

A Synopsis on

# **Android Malware Detection By Using Machine Learning**

Submitted in partial fulfillment of the requirements  
of the degree of

**Bachelor of Engineering**

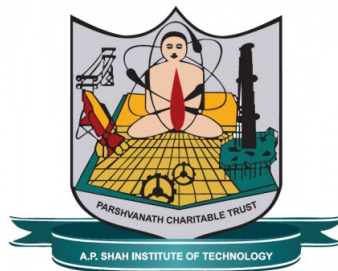
in

**Information Technology**

by

**Sonam Chavan (16104067)**

**Prof.Ganesh Gourshete  
Prof.Nahid Kausar Shaikh**



**Department of Branch Name**

A.P. Shah Institute of Technology

G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615

UNIVERSITY OF MUMBAI

2019-2020

## CERTIFICATE

This is to certify that the project Synopsis entitled “***Android Malware Detection By Using Machine Learning***” Submitted by “***Sonam Chavan (16104067)***” for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering in Information Technology***.to the University of Mumbai,is a bonafide work carried out during academic year 2019-2020

(Prof.Nahid Kausar Shaikh)  
Co-Guide

(Prof.Ganesh Gourshete)  
Guide

Prof. Kiran Deshpande  
Head Department of Information Technology

Dr. Uttam D.Kolekar  
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane

Date:

## Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---

(Sonam Chavan 16104067)

Date:

## **Abstract**

Malware is one of the major issues regarding the operating system or in the software world. The android system is also going through the same problems. We can see earlier, Signature-based detection techniques were used to detect unknown malware. But these techniques were not able to detect unknown malware. Despite a number of detection and analysis techniques are in place, the high detection accuracy of new malware is still a critical issue. In this paper, we study and highlight the existing detection and existing analysis methods used for the android malicious code. Along with studying, we propose a Machine learning algorithm that will be used to analyze such malware. We will be having a data set that will have both types of application malicious and benign which will be installed on an android device to analyze the behavior patterns. We will generate a system feature vector from each application by executing it using the algorithms. In the end, the user will be able to see how much malicious content is there in the application or the file which is analyzed.

## Introduction

Malware is nothing but the short name for malicious software, in general referred to many forms of hostile or intrusion creating software, spyware, Trojan horses, backdoors, and rootkits. Main aim of malware is to damage, steal, disrupt or do some bad actions. Due to popularity of android system has led to more spreading of android malware. This malware are spreading in market by the third parties developing application. The Google android market also doesn't promise to guarantee that all its listed applications are threat free. There are also such reports about download Trojans applications that download their malicious code after installation such applications cannot be easily detected by Google's technologies during publication in Google android market. The android threats include banking Trojans , spyware ,bots ,root exploits, SMS fraud, phishing, premium dialer fake installer. Penetration techniques commonly used for malware applications for installation activation running on the android system are repackaging, updating and downloading.

1.Repackaging:- It is among the common techniques for malware developers to install malicious applications on a android platform. Repackaging approach for popular applications and misuse them as a malware. The developer downloads such types of application and recode them and add their own malicious code and upload that application to the official android app store or on the different markets.

2.Updating:- This technique is much more difficult for detecting the malware. The malware developer may still use repackaging but instead of encoding the inflict code to the application, the developer may include a update component that will able to download malicious code at the runtime.

3.Downloading:- This is the most traditional attacking technique. The malware developer need to attract the user to download the interesting and attractive applications.

## Objectives

The objective behind developing this project is:

1. To have an malware detection interface.
2. To provide user convenience.
3. To make service available whenever they required.
4. Instead of relying what the permissions the user have granted our project will be able to list all the permissions over it.
5. Also the permissions will be classify later and according to it the malware will be detected from the datasets.

## Literature Review

1)Darus, Fauzi Mohd, Salleh Noor Azurati Ahmad, and Aswami Fadillah Mohd Ariffin. "Android Malware Detection Using Machine Learning on Image Patterns." 2018 Cyber Resilience Conference (CRC). IEEE, 2018.

-In this paper we have seen that they had 300 malware and 300 benign APK files, and they managed to generate only 183 malware and 300 benign gray-scale images. The other 117 malware samples could not be generated into images because the APK files are either corrupted or they did not contain classes.dex file. And their accuracy was less in all the algorithms they have used.

2)Varma, P. Ravi Kiran, Kotari Prudvi Raj, and KV Subba Raju. "Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms." 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE, 2017.

-In this paper different machine learning algorithms are used such as navies Bayes, j48, Random forest, Multi class classifier and multilayer perceptron to detect android malware and evaluate the performance of each algorithm. Here they implemented a framework for classifying android applications with the help of machine learning techniques to check whether it is malware or normal application. To access this model have to extract several features and permission from many downloaded applications from the android market. For validating, their system they collected 3258 samples of android apps and those have to be extracted for each and every application, extract their features and have to train the models going to be evaluated with the help of classification accuracy and time taken for the model.

3)Chang, Wei-Ling, Hung-Min Sun, and Wei Wu. "An Android Behavior-Based Malware Detection Method using Machine Learning." 2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). IEEE, 2016.

-In this they have proposed a Robotium program in an Android sandbox which can trigger Android application automatically and monitor behavior. A Robotium program is a UI-identification automatic trigger program which can click mobile applications in meaningful order. More importantly, they do large scale experiments. They build a decision model using behavior we collected and RandomForest algorithm. It can determine whether unknown application is a malware and show its condence value. Their evaluation indicators show permission and dynamic behavior we collected such as network activtiy, le read/wirte, phone call, sent SMS, information leak are suitable feature data for classication malware. The accuracy of their model is 97 and the FPR is less than 4. They also store classication result and condence value of unknown APK in their database.

4)Koli, J. D. "RanDroid: Android malware detection using random machine learning classifiers." 2018 Technologies for Smart-City Energy Security and Power (ICSESP). IEEE, 2018  
-In this, Android malware detection system is proposed which uses permission, APIs, and presence of others key apps information such as, dynamic code, reection code, native code, cryptographic code, database etc. as features to train and build classification model by using various techniques which automatically distinguish malicious Androidapps (malware) from legitimate ones.



## **Problem Defination**

The Existing system was working on the different Apk files where they were not getting the actual and the perfect output of the malware files as required. The existing users has to go on the terminal and they were not getting the accuracy as the datasets were not been trained. In this system there is more accuracy even the datasets to be taken is of bulk so the user can get the actual output needed for the particular malware of the particular Apk file. Also we are developing an User Interface where the user can interact and also he can be able to get the permissions which the user has provided to the particular application.

## Proposed System

In this system we are detecting the maximum accuracy by using the machine learning algorithms on image features generated from the APK samples. The image will be generated from the maximum APK's samples consisting of Malware and benign samples, and the features are extracted using the descriptor. For validating our system will collect the maximum samples of android apps and those have to be extracted for each and every application. The scope of this project is to detect the malware in the Android Application using machine learning against the threat of computer viruses like Trojans, spyware, bots, root exploits, SMS fraud, phishing, premium dialer, fake installer. We are introducing the architecture of malware detection system describing all the functions in details to show how the scheme works for Malware Detection.

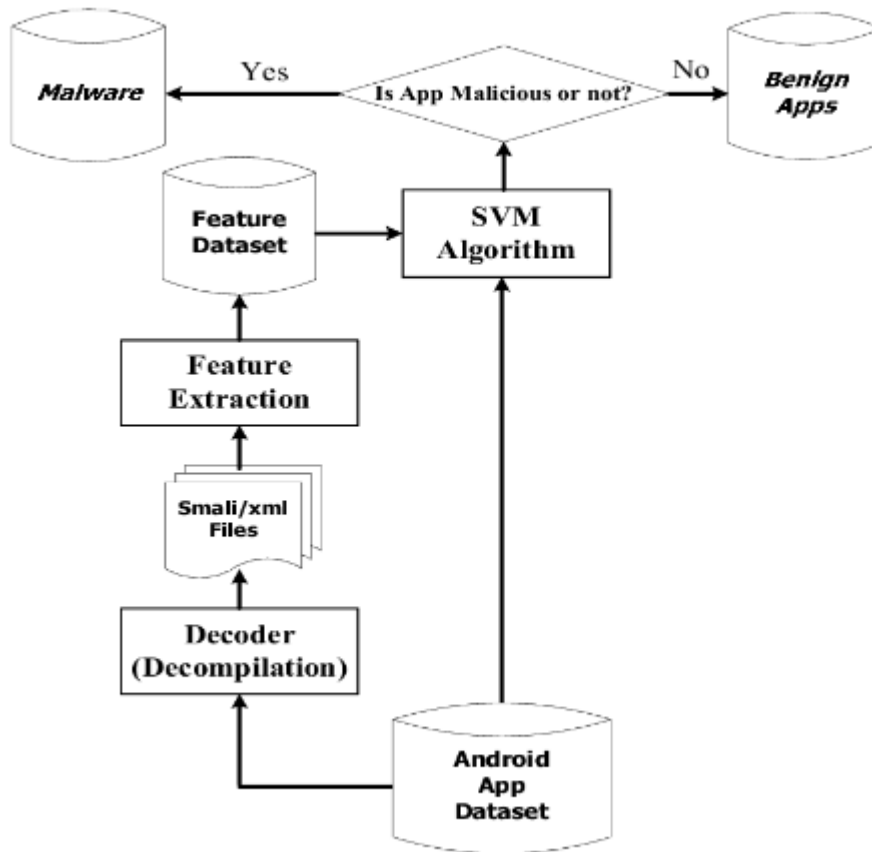


Figure 1:Proposed System

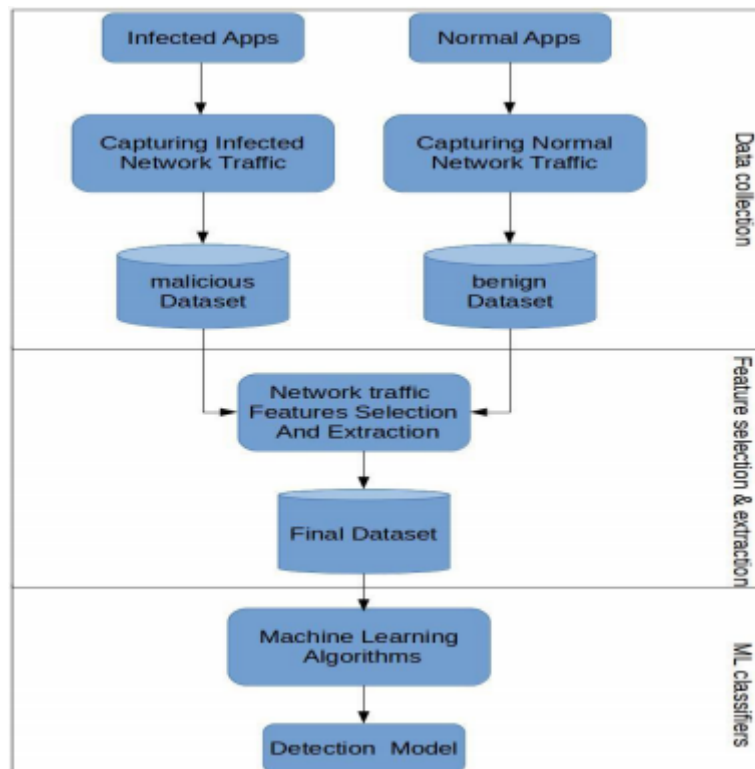


Figure 2:Workflow of Malware Detection

## Design

The Design of our system consists of the following:

- 1)The user will provide the APK files to the system.
- 2)The APK files will be then extracted and the permission granted to that particular Apk file will be listed down.
- 3)The Permissions for that APK file will be then matched with the Dataset which we have been created.
- 4)The Dataset will be a combination of both the Benign file as well as the mallicious files.
- 5)The dataset will be then match with the percent given to the permissions and according to it the supervised machine algorithm will work.
- 6)According to it the permissions which are not been useful for that particular APK file according to that percentage it will show us as that the particular file can affect on your system.
- 7)As it may result into an Malware file.

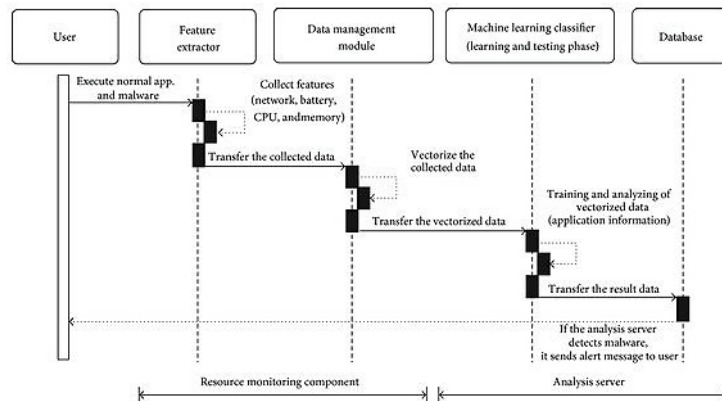


Figure 3:Sequence Diagram

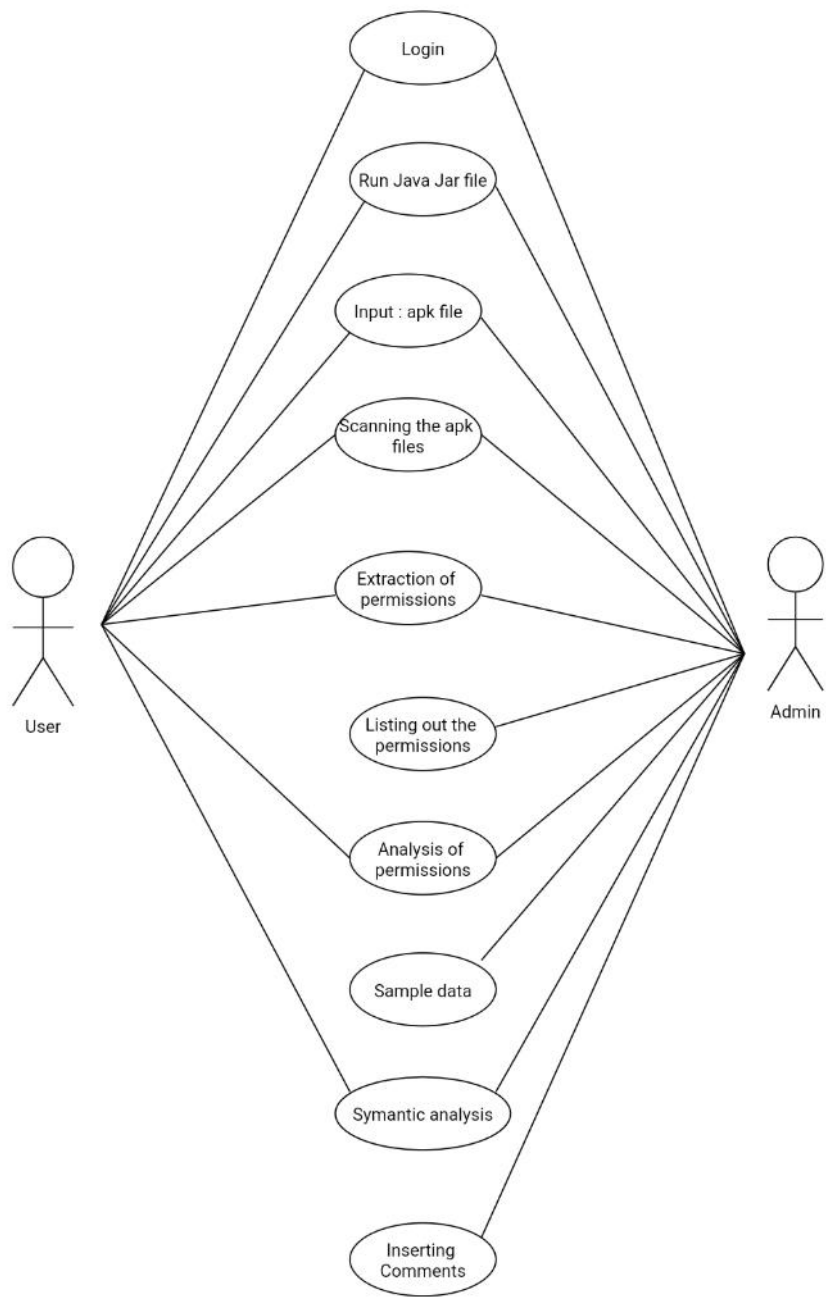


Figure 4:Use Case Diagram

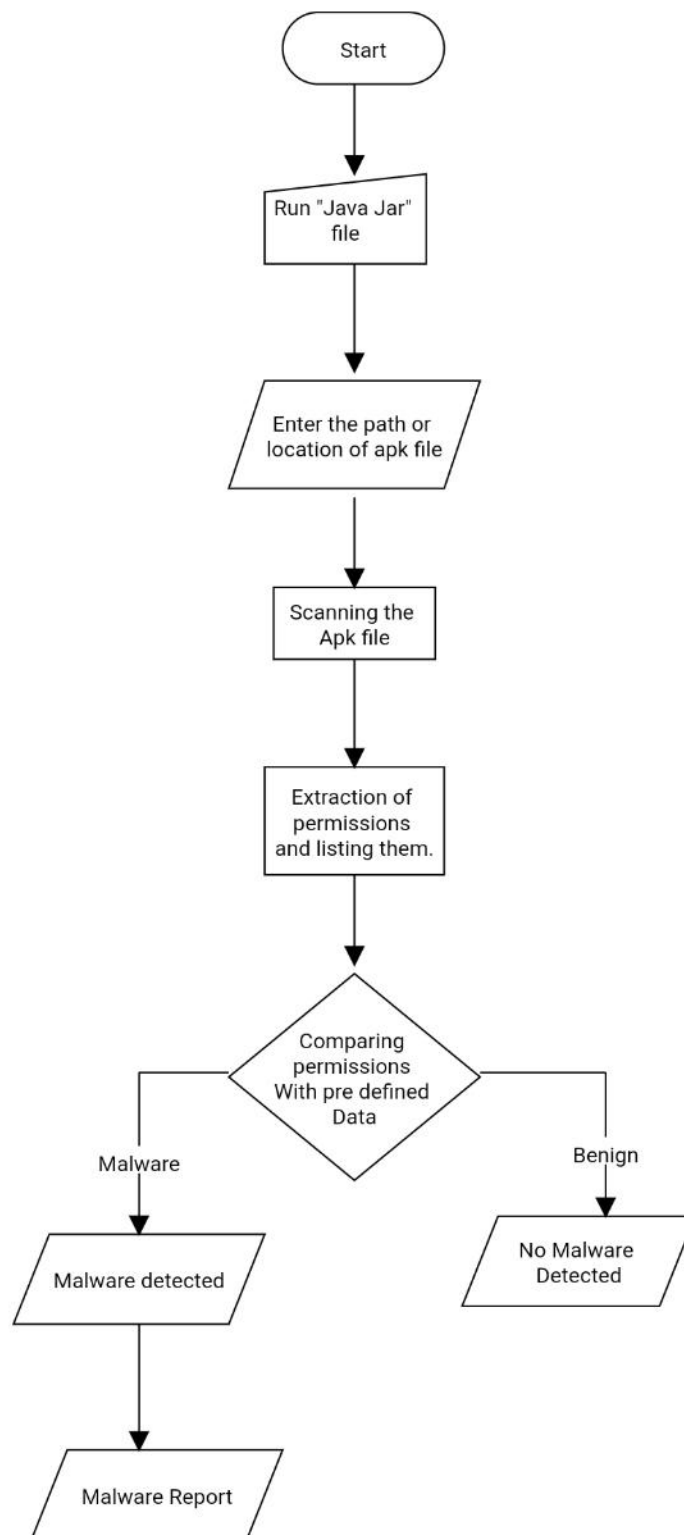


Figure 5:Permission Analysis

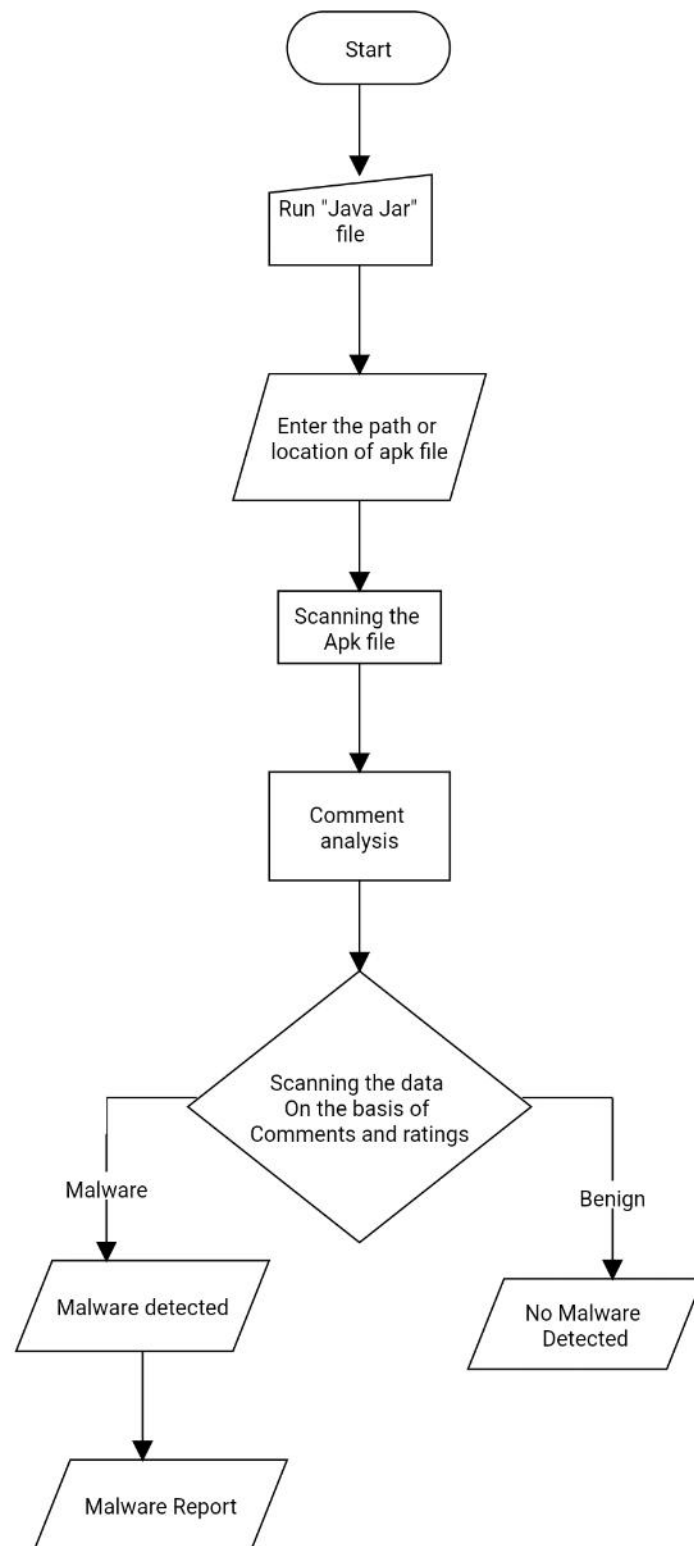


Figure 6: Semantic Analysis

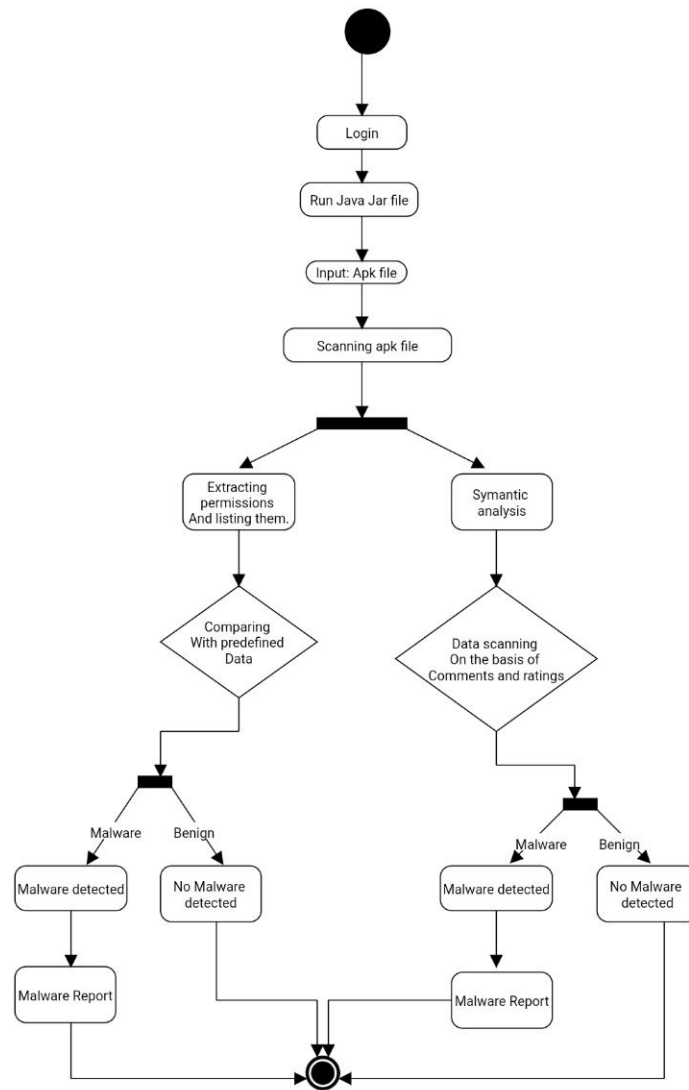
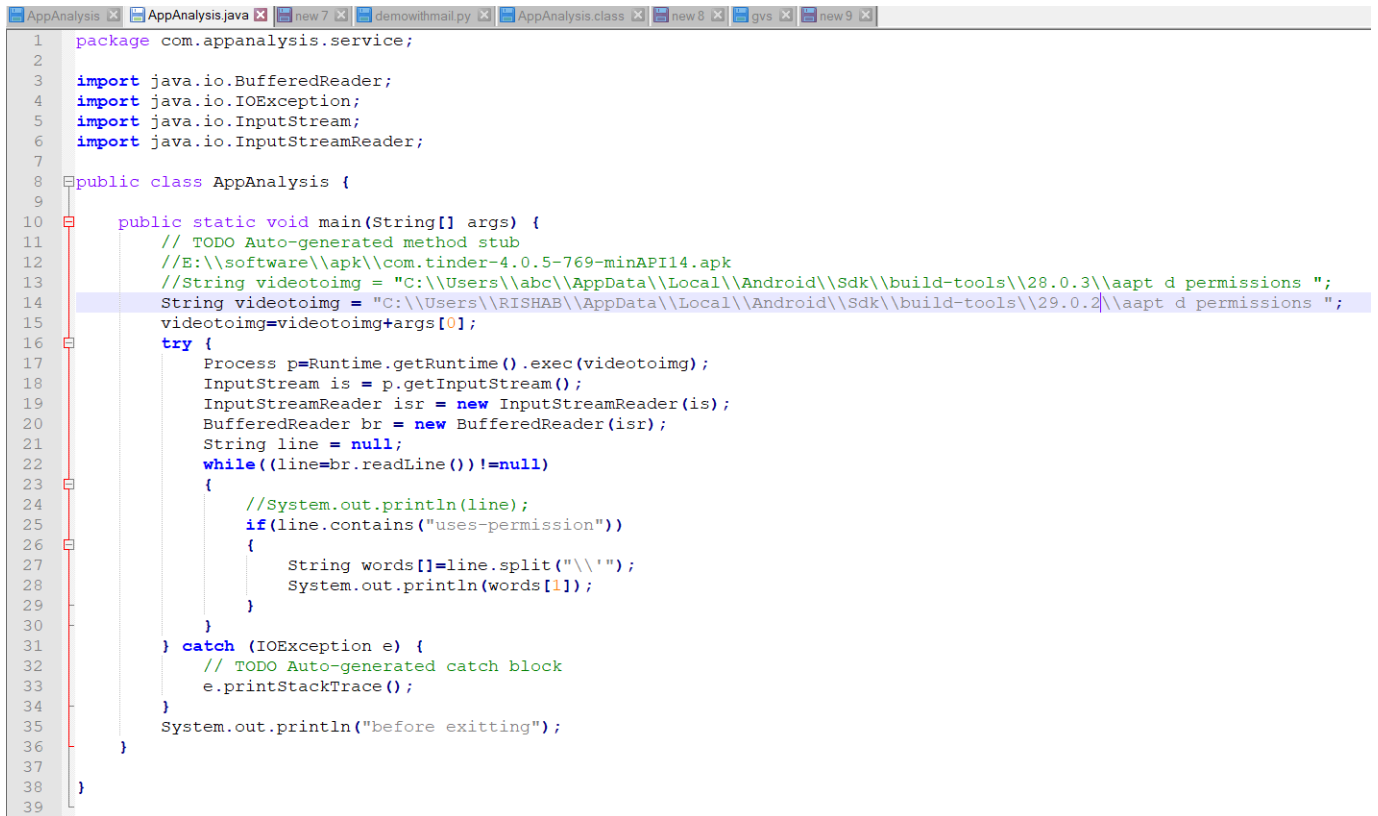


Figure 7:Activity Diagram



## Implementation



```
1 package com.appanalysis.service;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5 import java.io.InputStream;
6 import java.io.InputStreamReader;
7
8 public class AppAnalysis {
9
10     public static void main(String[] args) {
11         // TODO Auto-generated method stub
12         //E:\\software\\apk\\com.tinder-4.0.5-769-minAPI14.apk
13         //String videotointg = "C:\\Users\\abc\\AppData\\Local\\Android\\Sdk\\build-tools\\28.0.3\\aapt d permissions ";
14         String videotointg = "C:\\Users\\RISHAB\\AppData\\Local\\Android\\Sdk\\build-tools\\29.0.2\\aapt d permissions ";
15         videotointg=videotointg+args[0];
16         try {
17             Process p=Runtime.getRuntime().exec(videotointg);
18             InputStream is = p.getInputStream();
19             InputStreamReader isr = new InputStreamReader(is);
20             BufferedReader br = new BufferedReader(isr);
21             String line = null;
22             while((line=br.readLine())!=null)
23             {
24                 //System.out.println(line);
25                 if(line.contains("uses-permission"))
26                 {
27                     String words[]=line.split("\\'");
28                     System.out.println(words[1]);
29                 }
30             }
31         } catch (IOException e) {
32             // TODO Auto-generated catch block
33             e.printStackTrace();
34         }
35         System.out.println("before exiting");
36     }
37 }
38
39 }
```

Figure 8:Code

```
C:\Windows\System32\cmd.exe
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\RISHAB\Desktop\sw>java -jar apkextractionv1.jar C:\Users\RISHAB\Desktop\application\camscanner.apk
android.permission.FOREGROUND_SERVICE
android.permission.READ_EXTERNAL_STORAGE
android.permission.CAMERA
android.permission.WRITE_EXTERNAL_STORAGE
android.permission.ACCESS_NETWORK_STATE
android.permission.INTERNET
android.permission.ACCESS_WIFI_STATE
com.android.launcher.permission.INSTALL_SHORTCUT
com.intsig.camscanner.Account
android.permission.INTERNET
com.intsig.camscanner.permission.C2D_MESSAGE
com.google.android.c2dm.permission.RECEIVE
com.android.vending.BILLING
android.permission.REQUEST_INSTALL_PACKAGES
com.sony.mobile.permission.SYSTEM_UI_VISIBILITY_EXTENSION
android.permission.WAKE_LOCK
com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE
android.permission.MOUNT_UNMOUNT_FILESYSTEMS
before exiting

C:\Users\RISHAB\Desktop\sw>
```

Figure 9:Output

## References

- 1)Darus, Fauzi Mohd, Salleh Noor Azurati Ahmad, and Aswami Fadillah Mohd Ariffin. "Android Malware Detection Using Machine Learning on Image Patterns." 2018 Cyber Resilience Conference (CRC). IEEE, 2018.
- 2)Varma, P. Ravi Kiran, Kotari Prudvi Raj, and KV Subba Raju. "Android mobile security by detecting and classification of malware based on permissions using machine learning algorithms." 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE, 2017.
- 3)Chang, Wei-Ling, Hung-Min Sun, and Wei Wu. "An Android Behavior-Based Malware Detection Method using Machine Learning." 2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). IEEE, 2016
- 4)Koli, J. D. "RanDroid: Android malware detection using random machine learning classifiers." 2018 Technologies for Smart-City Energy Security and Power (ICSESP). IEEE, 2018.