e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:02/February-2022 Impact Factor- 6.752 www.irjmets.com

MOBILE BOTNET DETECTION: A MACHINE LEARNING APPROACH USING SVM

Akbar Shaikh*1, Ganesh Landage*2, Sanket Thorat*3, Prasad Shinde*4, Mamta Sharma*5

*1,2,3,4Student, Department Of Computer Engineering, P. K. Technical Campus, Savitribai Phule Pune University Kadachiwadi, Pune, Maharashtra India 410501.

*5Assistant Prof. Department Of Computer Engineering, P. K. Technical Campus, Savitribai Phule Pune University Kadachiwadi, Pune, Maharashtra India 410501.

ABSTRACT

We live in a digital era where a sea of data and information is produced and processed every day. Data and information so huge are evidently used in multiple ways and for different purposes such as Scientific and Medical research, news, blogs, statistical data that change every day. Hence the reading understanding and thereafter classifying these scores of data becomes increasingly difficult. Moreover, if done manually it is also susceptible to mistakes and human errors. This results in increased time consumption in reading and analyzing the information and consequentially superfluous information takes up the room supposed for essential information. It is a herculean task indeed for human beings to manually read, process, and reproduce the summary out of loads of information in a large document or text. This presents a problem for searching for the appropriate information from relevant documents. It thereby becomes necessary to develop a solution to gather and arrange concisely the important information in a swift and efficient manner. To fix the above-discussed complexities, a solution that summarizes the important text or information is the need. In this proposed mechanism we are implementing a text summarization model using a combined set of TFIDF and Textrank algorithm along with some natural language processing methods which shall provide more precise results as compared to preceding models of the same kind. This, in turn, will lead to convenience in various activities in all spheres of life in the computing and data processing world by providing users with relevant and usable information without any irrelevant information. This will also help in the swift and efficient identification of harmful botnets that can potentially steal your data and infect your system with dangerous viruses and malware. That is not all the time saved by this solution we develop will save human effort and a tremendous amount of money alike.

Keywords: Machine Learning, Support Vector Machine (SVM), Botnet Detection, Deep Learning.

I. INTRODUCTION

The arrangement of information into clear and concise bits has a crucial role in making decisions, solving problems, or even forecasting trends, and making predictions based on the available statistical data. It is no wonder that the correct pro- cessing and analyzing of data helps companies save a fortune, cut their costs, and drive their decisions and products just like it drives our day to day decisions as students businesses, and educators, it forms the basis of all the decisions made from statistical data. If not gathered, filtered, and arranged properly raw data can often be misleading and affect the decisions or further steps negatively. As a result, it became a necessity to study the problem to its core and to try to come up with innovative and sustainable solutions to solve this problem and make this work easier with an automated process. Owing to the increase in the number of mobile devices or smartphones and consequently to the skyrocketing number of android market share these days, there is widespread risk of android malware, hence we do not know what any application or file contains before we have installed it in our device. We through our system will be able to extract the important and characteristic features from the android APK files that can be used in the identification of the botnet and its details i.e the class and family it belongs to. We also shall be able to find a suitable machine learning algorithm to segregate the Android botnet family with a high recall. The final step is to develop a system called ABIS (Android Botnet Identification Furstem) which is a package containing the identification engine along with a web application and an android application for the users which will allow them to inspect and scan any application before it is installed. This will help users in two ways i.e. the users who need to segregate important information from raw data that will be done quickly, and the other being the security risk that earlier the users had to first install the application to not even actually inspect but just to guess whether the application or file installed on their system is safe or harmful, so both these objectives will be achieved in a fast and accurate manner.

Malicious botnet applications have become a big problem in recent years. Furthermore, their increasing use of advanced evasive strategies necessitates the development of more ef- ficient detection methods. As a result, in this research, we describe a deep learning strategy for Android botnet detection that uses Convolutional Neural Networks (CNN) [1]. To identify new or previously unknown applications as 'botnet' or 'normal,' the CNN model uses 342 static characteristics. The features are retrieved through automated reverse engineering of the applications and utilised to build feature vectors that are fed straight into the CNN model without any additional pre-processing or feature selection [1].

II. MOTIVATION

Machine Learning (ML) is used to solve issues in which the relationship between the input and output variables is unknown or difficult to establish. The term "learning" refers to the automatic acquisition of structural descriptions from instances of the described object. ML does not make assumptions about the right structure of the data model, which characterizes the data, unlike traditional statistical approaches. This property comes particularly handy for modeling complicated non-linear processes, such as a crop yield prediction function. To collect malware signatures and understand and analyze the motiva- tions and techniques behind the threat.

III. RELEVANCE

Botnet identification is significant not only on a local level where the problems of security usually occur but also globally and on a macro level. This work and study are important be- cause it contributes greatly in the fields of agriculture, energy and reserves management, management and reserves of fossil fuels, prediction of natural disasters like floods, hurricanes, tsunamis, and earthquakes, and general ease in management of industrial, technological, and healthcare affairs and by generally making peoples lives easier thereby contributing towards the greater good and to the welfare of the society. The study will be of utmost importance to the disaster management authorities by helping them carefully comb and process the data and information they have had all these years in raw and by collimating the necessary information to help forecast the imminent or incoming dangers. The Botnet detection will also be extremely useful in addressing the issue of data theft from poorly developed apps by the administrations providing greater privacy and security to people. It will also be helpful to the student community to help safeguard their academics and help with their academic research projects and activities. It will also assist people in planning and managing their social activities. It will also help the government in keeping a digital record of all the essential government data and information about the people generated in a huge amount every day.

IV. RELATED WORK

The adaptive technique is one of the most prominent ways to increase the performance of the PSO algorithm [2]–[4]. One of the most well-known algorithms introduced in this field (adaptive) is the APSO algorithm [2]. The goal of the APSO algorithm is to improve the efficiency of the PSO algorithm by adjusting the algorithm's primary parameters in response to changes in the search space. This part not only presents the APSO algorithm, but also goes through some of its drawbacks.

In paper [1] they uses deep learning approach using CNN (Convolutional Neural Networks). They tested the approach with 1,929 botnet applications and 4,387 clean apps in large- scale testing. On the same dataset, the model beats many famous machine learning classifiers. The findings show that our suggested CNN-based model can identify new, previously unseen Android botnets more accurately than the other models (Accuracy: 98.9%; Precision: 0.983; Recall: 0.978; F1-score:

0.981).

V. EASE OF USE

A. Trends of Android Botnets:

The complexity with which Android botnets are evolving is rapidly rising. Despite its similarities, techniques are always being employed, but the manner in which they are used is changing. The procedures that are used are continually evolving. Therefore, It is vital, from a security standpoint, to raise awareness about the developments in the development of Android Botnets.

The initial wave of Android botnet development was a ba- sic SMS Trojan. This Trojan was primarily responsible for delivering SMSs to premium rate phones and was inserted in a repackaged version of a legal program. The Trojan did not yet exhibit any significant botnet functionality, but it did demonstrate the prospect of malware running invisibly on Android smartphones.

Soon after, spyware with the capacity to interact with a remote server began to proliferate on Android handsets. This remote server, also known as a C and C server [5], is in charge of receiving information from the infected Android smartphone as well as transmitting orders to it. This is the first time in the evolution of malware that typical botnet capability has been demonstrated.

Trojan apps install new, but dangerous, software in addition to communicating with a C and C server. A malicious program is downloaded dynamically or the user is prompted to install it. Botnets on Android smartphones are becoming more likely as virus capability improves. As Android botnets became a reality, the attention switched to vulnerabilities that may increase the Trojan malware’s functioning. The ’rage against the cage’ vulnerability is a well-known attack that allows a user to achieve root access on carrier-locked Android handsets. Exploits like this open up new avenues for Android botnet development.

For months, Android malware was mostly found in unap- proved third-party app stores. Malware has recently managed to get past Android’s security gates and into their Official Market. The DroidDream spyware was one of the first to do so. The ability to infect applications on the Official Android Market allows botnet malware to propagate more quickly. The usage of SMSs to receive botnet orders is the newest trend in Android Botnet development. On mobile devices, the typical usage of IRC and HTTP-controlled botnets has become unworkable. SMS, which is available on nearly all mobile devices, expands C and C possibilities.

B. Characteristics of Android Botnets:

The following Android malware is tested in order to dis- cover probable Android characteristics: Base Bridge, BgServ, DroidDream, DroidKungFu, Geinimi, LeNa, Nickispy, Pjapps, Root Smart, and SMSspacem. ADRD, Droid Dream Light, Ton- clank, and Golddream were among the other malware samples examined. It was possible to identify common traits among the malware by analysing the technical reports of the afore- mentioned Android Malware. Repackaging an application, accepting commands, communicating, stealing information, programs discovered on third-party application markets, obtaining new material, and changing the Android Manifest file are among the characteristics. Receiving orders and steal- ing information, for example, are two of the discovered traits that are closely related to classic botnet activity. As a result, these detected traits may be used to detect botnets on Android smartphones [6].

1) Repackaged Application:

Malicious code is typically distributed in the form of an application to operate a botnet. These are well-known and genuine programs, however an attacker reverse engineered and repackaged the original code with malicious code. The user downloads the program but is uninformed of the device’s extra configurations. This feature is comparable to that of a Trojan horse, and it is the most typical way for botnet code to be distributed.

2) Receiving Commands:

A bot’s capacity to either automatically accept commands or to prompt a remote server for commands is a must-have feature. Android botnets utilise strategies that are quite similar to those used by classic botnets. The first method is to transmit orders to the Android bot directly from a C and C server as needed. The alternative solution is to have the Android bot periodically contact the C and C server to see if new commands are available. Any communication with a remote server is a clear sign that an Android botnet is at work.

3) Steal Information:

Android botnets not only receive data from a command-and-control server, but also send data about the infected device to the server. This sort of action normally occurs after the malicious application has been installed. The following are examples of information that Android botnets may collect:

- IMEI (International Mobile Equipment Identity) number
- IMSI (International Mobile Subscriber Identity) number
- GPS Location
- Phone Number

- SDK (Software Development Kit) Version
- Device Model
- Installed Packages

The botmaster can use the above stolen information to uniquely identify and operate a bot.

4) Messaging:

The classic definition of a botnet is one that is used to create devastation at a certain level or to make money. SMS messages are currently being used by Android botnets to collect money by sending messages to premium-rate lines. Premium-rate numbers are phone numbers that are used for a specific service and are charged at a higher rate than regular calls. The botnet may make a lot of money for its controllers by delivering SMS messages to these numbers at regular intervals.

5) Third Party Application Markets:

Malicious programs have traditionally only been found on unapproved third-party application stores. This is no longer the case, as fraudulent apps have just been discovered on the official Android Market. One such example is the DroidDream virus. Even if the chances of finding a malicious software in the Official Android Market are small, vigilance should still be used.

6) Additional Content Downloaded:

The capacity to download new content is the most recent feature of Android botnets. This content, which is generally harmful, assists and boosts the botnet's effectiveness. The app either downloads the new content dynamically or prompts the user to conduct the necessary download.

7) Features and Permissions:

The AndroidManifest.xml file is found in the root directory of every Android application. This file provides the Android system with critical information about the application. The usesfeature¿ and uses-permission¿ elements are among the components found in the AndroidManifest.xml file's structure. The usesfeature¿ element specifies a specific hardware or software feature that the program makes use of.

The following features are frequently used by Android botnets:

- android.hardware.telephony
- android.hardware.touchscreen
- android.hardware.location
- android.hardware.wifi

All of the functions listed above are self-explanatory, and they provide the Android botnet more control over the infected smartphone. The uses-permission¿ element asks for a permis- sion that the program needs to function properly.

The following permissions are frequently used by Android botnets:

- android.permission.READ CONTACTS _
- android.permission.WRITE CONTACTS _
- android.permission.SEND SMS _
- android.permission.WRITE SMS _
- android.permission.READ SMS _
- android.permission.RECEIVE SMS _
- android.permission.READ PHONE STATE _ _
- android.permission.INTERNET
- android.permission.WRITE INTERNAL STORAGE _ _

The AndroidManifest.xml file includes recognisable properties of an Android program and gives essential information to the user about a certain application [6].

VI. PROPOSED SYSTEM

The SVM was put forth by Cortes and Vapnik [7]. It is a supervised learning model based on structural risk minimiza- tion and the Vapnik–Chervonenkis dimension. An SVM is typically used in machine learning and for the purpose of solving classification or regression problems; therefore, the main purpose of an SVM is to identify the optimal hyperplane to analyze various classification data. The optimal hyperplane possesses the maximal margin associated with the various classification data, as shown in Figure Two black and three white points lie on the maximal margin line, which depicts two types of categorization data these points are known as support vectors. SVM selects the points or vectors that are extreme that actually constitute the hyperplane. These extreme cases are therefore referred to as the support vectors, that is why this algorithm is named as Support Vector Machine. Some of the works earlier research papers used the CNN algorithm for similar purposes but here we are employing the SVM for better accuracy and improved consistent results.

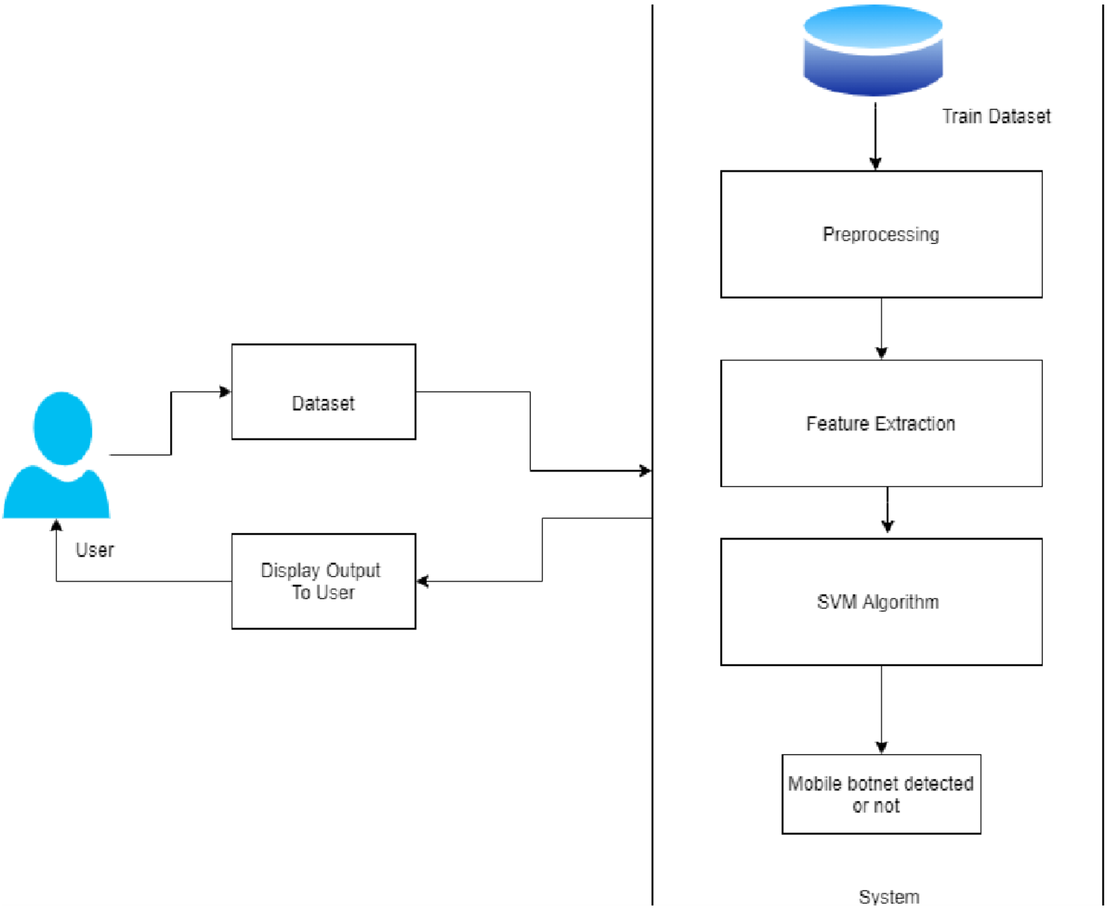

Fig 1: Architecture

SVM can be explained through an example as follows, suppose we encounter a strange weird cat that also has some attributes or characteristics of a dog, so now if we require a model that would precisely identify whether the given animal is a cat or a dog, such a model is perfectly possible by an SVM algorithm, Firstly we shall train our model with numerous images of cats and dogs, for it to learn about their different and distinguishing features. After that, we will test it with a strange creature, so now as explained earlier the SVM will create a boundary of decision between these two different pieces of data i.e. cat and dog, and chooses the extreme scenarios (support vectors), evidently, it will see the extreme instance of cat and a dog then basing its decision on support vectors, the classification will be done as a cat or a dog. The dimensions of a hyperplane are contingent on the char- acteristics in the dataset, this says if there exist two features, the hyperplane should be a straight line. And if there are 3 features, then it comes out to be a 2-dimensional plane.

These support vectors can classify new data. When the data is not linearly separable, the kernel function should be used to map the data into the Vapnik-Chervonenkis dimensional cavity.

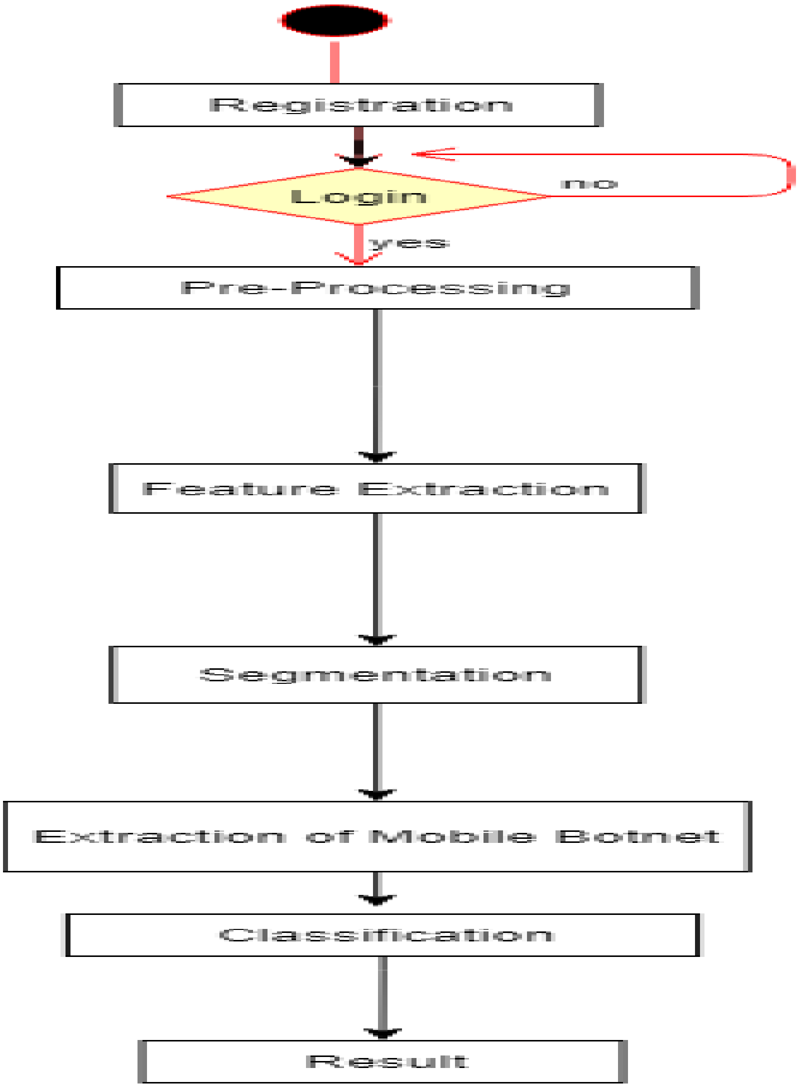

Fig 2: Activity Mobile Botnet

Three types of kernel functions exist, radial basis functions (RBFs), polynomials, and the sigmoids. Using the suitable kernel functions for transforming the data is imperative for increasing the classification speed.

SVM algorithm can be implemented for use cases like Face detection, image classification, text categorization, etc.

VII. ALGORITHM

SVM (Support Vector Machine) [7] is a supervised machine learning technique that may be utilized to solve classification and regression problems. It is, however, usually employed to solve categorization difficulties. Each data item is plotted as a point in an n-dimensional space (where n is the num- ber of features), with the value of each feature being the value of a particular coordinate in the SVM algorithm. Then classification is done by locating the hyper-plane that clearly discriminates the two classes

A. SVM WORKING

Support Vector Machine or SVM is one of the popular Super- vised Learning algorithms, which is used for Classification and Regression problems. However, primarily, its use for Classification problems in Machine Learning. The objective of the SVM algorithm is to create the best line or decision boundary that can segregate an n-dimensional space into classes to easily put the new data point In the correct categories in the future. This best decision boundary is a hyperplane. SVM is the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence algorithm is termed Support Vector Machine.

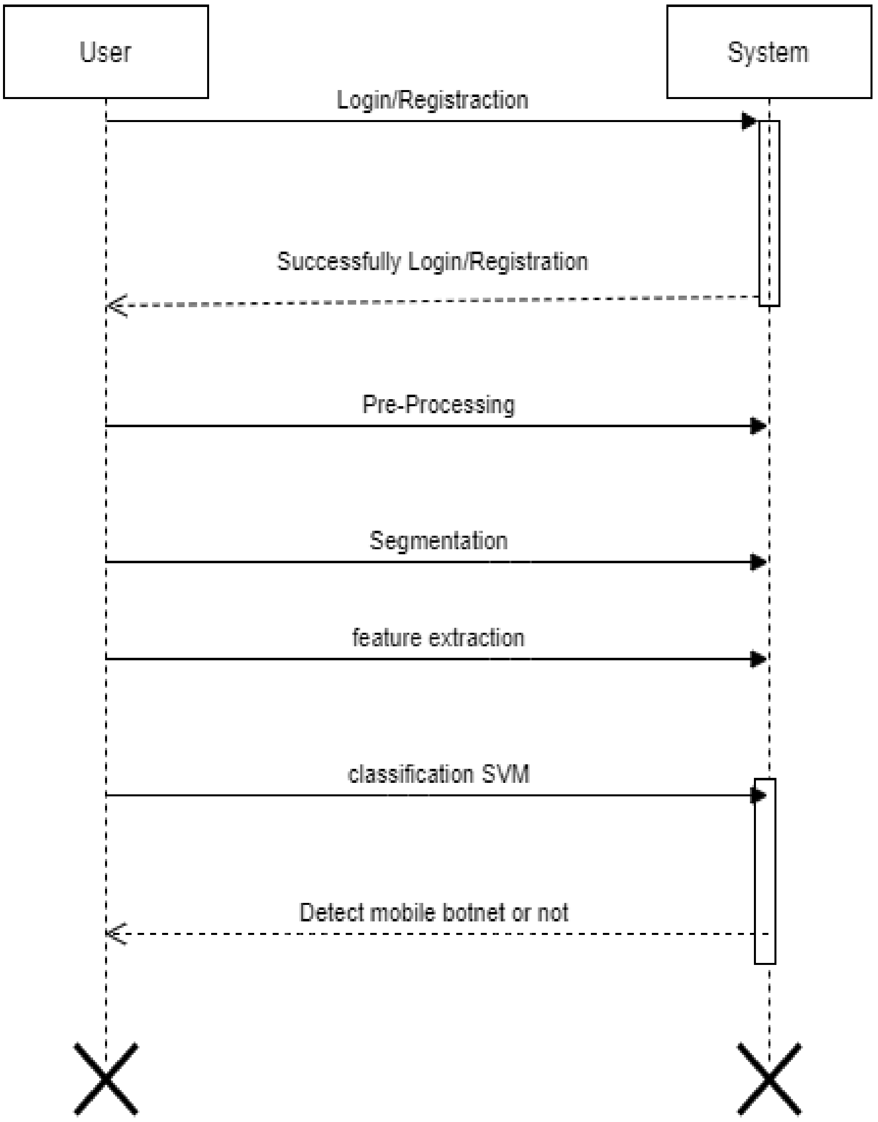

Fig 3: Sequence

B. CLASSIFICATION OF SVM

SVM can be of two major types:

- Linear SVM - Linear SVM as the name suggests it is used for any linearly separable data, meaning, when a data set is categorized into two classes by a straight line, then this data shall be referred to as linearly separable data, and the respective classifier employed is known as Linear SVM classifier [7].
- Non-linear SVM: Non-linear SVM too as the word sug- gests is specifically utilized for non-linearly separated data, implying if a dataset cannot be categorized by a straight line, then this data would be termed as non-linear data and the classifier used is called as Non-linear SVM classifier [7].

C. Why SVM over CNN

SVM (Support Vector Machine)

- SVM is able to extract the separate features of any given dataset.
- Moreover, SVM can also, to some extent, select the separate features of a dataset.
- SVM can smoothly operate with small datasets without issues like overfitting.
- The accuracy of SVM algorithm in binary classification is 80.95%
- SVM in multiple class classification has an accuracy of about 50%.

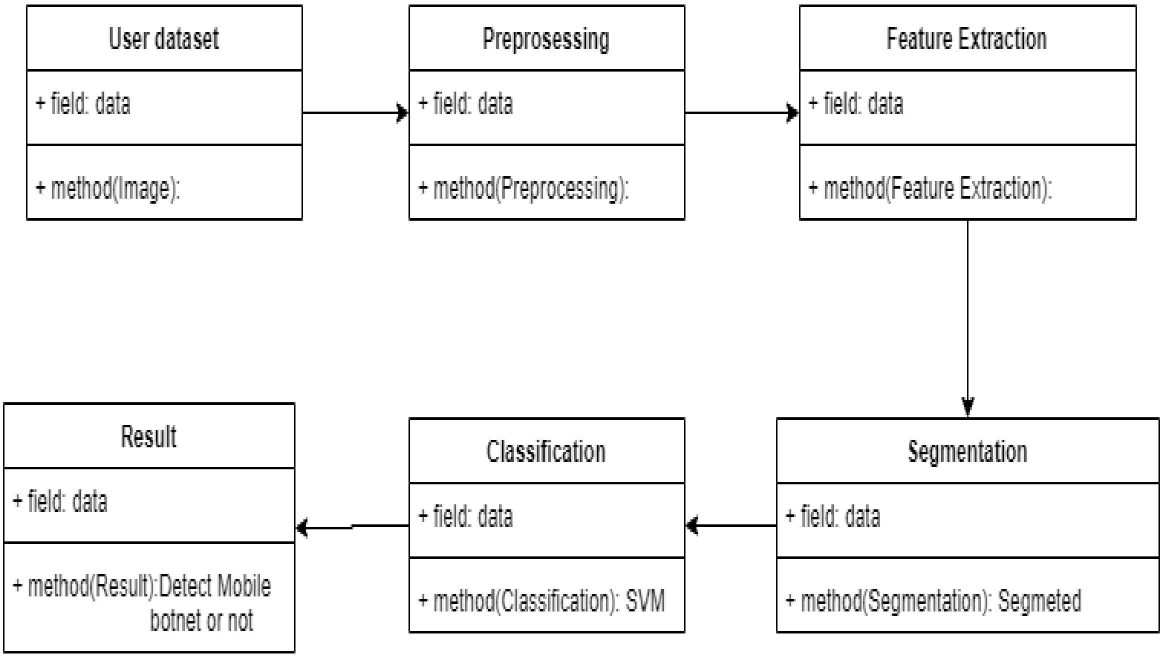

Fig 4: Class Diagram

VIII. CONCLUSION

Botnets are a menace among the malware in society. They are being used to damage and destroy systems, steal information, and Compromise Systems. They are difficult to detect and Eradicate. So, Our System Is a Useful tool to detect Mobile Botnets.

Future Scope for this model is, if additional malware or malicious programs are discovered in the future, we may add more datasets to make it even more precise and easier to use. With these features, it may be utilized in higher-level platforms such as Playstore and Appstore, even if it cannot hold or achieve that level. We can also use it in local company systems to ensure that all apps and programs are malware- free.

IX. REFERENCES

[1] Suleiman Y Yerima and Mohammed K Alzaylaee. Mobile botnet detection: A deep learning approach using convolutional neural networks. In 2020 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), pages 1–8. IEEE, 2020.

[2] Zhi-Hui Zhan, Jun Zhang, Yun Li, and Henry Shu-Hung Chung. Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(6):1362–1381, 2009.

[3] Pinkey Chauhan, Kusum Deep, and Millie Pant. Novel inertia weight strategies for particle swarm optimization. Memetic computing, 5(3):229– 251, 2013.

[4] Ahmad Nickabadi, Mohammad Mehdi Ebadzadeh, and Reza Safabakhsh. A novel particle swarm optimization algorithm with adaptive inertia weight. Applied soft computing, 11(4):3658–3670, 2011.

[5] Yuanyuan Zeng, Kang G Shin, and Xin Hu. Design of sms commanded- and-controlled and p2p-structured mobile botnets. In Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, pages 137–148, 2012.

[6] Heloise Pieterse and Martin S Olivier. Android botnets on the rise: Trends and characteristics. In 2012 information security for South Africa, pages 1–5. IEEE, 2012.

[7] Tristan Fletcher. Support vector machines explained. Tutorial paper, pages 1–19, 2009.