A Project Report on

DETECTION OF ANOMALOUS BEHAVIOR OF SMARTPHONE DEVICES USING CHANGE POINT ANALYSIS & MACHINE LEARNING

Submitted in partial fulfilment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

J.R. SAI LOHITHA	**(19KA1A0519)**
A. SATHWIKA	**(19KA1A0504)**
V. HARI SREE RANI	**(19KA1A0517)**
M. PREMALATHA	**(19KA1A0555)**

Under the esteemed guidance of

Mrs. K.R. LAVANYA, M. Tech.

Assistant Professor (Adhoc)

Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING: KALIKIRI

ANNAMAYYA (Dist.), ANDHRA PRADESH – 517234

2019-2023

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING, KALIKIRI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "**DETECTION OF ANOMALOUS BEHAVIOR OF SMARTPHONE DEVICES USING CHANGE POINT ANALYSIS & MACHINE LEARNING**" that is submitted by

J.R. SAI LOHITHA	**(19KA1A0519)**
A. SATHWIKA	**(19KA1A0504)**
V. HARI SREE RANI	**(19KA1A0517)**
M. PREMALATHA	**(19KA1A0555)**

in partial fulfilment of the requirements for the award of degree of Bachelor of Technology (**B. Tech**) in **Computer Science and Engineering (CSE)** from **Jawaharlal Nehru Technological University Anantapur College of Engineering, Kalikiri** during the academic year **2022-2023**.

Project Guide

Mrs. K. R. Lavanya, M. Tech

Assistant Professor (Adhoc),

Department of CSE, JNTUACEK

Kalikiri, Annamayya (Dist.)

Head of The Department

Mrs. Shaik Naseera, M. Tech., Ph.D.

Professor & Head of Department,

Department of CSE, JNTUACEK

Kalikiri, Annamayya (Dist.)

Internal Examiner

External Examiner

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING, KALIKIRI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We J.R. SAI LOHITHA(19KA1A0519), A. SATHWIKA(19KA1A0504), V. HARI SREE RANI(19KA1A0517), M. PREMALATHA(19KA1A0555) hereby declare that the project work entitled "**DETECTION OF ANOMALOUS BEHAVIOR OF SMARTPHONE DEVICES USING CHANGE POINT ANALYSIS & MACHINE LEARNING**" is a genuine work carried out by us under the guidance of **Mrs. K.R.LAVANYA**, M. Tech., Assistant Professor(Ad hoc), Department of CSE, in partial fulfilment for award of the degree of "**BACHELOR OF TECHNOLOGY**" from JNT University, Anantapur. The results embodied in this project work has not been submitted to any other university or institute for the award of any degree.

J.R. SAI LOHITHA	**(19KA1A0519)**
A. SATHWIKA	**(19KA1A0504)**
V. HARI SREE RANI	**(19KA1A0517)**
M. PREMALATHA	**(19KA1A0555)**

ACKNOWLEDGEMENTS

An endeavour over a long period can be successful only with advice and support of many well-wishers. The task would be incomplete without mentioning the people who have made it possible, because it is the epitome of hard work. So, with the gratitude, we acknowledge all those whose guidance and encouragement owned our efforts with success.

We are thankful to **Prof. S. V. SATYANARAYANA**, M. Tech., Ph.D., Principal and Professor of Chemical Department, JNTUACE, Kalikiri for his kind and timely help offered to us in projection of our studies and execution.

We are very much obliged to our beloved **Dr. SHAIK NASEERA**, M. Tech., Ph.D., HOD and Professor, Department of Computer Science & Engineering, JNTUACE, Kalikiri for the moral support and invaluable advice provided by her for the success of the project.

We wish to express grateful acknowledgement to our guide **Mrs. K.R. LAVANYA**, M. Tech, Assistant Professor (Adhoc), Department of Computer Science & Engineering, JNTUACE, Kalikiri for his inspiring guidance and continuous encouragement throughout the project.

Finally, we would like to extend our deep sense of gratitude to all the staff members, friends and last but not least we are greatly indebted to our parents who inspired us at all circumstances.

PROJECT ASSOCIATES

J.R. SAI LOHITHA (19KA1A0519)
A. SATHWIKA (19KA1A0504)
V. HARI SREE RANI (19KA1A0517)
M. PREMALATHA (19KA1A0555)

ABSTRACT

The use of smartphones has increased significantly over the years, and so has the risk of malware attacks on these devices. Traditional methods for detecting malicious activity on smartphones, such as static and dynamic analysis, have proven to be vulnerable and time-consuming. In this project, we propose a generic methodology that uses a data collector and analyser to detect anomalous behaviour on smartphones by analysing changes in power consumption, which can summarize software changes.

To create user inputs, the data collector employs an automated tool, and the data analyser uses change point analysis to extract features from power usage data. We train these features using machine learning algorithms such as support vector machine, decision tree, XG Boost, and AdaBoost. Two techniques are used in the data analyser step to extract features, utilizing both parametric and non-parametric change points.

Our methodology is designed to take less time to collect data than manual methods and is more accurate than previous approaches, both for simulated and actual malware. The system includes a user module and a user interface to facilitate data collection, pre-processing, model building, and result visualization. We present a data flow diagram and collaboration diagram to illustrate the system's modules and the interaction between them. Additionally, we discuss the use of XAMPP and Docker in the project and provide a detailed explanation of the selected machine learning algorithms.

Overall, our project proposes an efficient and accurate methodology for detecting anomalous behaviour on smartphones, which can aid in early detection and prevention of malware attacks

Table of Contents

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBRREVIATIONS

SVM	-	SUPPORT VECTOR MACHINE
XG BOOST	-	EXTREME GRADIENT BOOST
ADA BOOST	-	ADAPTIVE BOOST
IOT	-	INTERNET OF THINGS
AR	-	ARGUMENTED REALITY
LBMAR	-	LOCATION-BASED MOBILE AUGMENTED REALITY
DEX	-	DALVIK EXECUTABLE
AMD	-	ANDROID MALWARE DATASET
TCP	-	TRANSMISSION CONTROL PROTOCOL
IDE	-	INTEGRATED DEVELOPMENT ENVIRONMENT
SDLC	-	SOFTWARE DEVELOPMENT LIFE CYCLE
CSS	-	CASCADING STYLE SHEET
HTML	-	HYPERTEXT MAEKUP LANGUAGE
HTTP	-	HYPER TEXT TRANSFER PROTOCOL
FTP	-	FILE TRANSFER PROTOCOL
XSS	-	CROSS-SITE SCRIPTING
CSRF	-	CROSS-SITE REQUEST FORGERY
VS CODE	-	VISUAL STUDIO CODE

CHAPTER 1

INTRODUCTION

1.1 Introduction

Smartphones have become an essential part of modern life due to their widespread use for communication and access to information. However, these devices are also a target for cybercriminals, who develop malicious applications to steal user information or harm the performance of cellular networks. Researchers have been developing various methodologies to detect these malicious applications based on the analysis of dynamic characteristics of the device such as power consumption, network traffic, CPU activity, and temperature.

In this project, we propose a novel methodology to detection of anomalous behaviour on smartphones using power consumption as the primary feature. The hypothesis is that the power consumed by a device contains valuable information that can be used to identify the presence of abnormal activities. The proposed methodology uses an offline processing technique and off-device measurement, in which an external device collects the power consumption data to improve the resolution of the power traces.

To extract features from the non-stationary power consumption time series signal, we use the theory of changepoint detection. This theory identifies points in the time series where there is a significant change in the power consumption pattern, which can indicate the presence of a malicious application. The extracted features are used as input to a binary classification problem, where the goal is to detect the presence or absence of anomalous behaviour.

To solve this classification problem, we employ two machine learning algorithms, Support Vector Machines (SVM), Decision Trees, XG Boost and Ada Boost Classifier. SVM is a supervised learning algorithm that can effectively classify non-linear data by finding the optimal hyperplane that separates the two classes. Decision Trees, on the other hand, are a supervised learning algorithm that uses a tree-like model to classify data based on a set of decision rules. XG Boost (Extreme Gradient Boosting) and AdaBoost (Adaptive Boosting) are two popular machine learning algorithms that use boosting techniques to improve the accuracy of models by combining multiple weak classifiers into a stronger ensemble classifier.

We conduct experiments on a dataset of power consumption traces collected from a set of smartphones running both benign and malicious applications. Our results show that the proposed methodology can effectively detect malicious applications with high accuracy using both Ada Boost and Decision Trees. Furthermore, the Decision Trees outperformed all other algorithms in terms of accuracy, precision, recall, and F1-score.

In conclusion, we present a novel methodology for detecting anomalous behaviour of smartphones using change point analysis as the primary feature. We demonstrate that the proposed methodology, combined with Machine learning algorithms, can effectively detect anomalous applications with high accuracy.

1.2 Objectives

The project aims to address the growing concern of cyber security threats on smartphones, which are a common target for cybercriminals due to the sensitive information they store and transmit. The proposed methodology utilizes change point analysis, which involves detecting changes in the behaviour of a time series signal, such as the power consumption of a smartphone, to identify anomalous behaviour. The change points can provide useful features for machine learning algorithms, such as SVM, Decision Trees, XG Boost, and AdaBoost, to improve the accuracy of detecting malicious applications or other anomalous behaviour.

By comparing the performance of different machine learning algorithms, the project aims to identify the most effective approach for detecting anomalous behaviour in smartphones. The SVM algorithm is known for its ability to handle complex and high-dimensional data, while Decision Trees are easy to interpret and can handle both numerical and categorical data. XG Boost is a powerful algorithm that uses gradient boosting and has been shown to achieve state-of-the-art results in various machine learning tasks, while AdaBoost can improve the accuracy of weak classifiers by combining them into a strong ensemble classifier.

The project aims to contribute to the development of effective and efficient methodologies for detecting anomalous behaviour in smartphones, which can help to enhance their security and protect the privacy of their users.

CHAPTER 2

LITERATURE SURVEY

D. Evans, "The internet of things: How the next evolution of the internet is changing everything," CISCO white paper, Tech. Rep., 2011.

Nowadays, we experience an abundance of Internet of Things middleware solutions that make the sensors and the actuators are able to connect to the Internet. These solutions, referred to as platforms to gain a widespread adoption, have to meet the expectations of different players in the IoT ecosystem, including devices. Low-cost devices are easily able to connect wirelessly to the Internet, from handhelds to coffee machines, also known as Internet of Things (IoT). This research describes the methodology and the development process of creating an IoT platform. This paper also presents the architecture and implementation for the IoT platform. The goal of this research is to develop an analytics engine which can gather sensor data from different devices and provide the ability to gain meaningful information from IoT data and act on it using machine learning algorithms. The proposed system is introducing the use of a messaging system to improve the overall system performance as well as provide easy scalability.

Statista, "Number of mobile phone users worldwide from 2015 to 2020 (in billions)." [Online]. Available: https://www.statista.com/statistics/274774/forecast-ofmobile-phone-users-worldwide

The advancement of virtual reality has sharpened the concept of augmented reality (AR) to new dimension of perceptions of seeing, hearing and immersing in a real world. The evolution of mobile devices has pioneered AR as a state-of-the-art technology in the last decade giving rise to more and more location-based mobile AR (LBMAR) systems. However, notably there are very limited review studies that have focused on investigating factors such as: growth, types, characteristics, features, sensors, application domains and their respective challenges. This study presents a systematic review on location-based mobile augmented reality (LBMAR) system. A total of 35 studies published between the years 2013 and 2018 in top six most popular indexed databases are reviewed. The systematic review has been conducted using Kitchenham method, and the analysis of the findings was carried out using the PRISMA method. This chapter provides a major review of the current state of LBMAR system and outlines the research issues that require more investigation.

A. Arabo and B. Pranggono, "Mobile malware and smart device security: Trends, challenges and solutions," in 2013 19th International Conference on Control Systems and Computer Science, May 2013, pp. 526–531.

This work is part of the research to study trends and challenges of cyber security to smart devices in smart homes. We have seen the development and demand for seamless interconnectivity of smart devices to provide various functionality and abilities to users. While these devices provide more features and functionality, they also introduce new risks and threats. Subsequently, current cyber security issues related to smart devices are discussed and analyzed. The paper begins with related background and motivation. We identified mobile malware as one of the main issues in the smart devices' security. In the near future, mobile smart device users can expect to see a striking increase in malware and notable advancements in malware-related attacks, particularly on the Android platform as the user base has grown exponentially. We discuss and analysed mobile malware in details and identified challenges and future trends in this area. Then we propose and discuss an integrated security solution for cyber security in smart devices to tackle the issue.

T. Kim, B. Kang, M. Rho, and et. all, "A multimodal deep learning method for android malware detection using various features," IEEE Trans. on Info. Forensics and Security, vol. 14, no. 3, 2019.

With the widespread use of smartphones, the number of malwares has been increasing exponentially. Among smart devices, Android devices are the most targeted devices by malware because of their high popularity. This paper proposes a novel framework for Android malware detection. Our framework uses various kinds of features to reflect the properties of Android applications from various aspects, and the features are refined using our existence-based or similarity-based feature extraction method for effective feature representation on malware detection. Besides, a multimodal deep learning method is proposed to be used as a malware detection model. This paper is the first study of the multimodal deep learning to be used in the Android malware detection. With our detection model, it was possible to maximize the benefits of encompassing multiple feature types. To evaluate the performance, we carried out various experiments with a total of 41,260 samples. We compared the accuracy of our model with that of other deep neural network models.

Y.-S. Yen and H.-M. Sun, "An android mutation malware detection based on deep learning using visualization of importance from codes," Microelectronics Reliability, vol. 93, pp. 109–114, 2019.

The rapid proliferation of Android malware is challenging the classification of the Android malware family. The traditional static method for classification is easily affected by the confusion and reinforcement, while the dynamic method is expensive in computation. To solve these problems, this paper proposes an Android malware familial classification method based on Dalvik Executable (DEX) file section features. First, the DEX file is converted into RGB (Red/Green/Blue) image and plain text respectively, and then, the colour and texture of image and text are extracted as features. Finally, a feature fusion algorithm based on multiple kernel learning is used for classification. In this experiment, the Android Malware Dataset (AMD) was selected as the sample set. Two different comparative experiments were set up, and the method in this paper was compared with the common visualization method and feature fusion method. The results show that our method has a better classification effect with precision, recall and F1 score reaching 0.96. Besides, the time of feature extraction in this paper is reduced by 2.999 seconds compared with the method of frequent subsequence. In conclusion, the method proposed in this paper is efficient and precise in the classification of the Android malware family.

CHAPTER 3

PROBLEM IDENTIFICATION

3.1 Existing System

In the existing systems, implementation of machine learning algorithms is bit complex to build due to the lack of information about the data visualization.

Mathematical calculations are used in existing systems for model building this may takes the lot of time and complexity.

The existing systems has several machine learning models to classify whether there is anomalous behaviour or not in the android device, but none have adequately addressed this misdiagnosis problem.

Also, similar studies that have proposed models for evaluation of such performance classification mostly do not consider the heterogeneity and the size of the data

To overcome all this, we use machine learning packages available in the scikit-learn library

3.1.1 Disadvantages of Existing Systems

1. Lack of information about data visualization: The existing system faces challenges in implementing machine learning algorithms due to the lack of information about data visualization. This means that it is difficult to understand and visualize the data, which can make it harder to identify patterns or outliers that may be indicative of anomalous behaviour.
2. Use of mathematical calculations for model building: The existing system relies on mathematical calculations for building machine learning models. This approach can be time-consuming and complex, as it involves writing and executing complex mathematical equations to build the models.
3. Misdiagnosis problem: The existing system has several machine learning models for classifying whether there is anomalous behaviour in the Android device. However, these models have not adequately addressed the misdiagnosis problem, which refers to the incorrect identification of anomalous behaviour.

4. Ignoring heterogeneity and data size: Similar studies that have proposed models for evaluating the performance of classification mostly do not consider the heterogeneity and size of the data. The existing system may face similar challenges in addressing these issues.

To overcome these challenges, the proposed system will use machine learning packages available in the scikit-learn library. It will also incorporate change point analysis to detect anomalous behaviour in smartphones, which can improve the accuracy of the models.

3.2 Proposed System

The proposed project aims to provide an end-to-end methodology to automatically collect data and analyse them to detect anomalous behaviour in smartphones. The methodology involves collecting network traffic and TCP packets, filtering them, and then selecting and extracting features from them. The selected functions from various network features are then labelled and stored.

Machine learning classification is then used to build a detection model. The proposed project uses several machine learning models, including Support Vector Machines (SVM), Decision Trees, XG Boost Classifier, and AdaBoost Classifier, to classify whether there is anomalous behaviour or not in the smartphone. The selected features and labelled data are used to train the machine learning models, which are then used to predict whether there is anomalous behaviour in the smartphone.

The use of machine learning classification in the proposed project provides several advantages. First, it offers the highest accuracy compared to traditional methods. Second, it reduces time complexity by automating the process of analysing the collected data. Third, it is easy to use, making it accessible to a wide range of users.

SVM is a popular classification technique used in machine learning. It works by finding a hyperplane that maximizes the margin between the classes. SVM is effective

in handling high-dimensional datasets and has been used in many applications, including image recognition, natural language processing, and bioinformatics.

Decision Trees are another classification technique used in machine learning. They work by partitioning the feature space recursively based on the feature that provides the most information gain. Decision Trees are useful when dealing with categorical data and can handle missing data.

XG Boost Classifier is a powerful classification technique that is particularly effective when dealing with large datasets. It works by combining multiple decision trees and correcting their errors through a process called boosting. XG Boost is widely used in various applications, including speech recognition, natural language processing, and computer vision.

Ada Boost is another boosting technique that combines weak classifiers to create a strong classifier. It is particularly useful in handling imbalanced datasets and has been used in applications such as credit scoring, spam filtering, and face detection.

The proposed methodology has the potential to provide an effective and efficient approach to detecting anomalous behaviour in smartphones. The use of machine learning classification techniques can improve the accuracy of the detection and reduce the time and complexity of the analysis process. The proposed approach is also easy to use, making it accessible to a wide range of users.

3.2.1 Advantages of Proposed System

The proposed system for anomalous detection of smartphone applications using changepoint analysis and machine learning techniques offers several advantages over the existing system. Some of the main advantages are:

1. Higher Accuracy: The proposed system utilizes changepoint analysis to extract features from power consumption data, and machine learning techniques to train a classifier. This enables the system to identify malicious behaviour more accurately, leading to better detection rates and fewer false positives.

2. Faster Detection: The proposed system can recognize malware acting in short periods of time which is a disadvantage of the other methodologies. This leads to faster detection of malicious behaviour, reducing the time taken for detection and remediation.
3. Greater Flexibility: The proposed system can be easily adapted to detect anomalous behaviour in different applications, enabling it to be used in a variety of scenarios. This makes it a more versatile solution compared to existing systems that may be limited to specific applications.
4. Reduced Human Effort: The proposed system employs an automated tool to create user inputs, and the data analyser is more accurate than previous methods. This reduces the human effort required to collect data, increasing efficiency and reducing costs.
5. Improved Security: By detecting malicious behaviour more accurately and quickly, the proposed system can help to improve the overall security of smartphones. This can help to prevent sensitive data from being compromised, reducing the risk of data breaches and other security incidents.

The proposed system offers significant advantages over existing systems, with higher accuracy, faster detection, greater flexibility, reduced human effort, and improved security. These advantages make it a more effective and efficient solution for anomalous detection of smartphone applications.

3.3 Scope

1. The scope of our project is to develop a system for anomalous detection of smartphones using change point analysis and machine learning algorithms. This involves designing and implementing a data collector and analyser that can identify harmful activity through data analysis and power consumption monitoring.
2. Our project also includes the development of a user module that allows users to interact with the system. The user module should be intuitive and easy to use, making it accessible to a wide range of users.
3. Another important aspect of our project is the selection and implementation of machine learning algorithms. We will need to carefully evaluate various algorithms and select

the ones that are best suited for detecting anomalous behaviour in smartphone applications. This will require a deep understanding of machine learning principles and techniques.

4. Additionally, our project will require the use of various tools and technologies, such as Python, Scikit-learn, Pandas, NumPy, Flask, and Docker. You will need to be proficient in using these tools and technologies, as well as in programming in general.
5. Finally, the scope of our project includes testing and validation of the system to ensure that it is accurate, reliable, and effective. We will need to design and implement various tests to evaluate the performance of the system, and make any necessary adjustments and improvements to ensure that it meets the required standards.

The scope of our project is broad and requires a range of skills and expertise. By developing a system for anomalous detection of smartphones, we will be contributing to the ongoing efforts to improve cybersecurity and protect users from malicious activity. Our project has the potential to have a significant impact on the field of cybersecurity, and could be used to develop more advanced and effective detection systems in the future.

3.4 Target

The primary target of this project is to develop a methodology for identifying malicious activity on smartphones in real-time. With the increasing number of malware attacks on smartphones, it is important to have a system that can detect these attacks quickly and accurately. The proposed methodology uses changepoint detection theory and machine learning algorithms to analyse the power consumption of a smartphone and identify any anomalous behaviour. The aim is to achieve a high level of accuracy in detecting malware and other malicious activity on smartphones.

Another target of this project is to provide a user-friendly interface for smartphone users to monitor their devices for malicious activity. The user interface will display information about the smartphone's power consumption and highlight any anomalous behaviour. This will help users to detect any potential threats and take necessary actions to prevent further damage. The user interface will also provide guidance on how to safeguard their devices against malware attacks and other security threats.

A key target of this project is to provide a scalable and efficient system that can handle large volumes of data. The proposed methodology uses machine learning algorithms to analyse the power consumption data collected from smartphones. To achieve scalability, we will use distributed computing techniques and cloud-based platforms to process the data.

Another target of this project is to enable researchers to study malware attacks on smartphones and develop new techniques to detect and prevent these attacks. The proposed methodology provides a framework for collecting and analysing data from smartphones, which can be used by researchers to develop new machine learning algorithms and other techniques for detecting malware. The system can also be used to analyse the effectiveness of existing malware detection techniques and identify areas for improvement.

Finally, a key target of this project is to contribute to the development of a more secure and resilient mobile ecosystem. The proposed methodology can help to improve the security of smartphones and prevent cyber-attacks. By providing a reliable and efficient system for detecting and preventing malware attacks, we can help to protect users' personal information and prevent financial losses. Ultimately, the goal of this project is to enhance the overall security and privacy of smartphone users and contribute to a safer and more secure mobile ecosystem.

3.5 Feasibility Study

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY
- OPERATIONAL FEASIBILITY

3.5.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Economic feasibility refers to the cost-benefit analysis of the proposed solution. This project aims to develop a methodology that can detect anomalous behaviour in smartphone applications, which can help prevent malware attacks and other cyber threats. The cost of developing the proposed solution includes the cost of hiring skilled professionals and purchasing hardware and software. However, the benefits of the solution outweigh the cost, as it can prevent data breaches, identity theft, and other cyber threats. Therefore, the economic feasibility of the proposed solution is high.

3.5.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system

Technical feasibility refers to the capability of the technology to achieve the objectives of the project. In this project, we aim to develop a methodology to detect anomalous behaviour in smartphone applications using changepoint detection theory and machine learning techniques. The proposed methodology utilizes commonly available technology, such as smartphones, automated tools for creating user inputs, and machine learning libraries. Therefore, the technical feasibility of this project is high.

3.5.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel

threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

Social feasibility refers to the impact of the proposed solution on society. The proposed methodology can have a significant impact on society, as it can help prevent cyber threats, which have become prevalent in recent times. It can also help protect user privacy and confidential data. However, the methodology may require users' consent to collect power consumption data from their smartphones. Therefore, the social feasibility of the proposed solution is moderate.

3.5.4 Operational Feasibility

Operational feasibility refers to the practicality of implementing the proposed solution. This project aims to develop a methodology for detecting anomalous behaviour in smartphone applications, which requires collecting power consumption data and training a machine learning model. The data collection process may require user inputs and can be automated. Additionally, the trained model can be deployed on a smartphone or in the cloud for real-time detection of anomalous behaviour. Therefore, the operational feasibility of the proposed solution is high.

3.6 Requirements

The requirement analysis of the project "Anomalous behaviour of smartphone using change point analysis and machine learning algorithms" involves identifying the necessary components and functionalities to achieve the project goals. Firstly, data collection and pre-processing of network traffic and TCP packets are required to extract relevant features for analysis. This involves filtering and labelling of the collected data to enable effective analysis using machine learning algorithms. Secondly, change point analysis is used to detect any abnormalities in the collected data. Change point analysis identifies sudden changes or shifts in the data patterns, which could indicate the presence of an anomaly in the smartphone behaviour.

The project also requires the implementation of various machine learning algorithms, including Support Vector Machines, Decision Trees, XG Boost, and AdaBoost. These algorithms are used to build classification models that can detect anomalous behaviour in

smartphones. Each of these algorithms has unique strengths and limitations, and their performance is evaluated to determine the most suitable algorithm for the project. Additionally, the accuracy and efficiency of the proposed methodology are crucial requirements of the project, as they determine the effectiveness and practicality of the anomaly detection system.

The project involved analysing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

3.6.1 Functional Requirements

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1) Authentication of user whenever he/she logs into the system
2) System shutdown in case of a cyber-attack
3) A verification email is sent to user whenever he/she register for the first time on some software system.

3.6.2 Non-Functional Requirements

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project together. They are also called non-behavior requirements. They basically deal with issues like:

- Portability
- Security
- Maintainability

- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

1) The processing of each request should be done within 10 seconds
2) The site should load in 3 seconds whenever of simultaneous users are > 10000

3.6.3 Hardware & Software Requirements

a) Hardware:

For developing the application, the following are the Software Requirements:

- Coding Language : Python 3.9.4
- IDE : PyCharm 2022.2.2
- Frameworks : Flask 2.2
- Tool : WinRunner, Docker, XAMPP Server 3.2.0
- Operating System : Windows 10
- Debugger and Emulator : Any Browser (Particularly Chrome)

b) Software:

For developing the application, the following are the Hardware Requirements:

- Processor : Pentium IV or higher
- RAM : 8 GB
- Hard Disk or SSD : More than 500 GB

CHAPTER 4

SYSTEM DESIGN

4.1 Software Development Life Cycle (SDLC)

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.

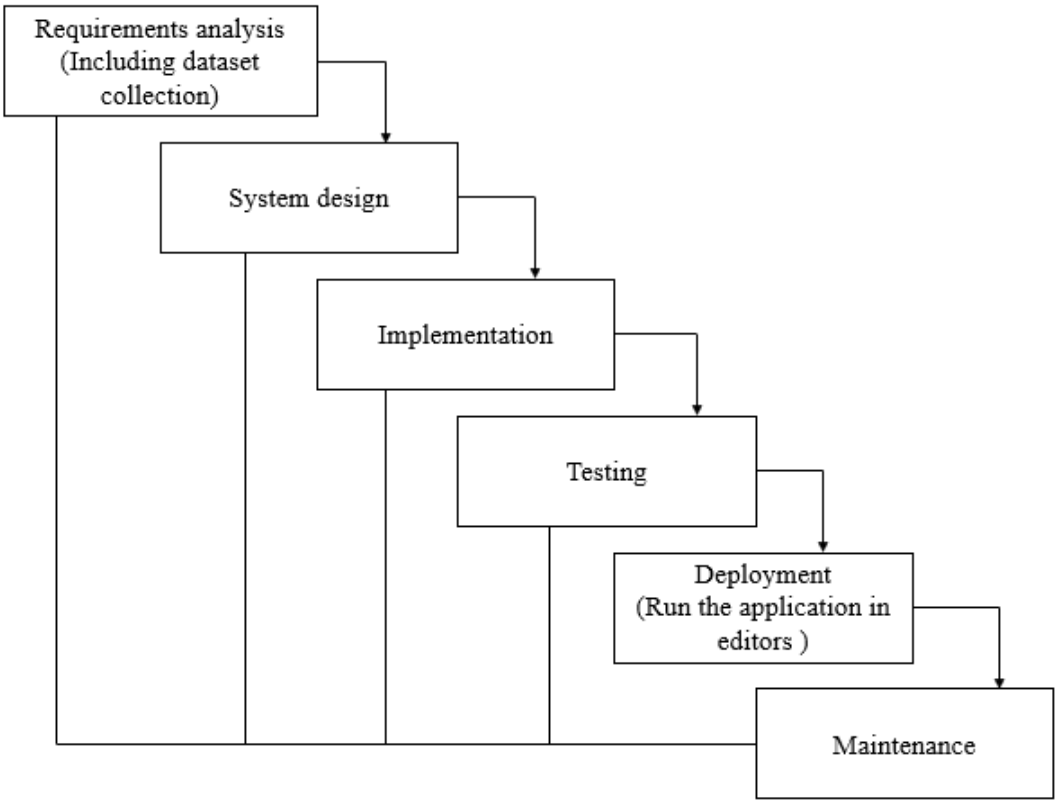

Figure 1 Waterfall

- Requirement Gathering and analysis − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- System Design − the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- Implementation − with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- Integration and Testing − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- Deployment of system − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- Maintenance − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

4.1.1 Need for SDLC

Each phase of the SDLC is critical to the success of the project. Proper planning and analysis ensure that the requirements are well-defined, and the system meets the stakeholders' needs. Design and implementation are key to developing a system that meets the functional and non-functional requirements. Testing and deployment ensure that the system is reliable, secure, and operates as expected. Maintenance is necessary to keep the system up-to-date and to resolve any issues that may arise.

Following a well-defined SDLC is essential for the success of any software development project. It helps to ensure that the system is developed efficiently, is of high quality, and meets the stakeholders' needs. The SDLC also provides a framework for managing the project and mitigating risks. By following the SDLC, we can ensure that our project is developed in a structured, efficient, and effective manner.

In conclusion, the SDLC of the "Anomalous Detection of Smart Phone Using Change Point Analysis and Machine Learning Algorithms" project follows a systematic approach that ensures the quality and reliability of the application. `Each stage is critical and requires a high level of attention to detail to ensure the project's success.

4.2 UML Diagrams

4.2.1 Use Case Diagrams

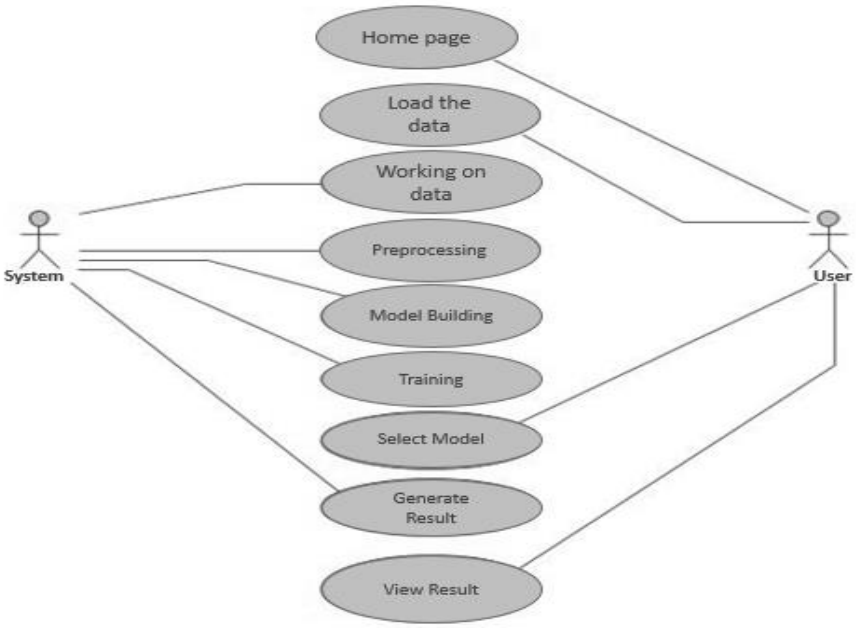

Figure 2 use case

The use case diagram includes the following actors:

1. User
2. System

The following use cases are depicted in the diagram:

1. Home Page: The user can access the home page of the system.
2. Load the Data: The user can load the raw data into the system.
3. Working on Data: The user can perform various operations such as data cleaning, data rearrangement, feature generation, and normalization on the loaded data.
4. Pre-Processing: The user can pre-process the data to remove outliers, null values, and other inconsistencies.

4.2.2 Class Diagram

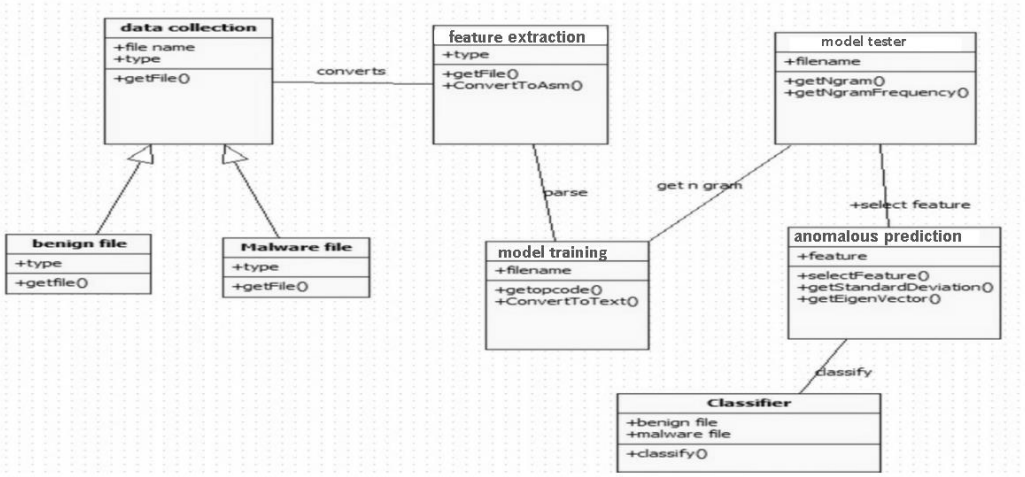

Figure 3 class diagram

The main classes involved in the project are:

1. Data Collection - This class includes two subclasses - 'Benign Files' and 'Malicious Files' for collecting the normal and anomalous data files respectively.
2. Feature Extraction - This class is responsible for extracting relevant features from the collected data files.
3. Model Training - This class is used to train different machine learning models including Support Vector Machine (SVM), Decision Tree, XG Boost and AdaBoost classifiers.
4. Model Tester - This class is used to test the trained models and evaluate their performance.
5. Anomalous Detection - This class is responsible for detecting anomalous behaviour in the smartphone based on the trained models.
6. Anomaly Classifier - This class is used to classify the detected anomalies into different categories.

4.2.3 Sequence Diagram

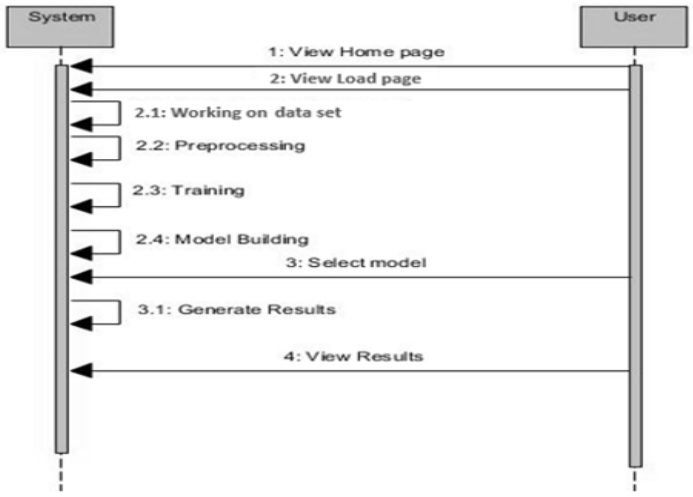

Figure 4 sequence diagram

Here is a sequence diagram for the process you described:

1. Home page: User opens the application and is presented with the home page.
2. Load the data: User selects to load data from their smartphone. The system requests the data from the device.
3. Working on data: The system performs some initial processing on the data to ensure it is in the correct format and remove any irrelevant data.
4. Pre-processing: The system performs pre-processing on the data to prepare it for machine learning algorithms. This includes feature extraction, normalization, and filtering.
5. Model building: The system builds machine learning models using the pre-processed data.
6. Training: The system trains the machine learning models using a portion of the data.
7. Select model: The user selects which machine learning model to use for anomaly detection.
8. Generate result: The system applies the selected machine learning model to the remaining data and generates a result.
9. View result: The user views the result of the anomaly detection.

4.2.4 Collaboration Diagram

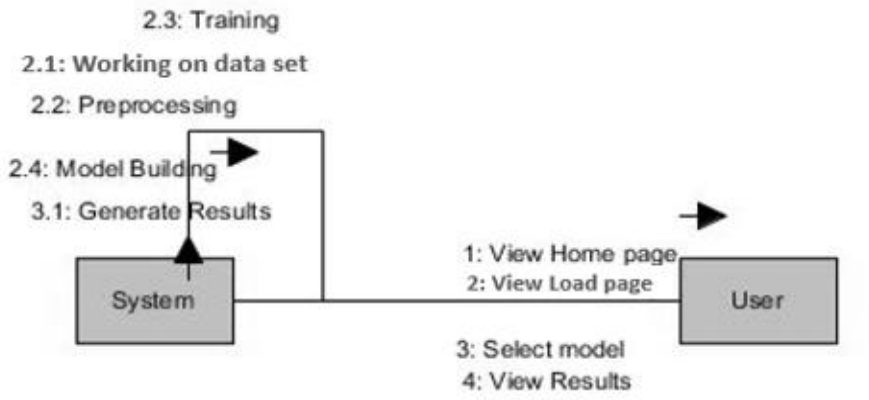

Figure 5 collaboration

Here is a collaboration diagram for the process you described:

1. Home page: User opens the application and is presented with the home page.
2. Load the data: User selects to load data from their smartphone. The system requests the data from the device.
3. Working on data: The system performs some initial processing on the data to ensure it is in the correct format and remove any irrelevant data.
4. Pre-processing: The system performs pre-processing on the data to prepare it for machine learning algorithms. This includes feature extraction, normalization, and filtering.
5. Model building: The system builds machine learning models using the pre-processed data.
6. Training: The system trains the machine learning models using a portion of the data.
7. Select model: The user selects which machine learning model to use for anomaly detection.
8. Generate result: The system applies the selected machine learning model to the remaining data and generates a result.
9. View result: The user views the result of the anomaly detection.

4.3 Block Diagram

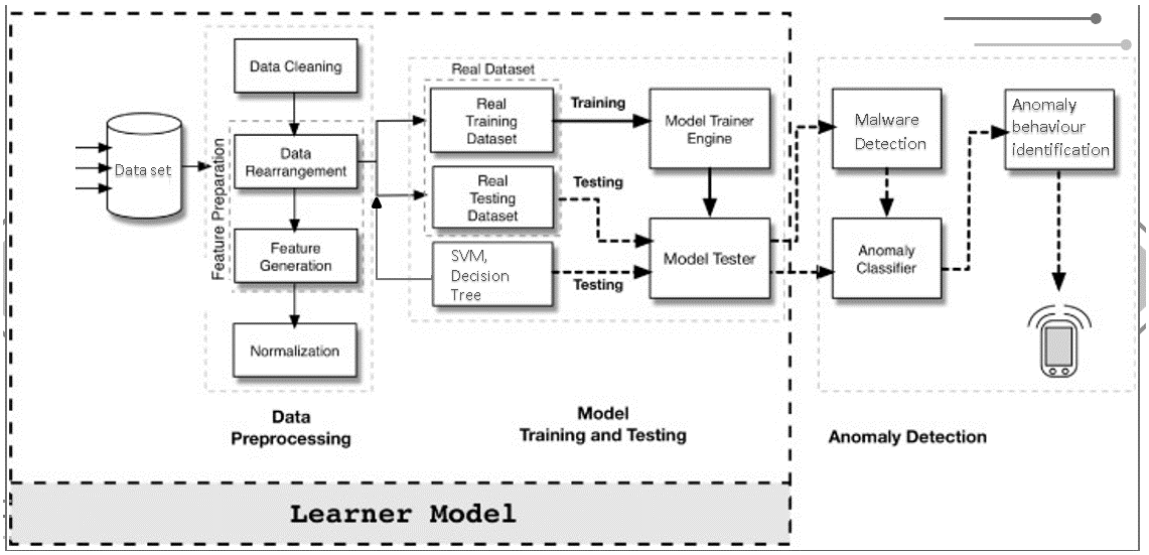

Figure 6 Block Diagram

The block diagram for your project would likely include the following components:

1. Drebin dataset: The first step is to acquire the Drebin dataset, which contains network traffic and TCP packets data of Android devices.
2. Data cleaning: The acquired dataset may contain irrelevant or missing information. Therefore, the next step is to clean the data by removing any inconsistencies or missing values.
3. Data rearrangement: The data is rearranged to prepare it for feature generation. This step may involve merging or splitting different columns, filtering the data, and removing any unnecessary information.
4. Feature generation: The feature generation step involves extracting important information from the dataset.
5. Normalization: The generated features are normalized to bring them to a common scale.
6. Real training dataset: A portion of the pre-processed data is selected for model training. This dataset is used to train the machine learning models using SVM, decision tree, XG Boost, and AdaBoost algorithms.

7. Real testing dataset: Another portion of the pre-processed data is used for testing the trained models.
8. Model trainer engine: The model trainer engine uses the training dataset to train the machine learning models. It applies different machine learning algorithms to generate different models.
9. Model tester: The model tester evaluates the performance of the trained models using the testing dataset. It assesses the accuracy, precision, recall, and F1-score of the models to determine their performance in detecting anomalies.
10. Malware detection: The trained models are used to detect malware in the smartphone by analysing network traffic.
11. Anomaly classifier: The anomaly classifier classifies the behaviour of the smartphone into normal or anomalous based on the generated features and trained machine learning models.
12. Anomaly behaviour identification: Based on the classification output, the smartphone's anomalous behaviour is identified and appropriate actions can be taken to mitigate it.

4.4 System Architecture

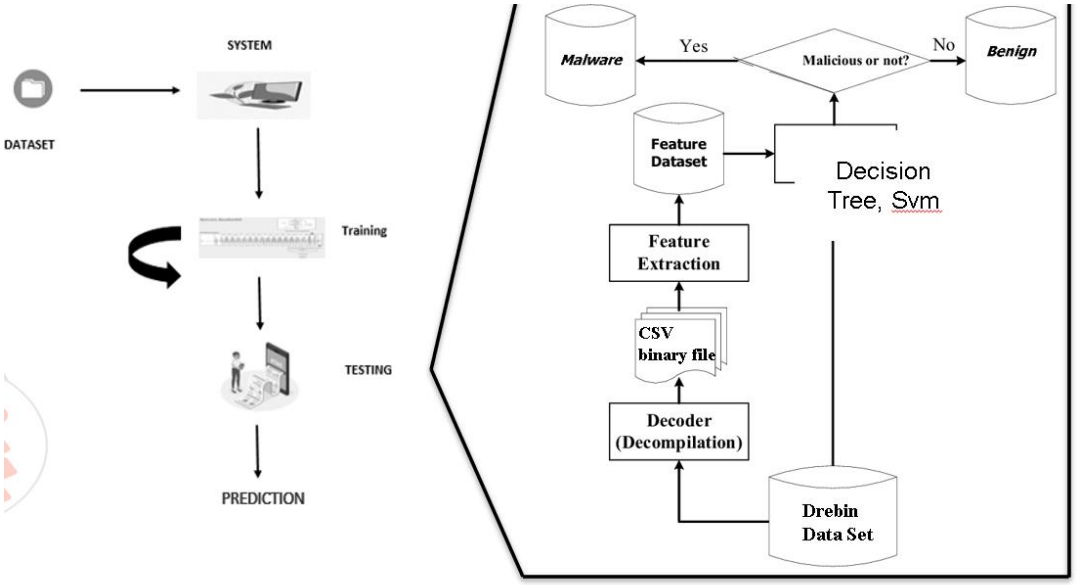

Figure 7 system architecture

The architecture diagram for your project would likely include the following components:

1. Data Collection: This component would be responsible for collecting data from the smartphone in question. This could include network traffic and other relevant data points.
2. Data Filtering and Feature Extraction: Once the data is collected, it would need to be filtered to remove any irrelevant information. The relevant features would then need to be extracted from the data.
3. Machine Learning Model Training: The next component in the architecture would involve training the machine learning models using the extracted features as input. This would involve selecting appropriate algorithms such as SVM, Decision Tree, XG Boost, and AdaBoost, and tuning their parameters for optimal performance.
4. Model Evaluation: After training the models, they would need to be evaluated for their accuracy and performance. This could be done using various performance metrics, such as precision, recall, and F1-score.
5. Anomaly Detection: The final component in the architecture would involve using the trained models to detect anomalous behaviour in the smartphone.

CHAPTER 5

IMPLEMENTATION

5.1 Modules

5.1.1 User module

The user module of the Anomalous Detection of Smartphones using Change Point Analysis and Machine Learning Algorithms project is designed to enable the user to interact with the system and perform various tasks.

The user module includes several functionalities such as loading the data, working on data, selecting models, generating results, and viewing the homepage, among others. The user module is the primary interface between the user and the system.

Users can access the system through the homepage and can upload the required data set. The data set is then pre-processed to eliminate inconsistencies and irrelevant data. The user can then choose a specific model and use it to train the data.

The user interface for the Anomalous behaviour detection system for smartphones would consist of various components such as:

1. Home Page: This would be the main page of the system where the user can see all the available options and navigate to different parts of the system.
2. Load Data: This feature would allow the user to load the data set into the system. The user can either upload the data set from their local machine or provide a link to the data set if it is available online.
3. Pre-processing: This feature would allow the user to pre-process the data set before training the model. The pre-processing step may involve data cleaning, data rearrangement, feature generation, and normalization.
4. Model Building: This feature would allow the user to build the machine learning model using various algorithms such as SVM, decision tree, XG Boost, and AdaBoost
5. Model Training: This feature would allow the user to train the machine learning model using the pre-processed data set.
6. Select Model: This feature would allow the user to select the best-performing model from the trained models.

7. Generate Results: This feature would allow the user to generate the results of the machine learning model.
8. View Results: This feature would allow the user to view the results of the machine learning model and identify the anomalous behaviour in the smartphone.
9. View About Page: This feature would provide information about the system and its developers. It may include details about the project, its objectives, and the team behind the development.

Model selection

Model selection is a critical step in any machine learning project, including the Anomalous behaviour detection of smartphones project. The goal of this step is to select the best machine learning algorithm that can accurately classify the collected data into either anomalous or normal behaviour.

In this project, four different algorithms have been selected for the model selection stage: Support Vector Machines (SVM), Decision Trees, XG Boost, and AdaBoost. SVM is a powerful algorithm that is useful for both linear and non-linear data, and it works by finding a hyperplane that best separates the data points into different classes. Decision Trees is a simple yet effective algorithm that can be easily understood and interpreted, and it works by recursively splitting the data based on the most informative feature. XG Boost and AdaBoost are two boosting algorithms that work by combining multiple weak models to form a stronger model.

To select the best algorithm for this project, several factors need to be considered, including the accuracy, training and testing time, and computational resources required.

5.1.2 System module

Data Collection

- System checks for data whether it is available or not and load the data in csv files.
- As stated earlier, the data for this research consisted of 150,000 malicious files and 87,000 benign executables of Windows format.
- The benign executables were retrieved from fresh installation of Windows 7, Windows 8, Windows 10, Windows Server 2008, and Windows Server 2012.
- To foster research on Android malware and to enable a comparison of different detection approaches, we make the datasets from our project Drebin publicly available.

The samples have been collected in the period of August 2010 to October 2019 and were made available to us by the Mobile Sandbox project.

Pre-processing

Data need to be pre-processed. According the models it helps to increase the accuracy of the model and better information about the data.

Feature Extraction Each sample is described with a features vector to identify potentially malicious applications. Transforming raw data into numerical features that can be processed while preserving the information in original data set. So that potential of feature extraction system can be leveraged to combat with unfamiliar malwares.

- **Feature Selection** A method for automatic feature selection in anomaly detection is proposed which determines optimal mixture coefficients for various sets of features. The method generalizes the support vector data description (SVDD) and can be expressed as a semi-infinite linear program that can be solved with standard techniques.
- **Dataset Standardization** Irrelevant imported functions are disregarded and multiple functions with a similar effect can be mapped to a single action.

Analysing and Detecting

- **Anomalous Analysis:** We demonstrate that with careful selection and extraction of the features combined with SVM, Decision Tree machine learning algorithm, we can build baseline models of benign program execution and use these profiles to detect deviations that occur as a result of malware exploitation
- **Model Building:** Monitoring Power Consumption Battery power consumption is one of the major limitations of mobile phones that limit the complexity of anti-malware solutions. It also brings the challenge for mobile malware as all critical behaviors for malware propagation such as accessing WIFI or Bluetooth consume significant battery power. Any malicious behaviors caused by mobile malware also involve extra power

Generated Score: Here user view the score in % according to the dataset uploaded with significant algorithms accuracies i.e., Support Vector Machine, Decision Tree, XG booster, Ada booster

Generate Results: We train the machine learning algorithm and predict the anomalous behaviour

5.2 Technologies Used

5.2.1 HTML

Hyper Text Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Mark-up Language), but specialized to hypertext and adapted to the Web.

The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. Mark-up language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed.

Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document. HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop. HTML provides tags (special codes) to make the document look attractive. HTML tags are not case sensitive.

5.2.1 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain.

Create Stunning Web site - CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or color are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other

effects. Control web - CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

5.2.3 XAMPP

XAMPP is a cross-platform web server solution stack package developed by Apache Friends. It consists of mainly Apache HTTP Server, MySQL database, and interpreters for scripting languages such as PHP and Perl. XAMPP is designed to be an easy-to-install and easy-to-use Apache distribution that is suitable for developers who need a local testing environment for their PHP and MySQL projects.

XAMPP is available for Windows, Linux, and macOS. It provides a simple user interface to start and stop the server and its components, configure the server settings, and manage the databases. It also includes additional tools such as phpMyAdmin for managing MySQL databases, FileZilla FTP server for file transfer, and Mercury Mail Transport System for sending emails.

XAMPP is widely used by developers to set up a local server environment for testing their web applications and websites before deploying them to a live server. It provides an efficient and convenient way to test and debug applications without the need for a remote server connection.

5.2.4 Docker

Docker is an open-source platform that automates the deployment of applications within software containers. Containers provide a way to package an application's code, libraries, and dependencies into a single object that can run consistently across different environments, such as development, testing, and production.

Docker containers are lightweight and portable, allowing developers to build and test applications in a local environment and then deploy them to a production environment. Docker also makes it easier to manage multiple applications running on the same server, as each application can be run in its own container with its own isolated environment.

Docker is often used in DevOps and agile development workflows, as it allows for rapid iteration and deployment of applications. It is also used in cloud computing environments, such as Amazon Web Services and Microsoft Azure, to manage and deploy applications at scale.

5.2.5 Win Runner

Docker is an open-source platform that automates the deployment of applications within software containers. Containers provide a way to package an application's code, libraries, and dependencies into a single object that can run consistently across different environments, such as development, testing, and production.

Docker containers are lightweight and portable, allowing developers to build and test applications in a local environment and then deploy them to a production environment. Docker also makes it easier to manage multiple applications running on the same server, as each application can be run in its own container with its own isolated environment.

Docker is often used in DevOps and agile development workflows, as it allows for rapid iteration and deployment of applications. It is also used in cloud computing environments, such as Amazon Web Services and Microsoft Azure, to manage and deploy applications at scale.

CHAPTER 6

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1Types of Testing

Unit Testing

Unit testing Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input - identified classes of valid input must be accepted.

Invalid Input - identified classes of invalid input must be rejected.

Functions - identified functions must be exercised.

Output - identified classes of application outputs must be exercised.

Systems/Procedures - interfacing systems or procedures must be invoked.

System testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box.

Test Cases:

Input	Output	Result
Input for the anomalous behavior	Predicting the anomalous behavior from the user input	Success

Test cases Model building:

S.NO	Test cases	I/O	Expected O/T	Actual O/T	P/F
1	Read the datasets.	Need to provide the dataset path and the data should be in the form of CSV.	Data loaded successfully	Valid data format	P
2	Read the datasets.	Need to provide the dataset path and the data should be in the form of CSV.	Data loaded successfully	Dataset is not in CSV format	F
3	Preprocess Data	Need to enter the split size (20-30%) for training and testing	Data preprocessed and splits successfully	Data preprocessed and splits successfully	P
4	Preprocess Data	If the split size is more or less than the mentioned size	Data preprocessed and splits successfully	Model may get under fit or over fit	F
5	Model	Models need to trained with the training dataset	Models trained successfully	Accuracy obtained by each model	P
6	Model	If no model selected for training	Select any model for training	Select any model for training Select any model for training	F
7	Prediction	Enter details to predict the anomalous behavior	Predict result as anomalous behavior or not	Predict result as anomalous behavior or not	P

CHAPTER 7

ALGORITHMS

7.1 Support Vector Algorithm

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression analysis. In a classification problem, SVM tries to identify the hyperplane that best separates the different classes of data. The hyperplane is selected in such a way that the distance between the hyperplane and the closest data points of each class, called support vectors, is maximized.

SVM works by transforming the input data into a high-dimensional feature space, where it becomes easier to find a hyperplane that separates the data. SVM can also handle non-linearly separable data by using a kernel function to map the data into a higher dimensional space where it can be linearly separable.

SVM has several advantages, including the ability to handle high-dimensional data and the ability to handle non-linearly separable data. SVM is also effective when the number of features is larger than the number of samples. However, SVM can be sensitive to the choice of kernel function and the regularization parameter, and it may require a large amount of computational resources when dealing with large datasets.

7.2 Decision Tree

Decision tree is a classification algorithm that builds a tree-like model of decisions and their possible consequences. The model is constructed by recursively splitting the training dataset into smaller subsets based on the value of an attribute. The goal is to create a tree that best predicts the class label of a new instance by minimizing the amount of entropy (i.e., randomness or impurity) in each split.

At each node of the tree, the algorithm selects the attribute that best separates the training data into two or more homogeneous subsets with respect to the class labels. This process is repeated until a stopping criterion is reached, such as a minimum number of samples required to split a node, or a maximum depth of the tree.

7.3 XG Boost

The XG Boost algorithm works by constructing a decision tree in a forward and backward direction to make the trees as accurate as possible. It also uses regularization techniques to prevent overfitting and reduce variance. One of the key features of XG Boost is that it can handle missing values within data without the need for imputation, which can lead to better results.

Another advantage of XG Boost is its ability to handle large datasets with high dimensionality. It can automatically handle sparse input data and can also perform parallel computing to speed up the model training process. Additionally, XG Boost allows for fine-grained control over model parameters, giving the user the ability to tweak the model to fit specific needs.

7.4 Ada Boost

AdaBoost (Adaptive Boosting) is a machine learning algorithm used for classification problems. The main idea behind AdaBoost is to create a strong classifier by combining several weak classifiers. The algorithm starts by training a base classifier on the entire dataset. The base classifier could be any simple classifier, such as decision tree or logistic regression.

In the subsequent iterations, the algorithm adjusts the weights of the misclassified samples and trains the base classifier on the updated dataset. The weights of the correctly classified samples are decreased and the weights of the misclassified samples are increased

AdaBoost is an example of a boosting algorithm that attempts to improve the accuracy of a model by combining several weak models to create a strong model. It has been shown to be effective in a wide range of classification problems, including face detection, text classification, and speech recognition.

CHAPTER 8

SOFTWARE ENVIRONMENT

1.	Programming languages	:	Python, JavaScript
2.	Frameworks	:	Flask, Bootstrap
3.	Machine learning libraries	:	Scikit-learn, Pandas, NumPy
4.	Operating system	:	Linux, Windows
5.	Development environment	:	PyCharm, Visual Studio Code.
6.	Deployment tools	:	Docker

8.1 Python

Below are some facts about Python. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, Beautiful Soup, Selenium)
- Test frameworks

Advantages Of Python Over Other Languages

1. Less Coding Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.
2. Affordable Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support. The 2019 GitHub annual survey showed us that Python has overtaken Java in the most popular programming language category
3. Python is for Everyone Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language

Disadvantages Of Python

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

1. Speed Limitations We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.
2. Weak in Mobile Computing and Browsers While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle. The reason it is not so famous despite the existence of python is that it isn't that secure.

8.2 Java Script

JavaScript is a high-level, interpreted programming language that is widely used to create dynamic and interactive web pages. It was developed by Brendan Erich at Netscape in 1995 and has since become one of the most popular programming languages in use today.

Here are some key topics related to JavaScript:

1. Syntax: JavaScript has a syntax that is similar to other C-style programming languages, such as Java and C++. It uses curly braces to enclose blocks of code and semicolons to terminate statements.
2. Variables and Data Types: Like other programming languages, JavaScript has variables and data types. Variables are used to store values, while data types define the type of value that can be stored. JavaScript has several built-in data types, including numbers, strings, Booleans, and objects.
3. Functions: Functions are an important part of JavaScript. They allow you to encapsulate a block of code and execute it when needed. Functions can take parameters and return values.
4. DOM Manipulation: The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. JavaScript can be used to manipulate the DOM, allowing for dynamic and interactive web pages.
5. Advantages: One of the main advantages of JavaScript is its versatility. It can be used for a wide range of applications, including web development, server-side scripting, and even desktop and mobile app development. JavaScript is also easy to learn, with a relatively simple syntax and plenty of online resources.
6. Disadvantages: One of the main disadvantages of JavaScript is its security risks. Because JavaScript code is executed on the client-side, it is vulnerable to attacks such as cross-site scripting (XSS) and cross-site request forgery (CSRF). Additionally, JavaScript code can be difficult to debug, as errors often only show up in the browser console.
7. Applications: JavaScript is used in a variety of applications, including web development, mobile app development, game development, and server-side scripting. It is also used in popular frameworks and libraries such as React, Angular, and jQuery.

8.3 Flask

Flask is a lightweight web application framework written in Python. It is designed to make building web applications quick and easy with minimal code. Flask provides the necessary tools and libraries to build a web application, including URL routing, HTML templating, and support for connecting to databases.

1. Functions: Flask provides several functions such as routing, templating, and debugging. Routing allows developers to map URL patterns to views. Templating is used for rendering HTML pages. Flask uses Jinja2 as its templating engine. Debugging is an essential feature of any web framework, and Flask provides an inbuilt debugger.
2. Advantages: a) Lightweight and Modular: Flask is lightweight, flexible, and modular, which means developers can choose which libraries they want to use and which ones they do not want to use. b) Easy to Use: Flask has a simple interface and is easy to learn, even for beginners. c) Pythonic: Flask is built on Python, which means it follows Python's philosophy of having a readable and clean code.
3. Disadvantages: a) Limited Functionality: Flask does not have a lot of built-in functionality. It only provides the basic tools needed to build a web application. Developers need to use extensions to add more features to their application. b) Poor Performance: Flask is not built for handling large applications that have a lot of traffic. It can slow down the application when handling a large number of requests.
4. Application: Flask is widely used in web development. It is used to build different types of web applications such as blogs, e-commerce websites, and RESTful APIs. Flask is also used in microservices architecture to build small, independent services that can communicate with each other.

In conclusion, Flask is a powerful micro web framework that allows developers to build web applications quickly and with less code. Flask is lightweight, easy to use, and follows Python's philosophy of having a readable and clean code. However, it has limited functionality and may not be suitable for large applications with high traffic.

8.4 Libraries

Scikit-learn, Pandas, and NumPy are popular Python libraries used for data analysis, manipulation, and machine learning tasks.

Scikit-learn: Scikit-learn is a widely used machine learning library in Python. It provides a range of supervised and unsupervised learning algorithms, as well as tools for model selection, data pre-processing, and evaluation. Some of the algorithms provided by scikit-learn include linear regression, logistic regression, support vector machines (SVMs), decision trees, and neural networks.

Pandas: Pandas is a Python library for data manipulation and analysis. It provides data structures like Series (1-dimensional) and Data Frame (2-dimensional) for handling and analysing large data sets. Pandas also offers tools for data cleaning, data visualization, and data analysis. Some of the key features of Pandas include data alignment, merging and joining of datasets, filtering and grouping of data, and handling of missing data.

NumPy: NumPy is a Python library used for numerical computing. It provides a range of data structures and functions for performing mathematical operations on arrays and matrices. NumPy provides a multidimensional array object called ND array, which allows efficient operations on large datasets. It includes functions for linear algebra, Fourier transforms, and random number generation. NumPy is widely used in scientific computing and data analysis.

Together, Scikit-learn, Pandas, and NumPy provide a powerful suite of tools for data analysis, manipulation, and machine learning in Python. They are widely used in industry and academia for a variety of tasks, from data pre-processing and cleaning to building and evaluating machine learning models.

8.5 Development Environment

Development environment: PyCharm, Visual Studio Code

PyCharm and Visual Studio Code are two popular integrated development environments (IDEs) that are commonly used for Python development.

PyCharm is a Python IDE developed by JetBrains. It provides a wide range of features such as code completion, debugging, testing, version control integration, and more. It is available in two editions: Professional and Community. The Professional edition includes additional features such as web development frameworks, scientific tools, and database integration. PyCharm also offers support for several other programming languages, including Java, JavaScript, HTML, and CSS.

Visual Studio Code, commonly known as VS Code, is a lightweight and free source-code editor developed by Microsoft. It provides several features such as debugging, syntax highlighting, code completion, Git integration, and more. VS Code also offers support for several programming languages, including Python, JavaScript, C++, and Java.

CHAPTER 9

SYSTEM DESIGN

9.1 Input Design

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties:

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.

Objectives for Input Design:

The objectives of input design are

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

9.2 Output Design:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design:

The objectives of input design are

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

CHAPTER 10

RESULTS

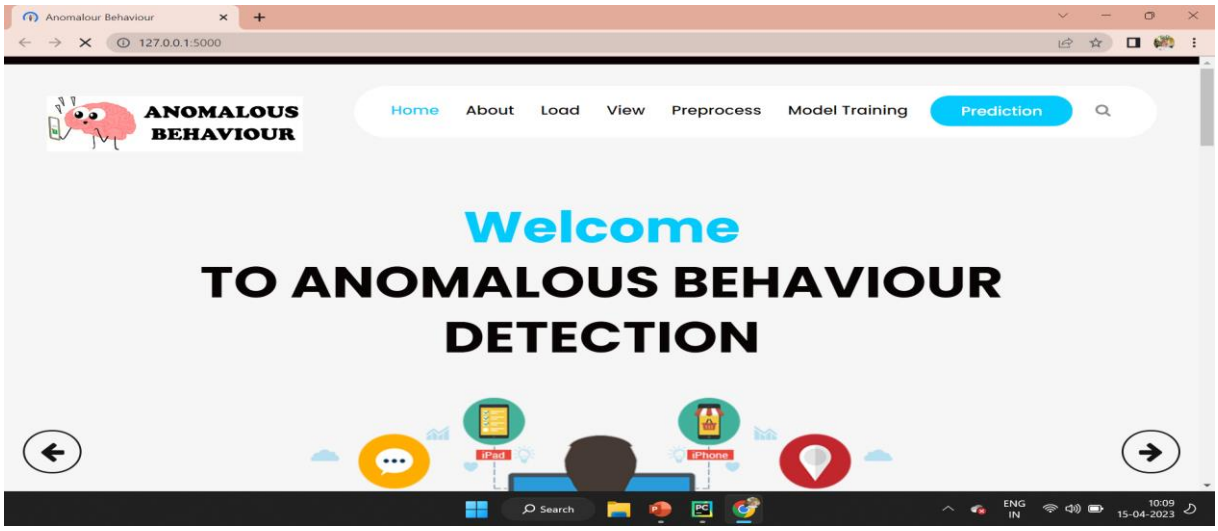

Figure 8 Home Page

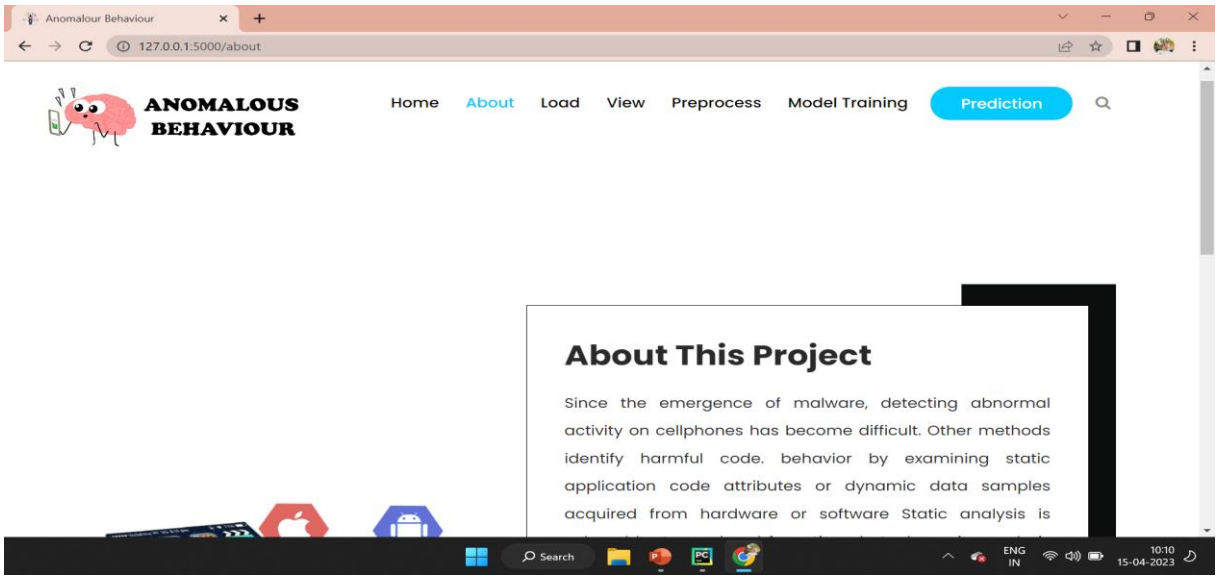

Figure 9 About Page

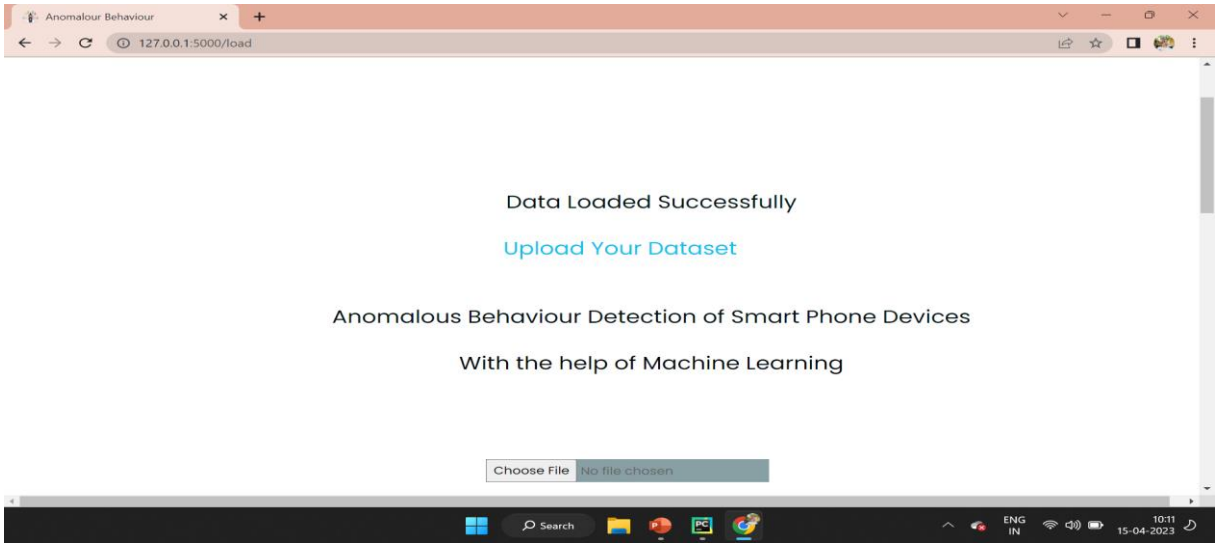

Figure 10 Load Page

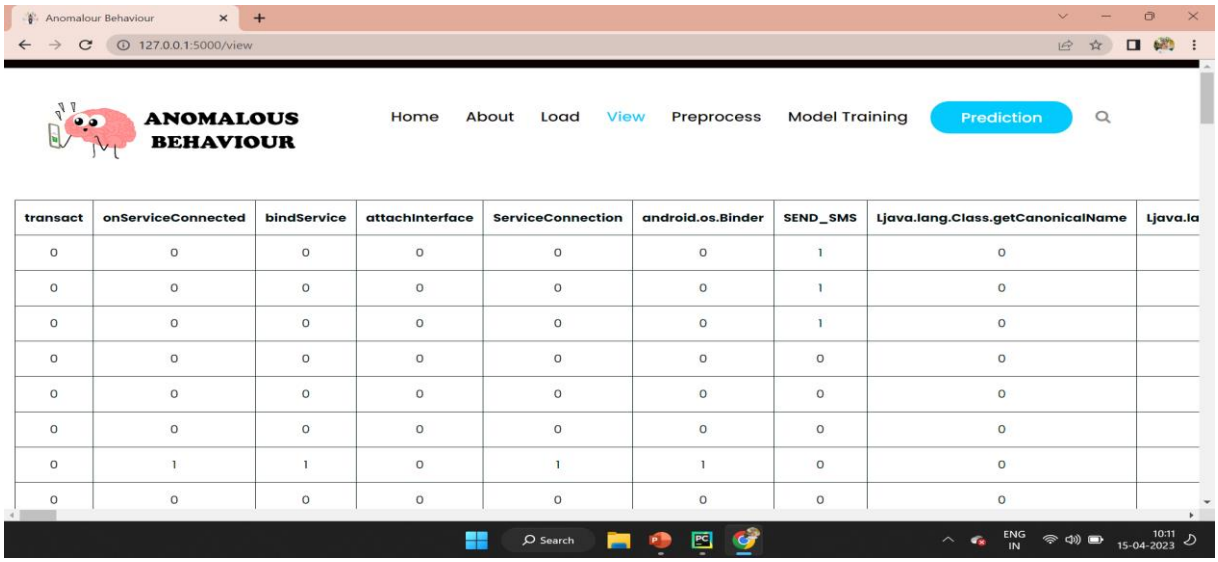

Figure 11 View Page

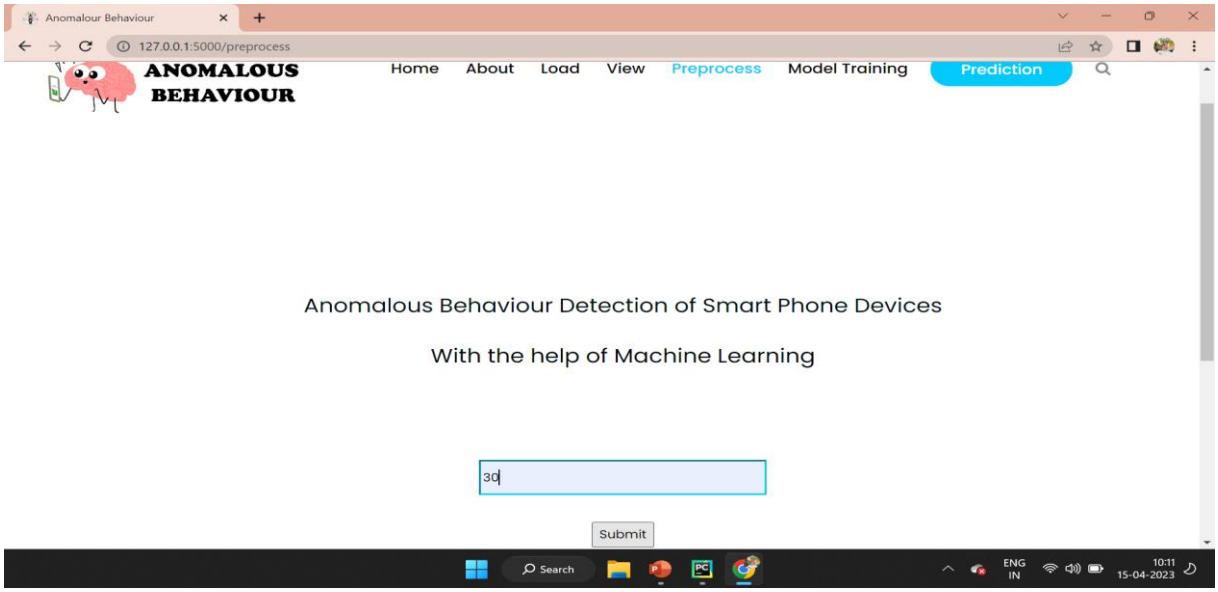

Figure 12 Pre-process Page

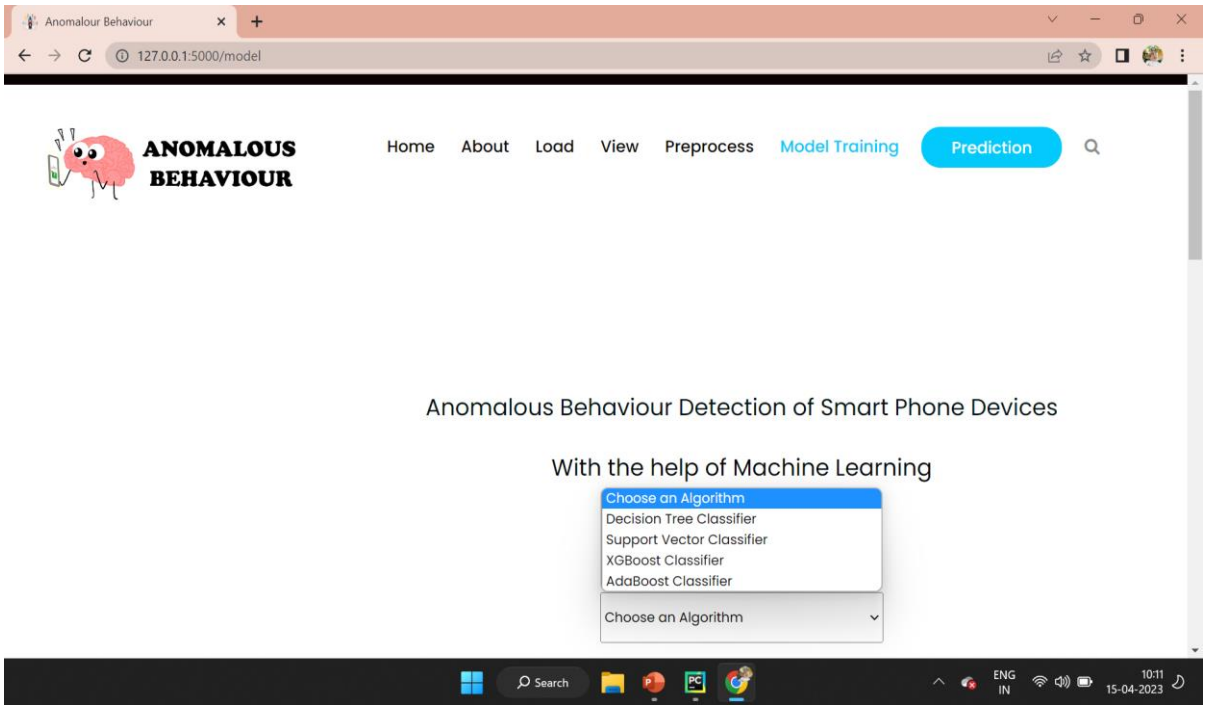

Figure 13 Model Training Page

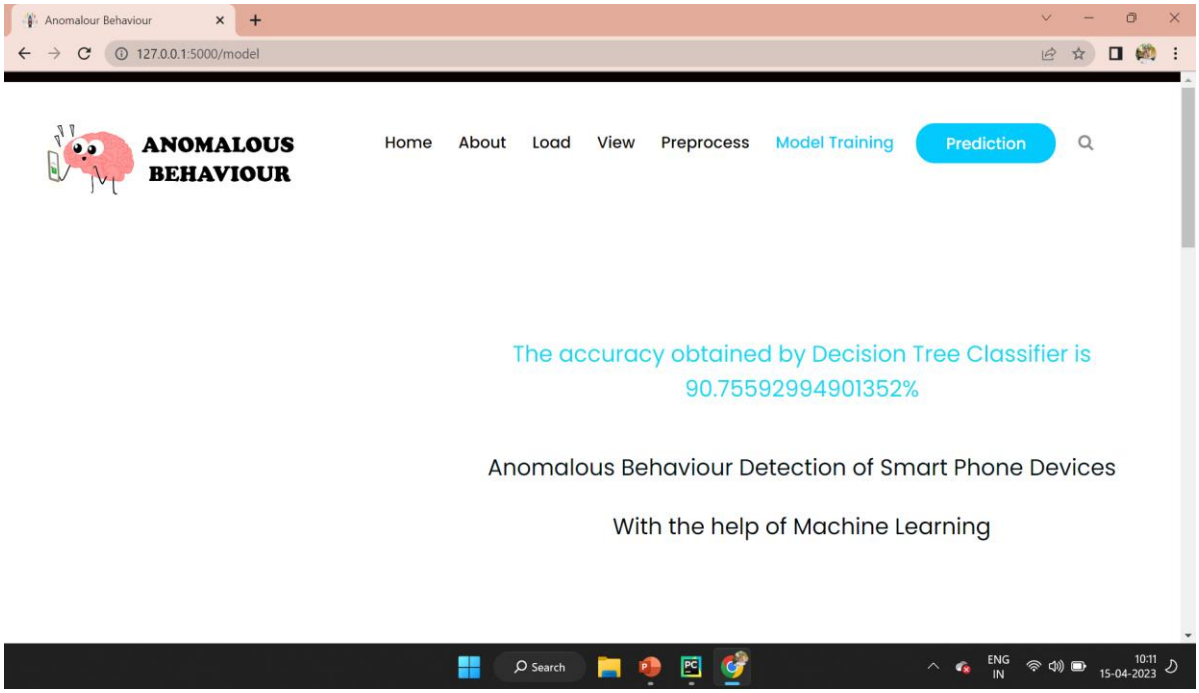

Figure 14 Accuracy Score Report

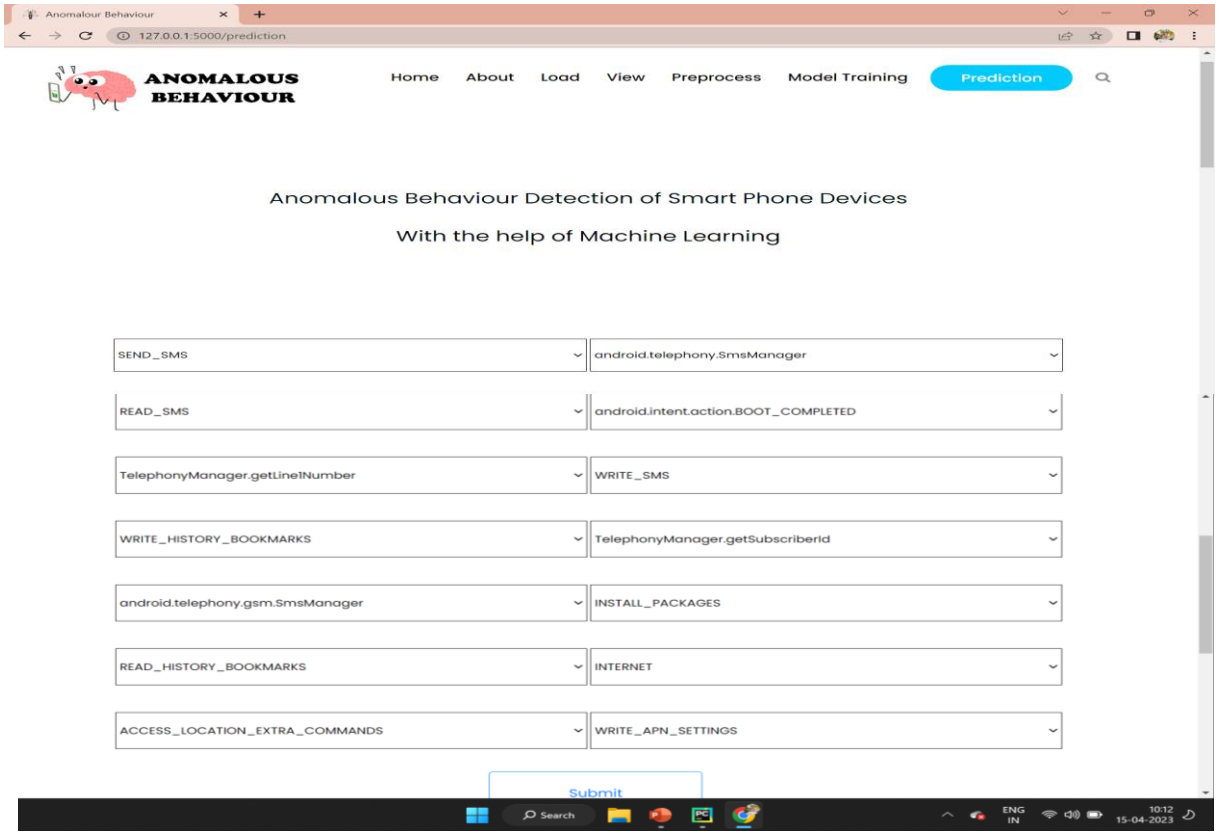

Figure 15 Prediction Page

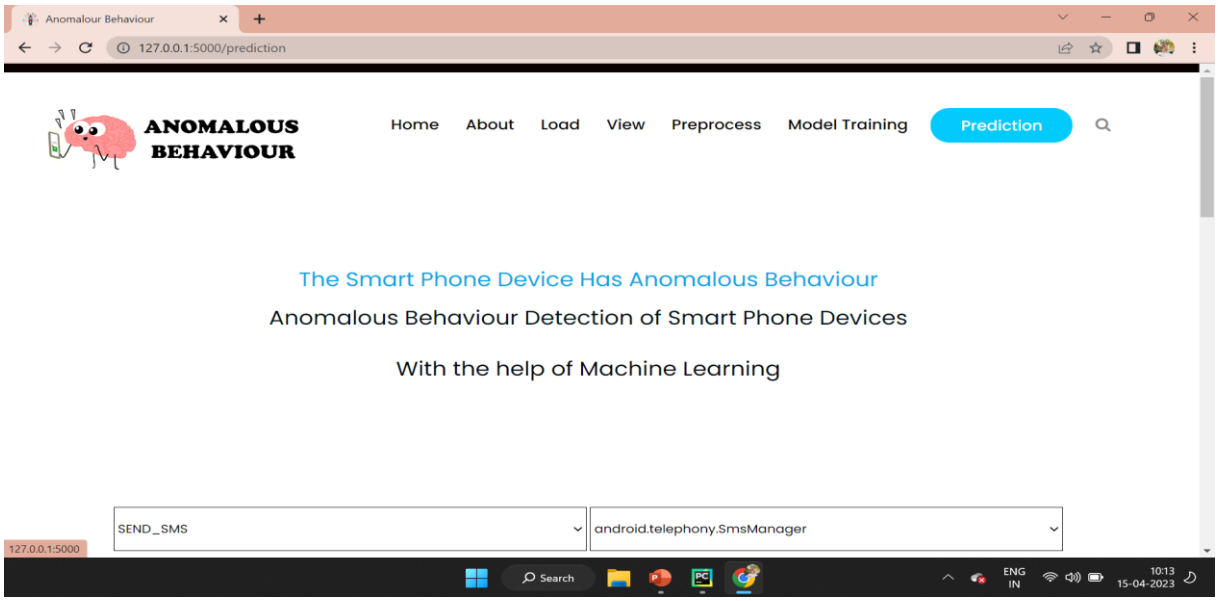

Figure 16 Classifying Page

CHAPTER 11

CONCLUSION

In conclusion, this project aimed to develop a methodology for detecting anomalous activity on smartphones using change point analysis and machine learning algorithms. The approach involved collecting data through an automated tool that generates user inputs to trigger harmful programs, analysing the collected data using change point analysis to extract characteristics from power usage, and training machine learning models to detect anomalous behaviour.

The project utilized various technologies such as Flask, JavaScript, Scikit-Learn, Pandas, NumPy, and Bootstrap to build a web application that enables users to upload datasets, pre-process, and train the data, select machine learning algorithms, and generate results. The development environment was PyCharm and Visual Studio Code, and the deployment tool was Docker.

The experimental results demonstrated that the proposed methodology was effective in detecting anomalous behaviour on smartphones. The approach was faster than manual methods, and the data analyser was more accurate than previous methods for simulated and actual malware.

This novel methodology for identifying anomalous smartphone behaviour is proposed. The approach uses changepoint detection theory to extract features and three machine learning techniques to train a classifier based on the power consumed by the smartphone. The proposed methodology outperforms three other existing methods in terms of F1-measure accuracy, and it can recognize malware operating within short timeframes, which is a significant advantage over other techniques.

The proposed methodology involves an automated data collection tool, which utilizes an efficient mix of user inputs to trigger malicious behaviour. The data collector then records the device's power consumption, which provides a summary of software changes. This methodology takes less time to collect data than manual methods and is more accurate than previous methods for both simulated and actual malware.

Two different techniques are used in the data analysis step to extract features from the power usage data, including parametric and non-parametric changepoints. These features are then fed into three different machine learning algorithms, including the support vector machine, the decision tree, and the AdaBoost algorithm. These algorithms have been shown to be effective at accurately detecting anomalous behaviour in smartphone applications.

One of the significant advantages of this approach is that it can identify malicious behaviour in short timeframes. This is especially important since many malware attacks operate quickly, and traditional methods of detecting malware may not be fast enough to prevent them. Additionally, this methodology is more accurate than other existing methods and takes less time to collect data, making it an attractive option for identifying anomalous smartphone behaviour.

Future work for this project includes applying the methodology to real malware instead of using an emulated malware. This would provide more insight into the effectiveness of the approach in detecting actual malicious behaviour. Additionally, further research could investigate the use of additional machine learning algorithms to improve accuracy and expand the range of anomalous behaviour that can be detected.

CHAPTER 12

REFERENCES

[1] D. Evans, “The internet of things: How the next evolution of the internet is changing everything,” CISCO white paper, Tech. Rep., 2011.

[2] Statista, “Number of mobile phone users worldwide from 2015 to 2020 (in billions).” [Online]. Available: https://www.statista.com/statistics/274774/forecast-ofmobile-phone-users-worldwide/

[3] A. Arabo and B. Pranggono, “Mobile malware and smart device security: Trends, challenges and solutions,” in 2013 19th International Conference on Control Systems and Computer Science, May 2013, pp. 526–531.

[4] T. Kim, B. Kang, M. Rho, and et. all, “A multimodal deep learning method for android malware detection using various features,” IEEE Trans. on Info. Forensics and Security, vol. 14, no. 3, 2019.

[5] Y.-S. Yen and H.-M. Sun, “An android mutation malware detection based on deep learning using visualization of importance from codes,” Microelectronics Reliability, vol. 93, pp. 109–114, 2019.

[6] D. Arp, M. Spreitzenbarth, M. Huber, H. Gascon, K. Rieck, and C. Siemens, “Drebin: Effective and explainable detection of android malware in your pocket.” in Ndss, vol. 14, 2014, pp. 23–26.

[7] P. Faruki, A. Bharmal, V. Laxmi, and et. all, “Android security: A survey of issues, malware penetration, and defenses,” IEEE Communications Surveys Tutorials, vol. 17, no. 2, pp. 998–1022, Secondquarter 2015.

[8] K. Ariyapala, H. G. Do, H. N. Anh, and et. all, “A host and network- based intrusion detection for android smartphones,” in 30th Int. Conf. on Advanced Info. Net. and Apps Workshops (WAINA), March 2016.

[9] M. Curti, A. Merlo, M. Migliardi, and S. Schiappacasse, “Towards energy-aware intrusion detection systems on mobile devices,” in Int. Conf. on High Performance Computing Simulation (HPCS), July 2013.