by using antivirus and signature-based scanner. The statistical analysis of the content in binary files to use the malware detection features [1].

DroidFusion is designed to induce a classification model for malware detection by training a number of base classifiers at the lower level. A set of ranking algorithms at a higher level to enable fusion using pairwise combination of ranking algorithms and find best fusion average accuracy. Using this information, the authors evaluated algorithms available in the WEKA framework [2].

Yerima proposed malware detection is difficult to find in mobiles and network. This is because of the malware techniques that are malicious payload, usually within apps that provide functionality or malicious action; sending spam or encrypting data and deleting data, etc., Polymorphic techniques and encrypting the malicious payload and signature-based scanner. The comparison of fusion classifier methods are majority vote, simple logistic classifiers and average of probability based on J48 [3].

Coronado-De-Alba proposed an analyze machine learning algorithms for classification over a dataset with binary attributes and nominal class (malware and benign) values using WEKA. A classification with an accuracy of 97.5620 using a feature selection algorithm over 660 features, for an unbalanced dataset of 1531 malware and 765 benign samples which exposes a reliable method. A classifier fusion method based on random forest and random committee ensemble classifiers [4].

Dehghantanha proposed classifier fusion approach with static analysis based on Android permissions and source code based analysis. They used SVM, random forests, decision trees, random tree, JRip, and linear regression classifiers. An experimented with machine learning algorithms that contained odd combinations of the three and five classifiers using the majority voting fusion method. The best fusion model achieved a better accuracy rate of 95.6 using the source code features [5].

Wang discussed approaches are the permissions and sources code analysis based on words of bag model representation. The permission based model is computationally inexpensive and is implemented as the OWASP Seraphim droid Android app that can be obtained from Google Play Store. An extracted 11th types of the static features and employed multiple classifiers in a majority vote classifier fusion approach. The classifiers include SVM, naive Bayes, classification & regression tree (CART), and random forest [6].

Rahulamathavan discussed android malware detection is combinations of intents and permissions which are classifiers benign and malware. They evaluated the average efficacy by using machine learning algorithms. A utilized intents and permissions as features to train machine learning models and applied classifier fusion for improved performance [7].

Shijo discussed malware classification is using dynamic or static analysis technique. The proposed are utilizes, benefits of both methods, dynamic and static analysis. Static analysis features are an extracted from the android app code files. The malware executable is collected from the VirusShare community website. Printable strings information (PSI) is extracted from the android app code files and which is used as a static analysis method and Dynamic analysis technique is done by using the cuckoo sandbox tool. The dynamic analysis technique is focused on API and system call. Combining the features of source code extracted and the app behavior of the files in execution time and better classification result [8].

III.PROPOSED METHODOLOGY

In this section, we describe the malware classification are using static analysis or dynamic analysis methods, Architecture of the system and experimental algorithms

A. Android Malware Analysis Methods

1) Static Analysis: Static analysis is observing malicious behavior by analyzing the source code. Malware detection based on static analysis reduces time and resources and without running the android app on a mobile device to extract the app features. The disadvantage of static analysis is source code obfuscations because of the detecting the malware in the android app. Static analysis can detect logical inconsistencies and runtime errors. The commonly used static app features are the Permission and API calls [8].

There are two methods

a) Signature Based Analysis and

b) Permission Based Analysis.

2) Dynamic Analysis: The dynamic analysis technique is the any software or application that is appeared after actually executing the programs. Dynamic analysis means that the application is executed and monitored, mostly in an environment also called a

sandbox. Behaviour based analysis monitors the actions performed while the applications executed. One way of monitoring the actions is to hook the API of the OS. The hooking technique allows monitoring the behavior of Android app by intercepting API calls [8].

There are methods

a) Taint Droid: It is focuses on apps that are shared to get information. TaintDroid are tracking the information of data labeling in memory and detecting privacy leaking in android app.

b) Anomaly: It is noticed different acts from the smart phones and then apply Machine Learning anomaly to classify and collected data as malicious or benign.

c) Crow Droid: Dynamic analysis of the app behaviour, malicious apps in Android OS. The developer is the collection of data set for the users of crowd and detection their behaviours by system logged calls.

B. Architecture

In fig. 1 system flow first we provide input as apk to extract features and code to detect malware using the Static Analyzer. At Lower level to classifier malware and benign using machine learning algorithms. And higher level, we applied ranking algorithms to detect accuracy and use classifier fusion approach using different dataset as, malgenome 215, Drebin 215, McAfee 100 and McAfee 350 then got a final output an Average accuracy.

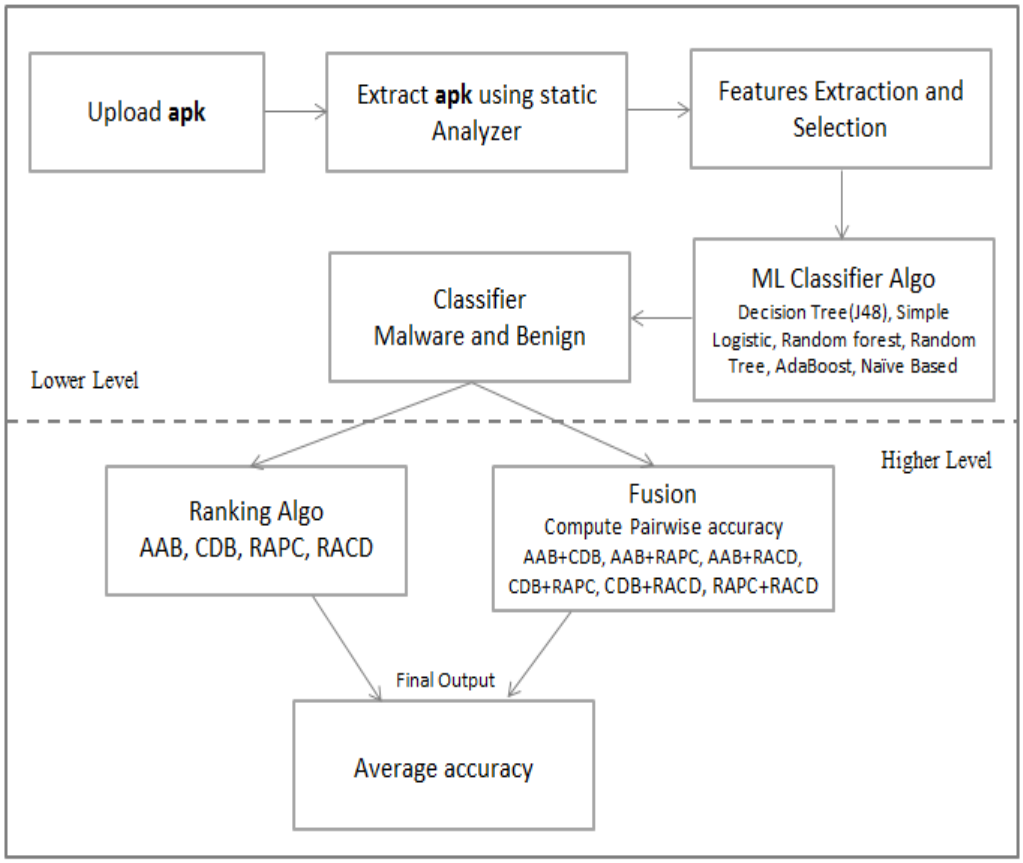


Fig.1: Proposed System Flow

1) App Feature Extraction for Machine Learning: The features used an automated static analysis tool developed with Python. The tool is an extract the intents and permissions from the android app manifest file after decompiling using AXMLprinter2. In API calls is extraction using reverse engineering of APK and the get .dex files then convert into human readable files using Baksmali disassembler. There are three features categories of 1) Permissions 2) API Calls and 3) Operating System commands. The API calls features are obtained from Classes.dex files. They are consisted of feature which enable of the API calls and selected Java API calls used to android app's files. The permission features are the extraction and detection by using android app manifest file in declaring permission [8].

Table 1.Example of features extraction

Type	Features (keywords)
API calls Related	getDeviceId; abortBroadcast; getSubscriberId; getSimSerialNumber;getLineNumber;getCallState; getNetworkOperator; getSimCountryIso; Runtime.exec(); getSimOperator; android.provider.Contacts; getPackageManager; HttpPost_init; android.provider.ContactsContract; HttpUriRequest; HttpGet_init; SMSReceiver; onActivityResult; FindClass; bindService ; DexClassLoader; ;SecretKey; createSubprocess sendMultipartTextMessage; reflectgetClass; System.loadLibrary; getMethod; intent.action.BOOT_COMPLETED; registerReceiver; intent.action.RUN
Command Related	mount; remount; chmod; chown; /system/bin/sh; /system/app; /res; /system/bin; .jar; .apk; GET_META_DATA; GET_SERVICES; GET_RECEIVERS; GET_PERMISSIONS; GET_SIGNATURES;
Permissions	INTERNET; ACCESS_COARSE_LOCATION; ACCESS_FINE_LOCATION; WRITE_SMS; WRITE_CALL_LOG; SEND_SMS; WRITE_APN_SETTINGS; CHANGE_NETWORK_STATE; BROADCAST_SMS; CHANGE_WIFI_STATE; WRITE_EXTERNAL_STORAGE; RECEIVE_MMS; CAMERA; RECEIVE_SMS; RECORD_AUDIO; CLEAR_APP_CACHE; INSTALL_PACKAGES; CALL_PHONE;

2) *Feature Selection:* Feature selection and ranking is based on the reduction technique which is the lowers model computational cost. The system used for evaluation of the datasets is derived from feature reduction based on the information gain (IG) feature ranking approach to the rank features and selecting the top n features. Given a feature A, IG is expressed as

$$IG = E(A) - E(A|B) \quad (1)$$

where E (X) and E (X / Y) represent entropy of feature X observing the feature Y, respectively. The entropy of feature is

$$E(A) = \sum_{x \in X} p(x) \log 2 (p(x)) \quad (2)$$

where p(x) is the probability function of random variable A. Similarly is entropy of A relative to B.

$$E(B|A) = \sum_{x \in X} p(x) \sum_{x \in X} p(x|y) \log 2 (p(x|y)) \quad (3)$$

where p(x |y) is the probability of conditional is x given y. The higher the reduction of the entropy of feature A, the greater the significance of the feature.

C. Algorithms

The Classifier malware and benign detection based on machine learning algorithms. Those are described in paper [5] and [9].

1) *Decision Tree (J48):* A decision tree used classification and regression problems. The decision tree is the decides target of new, sample value and it's based on the number of attribute values of available datas.
2) *Random Forest:* A machine learning methods that are generates many individual learners and aggregates the results. The best parameters at every node in decision tree and it are randomly selected one or more features to generates.
3) *Naive Bayes:* It is classifier is the collections of the classification algorithms are based on Bayes Rules. Naive Bayes is not an individual algorithm, but a set of algorithms where they are the share a common attributes, i.e., each pair of the features is classified and important to each other.
4) *Simple Logistic:* This is a machine learning algorithm. To evaluate the base learners this approach utilizes logistic regression methods based on simple regression functions. The linear regression is trying to find a function that will fit of the training data well by computing the large weights of the log- likelihood of the logistic regression function. In this algorithm, the training phase is relatively longer than the testing phase.
5) *Ada Boost:* AdaBoost is that Adaptive Boosting and is also known as AdaBoost.M1. AdaBoost can use for the boost performance of the algorithms. AdaBoost achieve average accuracy on classification problems.

Ranking based algorithms are used to a higher level to classification fusion they are described in paper [2].

a) Average Accuracy-Based Ranking Scheme (AAB): The AAB ranking scheme is the directly proportional of the average accuracies in android app classes. In this case, base classifiers with larger overall accuracy, performance will rank higher.

b) Class Differential-Based Ranking Scheme (CDB): The CDB ranking scheme is the directly proportional of the predictives accuracy and inversally proportional of the absolute value to the performance difference between the classes.

c) Ranked Aggregate of Per Class Accuracies Based Scheme (RAPC): The RAPC ranking scheme is direct proportional to the total number ranking class of the average accuracies classifiers. This method assigns a large weight to a base class classifier.

d) Ranked Aggregate of Average Accuracy and Class Differential Scheme (RACD): The ranking scheme is directly proportional to the total number of rankings the average performance accuracies and the rankings of the difference in performance between the classes. This method is assigned a large weight to base classifiers with good accuracy.

IV. CONCLUSIONS

We have discussed a study based on multilevel classifier fusion approach for Android malware detection. The Classifier fusion is based on machine learning algorithms that enable to lower level to classifier malware or benign and set of ranking algorithms at higher level to enable fusion based on pairwise combination of ranking algorithms. The performance of system will be measured in terms of accuracy with existing technique.

REFERENCES

[1] Ralescu A and Baskaran B,"A study of android malware detection techniques andmachine learning", Modern Artificial Intelligence and Cognitive Science; pp. 15-23, 2016 -April.

[2] S. Sezer and S. Y. Yerima, "DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection", IEEE Transaction on Cybernetics, pp.2168-2267, 2018 -Jan.

[3] S. Y. Yerima, I. Muttik, andS. Sezer, "Android malware detection using parallel machine learning classifiers, inProc. 8th Int. Conf. Next Gener. Mobile Apps Services Technol. (NGMAST) , Oxford, U.K., pp. 37-42, 2014 -Sept.

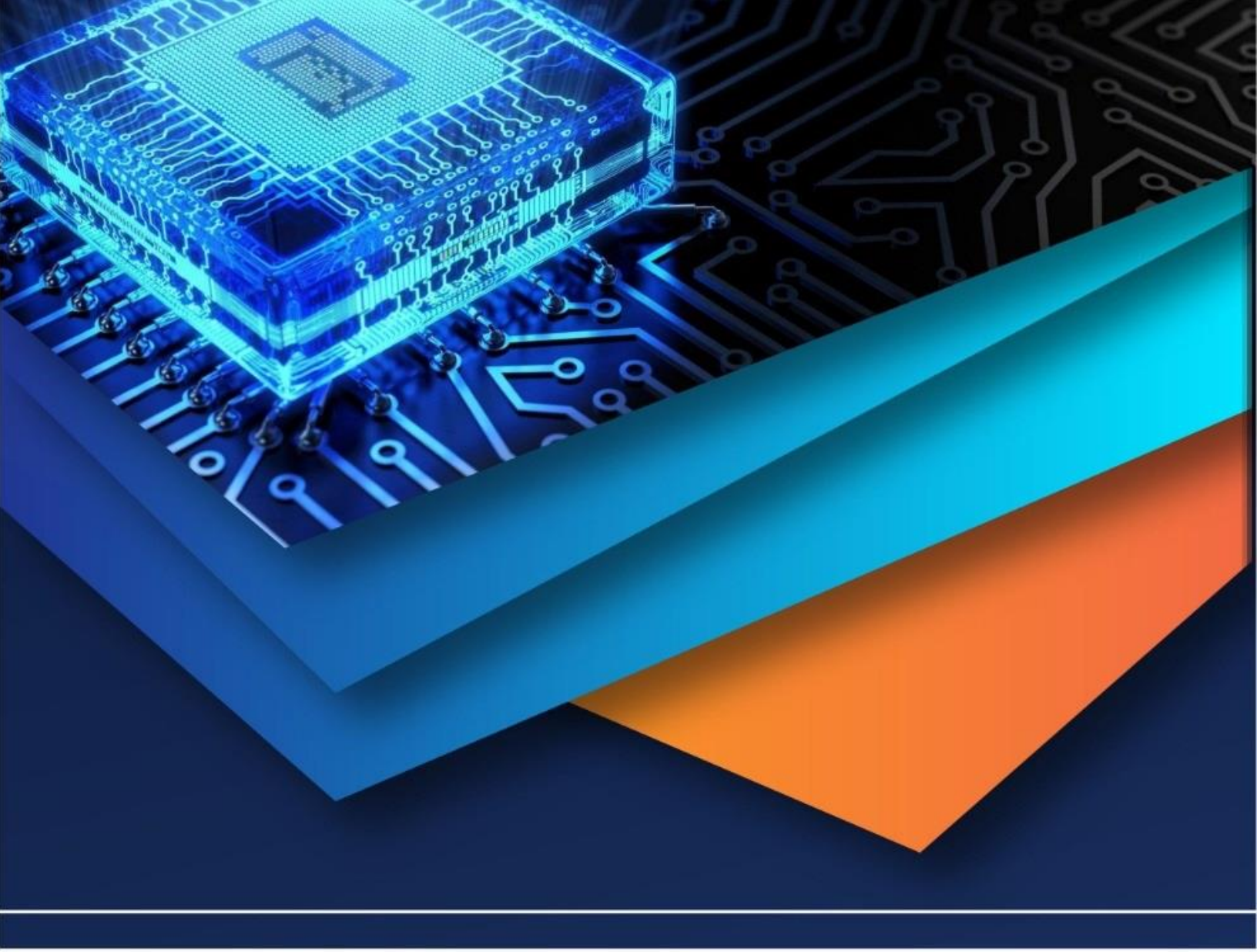
[4] A. Rodriguez-Mota, L. D. Coronado-De-Alba,and P. J. Escamilla-Ambrosio, "FeatureSelection and ensemble of classifiers for Android malware detection", inProc. 8th IEEE Conf. Commun. (LATINCOM),pp. 1-6, 2016 Nov.

[5] A. Dehghantanha,N. Milosevic and K.-K. R. Choo, "Machine learning aided Androidmalware classification", Comp. Elect. Engg., vol. 61, pp. 266-274, 2017 -Jul.

[6] Y. Li, X. Wang,X. Zhang,J. Liu, and W. Wang, "Detecting Android malicious appsand categorizing benign apps with ensemble of classifiers", Future Gener. Comp. Syst. , vol. 78, pp. 987-994, 2017-Jan.

[7] M. Rajarajan, Y. Rahulamathavan , T. M. Chen, and F. Idrees, "PIn-droid: A novel Android malware detection system using ensemble learning methods",Comput. Security, vol. 68, pp. 36-46, Jul. 2017.

[8] A. Salim andP. V. Shijo, "Integrated static and dynamic analysis for malware detection", Procedia Comp. Sci. 46, pp. 804–811, 2014.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY