International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

"Mobile Botnet Detection"

Aftab Mulla[1], PranayJadhav[2], GauravBhoi[3], SumitRaj[4], Mrs.SinuNambiar[5],Ms.Anamika Wasnik[6]

[1,2,3,4,5,6] Dept. of Computer Science Engineering

[1,2,3,4,5,6] Dr. D.Y. Patil Institute of Technology, Pune, Maharashtra, India

---

Abstract: ***Android, being the most widespread mobile operating systems is increasingly becoming a target for malware. Malicious apps designed to turn mobile devices into bots that may form part of a larger botnet have become quite common, thus posing a serious threat. This calls -for more effective methods to detect botnets on the Android plat- form. Hence, in this paper, we present a deep learning approach for Android botnet detection based on Support vector machine (SVM). Our proposed botnet detection system is implemented as a svm- based model that is trained on 342 static app features to distinguish between botnet apps and normal apps.***

Keywords: SVM or Support Vector Machine, SQLite Database, botnet, dataset.

I. INTRODUCTION

A. Overview

A botnet consists of a number of Internet-connected devices under the control of a malicious user or group of users known as botmaster(s). It also consists of a Command and Control (CC) infrastructure that enables the bots to receive commands, get updates and send status information to the malicious actors. Since smartphones and other mobile devices are typically used to connect to online services and are rarely switched off, they provide a rich source of candidates for operating botnets. Thus, the term 'mobile botnet' refers to a group of compromised smartphones and other mobile devices that are remotely controlled by botmasters using CC channels.

Why Android?

✔ Android popularity

✓ Ease of use of malicious applications

✔ Lack of proper defense

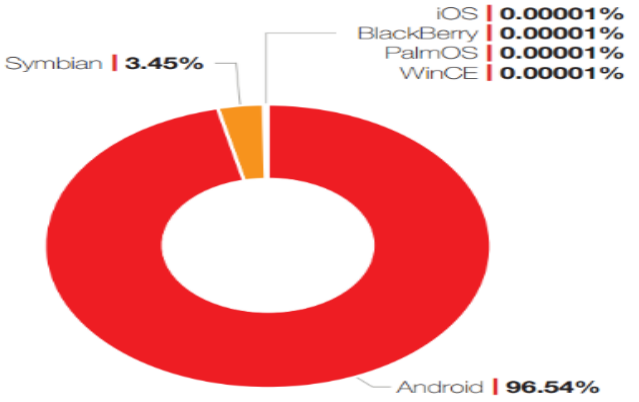

Fig1. Mobile users based on OS

What is Mobile Botnet?

Botnet = roBOT NETwork Collection of compromised Mobile devices (called iBots) which are controlled remotely by a BotMaster through the C&C server.

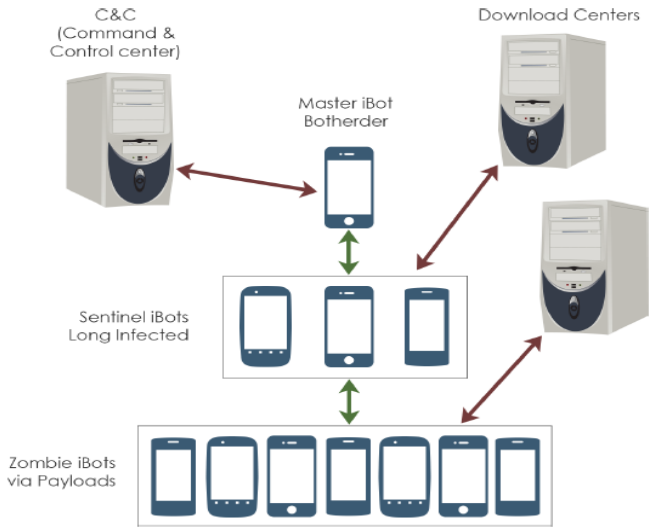

Fig.1.2. Mobile botnet architecture

B. *Project Scope*

A botnet is a network of agreed nodes spreading malware software, usually installed by all varieties of attacking methods likes worms, Trojan horses, and viruses. Many techniques have recently been proposed to block mobile malware or detect it.

C. *Motivation*

They have a strong ability to detect security threats, to collect malware signatures and to understand the motivation and technique behind the threat.

D. *Objective*

The goal is to set the user up for being unknowingly exposed to a malware infection. You'll commonly see hackers exploit security issues in software or websites or deliver the malware through emails and other online messages.

E. *Problem Statement*

In This project We Detect Botnet App. Botnet App Means Some malware are in- stalled in the App through the mobile. That Time loss Your Important Mobile Data. So we Avoid All The loss. Our proposed botnet detection system is implemented as a SVM-based model that is trained on app features to distinguish between botnet apps and normal apps.

F. User Classes and Characteristics.

This Proposed System is we present a deep learning approach for Android botnet detection based on SVM algorithm.

G. Assumptions and Dependencies.

Assumption :

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can eas- ily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Dependencies: Python:

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple program-ming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" lan- guage due to its comprehensive standard library. Python was created in the late.

II. SYSTEM REQUIREMENTS

A. *Database Requirements*

SQLite is one of the most popular and easy-to-use relational database systems. It possesses many features over other relational databases. Many big MNCs such as Adobe, use SQLite as the application file format for their Photoshop Lightroom product. SQLite is an embedded, server-less relational database management system. It is an in-memory open-source library with zero configuration and does not require any installation. Also, it is very convenient as it's less than 500kb in size, which is significantly lesser than other 7410WDSAdatabase management systems.

B. *Software Requirements*

Anaconda Navigator: Anaconda is an open-source distribution of the Python and R programming languages for data science that aims to simplify package management and deployment. Package versions in Anaconda are managed by the package management system, conda, which analyzes the current environment before executing an installation to avoid disrupting other frameworks and packages. The Anaconda distribution comes with over 250 packages automatically installed. Over 7500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI (graphical user interface), Anaconda Navigator, as a graphical alternative to the command line interface. Anaconda Navigator is included in the Anaconda distribution, and allows users to launch applications and manage anaconda packages, environments and channels without using command-line commands. Navigator can search for packages, install them in an environment, run the packages and update them.

C. *Hardware Requirements*

RAM: 8 GB

As we are using Machine Learning Algorithm and Various High Level Libraries Laptop RAM minimum required is 8 GB.

Hard Disk : 500 GB

DataSet is to be used hence minimum 40 GB Hard Disk memories required.Processor : Intel i5 Processor

IDE : Spyder.

III. ANALYSIS MODELS: SDLC MODEL TO BE APPLIED.

The software development cycle is a combination of different phases such as designing, implementing and deploying the project. These different phases of the software development model are described in this section. The SDLC model for the project development can be understood using the following figure the chosen SDLC model is the waterfall model which is easy to follow and fits bests for the implementation of this project.

1) *Requirements Analysis:* At this stage, the business requirements, definitions of use cases are studied and respective documentations are generated.
2) *Design:* In this stage, the designs of the data models will be defined and different data preparation and analysis will be carried out.
3) *Implementation:* The actual development of the model will be carried out in this stage. Based on the data model designs and requirements from previous stages, appropriate algorithms, mathematical models and design patterns will be used to develop the agent's back-end and front-end components.
4) *Testing:* The developed model based on the previous stages will be tested in this stage. Various validation tests will be carried out over the trained model.
5) *Deployment:* After the model is validated for its accuracy scores its ready to be deployed or used in simulated scenarios.
6) *Maintenance:* During the use of the developed solution various inputs/scenarios will been countered by the model which might affect the models overall accuracy. Or with passing time the model might not fit the new business requirements. Thus, the model must be maintained often to keep its desired state of operation.

A. *Mathematical Model*

Let S be the Whole system

S= I,P,O I-inputP-procedure

O-output Input(I)

I= Medical Chatbot dataset Where,

Dataset- Text to speech data, Voice to voice, Language Translation Procedure (P), P=I, Using I System perform operations andcalculate the prediction

Output(O)-O=System detect chatbot

B. *Proposed Algorithm*

SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many prac- tical problems. The idea of SVM is simple: The algorithm creates a line or a hyper plane which separates the data into classes.

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems.

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 nonlinear 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%

IV. SYSTEM ARCHITECTURE

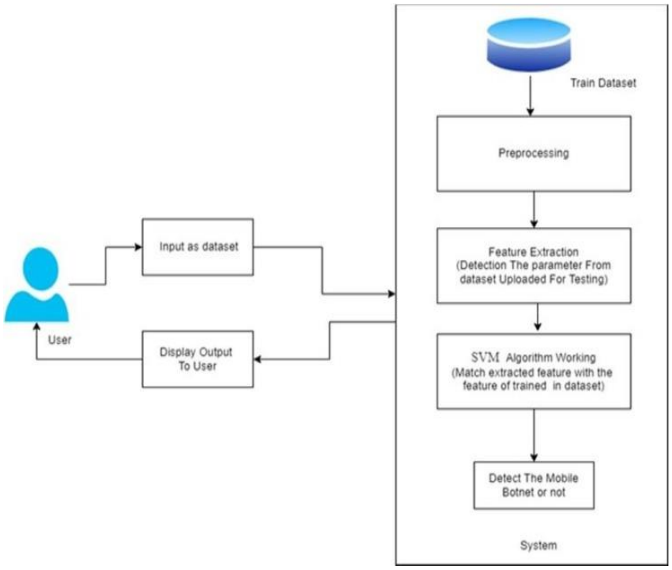

Figure 4.1: System Architecture

A. Data Flow Diagram

In Data Flow Diagram,we Show that flow of data in our system in DFD0 we show that base DFD in which rectangle present input as well as output and circle show our system,In DFD1 we show actual input and actual output of system input of our system is textor image and output is rumor detected like wise in DFD 2 we present operation of user as well as admin.

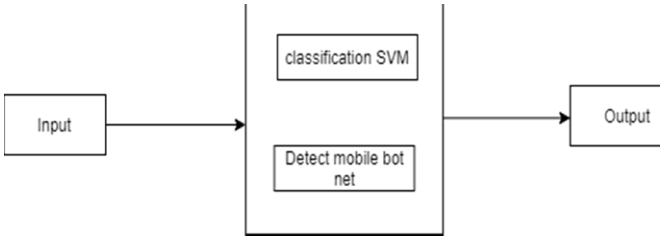

Figure 4.2: Data Flow(0) diagram

B. UML Diagram

Unified Modeling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of a software intensive system. UML is process independent ,although optimally it should be used in process that is use case driven, architecture centric, i terative, and incremental. The Number of UML Diagram is available.

1) Class Diagram

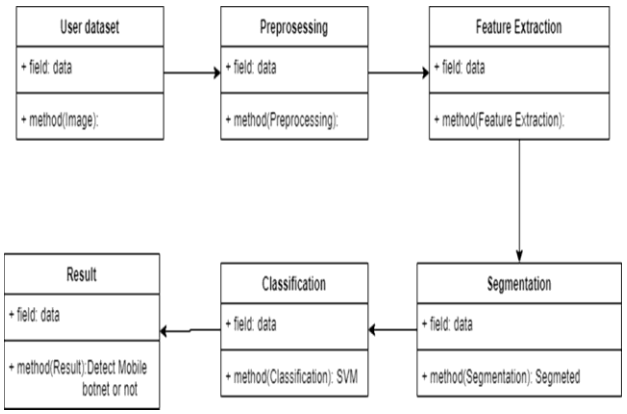

Figure 4.5: Class Diagram Diagram.

2) Use Case Diagram.

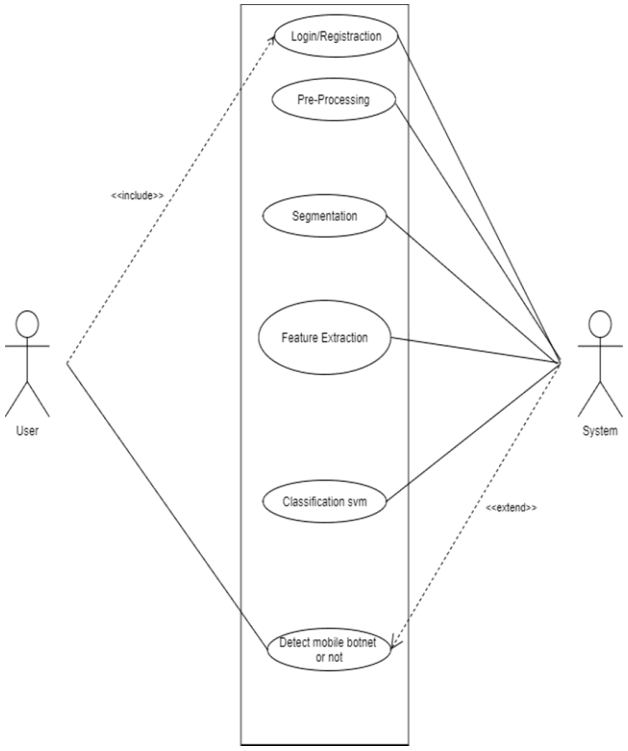

Figure 4.6: Use case Diagram

3) *Activity Diagram*

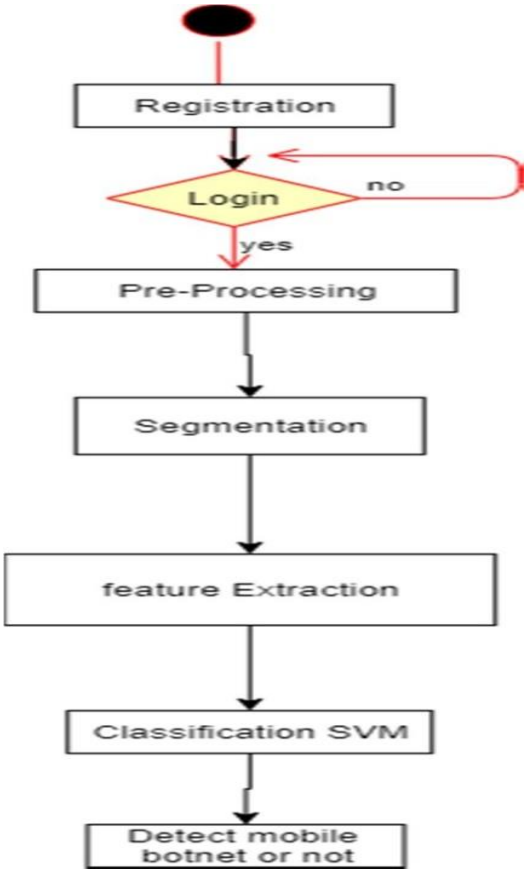

Figure 4.7: Activity Diagram

4) *Sequence Diagram*

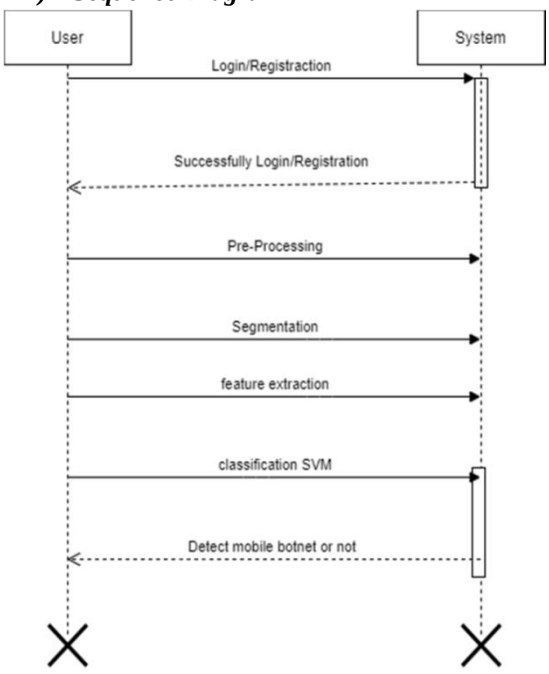

Figure 4.8: Sequence Diagram

V. CONCLUSION

Botnets are a Dangerous evolution in the malware world. They are being used to damage systems, steal information and Comprise Systems. They are hard to detect and eliminate. So Our System Is Useful To detect Mobile Botnet.

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.

VI. REFERENCES

[1] S. Y. Yerima and S. Khan “Longitudinal Perfomance Anlaysis of Machine Learning based Android Malware Detectors” 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), IEEE.

[2] H. Pieterse and M. S. Olivier, ”Android botnets on the rise: Trends and charac- teristics,” 2012 Information Security for South Africa, Johannesburg, Gauteng, 2012, pp. 1-5.

[3] Letteri, I., Del Rosso, M., Caianiello, P., Cassioli, D., 2018. Performance of botnet detection by neural networks in software-dened networks, in: CEUR WORKSHOP PROCEEDINGS, CEUR-WS.

[4] Kadir, A.F.A., Stakhanova, N., Ghorbani, A.A., 2015. Android botnets: What urls are telling us, in: International Conference on Network and System Secu- rity, Springer. pp. 78–91.

[5] ISCX Android botnet dataset. Available from https://www.unb.ca/cic/datasets/android- botnet.html. [Accessed 03/03/2020]

[6] M. Eslahi, M. V. Naseri, H. Hashim, N. M. Tahir, and E. H. M. Saad, ”BYOD: Current State and Security Challenges,” presented at the IEEE Symposium on Computer Applications Industrial Electronics, Peneng, Malaysia, 2014

[7] S. S. C. Silva, R. M. P. Silva, R. C. G. Pinto, and R. M. Salles, ”Botnets: A survey,” Computer Networks, vol. 57, pp. 378-403, 2013.

[8] G. Gu, J. Zhang, and W. Lee, ”BotSniffer: Detecting botnet command and control channels in network traffic,” in Proceedings of the 15th Annual Net- work and Distributed System Security Symposium (NDSS’08), 2008

[9] C. Byungha, C. Sung-Kyo, and C. Kyungsan, ”Detection of Mobile Botnet Using VPN,” in Proceedings of the Seventh International Conference on Inno- vative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013, pp. 142-148