A PROJECT REPORT ON

"An Efficient Spam Detection Technique for IoT Devices using Machine Learning"

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE
OF

BACHELOR OF ENGINEERING
IN
INFORMATION TECHNOLOGY

BY

Nikhil Bhor	Exam No:
Gaurav Bhise	Exam No:
Jagdish Bhagwat	Exam No:
Tanvi Alkute	Exam No:

Under the Guidance of

Prof. Gayke Mam

DEPARTMENT OF INFORMATION TECHNOLOGY

Dr. Vithalrao Vikhe Patil College of Engineering, Ahmednagar, Ahmednagar-11

2022-2023

CERTIFICATE

This is to certify that the project report entitles

"An Efficient Spam Detection Technique for IoT Devices using Machine Learning"

Submitted by

Nikhil Bhor	Exam No:
Gaurav Bhise	Exam No:
Jagdish Bhagwat	Exam No:
Tanvi Alkute	Exam No:

is a bonafide work carried out by them under the supervision of **Prof. Gayke Mam** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of Bachelor of Engineering (Information Technology).

This project report has not been earlier submitted to any other Institute or University for the award of any degree or diploma.

Prof. Gayke Mam
Internal Guide
Department of IT

Dr. Dipak Vidhate
Head of Dept.
Department of IT

Prof. ABC
External Examiner
Date:-

Dr. Uday Naik
Principal
PVP College, Ahmednagar

Acknowledgments

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to ***Prof. Gayke Mam*** *for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.*

We would like to express our gratitude towards my parents & member of the Information Technology Department of Dr. Vithalrao Vikhe Patil College of Engineering, Ahmednagar for their kind cooperation and encouragement which help us in the completion of this project.
We would like to express our special gratitude and thanks to All other Professors in the Department for giving us such attention and time.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

Last but not the least, we are grateful to all our friends and our parents for their direct or indirect constant moral support throughout the course of this project.

Nikhil Bhor
Gaurav Bhise
Jagdish Bhagwat
Tanvi Alkute
(B.E. IT)

Abstract

The Internet of Things (IoT) is a group of millions of devices having sensors and actuators linked over wired or wireless channel for data transmission. IoT has grown rapidly over the past decade with more than 25 billion devices are expected to be connected by 2020. The volume of data released from these devices will increase many-fold in the years to come. In addition to an increased volume, the IoT devices produces a large amount of data with a number of different modalities having varying data quality defined by its speed in terms of time and position dependency. In such an environment, machine learning algorithms can play an important role in ensuring security and authorization based on biotechnology, anomalous detection to improve the usability and security of IoT systems. On the other hand, attackers often view learning algorithms to exploit the vulnerabilities in smart IoT-based systems. Motivated from these, in this paper, we propose the security of the IoT devices by detecting spam using machine learning. Here we collect a dataset from Kaggle and perform an operation on it and finally predict the result. To achieve this objective, Spam Detection in IoT using Machine Learning framework is proposed. In this framework, machine learning models are evaluated using various metrics with a large collection of inputs features sets. Each model computes a spam score by considering the refined input features. This score depicts the trustworthiness of IoT device under various parameters. The results obtained proves the effectiveness of the proposed scheme in comparison to the other existing schemes.

Keywords:- Internet of Things (IoT) Technology, Machine Learning Algorithms, Spam Detection, Dataset, Security and Authorization, etc.

INDEX

List of Figures

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

IoT is taken into account as an interconnected and distributed network of embedded systems communicating through wired or wireless communication technologies. Massive growth and rapid development in the field of the Internet of Things (IoT), makes the presence of IoT devices prevalent in smart homes and smart cities. It is also defined because the network of physical objects or things empowered with limited computation, storage, and communication capabilities is also embedded with electronics (such as sensors and actuators), software, and network connectivity that permits these objects to gather, sometimes process, and exchange data. The things in IoT ask the objects from our lifestyle starting from smart household devices like a smart bulb, smart adapter, smart meter, smart refrigerator, smart oven, AC, temperature sensor, smoke detector, IP camera, to more sophisticated devices like frequency Identification (RFID) devices, heartbeat detectors, accelerometers, sensors in the parking zone, and a variety of other sensors in automobiles, etc. we collect dataset from Kaggle and perform an operation on it and finally predict the result.

There are various large amounts of applications and services offered by the IoT ranging from critical infrastructure to agriculture, military, home appliances, and personal health care. As the usage of IoT devices increases the anomalies generated by these devices also grow beyond the count. IoT applications need to ensure information protection to fix security issues like interruptions, spoofing attacks, Dos attacks, jamming, eavesdropping, spam, and malware. The maximum care to be taken is with web-based devices as the maximum number of IoT devices are web-dependent. It is common in the work environment that the IoT devices introduced in an association can be utilized to execute security and protection includes proficiently. For example, wearable devices that collect and send user's health data to a connected smartphone should prevent leakage of data to ensure privacy. It has been found in the market that 25-30% of working employees connect their Personal IoT devices with the organizational network. The expanding nature of IoT attracts both the audience, i.e., the users and therefore the attackers.

However, with the emergence of ML in various attack scenarios, IoT devices choose a defensive strategy and decide the key parameters in the security protocols for a trade-off between security, privacy, and computation. This work enhances the algorithm to affect the time-series regression model rather than a classification model and may also execute ML models in parallel. This proposed paper focuses on determining the trustworthiness of the IoT device within the smart home network. The algorithm scores an IoT device with a spamicity score to secure

smart devices by calculating spam scores using different machine learning models.

1.2 PURPOSE

As part of the recommended approach, the spammy characteristics are detected. ML models are used in Internet of things. This is the IoT data. it is pre-processed with the aid of pattern development method. By playing around with the structure, each IoT device is rewarded with ML models. T he amount of spamming that has been detected As a result, the criteria for success have been refined. IoT equipment operating in a smart house As we go forward, will take into account meteorological conditions as well as the environment IoT devices more secured and reliable.

1.3 MOTIVATION

The use of new innovations give incredible advantages to people, organizations, and governments, be that as it may, messes some up against them. For instance, the protection of significant data, security of put away information stages, accessibility of information and so forth. Contingent upon these issues, digital fear-based oppression is one of the most significant issues in this day and age. Digital fear, which made a great deal of issues people and establishments, has arrived at a level that could undermine open and nation security by different gatherings, for example, criminal association, proficient people and digital activists. Along these lines, Intrusion Detection Systems (IDS) has been created to maintain a strategic distance from digital assaults.

1.4 OBJECTIVE

- The proposed scheme of spam detection is validated using five different machine learning models.
- An algorithm is proposed to compute the spamicity score of each model which is then used for detection and intelligent decision making.
- Based upon the spamicity score computed in previous step, the reliability of IoT devices is analyzed using different evaluation metrics.
- To detect network attacks by applying machine learning methods.

- To reduce operational time.
- To increase accuracy and reliability.
- To increase operational efficiency.
- To provide data security.

1.5 PROBLEM STATEMENT

To address the aforementioned challenges, we proposed a novel algorithm and build an web-based application for the detection of spam from different IoT devices dataset. In Proposed studies shows that the problem definition gets more specific for any attack type and includes an expanded definition of the attack and its behavior. Further, we confirmed that the performance of neural network increases with increase in accuracy and performance of algorithms.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

1. Dr. Aaisha Makkar, Dr. Neeraj Kumar, Prof. Ahmed Ghoneim, "An Efficient Spam Detection Technique for IoT Devices using Machine Learning". The Internet of Things (IoT) is a group of millions of devices having sensors and actuators linked over wired or wireless channel for data transmission. IoT has grown rapidly over the past decade with more than 25 billion devices are expected to be connected by 2020. The volume of data released from these devices will increase many-fold in the years to come. In addition to an increased volume, the IoT devices produces a large amount of data with a number of different modalities having varying data quality defined by its speed in terms of time and position dependency. In such an environment, machine learning algorithms can play an important role in ensuring security and authorization based on biotechnology, anomalous detection to improve the usability and security of IoT systems. On the other hand, attackers often view learning algorithms to exploit the vulnerabilities in smart IoT-based systems. Motivated from these, in this paper, we propose the security of the IoT devices by detecting spam using machine learning. To achieve this objective, Spam Detection in IoT using Machine Learning framework is proposed. In this framework, five machine learning models are evaluated using various metrics with a large collection of inputs features sets. Each model computes a spam score by considering the refined input features. This score depicts the trustworthiness of IoT device under various parameters. REFIT Smart Home dataset is used for the validation of proposed technique. The results obtained proves the effectiveness of the proposed scheme in comparison to the other existing schemes.

2. Nutjahan, Farhana Nizam, Shudarshon Chaki, Shamim Al Mamun, M. Shamim Kaiser," Attack Detection and Prevention in the Cyber Physical System". [2016] [1] In this paper proposes Cyber Physical System cyber-attack detection and prevention To detect distributed denial of service and false data injection attacks, the Chi square detector and Fuzzy logic based attack classifier (FLAC) were utilised. Activity profiling, average packet rate, change point detection algorithm, cusum algorithm, unexpired user sessions, injected incomplete information, and reuse of session key are some of the fuzzy features used to choose the attacks described. An example scenario has been created using OpNET Simulator. Simulation results depict that the use of Chi-square detector and FLAC

are able to detect the mentioned cyber physical attacks with high accuracy. Compared to existing Fuzzy logic based attack detector, the proposed model outperforms the traditional distributed denial of service and false data detector.

3. Yong Fang, Cheng Huang, Yijia Xu and Yang Li, "RLXSS: Optimizing XSS Detection Model to Defend Against Adversarial Attacks Based on Reinforcement Learning". [2019] [2]. In this research, we introduce RLXSS, a reinforcement learning-based strategy for optimising the XSS detection model to defend against adversarial attacks. First, the adversarial samples of the detection model are mined by the adversarial attack model based on reinforcement learning. Secondly, the detection model and the adversarial model are alternately trained. After each round, the newly-excavated adversarial samples are marked as a malicious sample and are used to retrain the detection model. The proposed RLXSS model successfully mines adversarial samples that avoid black-box and white-box detection while retaining aggressive features, according to experimental data. Furthermore, by alternating training the detection model and the confronting assault model, the detection model's escape rate is continuously reduced, indicating that the model can increase the detection model's ability to defend against attacks.

4. Rishikesh Mahajan, Irfan Siddavatam, "Phishing Website Detection using Machine Learning Algorithms". [2018] [3] Phishing is the most basic method of obtaining sensitive information from unsuspecting consumers. The goal of phishers is to obtain sensitive information such as usernames, passwords, and bank account information. Cyber security professionals are now looking for dependable and consistent detection solutions for phishing websites. The purpose of this work is to discuss machine learning technology for detecting phishing URLs by extracting and analysing various aspects of authentic and phishing URLs. To detect phishing websites, the Decision Tree, Random Forest, and Support Vector Machine algorithms are used. The goal of this study is to detect phishing URLs as well as to narrow down the best machine learning method by analysing each algorithm's accuracy rate, false positive and false negative rate.

5. Vishnu. B. A, Ms. Jevitha. K. P., “Prediction of Cross-Site Scripting Attack Using Machine Learning Algorithms”. [2018] [4] Cross-site scripting (XSS) is one of the most frequently occurring types of attacks on web applications, hence is of importance in information security. XSS occurs when an attacker injects malicious code, usually JavaScript, into a web application such that it can be executed in the user’s browser. Detecting malicious scripts is an important aspect of an online application’s defence. This study studies the use of SVM, k-NN, and Random Forests to detect and limit known and undiscovered assaults on JavaScript code by developing classifiers. It shown that using an interesting feature set that combines language syntax and behavioural information resulted in classifiers that provide excellent accuracy and precision on huge real-world data sets without focusing solely on obfuscation.

6. Zohre Nasiri Zarandi, Iman Sharif, “Detection and Identification of Cyber-Attacks in Cyber- Physical Systems Based on Machine Learning Methods”. [2020] [5] The CPS is modelled in this study as a network of agents that move in unison with one another, with one agent acting as a leader and the other agents being ordered by the leader. In this study, the proposed strategy is to employ the structure of deep neural networks for the detection phase, which should tell the system of the existence of the attack in the early stages of the attack. In the leader-follower mechanism, the employment of robust control algorithms in the network to isolate the misbehaving agent has been examined. In the presented control method, after the attack detection phase with the use of a deep neural network, the control system uses the reputation algorithm to isolate the misbehave agent. Experiments reveal that deep learning algorithms outperform traditional approaches in detecting assaults, making cyber security simpler, more proactive, less expensive, and considerably more successful.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 INTRODUCTION

3.1.1 Project Scope

The scope of this project is to develop and implement an efficient spam detection technique specifically designed for IoT devices using machine learning algorithms. The project aims to address the growing concern of spam messages and malicious activities targeting IoT devices, which can compromise the security and functionality of these interconnected devices.

3.1.2 User Classes and Characteristics

To design products that satisfy their target users, a deeper understanding is needed of their user characteristics and product properties in development related to unexpected problems users face. These user characteristics en- compass cognitive aspect, personality, demographics, and use behavior. The product properties represent operational transparency, interaction density, product importance, frequency of use and so on. This study focuses on how user characteristics and product properties can influence whether soft us- ability problems occur, and if so, which types. The study will lead to an interaction model that provides an overview of the interaction between user characteristics, product properties, and soft usability problems.

3.1.3 Assumption and Dependencies

- End User application will be developed in Windows OS.
- Database will be in MySQL.
- All application code shall conform to the python standard.
- All scripts shall be written in JavaScript.
- Application design pattern shall be Singleton.
- Developer IDE shall be Eclipse Oxygen IDE.

3.2 FUNCTIONAL REQUIREMENTS

- System must validate the previous block before commit block.
- User can access the data over the internet 24*7.
- If any block has changed by third party attacker or unauthorized user, it must show during transaction current blockchain is invalid.
- It can recover the invalid blockchain using other data nodes, with the help of majority of trustiness.

3.3 EXTERNAL INTERFACE REQUIREMENTS

3.3.1 User Interface

- User interface of this program is the common windows interface, nothing additional is required.
- The system user interface should be intuitive, such that 99.9% of all new system users are able to use system application without any assistance.

3.3.2 Hardware Interface

The hardware should have following specifications:

- Ability to read gallery
- Ability to exchange data over network
- Touch screen for convenience
- Keypad (in case touchpad not available)
- Continuous power supply
- Ability to connect to network
- Ability to take input from user
- Ability to validate user

3.3.3 Software Interface

- The software interfaces are specific to the target other user's proposed Application or software systems.

3.3.4 Communication Interface

- Our Project belongs to web based, so connecting user at online with request and response form. For that HTTP protocol we are going to use. That is provided by tomcat server 7/8.
- HTTP protocol: The Hypertext Transfer protocol is an application protocol for distributed, collaborative, and hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text.

3.4 NON-FUNCTIONAL REQUIREMENTS

- System should be robustness
- The time required for processing the application will be less.
- The application must be scalable and reliable.
- Time saving i.e. time to generate detectors must be as minimum as possible.
- Accurate i.e. the accuracy of the anomaly detection system should be good as compared to proposed systems.

3.4.1 Performance Requirement

- System can produce results faster on 4GB of RAM.
- It may take more time for peak loads at main node.
- The system will be available 100% of the time. Once there is a fatal error, the system will provide understandable feed back to the user.

3.4.2 Safety Requirements

- Only administrators have access to the database of each individual user.
- All data will be backed-up every day automatically and also the system administrator can back-up the data as a function for him.
- This makes it easier to install and updates new functionality if required.
- For the safety purpose backup of the database must be required.

3.4.3 Security Requirements

- Our System is being developed in Java. Java is an platform independant, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Java's design philosophy emphasizes code readability with its notable use of significant whitespace.
  - At the time of deploying this software user have to register to system.
  - To use software user have to login and logout each time.

3.4.4 Software Quality Attributes

- **Runtime System Qualities:** Runtime System Qualities can be measured as the system executes.
- **Functionality:** The ability of the system to do the work for which it was intended.
- **Performance:** The response time, utilization, and throughput behavior of the system. Not to be confused with human performance or system delivery time.
- **Security:** A measure of systems ability to resist unauthorized attempts at usage or behavior modification, while still providing service to legitimate users.
- **Usability:** The ease of use and of training the end users of the system.
- **Sub qualities:** learn ability, efficiency, affect, helpfulness, control.
- **Interoperability:** The ability of two or more systems to cooperate at runtime.

3.5 SYSTEM REQUIREMENTS

3.5.1 Database Requirements

- **MySQL :**

  MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

  The MySQL Web site (http://www.mysql.com/) provides the latest information about MySQL software.

  MySQL is database software which is used to store all database related activities regarding to our project and it is easily stored and retrieve the data.

3.5.2 Software Requirements

- Operating system: Windows XP/7 Higher
- Programming Language: JAVA/J2EE
- Tools: Eclipse, Heidi SQL, JDK 1.8 or Higher
- Database: MySQL 5.1

3.6 HARDWARE REQUIREMENTS

- System : Pentium IV 2.4 GHz.
- Hard Disk : 240 GB (Min)
- Monitor : 15 VGA Colour.
- IO Devices : Mouse, Keyboard
- RAM : 4 GB (Min)

3.7 ANALYSIS MODEL

We are using waterfall model for our project:

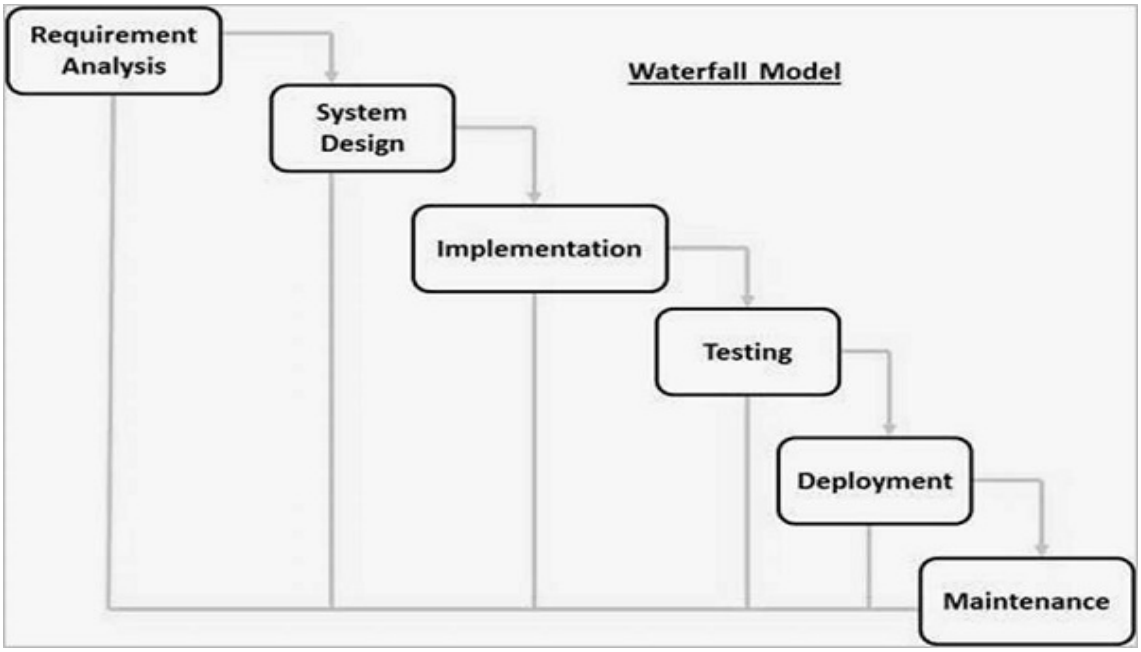

Figure 3.1: Waterfall Model

- **Requirement gathering and analysis:**

  In this step of waterfall we identify what are various requirements are need for our project such are software and hardware required, database, and interfaces.

- **System Design:**

  In this system design phase we design the system which is easily understood for end user i.e. user friendly. We design some UML diagrams and data flow diagram to understand the system flow and system module and sequence of execution.

- **Implementation:**

  In implementation phase of our project we have implemented various module required of successfully getting expected outcome at the different module levels.

  With inputs from 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.

- **Testing:**

  The different test cases are performed to test whether the project module are giving expected outcome in assumed time. 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.

- **Deployments 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.

  All these phases are cascaded to each other in which progress is seen as flowing steadily downwards like a waterfall through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name ”Waterfall Model”. In this model phases do not overlap.

3.8 SYSTEM IMPLEMENTATION PLAN

- **PERT Chart/ Gantt chart:**

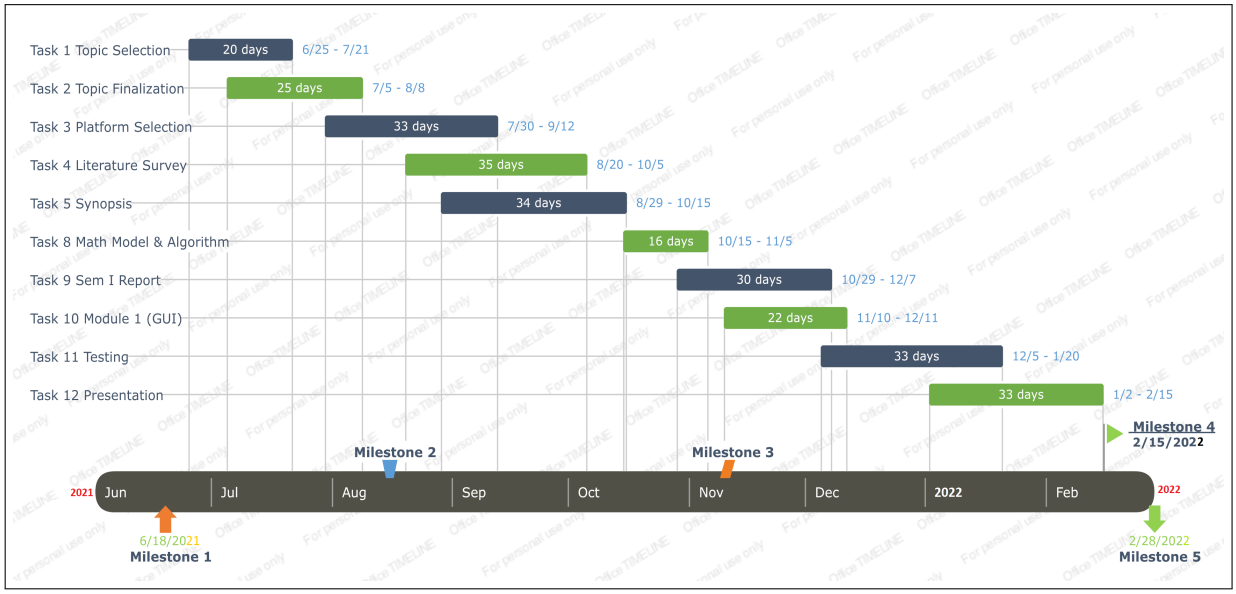

Figure 3.2: PERT Chart/ Gantt chart

- **Project Estimate**

  Estimating is figuring out how much time and money is required for completion. Usually developer will not really know the requirement. Usually management will have a number in mind when they ask for your estimate.

- **Cost Estimation**

The project cost can be found using any one of the model.

COCOMO-1 Model

COCOMO-2 Model

Model -1: The basic COCOMO model computes software development efforts as a function of program size expressed in estimated lines of code.

Model-2: The intermediate COCOMO model computes software development efforts as a function of program size and a set of cost drivers that include subjective assessment of the product, hardware, personnel, project attributes

Model-3: The advanced COCOMO model incorporates all characteristics of the intermediate version with a assessment of the cost drivers impact on each step of the software engineering process. Following is the basic COCOMO -2nd model.

Software Project	A(b)	B(b)	C(b)	D(b)
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.22	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Figure 3.3: COCOMO-2nd model

The basic COCOMO -2 model equations take form:

$E=A(b)KLOCB(b)$

$D=C(b)ED(b)$

Where E is the effort applied in person months. D is development time in chronological month. KLOC is estimated number of delivered lines of code for the project. This project can be classified as Semidetached software project. The rough estimate of number of lines of this project is 9.072k.Applying the above formula:

$E=3.0*(9.072)1.22$

= 44.20 person- months

D=2.5* 44.35

= 9.40 months

Hence according COCOMO -2nd model the time required for completion of the project is 9 (9.40) months.

- **Cost of Project :** Equation for calculation of cost of project using COCOMO – 2nd model is:

  C = D * Cp

  Where,

  C = Cost of project

  D = Duration in month

  Cp = Cost incurred per person-month, Cp=Rs.4000/- (per person-month) (approx.)

  C = 9 ∗ 4000

  = 36000/-

  Hence according COCOMO – 2nd model the cost of project is 36000/-(approx.)

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

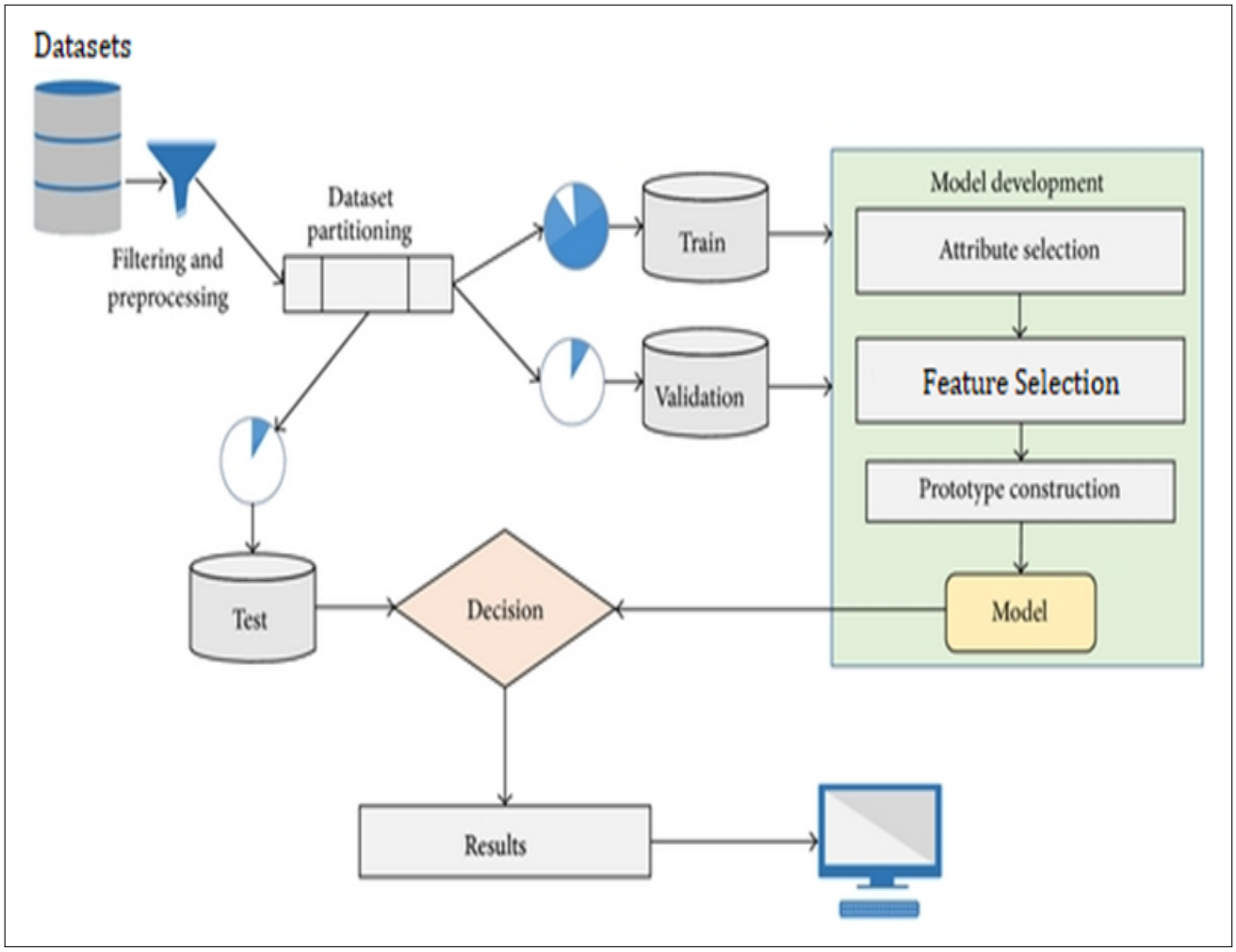

Figure 4.1: Architecture Diagram

Security and assurance systems are of the utmost importance in an environment like this because of the executive and control structure. Security challenges like as interruptions, spoofing attacks, denial-of-service attacks, stickiness, listening in, spam, and malware must be addressed in IoT applications. Depending on the amount and kind of relationship, IoT devices' security proportions vary. In order to participate, customers must be treated in a certain manner. As a result, we may argue that the location, nature, and application of are all factors.

The Internet of Things (IoT) allows demonstrations against the existing reality to be combined and executed regardless of where they take place. Executives and control in such an organisation make security and assurance systems the most vital and complex in this setting. IoT applications are needed to secure information from threats such interruptions, spoofing attacks, denial of service assaults, stickiness, listening in, spam, and malware. There is a direct correla-

tion between the size and kind of an association's influence on IoT device security... Customers are more likely to participate if they are handled well. It is therefore possible to claim that the place's location, character, and use are all factors that contribute to its significance.

The spammy characteristics are detection is proliferating everywhere exploiting every kind of vulnerability to the computing environment. Ethical Hackers pay more attention towards assessing vulnerabilities and recommending mitigation methodologies. The development of effective techniques has been an urgent demand in the field of the cyber security community. Most techniques used in today's IDS are not able to deal with the dynamic and complex nature of cyber-attacks on computer networks. Machine learning for cyber security has become an issue of great importance recently due to the effectiveness of machine learning in cyber security issues.

Machine learning techniques have been applied for major challenges in cyber security issues like intrusion detection, malware classification and detection, spam detection and phishing detection. Although machine learning cannot automate a complete cyber security system, it helps to identify cyber security threats more efficiently than other software-oriented methodologies, and thus reduces the burden on security analysts. Hence, efficient adaptive methods like various techniques of machine learning can result in higher detection rates, lower false alarm rates and reasonable computation and communication costs. Our main goal is that the task of finding attacks is fundamentally different from these other applications, making it significantly harder for the intrusion detection community to employ machine learning effectively.

4.2 DATA FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the

transformations that are applied as data moves from input to output.

4.2.1 Context Level Data Flow Diagram:

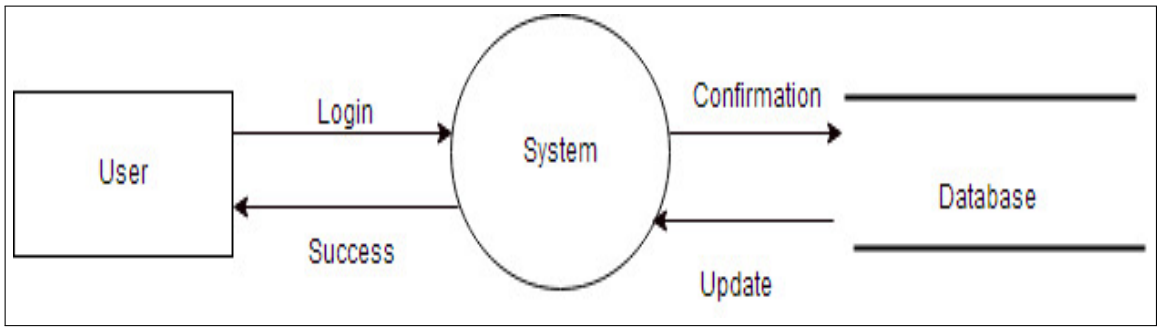

Figure 4.2: Context Level DFD

4.2.2 Multi Level Data Flow Diagram:

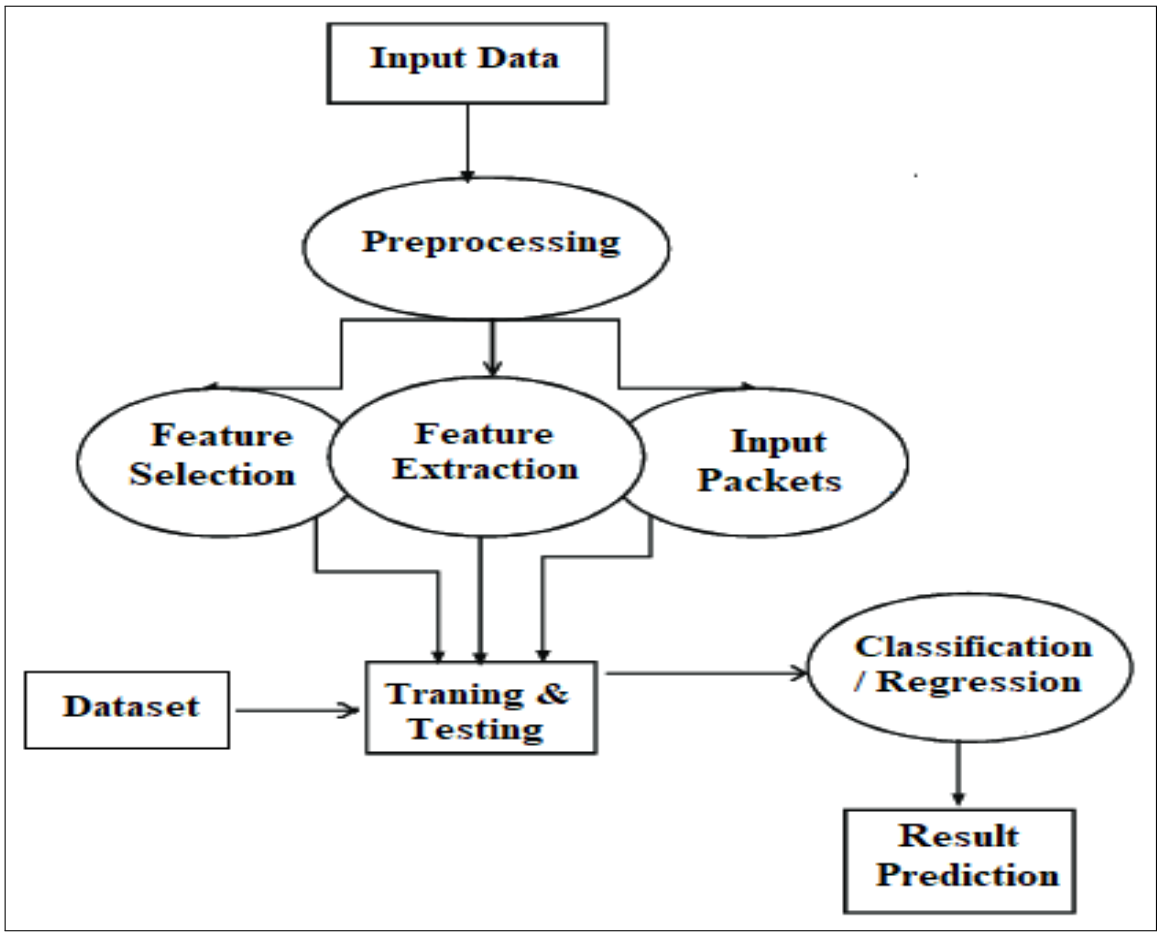

Figure 4.3: Multi Level DFD

4.3 UML DIAGRAM

4.3.1 Use case Diagram

Use case diagrams describe the interaction of any parson or external device with the system which is under design process. Use cases are often developed in collaborations between software developers and other users of the proposed system. The main purpose of the use case diagram is to help developing teams to visualize the functional requirements of the system. Use case diagram shows the relationship between actors and use cases. It consists of two elements: Use cases Actor The actor characterizes the interacting person or a thing. The use case describes the specific interaction of an actor with the system under design.

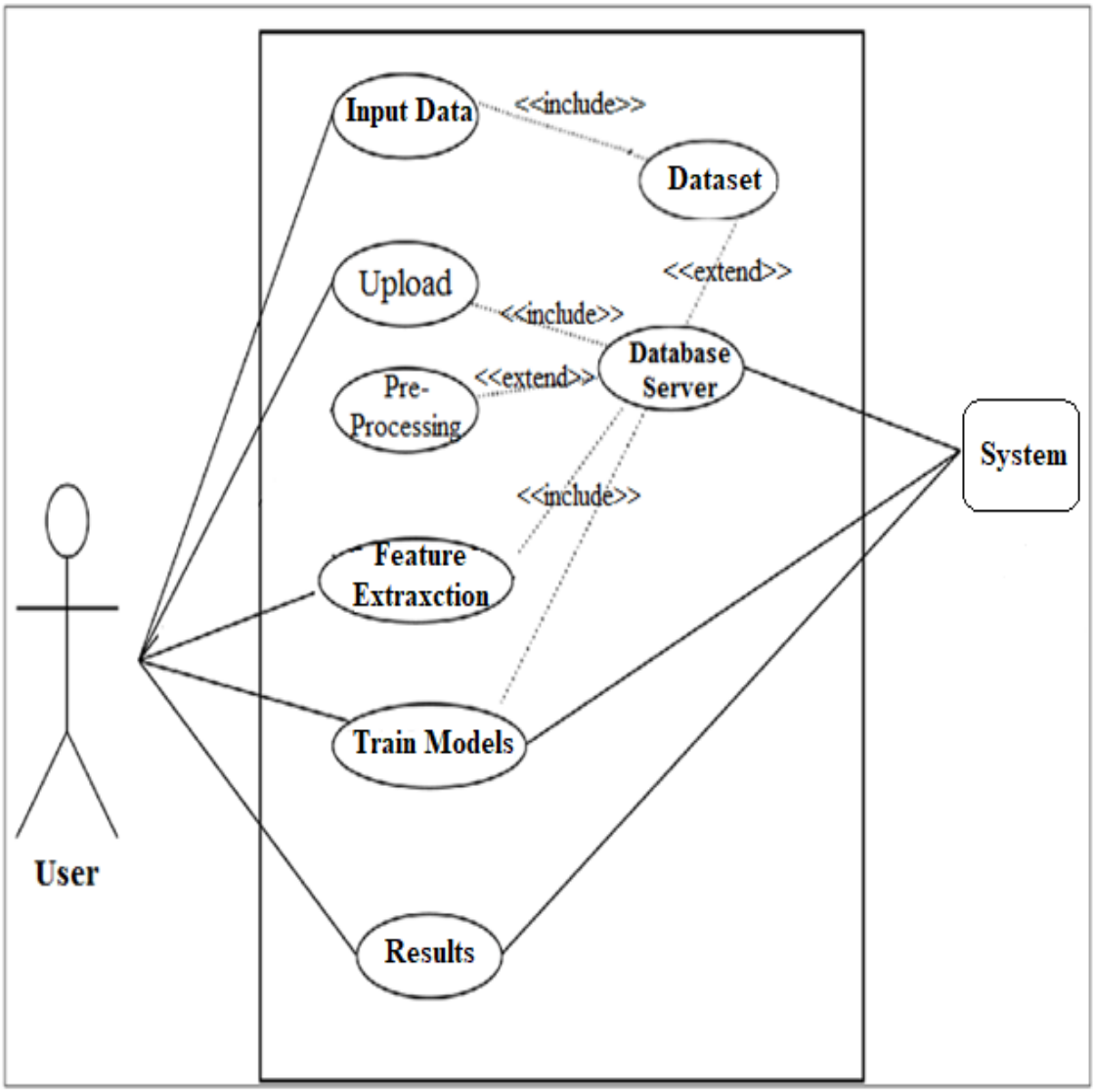

Figure 4.4: Use case Diagram

4.3.2 Class Diagram

In a class diagram classes are represented with boxes. Class diagram is the type of structure diagram which describe the structure of a system by showing classes, attributes, operations and relationships among the classes. Purpose of class diagram is to express the static structure of a system in terms of classes and relationships among those classes.

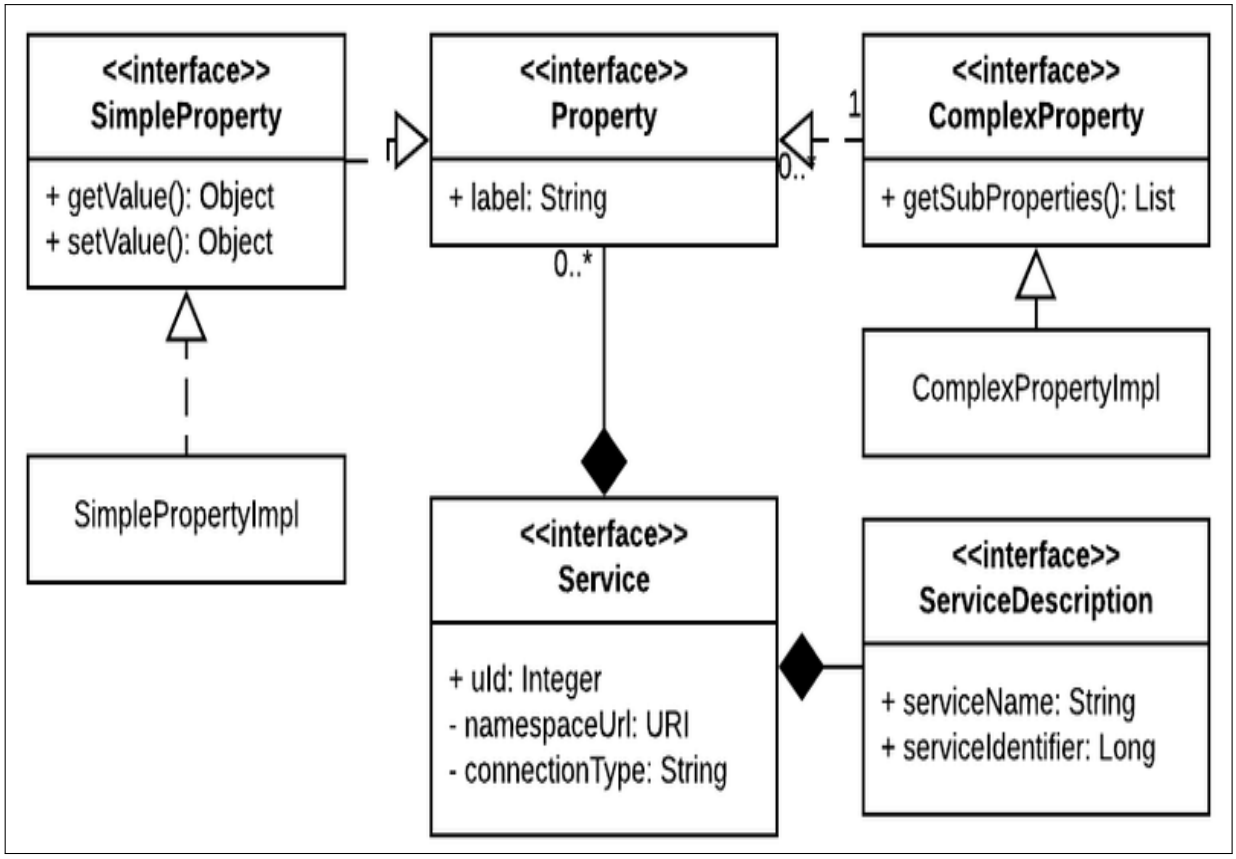

Figure 4.5: Class Diagram

4.3.3 Activity Diagram

Activity Diagram can be used to describe the dynamic aspects of the system. It is a flow chart which represents a flow from one activity to another activity. So activity diagram is considered as a flow chart. Main element used in this diagram is activity itself. An activity is a function performed by the system. Activity diagram is suitable for modeling the activity flow of the system.

Purpose:

- The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.
- Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.
- It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single. So the purposes can be described as:
  - Draw the activity flow of a system.
  - Describe the sequence from one activity to another.
  - Describe the parallel, branched and concurrent flow of the system.

  So before drawing an activity diagram we should identify the following elements:
  - Activities
  - Association
  - Conditions
  - Constraints

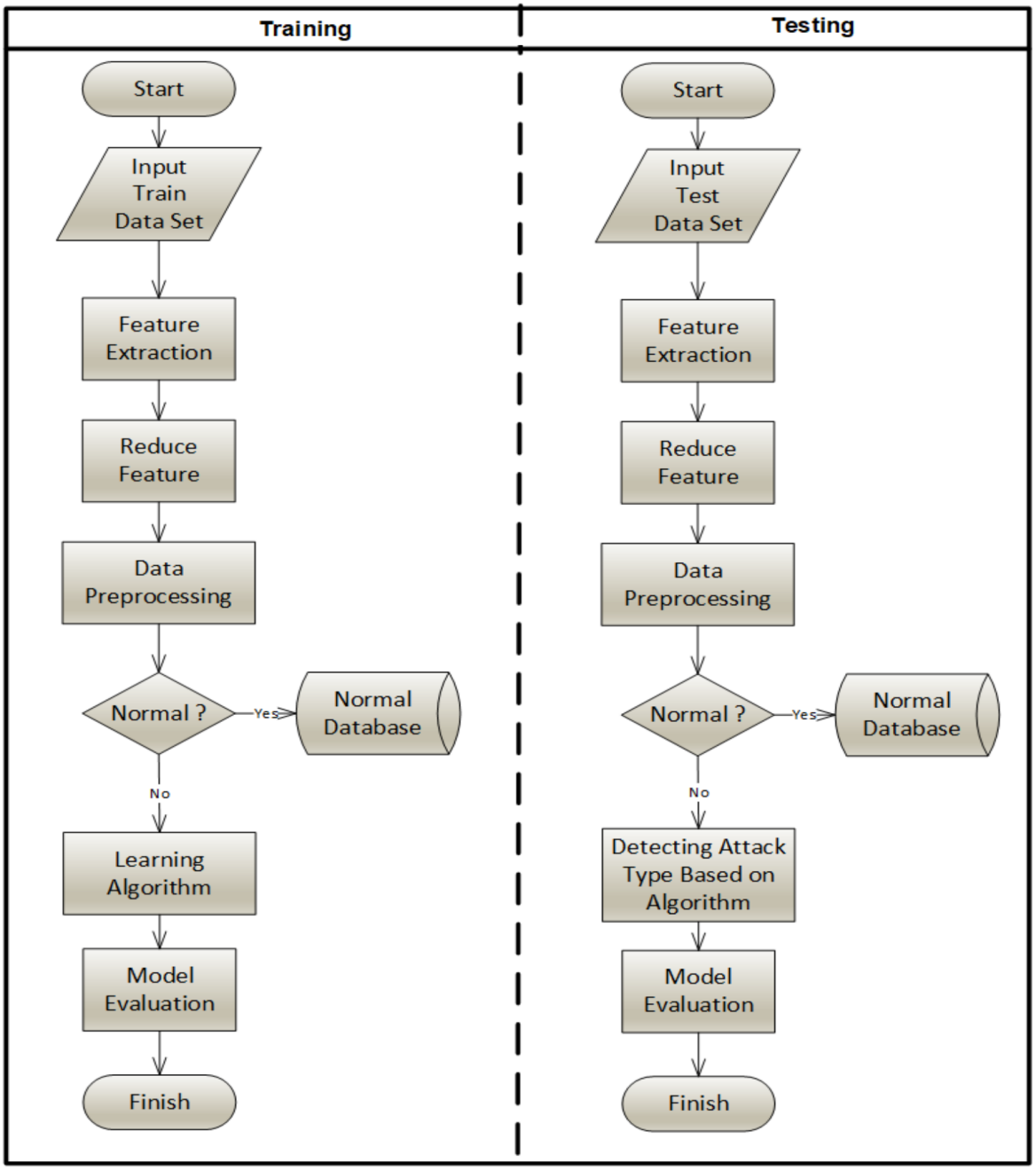

Figure 4.6: Activity Diagram

4.3.4 Component Diagram

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

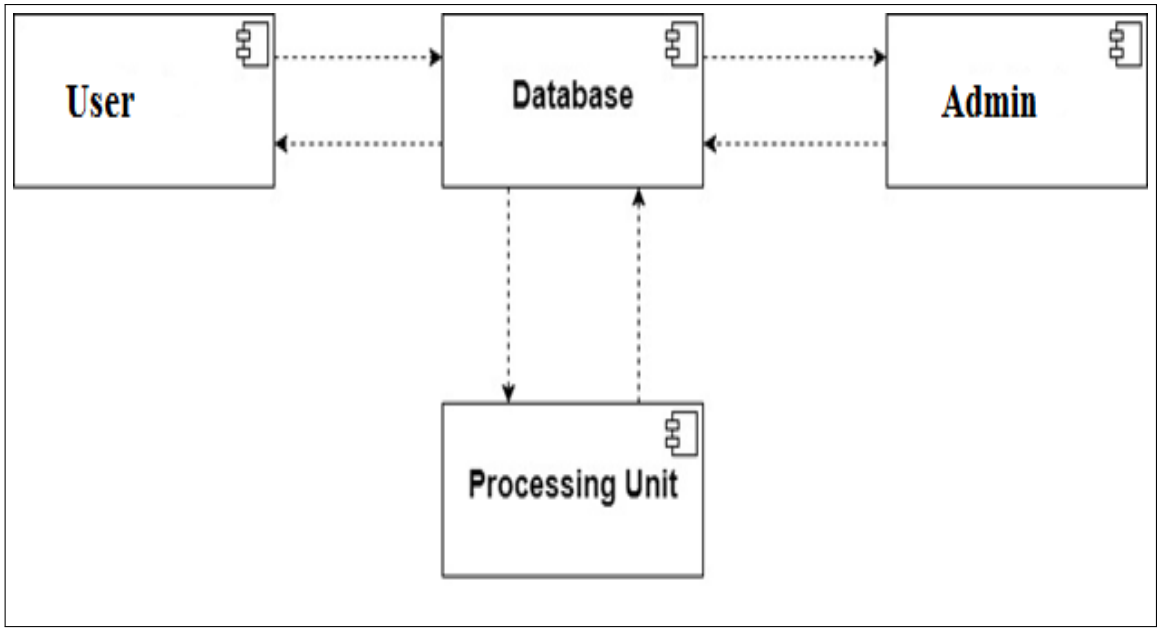

Figure 4.7: Component Diagram

4.3.5 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

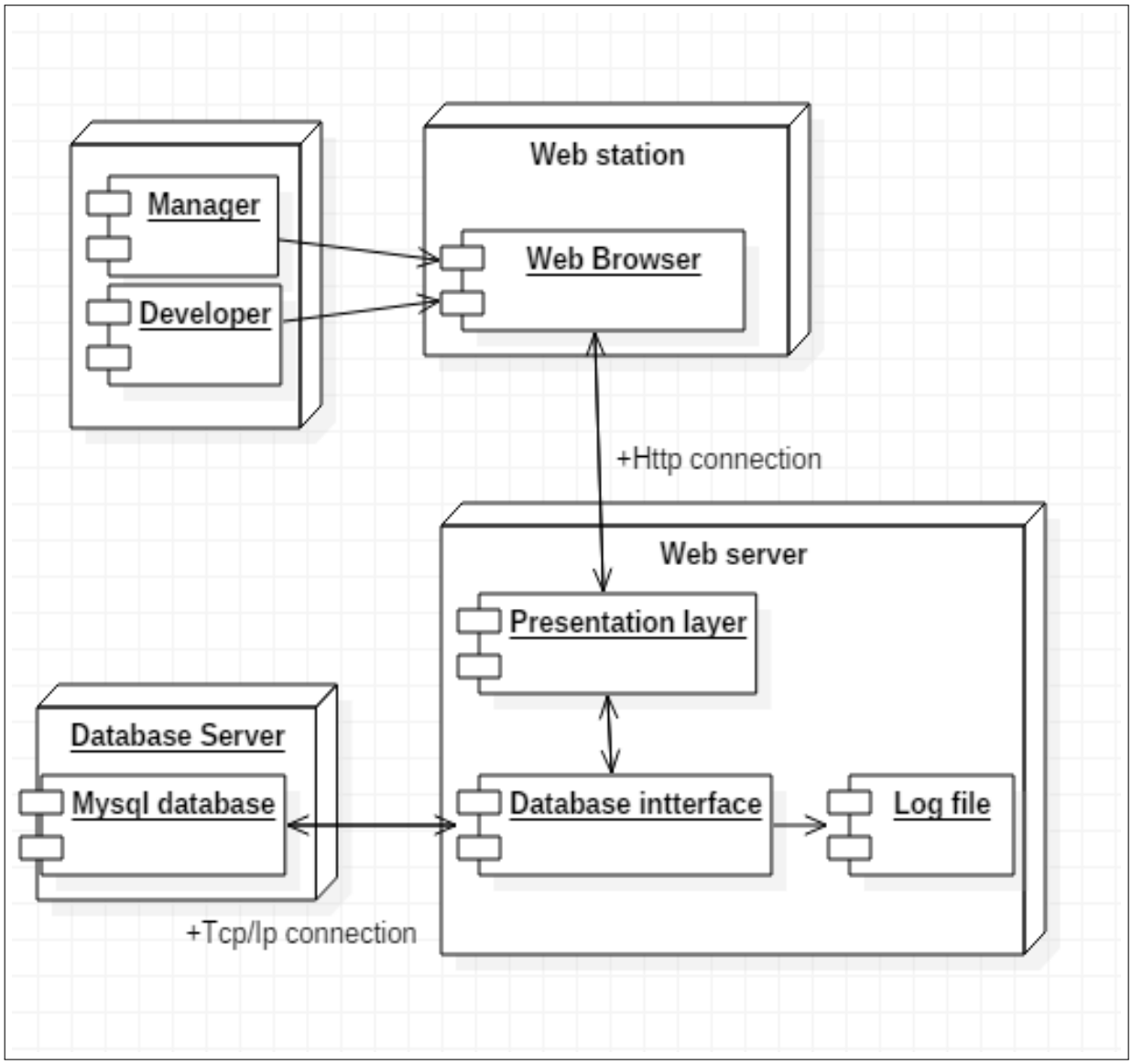

Figure 4.8: Deployment Diagram

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATES

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimate

Cost of project

C=N*Cp

C=3*8000

C=24,000

The Cost of the project is approximately up to 24000.

5.1.1.2 Time Estimates

Line of Code (LoC): Estimating LOC for this project is difficult at estimation stages this project is of innovative type project. Average estimation of this project is 10000 to 12000 line of code.

LOC based Estimation:

Efforts in Person in months

$E = 3.2 * (KLOC)^1.05$

$E = 3.2 * 9.0^1.05 to 11.0 * 4.2^1.05$

Function	Estimated KLOC
GUI design	1.1-1.3
Logical code	1.5-2.0
Location Based code	1.1-1.3
Directory matching code	1.0-1.3
Business logic	2.2-2.5
Testing	1.1-1.2
Re-correct Code	1.0-1.2
Total	9.0-10.11

Table 5.1: LOC Based estimation

5.1.1.3 Man Month Utilization:

Estimation of the man month is divide into following sub activities:
1-Technical training of the team member: This will take nearly 1 months. This will include Advance java, mysql, serialization etc.
2-Research: Being an innovative project research for the project is an important part currently it seems to have 1 to 1.5 months

5.1.2 Project Resources

- **Hardware Resources Required:**

    1. Processor: Intel i3
    2. Hard Disk: Minimum 100GB
    3. RAM: 4GB

- **Software Resources Required:**

    1. Platform: Windows7 and above.
    2. Backend: Mysql 5.5.0
    3. Front End: JAVA (J2EE).

5.2 RISK MANAGEMENT W.R.T. NP HARD ANALYSIS

When solving problems we have to decide the difficulty level of our problem. There are three types of classes provided for that. These are as follows:

- P class
- NP-Hard Class
- NP Complete Class

5.2.1 Risk Identification

For risks identification, review of scope document, requirements specifications and schedule is done.

1. Have top software and customer managers formally committed to support the project?
   Ans: All the required software's are freely available and hence development will be possible.

2. Are end-users enthusiastically committed to the project and the system/product to be built?
   Ans: The end user will be developers itself.

3. Are requirements fully understood by the software engineering team and its customers?
   Ans: Yes. All the requirements are fully understood by our team.

4. Have customers been involved fully in the definition of requirements?
   Ans: This is academic level project. So that whatever requirement be specify it should be by our team members and our guide.

5. Do end-users have realistic expectations?
   Ans: Yes.

6. Does the software engineering team have the right mix of skills?
   Ans: Yes, we have.

7. Are project requirements stable?
   Ans: All the basic requirements for this project are stable, from though some being variable but can be fulfilled.

8. Is the number of people on the project team adequate to do the job?
   Ans: Yes.

9. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?
   Ans: Yes.

5.2.2 Risk Analysis

The risks for the Project can be analyzed within the constraints of time and quality

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Deadline Risk	medium	Low	High	High
2	Technical skill Risk	medium	Low	High	High
3	Hardware Failure Risk	high	high	medium	high
4	Accuracy Risk	medium	medium	low	high

Table 5.2: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 5.3: Risk Probability definitions

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 5.4: Risk Impact definitions

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Following are the details for each risk.

Risk ID	1
Risk Description	Development Deadline Risk
Category	Development Environment.
Source	Software requirement Specification document.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Team Work distribution and Task plan.
Risk Status	Occurred

Risk ID	2
Risk Description	Technical skill risk
Category	Requirements
Source	Software Design Specification documentation review.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Self study and Internet will be best source for technology knowledge
Risk Status	Identified

Risk ID	3
Risk Description	Server Failure
Category	Requirements
Source	Software Design Specification documentation review.
Probability	Low
Impact	High
Response	Mitigate
Strategy	Check whether server is accurately working or not according to knowledge
Risk Status	Identified

5.3 PROJECT SCHEDULE

5.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1.1: Checking Feasibility of product
- Task 1.2: Scope of Product
- Task 1.3: Product Planing
- Task 1.4: Technical Risk
- Task 1.5: Proof of product
- Task 1.6: Implementation
- Task 1.7: Costumer Feedback

5.3.2 Task network

Project tasks and their dependencies are noted in this diagrammatic form.

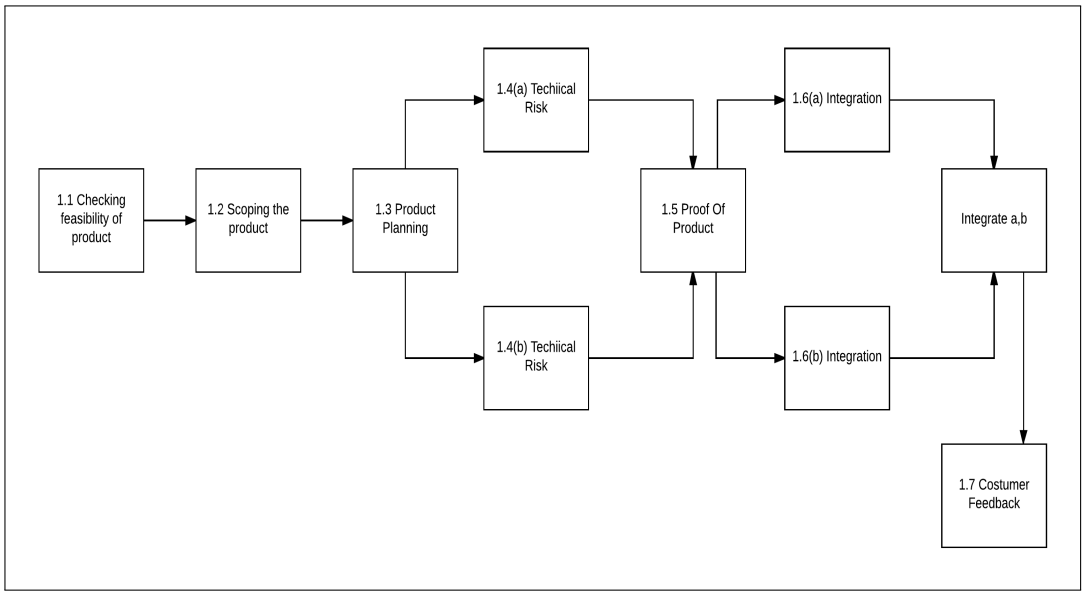

Figure 5.1: Task Network

5.3.3 Timeline Chart

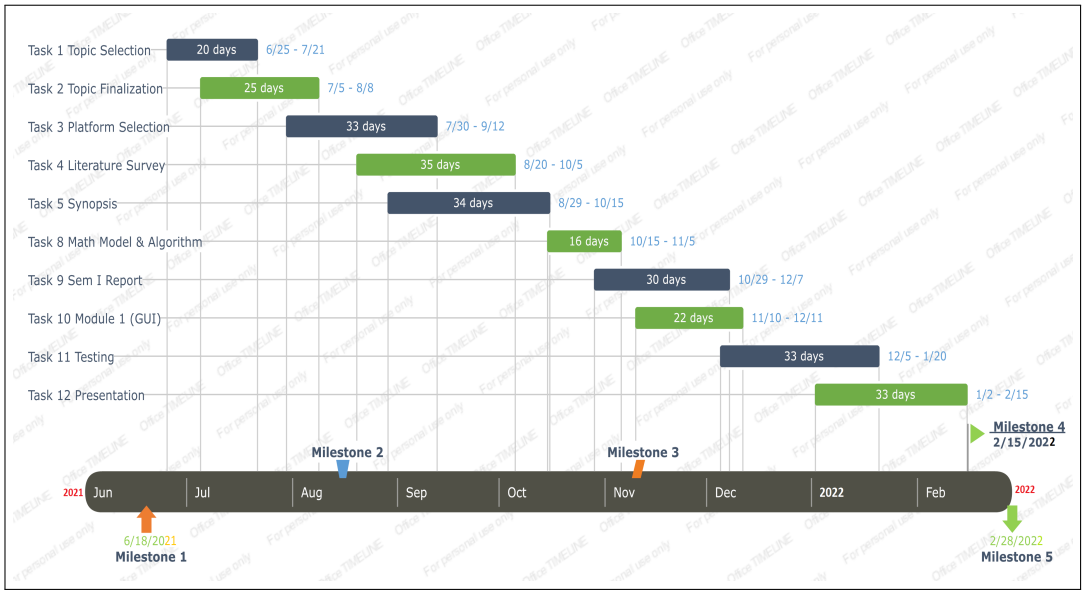

Figure 5.2: Time line Chart

5.4 TEAM ORGANIZATION

Team Member Details:-

1. Member 1
2. Member 2
3. Member 3
4. Member 4

5.4.1 Team structure

The team structure for the project is identified. Roles are defined. Our team have three members. We select this topic after discussing with each other. All the members performing all the task whatever tasks are assign to the members.

Name	Role
Member 1	Schedule all plan of project. Manage the team. Divide the work in team. The deadline are assign. Consider the all requirements and as per requirement gathering develop the module.
Member 2	Arrange the developing tool i.e platform, language, software, hardware and make the system architecture. Write the code of every module and apply the appropriate use case to test the plan.
Member 3 and Member 4	Test the each module if result is correct then combine all module and again test. After deployment manages the feedback report and correct some corrections.

Table 5.5: Team work distribution

5.4.2 Management reporting and communication

Sr.No	Reporting Date	Project Activity
1	22 June 2022	Decide project group member
2	29 June 2022	Submitted 3 Project Topic with IEEE Paper
3	13 Jul 2022	Discuss 5 point analysis of selected IEEE Paper
4	20 Jul 2022	3 Topics are presented and 1 topic selected
5	27 Jul 2022	Created and Submitted synopsis of a selected project
6	03 Aug 2022	Literature Survey and info gathering of a selected project
7	10 Aug 2022	30 percent project completion and presentation
8	31 Aug 2022	Draw UML diagram of a project
9	31 Aug 2022	50 percent project completion and presentation
10	07 Feb 2023	100 percent project completion and presentation
11	03 March 2023	Show the paper published
12	23 March 2023	Show the final report
13	6 April 2023	Show the final PPT
14	9 May 2023	Term 2nd Project overview

Table 5.6: Team work distribution

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 OVERVIEW OF PROJECT MODULES

The project "An Efficient Spam Detection Technique for IoT Devices using Machine Learning" typically involves managing and streamlining various aspects of the spam detection process for the system. Here is a general overview of the modules that could be included in such a project:

1. **Data Collection Module:**
    - Collect a diverse dataset of legitimate and spam messages specifically targeted at IoT devices.
    - Ensure the dataset represents different types of spam messages and includes sufficient samples of legitimate messages.
2. **Preprocessing Module:**
    - Clean and preprocess the collected dataset to remove noise, irrelevant information, and formatting inconsistencies.
    - Perform data normalization, tokenization, and feature extraction to transform the raw messages into a suitable format for machine learning algorithms.
3. **Feature Extraction Module:**
    - Identify and extract relevant features from the preprocessed messages that can effectively differentiate between legitimate and spam messages.
    - Features may include message content, sender information, device characteristics, network behavior, and any other relevant metadata.
4. **Model Selection and Training Module:**
    - Explore different machine learning algorithms suitable for spam detection in IoT devices, such as logistic regression, decision trees, random forests, support vector machines, or deep learning approaches like CNN or RNN.
    - Select the most appropriate algorithm(s) based on their performance and suitability for IoT devices.
    - Split the preprocessed dataset into training and testing sets.
    - Train the selected machine learning model using the training set, fine-tuning the model parameters as necessary.

5. **Evaluation Module:**

    - Evaluate the trained model using the testing set to measure its performance in detecting spam messages.
    - Assess the model's accuracy, precision, recall, F1 score, and other relevant metrics to evaluate its effectiveness.
    - Conduct cross-validation or other evaluation techniques to ensure the robustness of the model.

6. **Implementation Module:**

    - Implement the trained model on IoT devices, considering the resource-constrained nature of these devices.
    - Optimize the model for efficient deployment, taking into account computational requirements, memory usage, and energy consumption.

7. **Performance Evaluation Module:**

    - Conduct comprehensive performance evaluations of the implemented technique on a representative set of IoT devices.
    - Measure the detection accuracy, false positive rate, false negative rate, and other performance indicators under various scenarios and spam message variations.

8. **Comparison and Benchmarking Module:**

    - Compare the developed spam detection technique with existing methods for spam detection in IoT devices.
    - Benchmark the performance against state-of-the-art techniques and identify the advantages and limitations of the proposed approach.

By dividing the project into these modules, it becomes easier to manage and track the progress of each component, ensuring a systematic and organized approach towards developing an efficient spam detection technique for IoT devices using machine learning.

6.2 TOOLS AND TECHNOLOGIES USED

- **JAVA:** Java is one of the most popular and widely used programming language and platform. A platform is an environment that helps to develop and run programs written in any programming language.

  Java is fast, reliable and secure. From desktop to web applications, scientific supercomputers to gaming consoles, cell phones to the Internet, Java is used in every nook and corner.

  Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

  Java is a general-purpose, concurrent, object-oriented, class-based, and the runtime environment(JRE) which consists of JVM which is the cornerstone of the Java platform. This blog on What is Java will clear all your doubts about why to learn java, features and how it works.

- **XAMPP:** XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server is extremely easy as well. Web development using XAMPP is especially beginner friendly.

- **JDK:** The Java Development Kit (JDK) is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, macOS or Windows. The JDK includes a private

JVM and a few other resources to finish the development of a Java Application. Since the introduction of the Java platform, it has been by far the most widely used Software Development Kit (SDK).[citation needed] On 17 November 2006, Sun announced that they would release it under the GNU General Public License (GPL), thus making it free software. This happened in large part on 8 May 2007, when Sun contributed the source code to the OpenJDK.

- **Apache:** Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54 percent of all websites.

- **MySQL:** Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the world's most popular database management system. It powers everything from hobbyist websites to professional platforms like WordPress. You can learn how to master PHP with this free MySQL database for beginners course.

CHAPTER 7

SOFTWARE TESTING

7.1 TYPE OF TESTING

Types of Testing:

Along with the type of testing also mention the approach to be followed for the testing, that is, Manual Testing or Automated Testing. Use Automated Testing Plan for planning automation activities in details. The different types of testing that may be carried out in the project are as follows:

- **Unit Testing:**

  Individual components are tested independently to ensure their quality. The focus is to uncover errors in design and implementation, including.

  - Data structure in component
  - Program logic and program structure in a component
  - Component interface
  - Functions and operations of a component

- **Integration Testing :**

  A group of dependent components are tested together to ensure their quality of their integration unit. This approach is to do incremental integration to avoid "bigbang" problem. That is when the entire program is put together from all units and tested as a whole. The big-bang approach usually results in chaos which incremental integration avoids. Incremental integration testing can be done in two different way top down and bottom up. Then there is also the possibility of regression integration.

  The top down integration is when modules are integrated by moving downwards through the control hierarchy, beginning with the main control module. Modules subordinate to the main control module are incorporated into main structure n either depth-first or breadth-first manner. The top down integration verifies major controls or decision points early in the test process. If major control problems do exist, early recognition is essential. Bottom-up integration testing begins construction and testing with the lowest levels in the program structure. Because modules are integrated from the bottom-up,processing required for modules subordinate to a given level is always available and the need for test stubs is eliminated.

  The focus is to uncover errors in:

- Design and construction of software architecture
- Integrated functions or operations at sub-system level
- Interfaces and interaction and/or environment integration

- **System Testing :**

  The system software is tested as a whole. It verifies all elements mesh properly to make sure that all system functions and performance are achieved in the target environment. The focus areas are:

  - System functions and performance
  - System reliability and recoverability (recovery test)
  - System behavior in the special conditions (stress and load test)
  - System user operations (acceptance test/alpha test)
  - Hardware and software integration collaboration
  - Integration of external software and the system.

- **Validation Testing:**

  Validation can be defined in many ways, but a simple definition is that succeeds when software functions in a manner that can be reasonably expected by the customer. Software validation is achieved through a series of black-box tests that demonstrate conformity with requirements. A test plan outlines the classes of tests to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied, all behavioral characteristics are achieved, all performance requirements are attained, documentation is correct, and human engineered and other requirements are met.

- **White Box Testing:**

  White-box test design allows one to peek inside the “box”, and it focuses specifically on using internal knowledge of the software to guide the selection of test data. Synonyms for white-box include: structural, glass-box and clear-box.

White box testing is much more expensive than black box testing. It requires the source code to be produced before the tests can be planned and is much more laborious in the determination of suitable input data and the determination if the software is or is not correct. This testing is concerned only with testing the software product; it cannot guarantee that the complete specification has been implemented.

- **Black Box Testing:**

  Black-box test design treats the system as a "black-box", so it doesn't explicitly use knowledge of the internal structure. Black-box test design is usually described as focusing on testing functional requirements. Synonyms for black box include: behavioral, functional, opaque-box, and closed-box. Black box testing is concerned only with testing the specification; it cannot guarantee that all parts of the implementation have been tested. Thus black box testing is testing against the specification and will discover faults of omission, indicating that part of the specification has not been fulfilled.

- **GUI Testing:**

  Graphical User Interface (GUIs) present interesting challenges for software engineers. Because of reusable components provided as part of GUI development environments, the creation of the user interface has become less time consuming and more precise. But, the same time, the complexity of GUIs has grown, leading to more difficulty in the design and execution of the test cases. Because many modern GUIs have the same look and same feel, a series of test cases can be derived.

7.2 TEST CASES

Software to be tested:

After implementation of project software will be tested by tester.

Test Cases

Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

Input Field	Validation	Failure Cases (input)	Success Cases(input)
1. Username	Username should be present in database.	Raj or 1234abc	Raj or Raj@123
2. Password	Password should be present in database.	Raj or 1234abc	Raj or Raj@123
3. Login Button	N/A	Redirected to Login Failed page.	Redirected to User-Home page.

Table 7.1: Test case 1 for Login

Input Field	Validation	Failure Cases (input)	Success Cases(input)
1. Attack	Attack should be in text format and must be present in root directory.	cjjsksak	KASCNSNCJNJDNS
2. Submit Button	N/A	Attack is not visible in View Attack page.	Attack is visible in View Attack page.

Table 7.2: Test case 2 for Perform Attack

Input Field	Validation	Failure Cases (input)	Success Cases(input)
1. Initialization Mac	Check if the records are present in database	data not valid	All data are valid.
2. Result	Attack generated	Redirected to error page	Redirected to View Data page.

Table 7.3: Test case 3 for Attack Initialization

7.3 TEST RESULTS

Unit Testing:

It is the testing of individual software units of the application it is done after the completion of an individual unit before integration. 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. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

CHAPTER 8

RESULTS AND EVALUATION

8.1 RESULTS ANALYSIS

The results and evaluation of the project ”An Efficient Spam Detection Technique for IoT Devices using Machine Learning” involve assessing the performance of the developed technique in detecting spam messages. Here are the key aspects of results and evaluation:

- Performance Metrics: Measure the accuracy of the spam detection technique, which indicates the overall correctness of the classification.
  Calculate precision, recall, and F1 score to evaluate the technique's effectiveness in correctly identifying spam messages while minimizing false positives and false negatives.
  Assess the false positive rate (FPR) and false negative rate (FNR) to understand the rate of misclassification for both legitimate and spam messages.

- Comparison with Existing Techniques: Compare the performance of the developed technique with existing methods for spam detection in IoT devices.
  Benchmark the accuracy, precision, recall, and other relevant metrics against state-of-the-art techniques.
  Identify the strengths and weaknesses of the proposed approach compared to other approaches.

- Evaluation on Different Datasets: Test the spam detection technique on multiple datasets to assess its robustness and generalization capability.
  Evaluate the performance on different sets of legitimate and spam messages to validate the technique's effectiveness across various scenarios.

- Evaluation on IoT Devices: Deploy and evaluate the implemented technique on representative IoT devices.
  Measure the detection accuracy, false positive rate, and false negative rate specifically considering the resource-constrained nature of IoT devices.
  Assess the computational requirements, memory usage, and energy consumption of the deployed technique on IoT devices.

- Scalability and Efficiency: Evaluate the scalability of the spam detection technique to handle larger datasets and increasing volumes of spam messages.
  Measure the efficiency of the implemented technique in terms of processing time and resource utilization on IoT devices.

- Sensitivity Analysis: Perform sensitivity analysis by introducing variations in the spam messages, such as different content patterns, sender information, or network behavior. Assess the impact of these variations on the detection accuracy and false positive/false negative rates to understand the technique's robustness.

The results and evaluation phase of the project provide a comprehensive understanding of the developed spam detection technique's performance, its comparison with existing techniques, and its effectiveness in the context of IoT devices. It helps validate the proposed approach and provides insights for potential enhancements and future research directions.

8.2 ADVANTAGES

- Enhanced Security: The developed technique improves the security of IoT devices by effectively detecting and preventing spam messages. By identifying and filtering out spam, the technique reduces the risk of malicious activities and unauthorized access to IoT devices.
- Improved Reliability: With the implementation of the spam detection technique, the reliability of IoT devices is enhanced. Legitimate messages are correctly identified and delivered, ensuring the proper functioning of IoT applications and services.
- Customization for IoT Devices: The technique is specifically designed for IoT devices, considering their resource-constrained nature. It takes into account factors such as computational requirements, memory usage, and energy consumption, ensuring efficient deployment and optimal performance on IoT devices.
- Machine Learning Capabilities: By leveraging machine learning algorithms, the technique has the ability to learn and adapt to new spam patterns and variations. It can continuously improve its detection capabilities based on the evolving nature of spam messages, providing a proactive defense mechanism for IoT devices.
- Robustness and Generalization: The evaluation of the technique on multiple datasets and real-world IoT devices ensures its robustness and generalization. It demonstrates that the technique is effective across various scenarios, datasets, and IoT device environments.
- Reduced False Positives and False Negatives: The developed technique aims to minimize both false positives (legitimate messages wrongly identified as spam) and false negatives

(spam messages not detected). By optimizing the detection accuracy and minimizing misclassifications, the technique reduces the inconvenience and potential risks associated with mislabeled messages.

- Scalability: The spam detection technique is designed to scale with larger datasets and increasing volumes of spam messages. It can handle the growing demand for IoT devices and the accompanying increase in spam activity.
- Practical Implementation: The project focuses on the practical implementation of the spam detection technique on IoT devices. It considers the real-world constraints and requirements of IoT environments, making it feasible for integration into IoT systems and applications.
- Contribution to IoT Security: By addressing the specific security challenge of spam detection in IoT devices, the project contributes to the overall improvement of IoT security. It enhances the trustworthiness and reliability of IoT ecosystems, fostering the widespread adoption and utilization of IoT technologies.

8.3 LIMITATIONS

- Required proper project plan
- System required i.e. PC, Laptop, and Android Mobile Phone.
- Internet Connection required
- Database required.

8.4 APPLICATIONS

- IoT Security: The technique enhances the security of IoT devices by effectively detecting and mitigating spam messages. It helps prevent unauthorized access, data breaches, and potential malicious activities targeting IoT devices.
- Email Filtering: The technique can be applied to filter spam emails specifically targeted at IoT devices. It ensures that only legitimate and relevant emails are delivered to IoT devices, improving overall communication efficiency and reducing the risk of phishing attacks.

- Smart Home Security: By detecting and filtering spam messages, the technique improves the security of smart home systems. It prevents unauthorized commands, malicious control attempts, and potential breaches of privacy in smart home devices.
- Industrial IoT (IIoT): The technique can be employed in IIoT environments to detect and prevent spam messages targeting critical infrastructure, industrial sensors, and control systems. It ensures the integrity and reliability of industrial processes.
- Healthcare IoT: In healthcare IoT applications, the technique safeguards patient data and medical devices by identifying and blocking spam messages. It helps maintain the privacy and security of sensitive health information.
- Smart City Infrastructure: The technique contributes to the security of smart city infrastructure by identifying and mitigating spam messages targeting connected devices and systems. It helps protect critical infrastructure, transportation networks, and public services.
- Wearable Devices: The technique can be applied to wearable IoT devices, such as smartwatches and fitness trackers, to detect and filter spam notifications or messages. It ensures that users receive only relevant and legitimate notifications on their wearable devices.
- Automotive IoT: In the automotive IoT domain, the technique enhances the security and reliability of connected vehicles by detecting and filtering spam messages. It helps prevent malicious commands, unauthorized access, and potential safety risks.

CHAPTER 9

CONCLUSION

9.1 CONCLUSION

The proposed framework, detects the spam parameters of IoT devices using machine learning models. The IoT dataset used for experiments, is pre-processed by using feature engineering procedure. By experimenting the framework with machine learning models, each IoT appliance is awarded with a spam score. This refines the conditions to be taken for successful working of IoT devices in a smart home. In future, we are planning to consider the climatic and surrounding features of IoT device to make them more secure and trustworthy.

In conclusion, the project "An Efficient Spam Detection Technique for IoT Devices using Machine Learning" has successfully developed a tailored approach to address the challenge of spam detection in IoT devices. By leveraging machine learning algorithms and considering the resource-constrained nature of IoT devices, the technique offers enhanced security, reliability, and scalability. It effectively detects and filters spam messages, reducing the risk of malicious activities, unauthorized access, and compromising the functionality of IoT devices. The evaluation and comparative analysis have demonstrated the effectiveness and robustness of the developed technique in various scenarios and datasets, validating its applicability in real-world IoT environments.

In future research, more spams or attacks on agents can be considered, also data mining and other machine learning methods, such as support vector machine(SVM) algorithms or other types of neural networks such as recurrent neural networks to evaluate system performance improvements.

CHAPTER 10

REFERENCES

[1] K. Graves, Ceh: Official certified ethical hacker review guide: Exam 312-50. John Wiley & Sons, 2007.

[2] R. Christopher, "Port scanning techniques and the defense against them," SANS Institute, 2001.

[3] M. Baykara, R. Das¸, and I. Karado ?gan, "Bilgi g ¨uvenli ?gisistemlerindekullanilanarac¸ larinincelenmesi," in 1st International Symposium on Digital Forensics and Security (ISDFS13), 2013, pp. 231–239.

[4] S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans," Journal of Computer Security, vol. 10, no. 1-2, pp. 105–136, 2002.

[5] S. Robertson, E. V. Siegel, M. Miller, and S. J. Stolfo, "Surveillance detection in high bandwidth environments," in DARPA Information Survivability Conference and Exposition, 2003.Proceedings, vol. 1. IEEE, 2003, pp. 130–138.

[6] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "Iot security: ongoing challenges and research opportunities," in 2014 IEEE 7th international conference on service-oriented computing and applications. IEEE, 2014, pp. 230–234.

[7] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in 2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops). IEEE, 2017, pp. 618–623.

[8] E. Bertino and N. Islam, "Botnets and internet of things security," Computer, no. 2, pp. 76–79, 2017.

[9] F. A. Narudin, A. Feizollah, N. B. Anuar, and A. Gani, "Evaluation of machine learning classifiers for mobile malware detection," Soft Computing, vol. 20, no. 1, pp. 343–357, 2016.

ANNEXURE A

1. **Dr. Aaisha Makkar, Dr. Sahil (GE) Garg, Dr. Neeraj Kumar, Prof. M. Shamim Hossain, Prof. Ahmed Ghoneim, Dr. Mubarak Alrashoud, "An Efficient Spam Detection Technique for IoT Devices using Machine Learning".**

   The Internet of Things (IoT) is a group of millions of devices having sensors and actuators linked over wired or wireless channel for data transmission. IoT has grown rapidly over the past decade with more than 25 billion devices are expected to be connected by 2020. The volume of data released from these devices will increase many-fold in the years to come. In addition to an increased volume, the IoT devices produces a large amount of data with a number of different modalities having varying data quality defined by its speed in terms of time and position dependency. In such an environment, machine learning algorithms can play an important role in ensuring security and authorization based on biotechnology, anomalous detection to improve the usability and security of IoT systems. On the other hand, attackers often view learning algorithms to exploit the vulnerabilities in smart IoT-based systems. Motivated from these, in this paper, we propose the security of the IoT devices by detecting spam using machine learning. To achieve this objective, Spam Detection in IoT using Machine Learning framework is proposed. In this framework, five machine learning models are evaluated using various metrics with a large collection of inputs features sets. Each model computes a spam score by considering the refined input features. This score depicts the trustworthiness of IoT device under various parameters. REFIT Smart Home dataset is used for the validation of proposed technique. The results obtained proves the effectiveness of the proposed scheme in comparison to the other existing schemes.

ANNEXURE B

PLAGIARISMA

100% Unique

Total 3001 chars (2000 limit exceeded) , 290 words, 15 unique sentence(s).

Results	Query	Domains (original links)
Unique	blockchain itself is not designed as a huge-scale storage system	-
Unique	the slow dissemination of records exposes a potential security hollow for the malicious assaults	-
Unique	but, lengthy-time period solutions are nevertheless required	-
Unique	bitcoin peer-to-peer network topology can inevitable and used by malicious at	-
Unique	a blockchain system considered as a truely incorruptible cryptographic database where important clinical facts may	-
Unique	a network of computers that is available to all of us jogging the software	-
Unique	all transactions are uncovered to the public, although it is tamper-proof in the experience of	-
Unique	the get admission to manage of heterogeneous patients' healthcare statistics throughout multiple health institutions	-
Unique	inside the context of healthcare, a decentralized garage answer would significantly complement the weakness	-
Unique	the blockchain network as a decentralized machine is extra resilient in that there's no	-
Unique	people has get entry to, there already exist analytics tools that become aware of the	-
Unique	with popularity analytics, similarity or closeness amongst subjects within large extent of statistics may	-
Unique	as statistics flows among unique nodes in bitcoin network, bitcoin transaction is gradual because	-
Unique	like some other networks, bitcoin community is no exception when it comes to malicious	-
Unique	one of the high-quality varieties of attack against bitcoin network topology is eclipsing assault	-