

An industrial oriented major project report

on

**A USER-CENTRIC MACHINE LEARNING FRAMEWORK
FOR CYBER SECURITY OPERATIONS CENTER**

Submitted by

ANUSHKA KUMARI	19W91A0515
BASIREDDY RUDRATEJA REDDY	19W91A0529
BONGU SIKIRAN	19W91A0538
CHEPURI SHYNITHA	19W91A0548

Under the Esteemed Guidance of

Mr. M Nagendra Rao

Assistant Professor, CSE

TO

Jawaharlal Nehru Technological University, Hyderabad

In partial fulfilment of the requirements for award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MALLA REDDY INSTITUTE OF ENGINEERING AND TECHNOLOGY
(UGC AUTONOMOUS)**

(Sponsored by Malla Reddy Educational society)

(Affiliated to JNTU, Hyderabad)

Maisammaguda, Dhulapally post, Secunderabad-500014.

2022-2023



Department of Computer Science and Engineering

BONAFIDE CERTIFICATE

This is to certify that this is the bonafide certificate of major project report titled “**A USER CENTRIC MACHINE LEARNING FRAMEWORK FOR CYBER SECURITY OPERATIONS CENTER**” is submitted by **ANUSHKA KUMARI (19W91A0515), BASIREDDY RUDRATEJA REDDY (19W91A0529), BONGU SAIKIRAN(19W91A0538), CHEPURI SHYNITHA (19W91A0548)** of B. Tech in the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science and Engineering**, Dept. of Computer Science & Engineering and this has not been submitted for the award of any other degree of this institution.

Internal Guide Sign

Head of the Department Sign

External Examiner Sign

DECLARATION

We hereby declare that the Major Project report entitled “**A USER CENTRIC MACHINE LEARNING FRAMEWORK FOR CYBER SECURITY OPERATIONS CENTER**” submitted to Malla Reddy Institute of Engineering and Technology(Autonomous), affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH), for the award of the degree of Bachelor of Technology in Computer Science & Engineering is a result of original industrial oriented major project done by me.

It is further declared that the seminar report or any part thereof has not been previously submitted to any University or Institute for the award of degree or diploma

ANUSHKA KUMARI	19W91A0515
BASIREDDY RUDRATEJA REDDY	19W91A0529
BONGU SIKIRAN	19W91A0538
CHEPURI SHYNITHA	19W91A0548

ACKNOWLEDGEMENT

First and foremost, We are grateful to the Principal **Dr. M. ASHOK**, for providing me with all the resources in the college to make my project a success. We thank him for his valuable suggestions at the time of seminars which encouraged me to give my best in the project.

We would like to express my gratitude to **Dr.MD ASHFAKUL HASAN**, Head of the Department, Department of Computer Science and Engineering for his support and valuable suggestions during the dissertation work.

We offer my sincere gratitude to my project - coordinator **Dr. B. DHANALAXMI** and internal guide **Mr. M. NAGENDRA RAO**, Assistant Professor of Computer Science and Engineering department who has supported me throughout this project with their patience and valuable suggestions.

We would also like to thank all the supporting staff of the Dept. of CSE and all other departments who have been helpful directly or indirectly in making the project a success.

We are extremely grateful to our parents for their blessings and prayers for my completion of project that gave us strength to do our project

ANUSHKA KUMARI	19W91A0515
BASIREDDY RUDRATEJA REDDY	19W91A0529
BONGU SIKIRAN	19W91A0538
CHEPURI SHYNITHA	19W91A0548

TABLE OF CONTENTS

Abstract	I
List of Figures	II
List of Tables	III
List of Screens	IV
List of Abbreviations	V

CHAPTER NO	CONTENTS	PAGE NO
1	INTRODUCTION	1
	1.1 Motivation	2
	1.2 Problem Definition	2
	1.3 Objective	2
	1.4 Limitations	3
	1.5 Organization of Documentation	3
2	LITERATURE SURVEY	4
	2.1 Introduction	4
	2.2 Existing System	5
	2.3 Disadvantages of Existing System	5
	2.4 Proposed System	6
	2.5 Conclusion	7
3	ANALYSIS	8
	3.1 Introduction	8
	3.2 Software Requirements Specifications	8
	3.2.1 User Requirements	10
	3.2.2 Software Requirements	13
	3.2.3 Hardware Requirements	13
	3.3 Content Diagram	13
	3.4 Algorithms and Flowchart	14
	3.5 Conclusion	20
4	DESIGN	21
	4.1 Introduction	21
	4.2 UML Diagrams	22

	4.2.1 Class Diagram	23
	4.2.2 Use Case Diagram	25
	4.2.3 Sequence Diagram	27
	4.2.4 Collaboration Diagram	29
	4.2.5 Activity Diagram	30
	4.2.6 Component Diagram	32
	4.2.7 Deployment Diagram	34
	4.3 Module Design and Organization	35
	4.4 Conclusion	36
5	IMPLEMENTATION & RESULTS	37
	5.1 Introduction	37
	5.2 Explanation of Key Functions	37
	5.3 Method of Implementation	37
	5.4 Code Implementation	52
	5.4.1 Source Code	52
	5.4.2 Output Screens	58
	5.4.3 Result Analysis	58
	5.5 Conclusion	58
6	TESTING AND VALIDATION	59
	6.1 Introduction	59
	6.2 Design of Test Cases and Scenarios	59
	6.3 Validation	66
	6.4 Conclusion	66
7	CONCLUSION	67
	7.1 Project Conclusion	67
	7.2 Future Enhancement	67
8	REFERENCES	68

ABSTRACT

In order to ensure a company's Internet security, SIEM (Security Information and Event Management) system is in place to simplify the various preventive technologies and flag alerts for security events. Inspectors (SOC) investigate warnings to determine if this is true or not. However, the number of warnings in general is wrong with the majority and is more than the ability of SCO to handle all awareness. Because of this, malicious possibility. Attacks and compromised hosts may be wrong.

Machine learning is a possible approach to improving the wrong positive rate and improving the productivity of SOC analysts. In this article, we create a user-centric engineer learning framework for the Internet Safety Functional Center in the real organizational context. We discuss regular data sources in SOC, their work flow, and how to process this data and create an effective machine learning system. This article is aimed at two groups of readers. The first group is intelligent researchers who have no knowledge of data scientists or computer safety fields but who engineer should develop machine learning systems for machine safety. The second groups of visitors are Internet security practitioners that have deep knowledge and expertise in Cyber Security, but do Machine learning experiences do not exist and I'd like to create one by themselves.

At the end of the paper, we use the account as an example to demonstrate full steps from data collection, label creation, feature engineering, machine learning algorithm and sample performance evaluations using the computer built in the SOC production of Seyondike.

LIST OF FIGURES

S.no	Fig No	Name of the Figure	Page No
1	3.3.1	Content Diagram	13
2	3.4.1	Support Vector Machine Algorithm	14
3	3.4.2	Original Dataset	16
4	3.4.3	Data with separator added	16
5	3.4.4	Transformed data	16
6	3.4.5	Flowchart	19
7	4.2.1.1	Class diagram template	23
8	4.2.1.2	Dependency symbol	23
9	4.2.1.3	Generalization symbol	24
10	4.2.1.4	Assosiation	24
11	4.2.1.5	Aggregation	24
12	4.2.1.6	Class diagram of the system	25
13	4.2.2.1	User Use Case Diagram	26
14	4.2.2.2	Admin Use Case Diagram	26
15	4.2.3.1	Sequence Diagram	27
16	4.2.3.2	Sequence Diagram of the system	28
17	4.2.4.1	Collaboration Diagram of the system	29
18	4.2.5.1	user activity diagram of the system	31
19	4.2.5.2	Admin activity diagram of the system	31
20	4.2.6.1	notation of component	32
21	4.2.6.2	notation of node	32
22	4.2.6.3	User Component Diagram	33
23	4.2.6.4	Admin Component Diagram	33
24	4.2.7.1	Deployment Diagram	34
25	4.3.1	Module Design and Organization	36
26	5.3.1	Django Architecture	42

LIST OF TABLES

S.no	Table No	Name of the Table	Page No
1	4.2.5.1	Activity Diagram Elements	30
2	6.2.2.1	User Test Cases	64
3	6.2.2.2	Admin Test Cases	65

LIST OF SCREENS

S.no	Screenshot No	Name of Screenshot image	Page No
1	5.4.2.1	User Login Page	52
2	5.4.2.2	User Register Page	53
3	5.4.2.3	User Update Page	53
4	5.4.2.4	User Transaction Page	54
5	5.4.2.5	User Analyze Page	54
6	5.4.2.6	User Receive alert page	55
7	5.4.2.7	Admin Login page	55
8	5.4.2.8	Admin analyze page	56
9	5.4.2.9	Admin Risk User page	56
10	5.4.2.10	Admin pie chart Analysis page	57
11	5.4.2.11	Admin bar chart Analysis page	57
12	5.4.2.12	Admin column chart Analysis page	58

LIST OF ABBREVIATIONS

S.No	Abbreviation	Full Form
1	SOC	Security Operation Center
2	SIEM	Security Information and Event management
3	OTRS	Open Source Ticket Request System
4	DNS	Domain Name Server
5	DHCP	Dynamic Host Configuration Protocol
6	IDS	Intrusion Detection System
7	IPS	Intrusion Prevention System
8	ISMS	Information Security Management system
9	SCADA	Supervisory Control and Data Acquisition
10	SVM	Support Vector Machine
11	SRS	Software Requirements Specifications
12	HTML	Hypertext Markup Language
13	CSS	Cascading Styling Sheet
14	GUI	Graphical User Interface
15	UML	Unified Modelling Language

1. INTRODUCTION

Cyber security incidents will cause significant financial and reputation impacts on enterprise. In order to detect malicious activities, the SIEM (Security Information and Event Management) system is built in companies or government. The system correlates event logs from endpoint, firewalls, IDS/IPS (Intrusion Detection/Prevention System), DLP (Data Loss Protection), DNS (Domain Name System), DHCP (Dynamic Host Configuration Protocol), Windows/Unix security events, VPN logs etc. The security events can be grouped into different categories. The logs have terabytes of data each day. From the security event logs, SOC (Security Operation Centre) team develops so-called use cases with a pre-determined severity based on the analysts experiences. They are typically rule based correlating one or more indicators from different logs. These rules can be network/host based or time/frequency based.

If any pre-defined use case is triggered, SIEM system will generate an alert in real time. SOC analysts will then investigate the alerts to decide whether the user related to the alert is risky (a true positive) or not (false positive). If they find the alerts to be suspicious from the analysis, SOC analysts will create OTRS (Open Source Ticket Request System) tickets. After initial investigation, certain OTRS tickets will be escalated to tier 2 investigation system (e.g., Co3 System) as severe security incidents for further investigation and remediation by Incident Response Team.

However, SIEM typically generates a lot of the alerts, but with a very high false positive rate. The number of alerts per day can be hundreds of thousands, much more than the capacity for the SOC to investigate all of them. Because of this, SOC may choose to investigate only the alerts with high severity or suppress the same type of alerts. This could potentially miss some severe attacks. Consequently, a more intelligent and automatic system is required to identify risky users.

The machine learning system sits in the middle of SOC work flow, incorporates different event logs, SIEM alerts and SOC analysis results and generates comprehensive user risk score for security operation center. Instead of directly digging into large amount of SIEM alerts and trying to find needle in a haystack, SOC analysts can use the risk scores from machine learning system to prioritize their investigations, starting from the users with highest risks. This will greatly improve

their efficiency, optimize their job queue management, and ultimately enhance. Specifically, our approach constructs a framework of user centric machine learning system to evaluate user risk based on alert information.

This approach can provide security analyst a comprehensive risk score of a user and security analyst can focus on those users with high risk scores. To the best of our knowledge, there is no previous research on building a complete systematic solution for this application

1.1 MOTIVATION

Cyber Security is the set of technologies and processes designed to protect computers, network programs, and the data from attack, unauthorized access, change, or destruction. Cyber security systems are composed of network security systems and computer (host) security systems. Each of these has, at a minimum, a firewall, antivirus software, and an intrusion detection system (IDS). IDSs help discover, determine, and identify unauthorized use, duplication, alteration, and destruction of information systems. The security breaches include external intrusions (attacks from outside organization) and internal intrusions (attack from within the organization).

Misuse-based techniques are designed to detect known attacks by using signatures of those attacks. They are effective for detecting known type of attacks without generating overwhelming number of false alarms. They require frequent manual updates of the database with rules and signatures.

1.2 PROBLEM DEFINITION

Our approach constructs a framework of user centric machine learning system to evaluate user risk based on alert information. This approach can provide security analyst a comprehensive risk score of a user and security analyst can focus on those users with high risk scores.

1.3 OBJECTIVE

An advanced user-centric machine learning system is proposed and evaluated by real industry data to evaluate user risks. The system can effectively reduce the resources to analyse alert manually while at the same time enhance enterprise security.

This framework gives users an effortlessly explore through the application for more data in a most secure way. This framework gives simple access. The users can

do transactions safe and secure manner. And to get complete information about the risk of particular transactions respectively.

1.4 LIMITATIONS

- Majority of the users without annotations are left out of model, but they may have valuable information.
- Many machine learning models will not work well for highly unbalanced classification problem.

1.5 ORGANIZATION OF DOCUMENTATION

In this project documentation we have initially put the definition and objective of the project as well as the design of the project which is followed by the implementation and testing phases. Finally, the project has been concluded successfully and also the future enhancements of the project were given in this documentation.

2. LITERATURE SURVEY

2.1 INTRODUCTION

The input to an algorithm that learns a binary classifier normally consists of two sets of examples, where one set consists of positive examples of the concept to be learned, and the other set consists of negative examples. However, it is often the case that the available training data are an incomplete set of positive examples, and a set of unlabelled examples, some of which are positive and some of which are negative. The problem solved in this paper is how to learn a standard binary classifier given a non-traditional training set of this nature. Under the assumption that the labelled examples are selected randomly from the positive examples, we show that a classifier trained on positive and unlabelled examples predicts probabilities that differ by only a constant factor from the true conditional probabilities of being positive. We show how to use this result in two different ways to learn a classifier from a non-traditional training set.

The literature survey focuses on machine learning (ML) and data mining (DM) methods for cyber analytics in support of intrusion detection. Short tutorial descriptions of each ML/DM method are provided. Based on the number of citations or the relevance of an emerging method, each method were identified, read, and summarized. Because data are so important in ML/DM approaches, some well-known cyber data sets used in ML/DM are described. The complexity of ML/DM algorithms is addressed, discussion of challenges for using ML/DM for cyber security is presented, and some recommendations on when to use a given method are provided.

Network attacks have become more pervasive in the cyber world. There are various attacks such as denial of service, scanning, privilege escalation that is increasing day by day leading towards the requirement of a more robust and adaptable security techniques. Anomaly detection is the main focus of our paper. Support Vector Machine (SVM) is one of the good classification algorithm applied specially for intrusion detection. However, its performance can be significantly improved when it is applied in integration with other classifiers. In this system, we have performed a comparative analysis of SVM classifier's performance when it is

stacked with other classifiers like BayesNet, AdaBoost, Logistic, IBK, J48, RandomForest, JRip, OneR and SimpleCart.

(DNN). In this technique, in-vehicle network packets exchanged between electronic control units (ECU) are trained to extract low- dimensional features and used for discriminating normal and hacking packets. The features perform in high efficient and low complexity because they are generated directly from a bitstream over the network. The proposed technique monitors an exchanging packet in the vehicular network while the feature are trained off-line, and provides a real-time response to the attack with a significantly high detection ratio in our experiments.

2.2 EXISITING SYSTEM

Most approaches to security in the enterprise have focused on protecting the network infrastructure with no or little attention to end users. As a result, traditional security functions and associated devices, such as firewalls and intrusion detection and prevention devices, deal mainly with network level protection. Although still part of the overall security story, such an approach has limitations in light of the new security challenges described in the previous section.

Data Analysis for Network Cyber-Security focuses on monitoring and analyzing network traffic data, with the intention of preventing, or quickly identifying, malicious activity. Risk values were introduced in an information security management system (ISMS) and quantitative evaluation was conducted for detailed risk assessment. The quantitative evaluation showed that the proposed countermeasures could reduce risk to some extent. Investigation into the cost-effectiveness of the proposed countermeasures is an important future work. It provides users with attack information such as the type of attack, frequency, and target host ID and source host ID. Ten et al. proposed a cyber-security framework of the SCADA system as a critical infrastructure using real-time monitoring, anomaly detection, and impact analysis with an attack tree-based methodology, and mitigation strategies

2.3 DISADVANTAGES OF EXISTING SYSTEM

The disadvantage of Existing System are:

- Firewalls can be difficult to configure correctly.

- Incorrectly configured firewalls may block users from performing actions on the Internet, until the firewall configured correctly.
- Makes the system slower than before.
- Need to keep updating the new software in order to keep security up to date.
- Could be costly for average user.
- The user is the only constant.

2.4 PROPOSED SYSEM

User-centric cyber security helps enterprises reduce the risk associated with fast-evolving end-user realities by reinforcing security closer to end users. User-centric cyber security is not the same as user security. User-centric cyber security is about answering peoples' needs in ways that preserve the integrity of the enterprise network and its assets.

User security can almost seem like a matter of protecting the network from the user — securing it against vulnerabilities that user needs introduce. User-centric security has the greater value for enterprises. cyber-security systems are real-time and robust independent systems with high performances requirements. They are used in many application domains, including critical infrastructures, such as the national power grid, transportation, medical, and defence. These applications require the attainment of stability, performance, reliability, efficiency, and robustness, which require tight integration of computing, communication, and control technological systems.

Critical infrastructures have always been the target of criminals and are affected by security threats because of their complexity and cyber-security connectivity. These CPSs face security breaches when people, processes, technology, or other components are being attacked or risk management systems are missing, inadequate, or fail in any way. The attackers target confidential data. Main scope of this project in reduce the unwanted data for the dataset.

2.4.1 ADVANTAGES OF PROPOSED SYSTEM

Advantages of Proposed System includes:

- Protection against data from theft.
- Protects the computer from being hacked.

- Minimizes computer freezing and crashes.
- Gives privacy to users
- Securing the user-aware network edge
- Securing mobile users' communications ‘
- Managing user-centric security

2.5 CONCLUSION

User-centric cyber security helps enterprises reduce the risk associated with fast-evolving end-user realities by reinforcing security closer to end users. User-centric cyber security is not the same as user security. User-centric cyber security is about answering peoples' needs in ways that preserve the integrity of the enterprise network and its assets. User security can almost seem like a matter of protecting the network from the user — securing it against vulnerabilities that user needs introduce. User-centric security has the greater value for enterprises. cyber-security systems are real-time and robust independent systems with high performances requirements.

3. SYSTEM ANALYSIS

3.1 INTRODUCTION

Whatever we think need not be feasible. It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

3.1.1 Technical Feasibility

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already. The technical feasibility of the software involves the analysis of the software development environment such as software development resources, human resources, hardware resources.

3.1.2 Economical Feasibility

Development of this application is highly economically feasible. The organization needed not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. If we are doing so, we can attain the maximum usability of the corresponding.

3.2 SOFTWARE REQUIREMENT SPECIFICATION

The production of the requirements stage of the software development process is **Software Requirements Specifications (SRS)** (also called a **requirements document**). This report lays a foundation for software engineering activities and is constructing when entire requirements are elicited and analyzed. **SRS** is a formal report, which acts as a representation of software that enables the customers to review whether it (SRS) is according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. The two methods create entirely various situations and establish different purposes for the document altogether. The first case, SRS, is used to define the needs and expectation of the users. The second case, SRS, is written for various purposes and serves as a contract document between customer and developer.

System requirement specification is a structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the system development life cycle domain, typically performs a function between the business side of an enterprise and the information technology department or external service providers, Project are subject to three sorts of requirements:

- **Business Requirements**

Business requirements in the context of software engineering or the software development life cycle, is the concept of eliciting and documenting business requirements of business users such as customers, employees, and vendors early in the development cycle of a system to guide the design of the future system.

- **Product Requirements**

Product requirements prescribe properties of a system or product. Process requirements prescribe activities to be performed by the developing organization.

- **Process Requirements**

Process requirements specify the methodologies that must be followed, and constraints that the organization must obey.

Role of SRS

An SRS forms the basis of an organization's entire project. It sets out the framework that all the development teams will follow. It provides critical information to all the teams, including development, operations, quality assurance (QA) and maintenance, ensuring the teams are in agreement.

Using the SRS helps an enterprise confirm that the requirements are fulfilled and helps business leaders make decisions about the lifecycle of their product, such as when to retire a feature.

In addition, writing an SRS can help developers reduce the time and effort necessary to meet their goals as well as save money on the cost of development.

Scope:

SRS only describes the requirements of the system. It is meant for the user by the developers, and also be the basic for validating the final delivery of system. Any changes made to the requirements in the future will have to go through a formula change approval process. The developer is responsible for asking for clarification, where necessary, and will not make any alterations without the permission of the client.

Definitions, Acronyms and Abbreviations Software

Requirements Specification

It's a description of a particular software product, program or set of programs that performs a set of function in target environment.

References

IEEE Std. 830-1993. IEEE Recommended Practice for Software Requirements Specifications thy Sierra and Bert Bates.

Overview

The SRS contains the details of process, DFD's, functions of the product, user characteristics. The non-functional requirements if any are also specified.

3.2.1 User requirements

Understanding user requirements is an integral part of systems design and is critical to the success of a support system (Shen et al., 2004). The ISO 13407 standard specifies needs and requirements. A customer-friendly support-system design begins with a thorough understanding of the needs and requirements of the customer. The benefits can include increased productivity, enhanced quality of work, reductions in support and training costs and improved user satisfaction. Requirements analysis is not a simple process.

Functional Requirements

Functional requirement should include function performed by a specific screen outline work-flows performed by the system and other business or compliance requirement the system must meet. Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified. The functional specification describes what the system must do, how the system does it is described in the design specification. If a user requirement specification was written, all requirements outlined in the user requirements specifications should be addressed in the functional requirements.

Non Functional Requirements

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy (i.e. how precise are the systems numerical answers).

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the

- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

3.2.2 SOFTWARE REQUIREMENTS

- **Operating system** : Windows 7 Ultimate and above versions
- **Coding Language** : Python (version 3.7)
- **Front-End** : HTML, CSS , Javascript.
- **Data Base** : MySQL
- **Web Framework** : Django

3.2.3 HARWARE REQUIREMENTS

- **Processor** : Pentium IV 2.4 GHz.
- **Hard Disk** : 40 GB.
- **Monitor** : 14' Colour Monitor.
- **Mouse** : Optical Mouse.
- **Ram** : 512 Mb.

3.3 CONTENT DIAGRAM

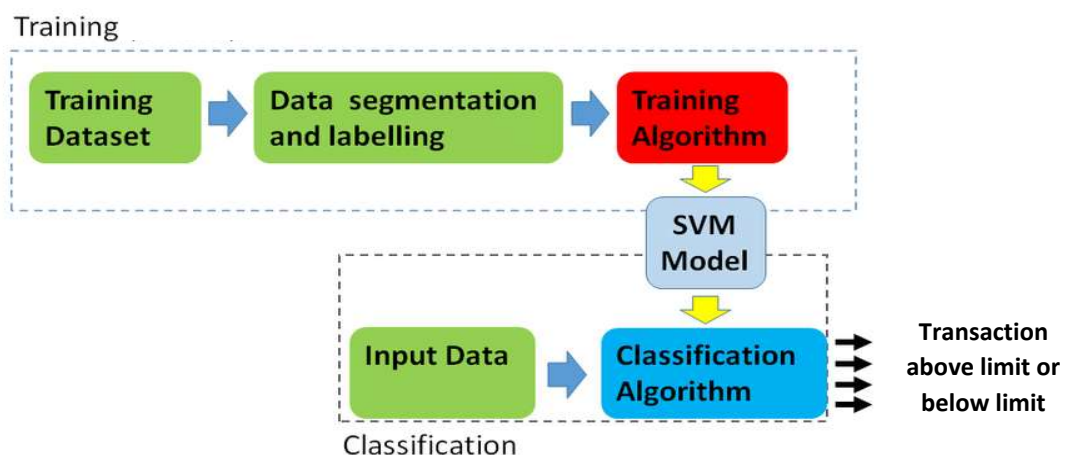


Fig 3.3.1 Content diagram

3.4 ALGORITHM AND FLOWCHART

Support Vector Machine (SVM) Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

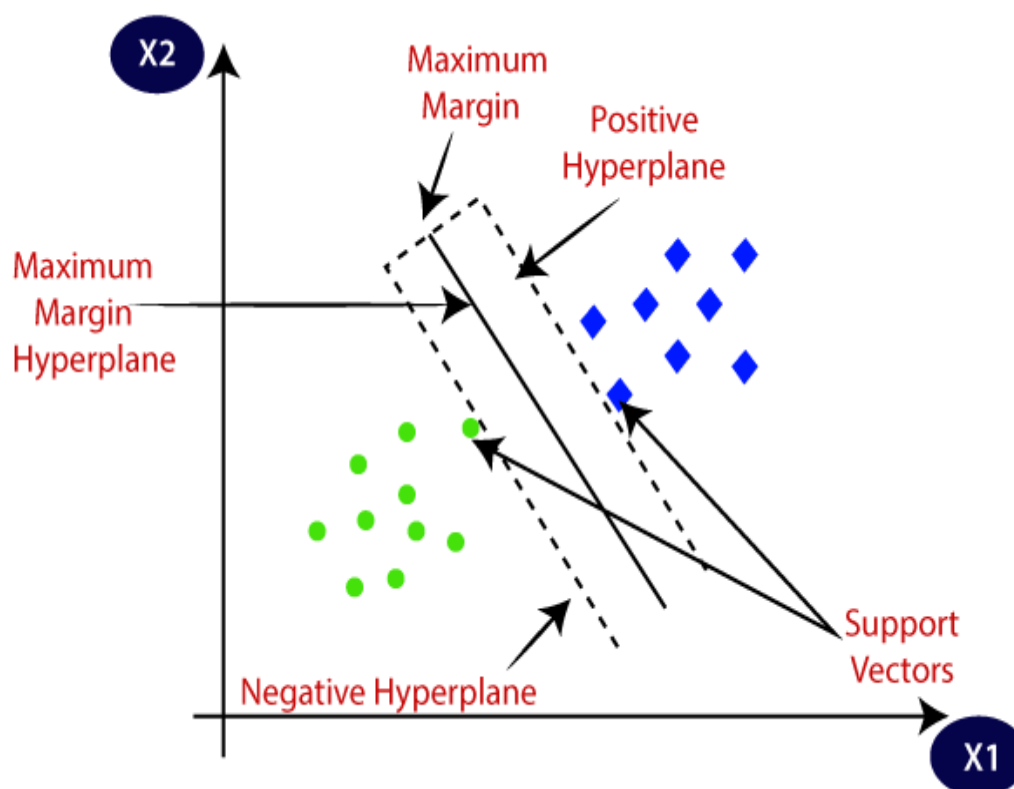


Fig 3.4.1 Support Vector Machine Algorithm

Assumptions of Support Vector Machine Algorithm

SVMs can be defined as linear classifiers under the following two assumptions:

- The margin should be as large as possible.
- The support vectors are the most useful data points because they are the ones most likely to be incorrectly classified.

Types of SVM

SVM can be of two types:

Linear SVM:

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM:

Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

Hyperplane and Support Vectors in the SVM algorithm:

Hyperplane:

There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

Support Vectors:

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

Why do we use SVM?

Below are some points that explains why should we use Support Vector Machine Algorithm:

- SVM works relatively well when there is a clear margin of separation between classes.
- SVM is effective in cases where the dimensions are greater than the number of samples.

How SVM Works?

SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

For example, consider the following figure, in which the data points fall into two different categories.

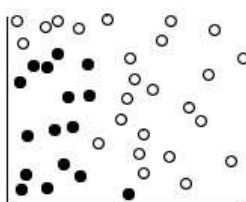


Fig 3.4.2 Original Dataset

The two categories can be separated with a curve, as shown in the following figure.

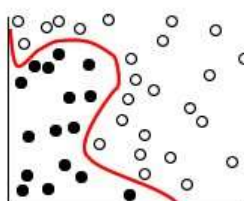


Fig 3.4.3 Data with separator added

After the transformation, the boundary between the two categories can be defined

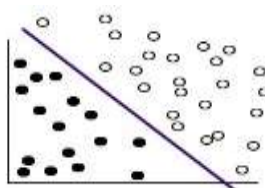


Fig 3.4.4 Transformed data

The mathematical function used for the transformation is known as the **kernel** function. SVM Modeler supports the following kernel types:

- Linear
- Polynomial
- Radial basis function (RBF)
- Sigmoid

A linear kernel function is recommended when linear separation of the data is straightforward. In other cases, one of the other functions should be used. You will need to experiment with the different functions to obtain the best model in each case, as they each use different algorithms and parameters.

Applications of Support Vector Machine

As we have seen, SVMs depends on **supervised learning** algorithms. The aim of using SVM is to correctly classify unseen data. SVMs have a number of applications. Some common applications of SVM are-

- **Face detection** – SVM classify parts of the image as a face and non-face and create a square boundary around the face.
- **Text and hypertext categorization** – SVMs allow Text and hypertext categorization for both inductive and transductive models. They use training data to classify documents into different categories. It categorizes on the basis of the score generated and then compares with the threshold value.
- **Classification of images** – Use of SVMs provides better search accuracy for image classification. It provides better accuracy in comparison to the traditional query-based searching techniques.
- **Bioinformatics** – It includes protein classification and cancer classification. We use SVM for identifying the classification of genes, patients on the basis of genes and other biological problems.
- **Protein fold and remote homology detection** – Apply SVM algorithms for protein remote homology detection.
- **Handwriting recognition** – We use SVMs to recognize handwritten characters used widely.
- **Generalized predictive control(GPC)** – Use SVM based GPC to control chaotic dynamics with useful parameters.

Implementation of SVM using Python

For implementing SVM in Python we will start with the standard libraries import as follows –

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

SVM Kernels

In practice, SVM algorithm is implemented with kernel that transforms an input data space into the required form. SVM uses a technique called the kernel trick in which kernel takes a low dimensional input space and transforms it into a higher dimensional space. In simple words, kernel converts non-separable problems into separable problems by adding more dimensions to it. It makes SVM more powerful, flexible and accurate. The following are some of the types of kernels used by SVM.

Linear Kernel

It can be used as a dot product between any two observations. The formula of linear kernel is as below –

$$K(x, x_i) = \sum(x * x_i)$$

From the above formula, we can see that the product between two vectors say x & x_i is the sum of the multiplication of each pair of input values.

Polynomial Kernel

It is more generalized form of linear kernel and distinguish curved or nonlinear input space. Following is the formula for polynomial kernel –

$$k(X, X_i) = 1 + \sum(X * X_i)^d$$

Here d is the degree of polynomial, which we need to specify manually in the learning algorithm.

Radial Basis Function (RBF) Kernel

RBF kernel, mostly used in SVM classification, maps input space in indefinite dimensional space. Following formula explains it mathematically –

$$K(x, x_i) = \exp(-\gamma \sum(x - x_i)^2)$$

Here, γ ranges from 0 to 1. We need to manually specify it in the learning algorithm. A good default value of γ is 0.1.

As we implemented SVM for linearly separable data, we can implement it in Python for the data that is not linearly separable. It can be done by using kernels.

Advantages of SVM

SVM classifiers offers great accuracy and work well with high dimensional space. SVM classifiers basically use a subset of training points hence in result uses very less memory.

Disadvantages of SVM

They have high training time hence in practice not suitable for large datasets. Another disadvantage is that SVM classifiers do not work well with overlapping classes.

FLOWCHART

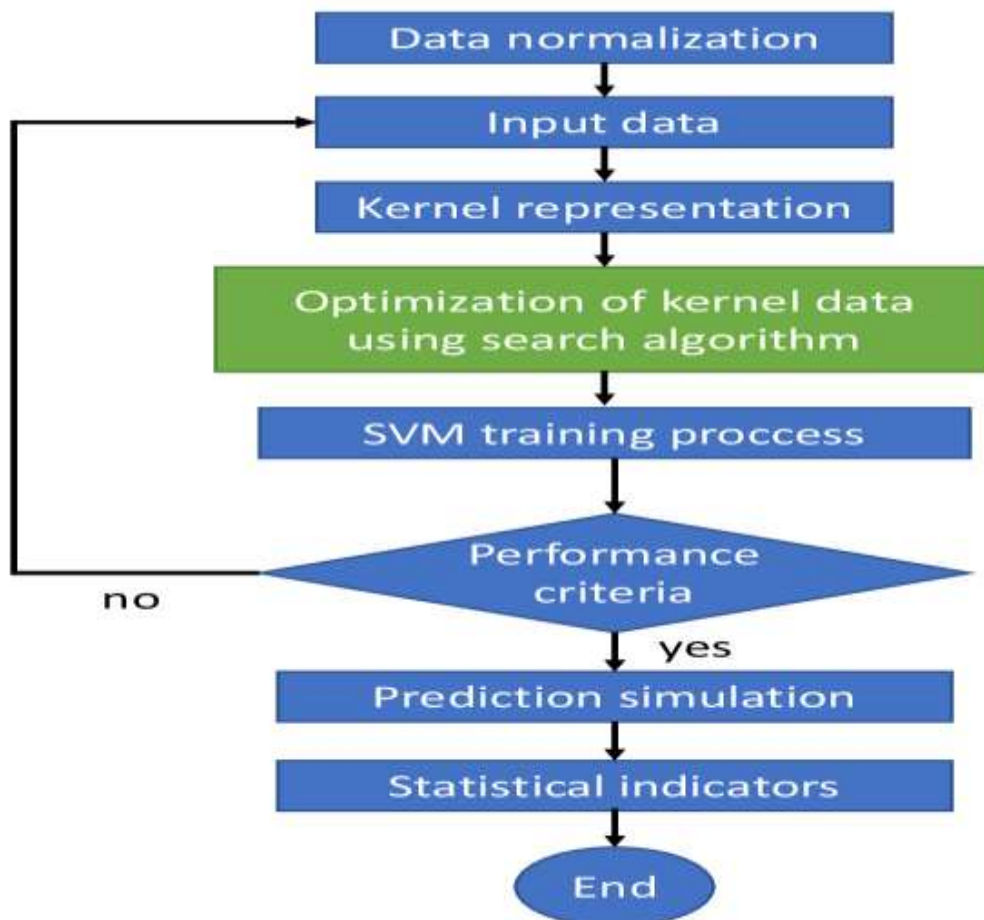


Fig 3.4.5 Flowchart

3.5 CONCLUSION

The analysis tells us the requirement specification of the project. The software requirements tell the required software and supporting files to process the data. The hardware requirements tell about the hardware components required to run the software. The various requirements of the system are selected through rigorous survey: the development is done in such a way that we ensure that all the requirements are met, and the software is up to the standards of professional software.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

4. DESIGN

4.1 INTRODUCTION

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analysed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage. The purpose of the design phase is to plan a solution of the problem specified by the requirement document.

This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided. During, Detailed Design, the internal logic of each of the modules specified in system design is decided. During this phase,

the details of the data of a module is usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented.

In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules. In other words, in system design the attention is on what components are needed, while in detailed design how the components can be implemented in software is the issue. Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified.

During the system design activities, Developers bridge the gap between the requirements specification, produced during requirements elicitation and analysis, and the system that is delivered to the user. Design is the place where the quality is fostered in development.

4.2 UML DIAGRAMS

The Unifies Modelling Language (UML) is a standard language for writing software blueprint. The UML may be used to visualize, specify, construct, and document the artifacts of a software intensive system. It is a very expressive language, addressing all the views needed to develop and then deploy such systems. The UML has its efficient use in the design phase of a system.

The vocabulary of the UML encompasses three kinds of building blocks:

- Things
- Relationships
- Diagrams

The diagrams that are employed to design this project are:

- Class diagram
- Use Case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- Component diagram

- Deployment diagram

All these Diagrams give the users a clear idea of design of the system.

4.2.1 CLASS DIAGRAM

Class diagram are widely used to describe the types of objects in a system and their relationships. Class diagrams describe three different perspectives when designing a system, conceptual, specifications and implementation.

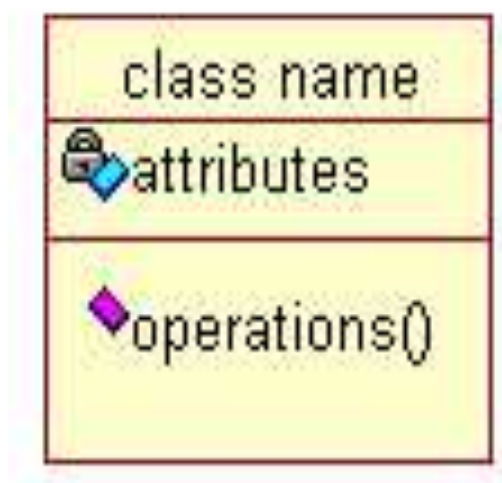


Fig 4.2.1.1 class diagram template

Class: Classes are composed of three things: names, dependency, generalization, and association.

Dependency: It is a using relationship that states a change in specification of one thing may affect another thing that uses it, but not necessarily the reverse.

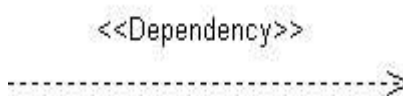


Fig 4.2.1.2 Dependency symbol

Generalization:

It is a relationship between general things and a more specific kind of those things that may be used to replace the general things. It is sometimes called as “is a kind

of' relationship. This gives the parent and child relation between actors in a usecase diagram.

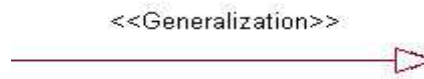


Fig 4.2.1.3 Genralization symbol

Association:

It is a structural relationship that objects of one thing are connected to objects of another. Aggregation is a plain association between two classes representing a structural relationship between peers; it is also called as whole part relationship in which one class represents a larger thing consists of smaller things.



Fig 4.2.1.4 Assosiation



Fig 4.2.1.5 Aggregation

Class Diagram of the system

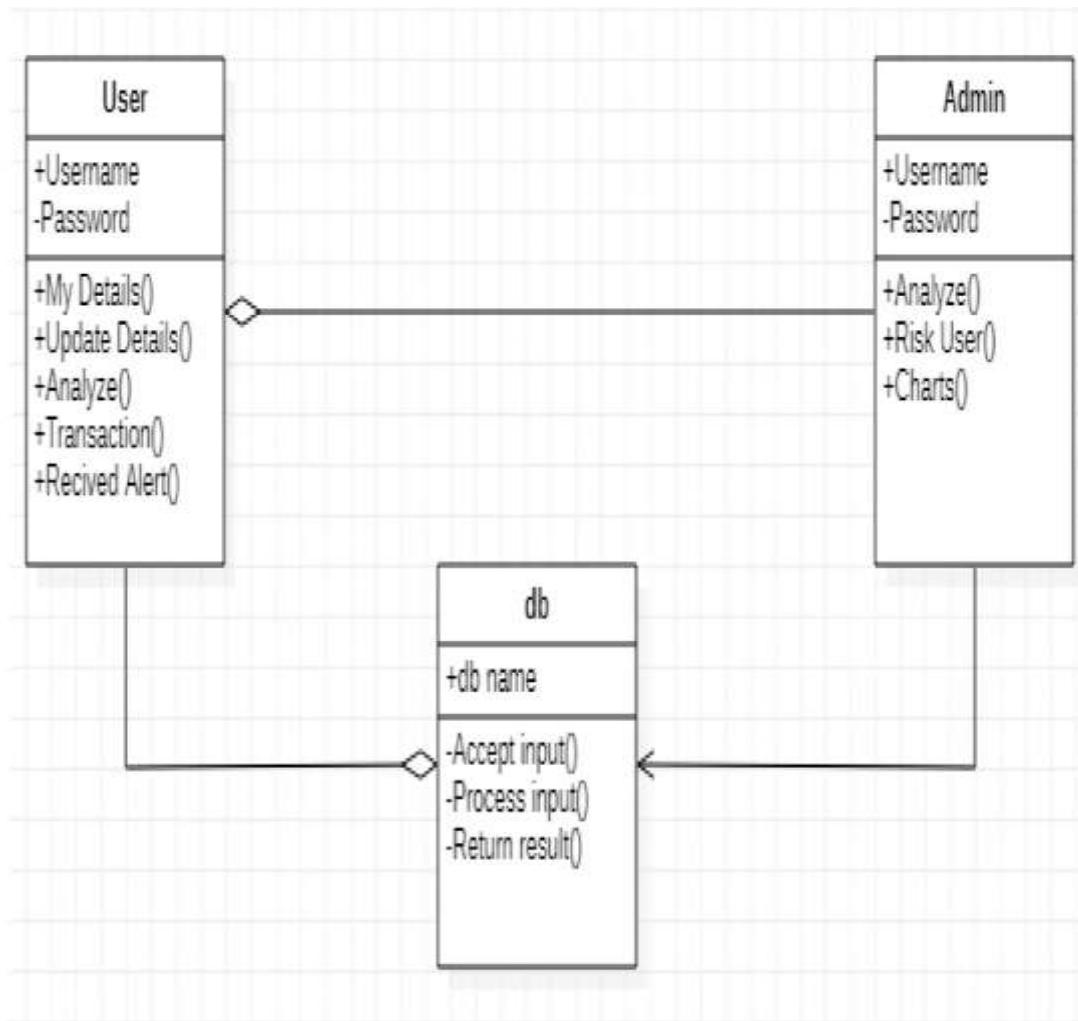
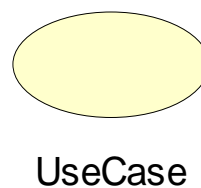
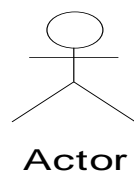


Fig 4.2.1.6 Class diagram of the system

4.2.2 USE CASE DIAGRAM

A use case is a set of scenarios that describe the interaction between the user and a system. These are used for modeling and organizing the behaviors of a system. A use case diagram displays the relationship among actors and use cases.

The two main components of a use case diagram are:



4.2.2 Use Case diagram

User:

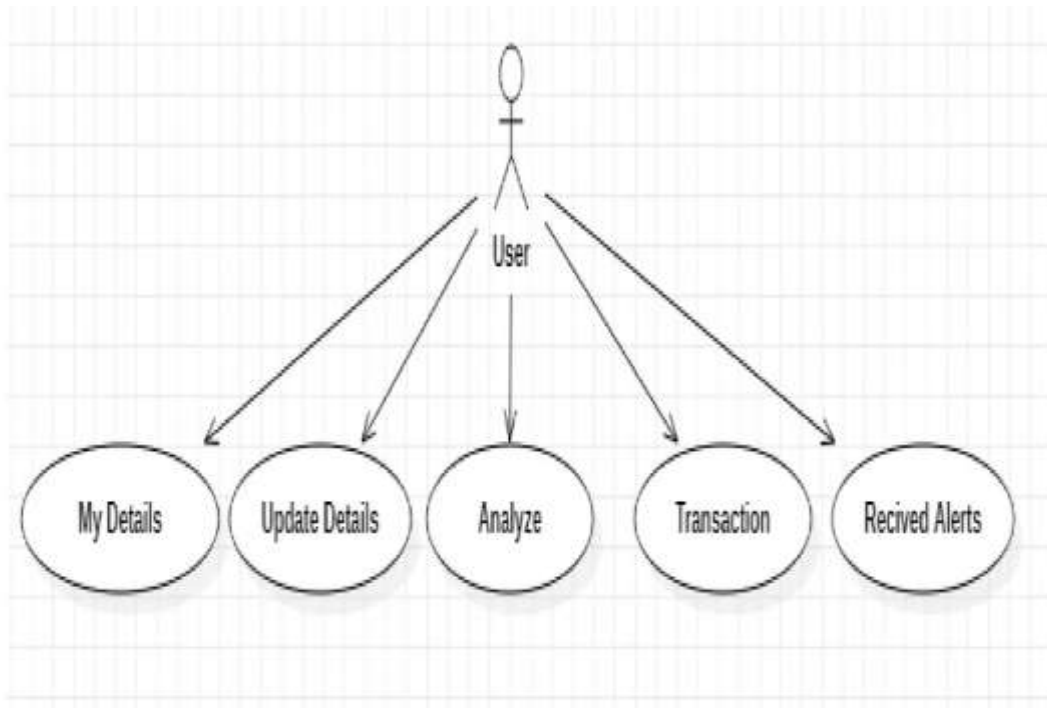


Fig 4.2.2.1 User Use Case Diagram

Admin:

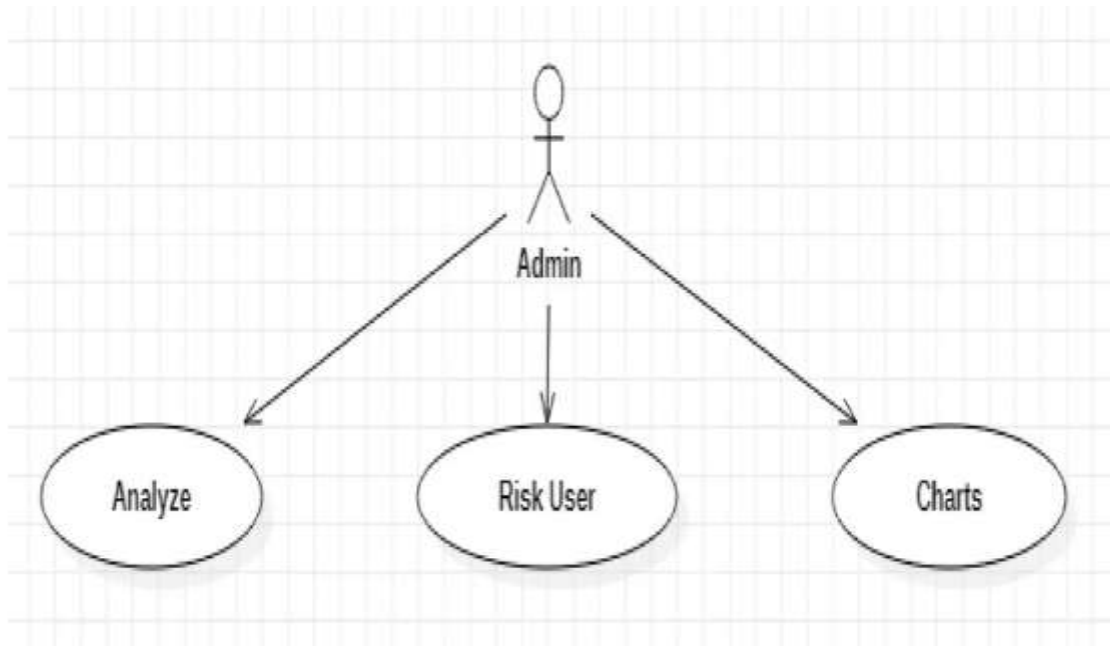


Fig 4.2.2.2 Admin Use Case Diagram

4.2.3 SEQUENCE DIAGRAM

The Sequence Diagrams emphasizes the Time ordering of messages between objects. It models collaboration of objects based on a time sequence. Sequence diagrams show a detailed flow for a specific use case or even just part of a specific use case. They are almost self-explanatory; they show the calls between the different objects in their sequence and can show, at a detailed level, different calls to different objects.

Object Lifeline: An object lifeline is the vertical dashed line that represents the existence of an object over a period of time. Objects may be created during the interaction, their lifelines start with the receipt of the message stereotyped as create. Objects may be destroyed during the interaction. Their lifeline end with the receipt of the message stereotyped as destroys.

Focus of control: Focus of control is a tall thin rectangle that shows the period of time during which an object is performing an action, either directly or through a subordinate procedure, the top of the rectangle is aligned with the start of the action; the bottom is aligned with its completion.

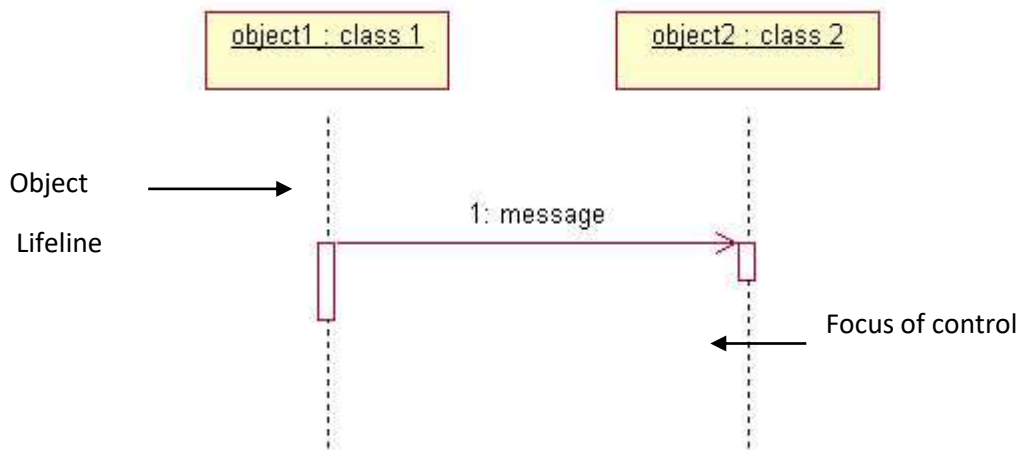


Fig 4.2.3.1 Sequence diagram

Sequence Diagram of the system

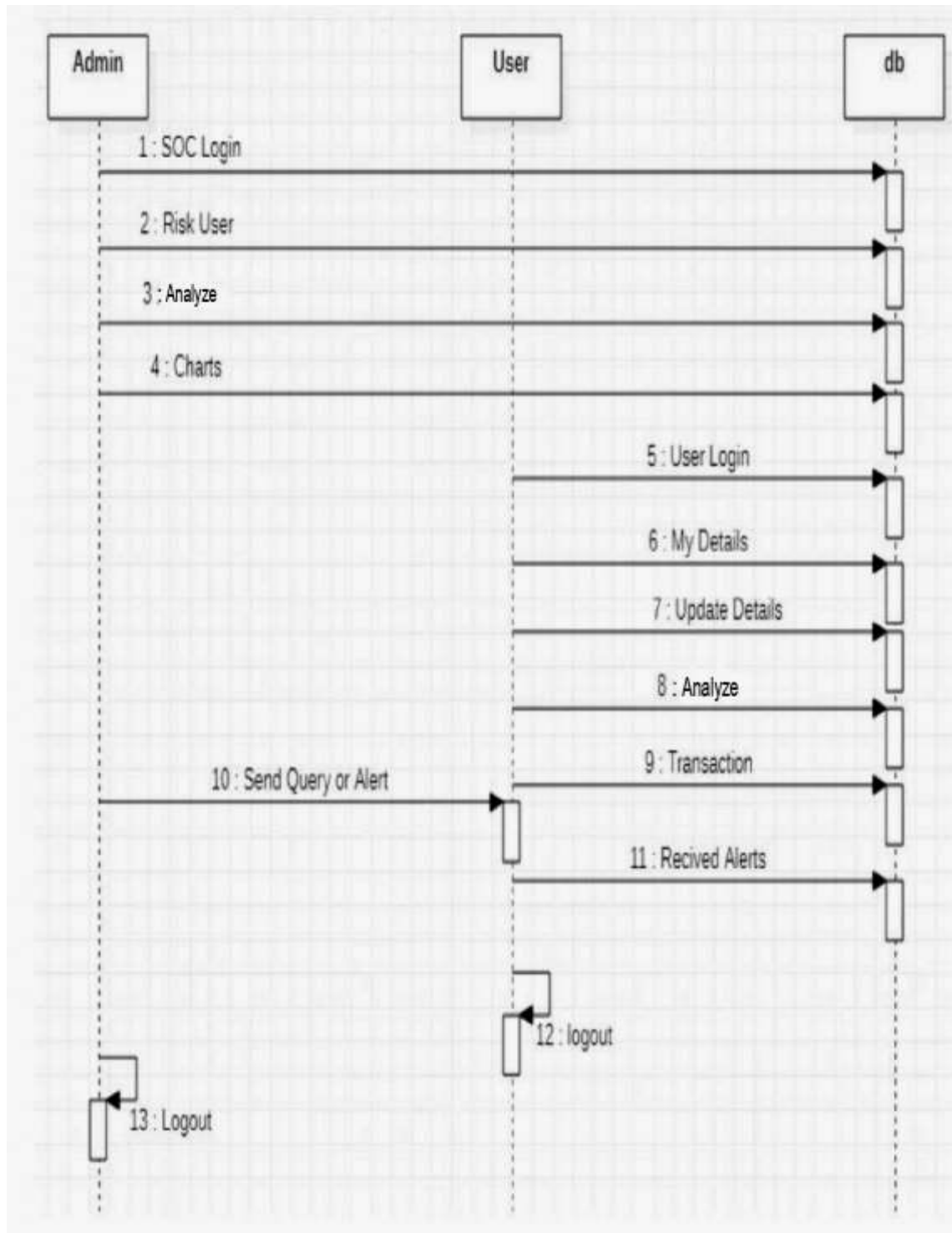


Fig 4.2.3.2 Sequence diagram of the system

4.2.4 COLLABORATION DIAGRAM

A collaboration diagram emphasizes the structural organization of objects that send and receive messages. It describes messages, interactions among objects in terms of sequenced messages.

Collaboration diagram represent a combination of information taken from class, sequence, and use case diagrams, describing both the static structure and dynamic behaviour of a system.

Collaboration diagram of the system

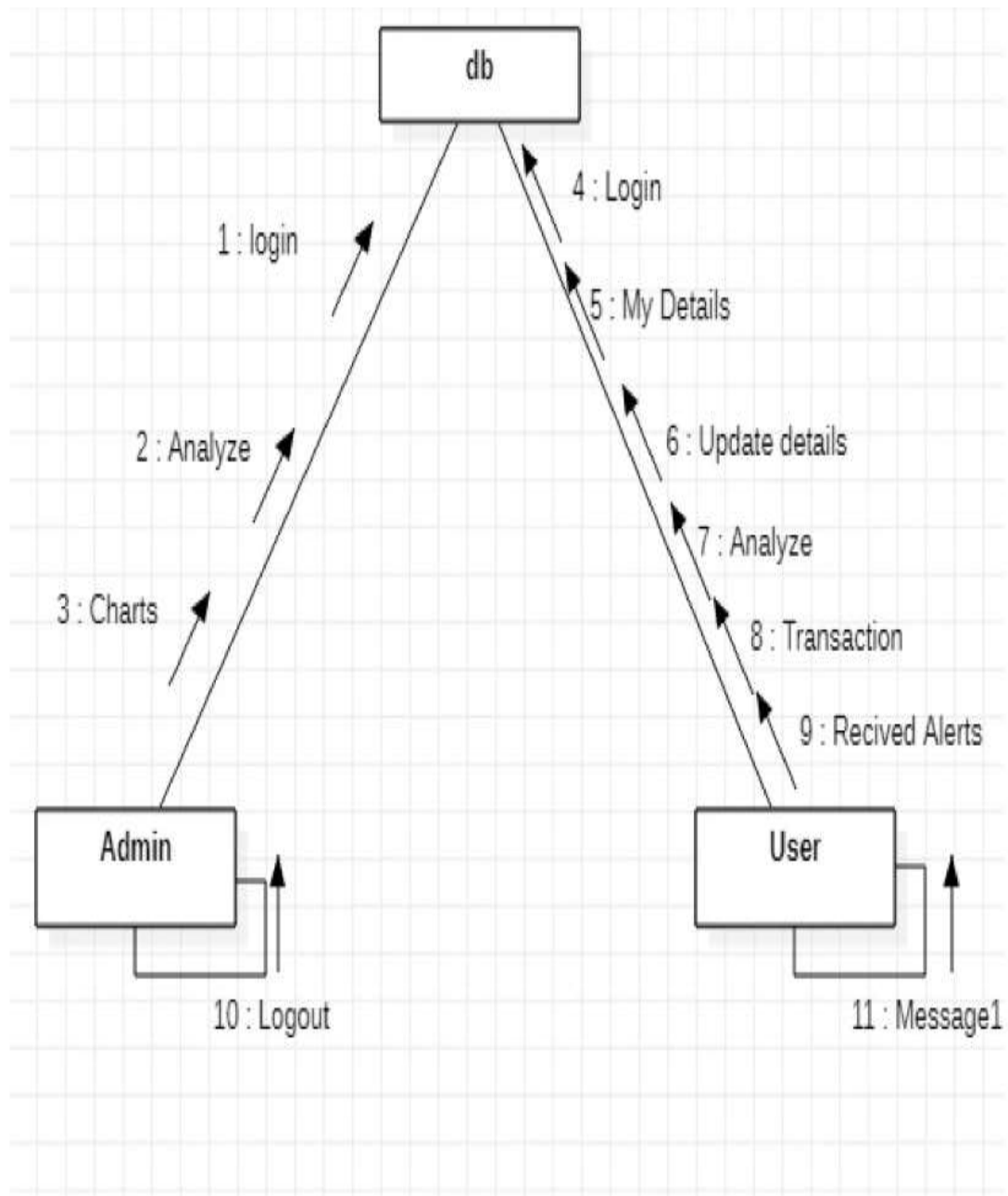


Fig 4.2.4.1 Collaboration diagram of the system

4.2.5 ACTIVITY DIAGRAM

Activity diagram shows the flow of activity to activity. It describes the workflow behaviour of a system. Activity diagrams are similar to state chart diagram because activities are the states of doing something. The diagrams describe the state of activity by showing the sequence of activity performed.







Symbol	Symbol Name	Meaning
	Start State	This is the starting of an activity
	Stop State	This is the ending of an activity
	Action State	An action state is an activity state that cannot be further decomposed
	Activity State	Activity states can be further more decomposed (non atomic).
	Branch	This specifies alternate paths taken based on some Boolean expression
	Transition/Link	This shows flow of control from one activity/action state to another.

Table 4.2.5.1 Activity diagram Elements

Activity diagram of the System

User

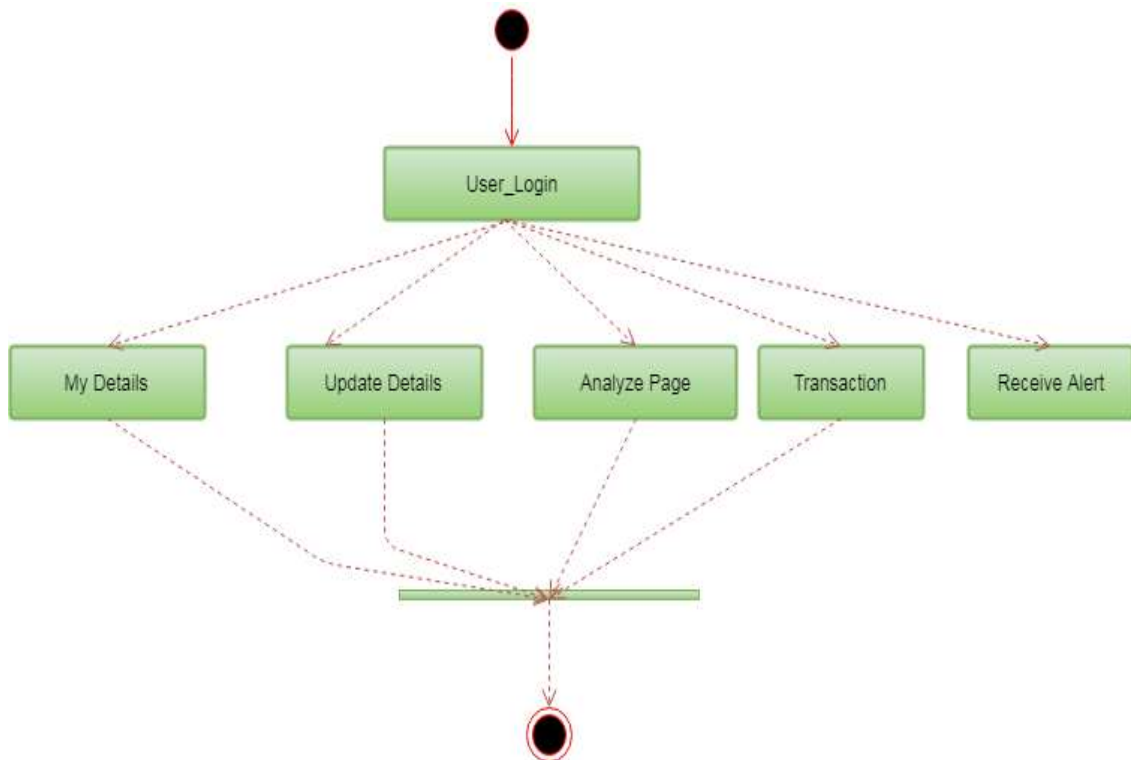


Fig 4.2.5.1 user activity diagram of the system

Admin

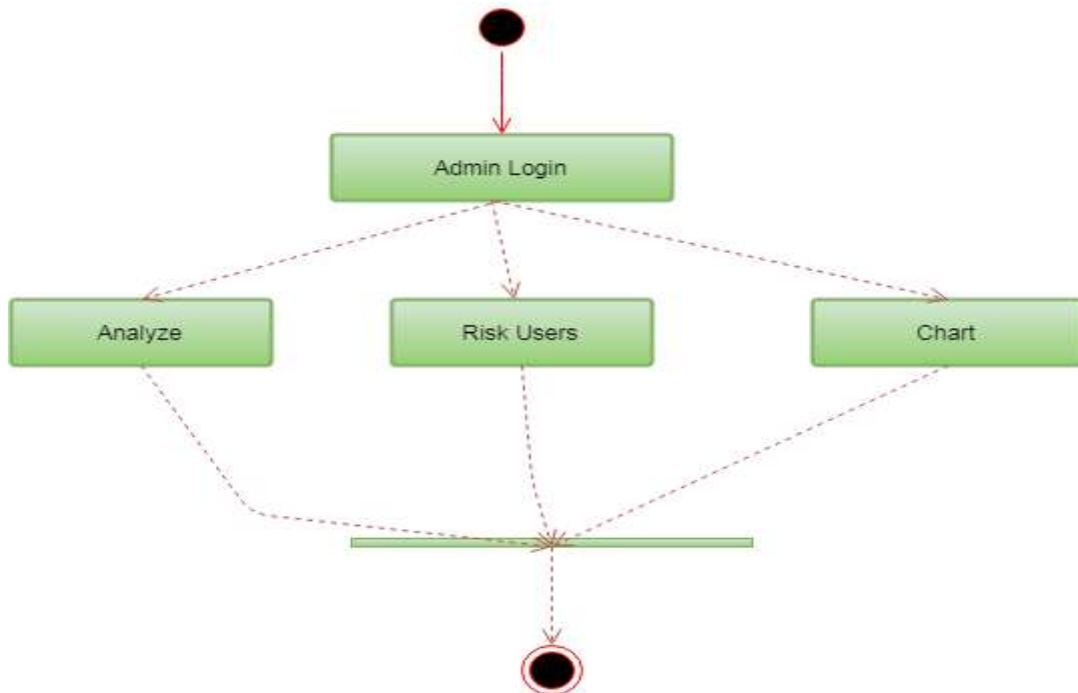


Fig 4.2.5.2 Admin Activity diagram of the system

4.2.6 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.

Notation of a Component Diagram

Component

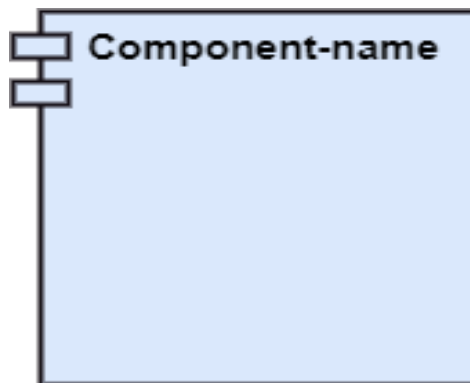


Fig 4.2.6.1 notation of component

Node

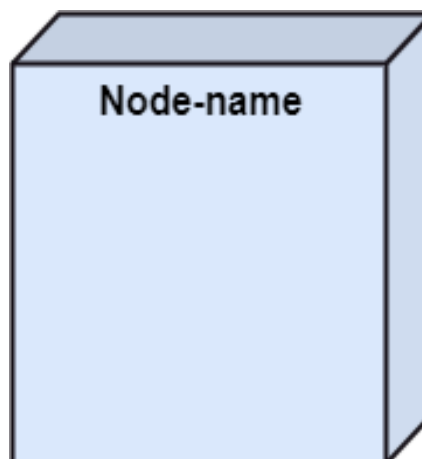


Fig 4.2.6.2 notation of node

Component Diagram of the system

User

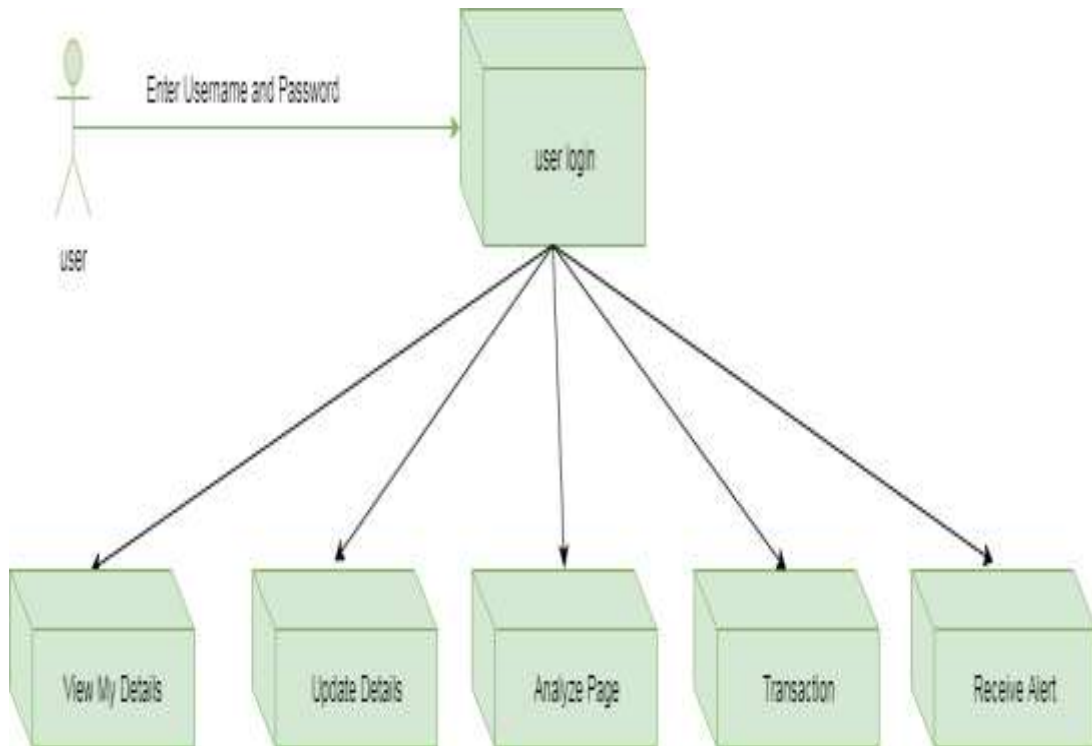


Fig 4.2.6.3 User Component Diagram

Admin

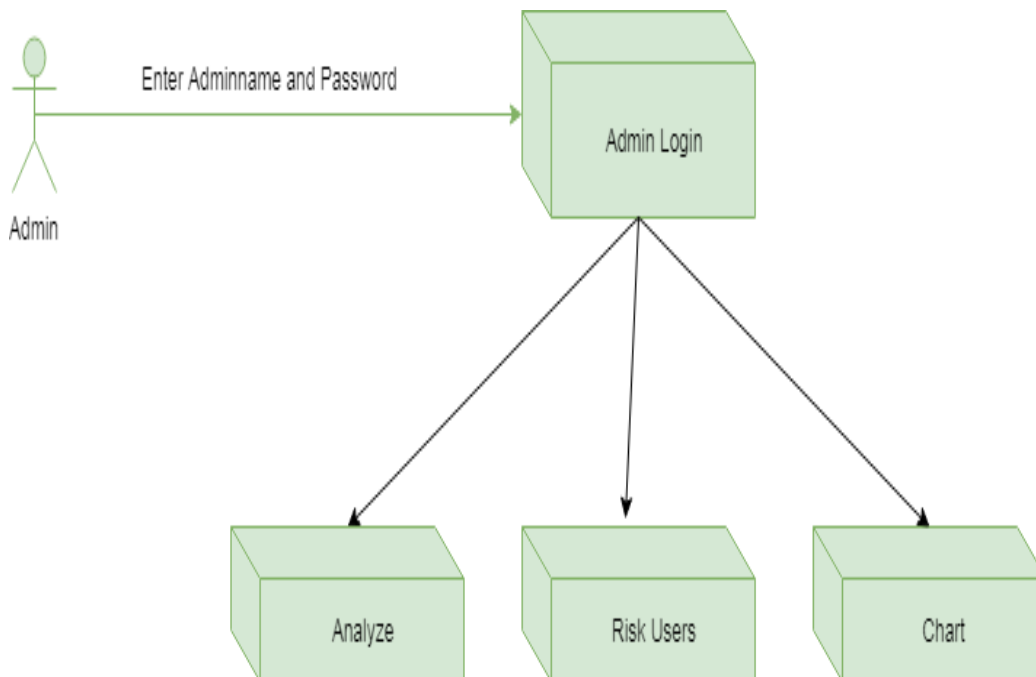


Fig 4.2.6.4 Admin Component Diagram

4.2.7 DEPLOYMENT DIAGRAM

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

The main purpose of the deployment diagram is to represent how software is installed on the hardware component. It depicts in what manner a software interacts with hardware to perform its execution. The deployment diagram does not focus on the logical components of the system, but it put its attention on the hardware topology.

Deployment Diagram of the System

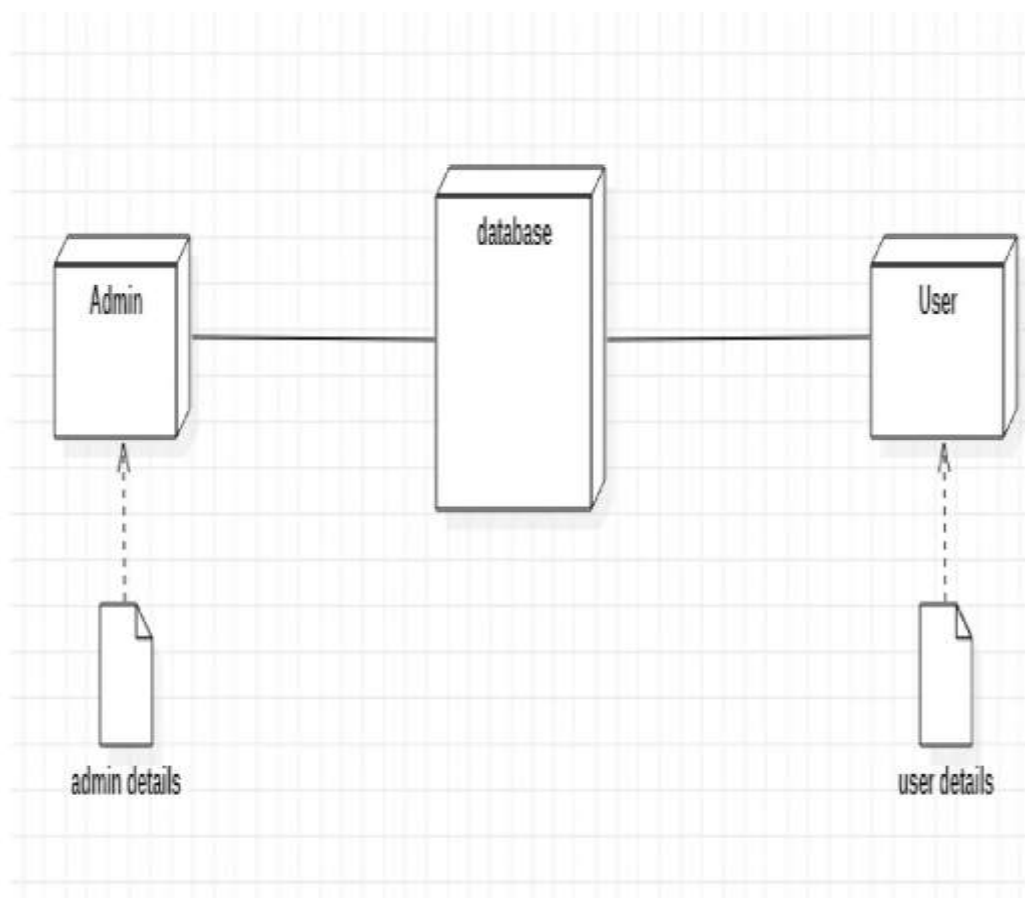


Fig 4.2.7.1 Deployment Diagram

4.3 MODULE DESIGN AND ORGANIZATION

Following modules are utilized for designing of our system, They include:

- Cyber Analysis
- Dataset modification
- Data Reduction
- Risky user Detection

Cyber Analysis

Cyber threat_analysis is a process in which the knowledge of internal and external information vulnerabilities pertinent to a particular organization is matched against real-world cyber-attacks. With respect to cyber security, this threat-oriented approach to combating cyber-attacks represents a smooth transition from a state of reactive security to a state of proactive one. Moreover, the desired result of a threat assessment is to give best practices on how to maximize the protective instruments with respect to availability, confidentiality and integrity, without turning back to usability and functionality conditions. CYBER ANALYSIS.

A threat could be anything that leads to interruption, meddling or destruction of any valuable service or item existing in the firm's repertoire. Whether of "human" or "nonhuman" origin, the analysis must scrutinize each element that may bring about conceivable security risk.

Dataset Modification

If a dataset in your dashboard contains many dataset objects, you can hide specific dataset objects from display in the Datasets panel. For example, if you decide to import a large amount of data from a file, but do not remove every unwanted data column before importing the data into Web, you can hide the unwanted attributes and metrics,

To hide dataset objects in the Datasets panel, To show hidden objects in the Datasets panel, To rename a dataset object, To create a metric based on an attribute, To create an attribute based on a metric, To define the geo role for an attribute, To create an attribute with additional time information, To replace a dataset object in the dashboard

Data Reduction

Improve storage efficiency through data reduction techniques and capacity optimization using data deduplication, compression, snapshots and thin provisioning. Data reduction via simply deleting unwanted or unneeded data is the most effective way to reduce a storing's data

Risky User Detection

False alarm immunity to prevent customer embarrassment, High detection rate to protect all kinds of goods from theft, Wide-exit coverage offers greater flexibility for entrance/exit layouts, Wide range of attractive designs complement any store décor, Sophisticated digital controller technology for optimum system performance.

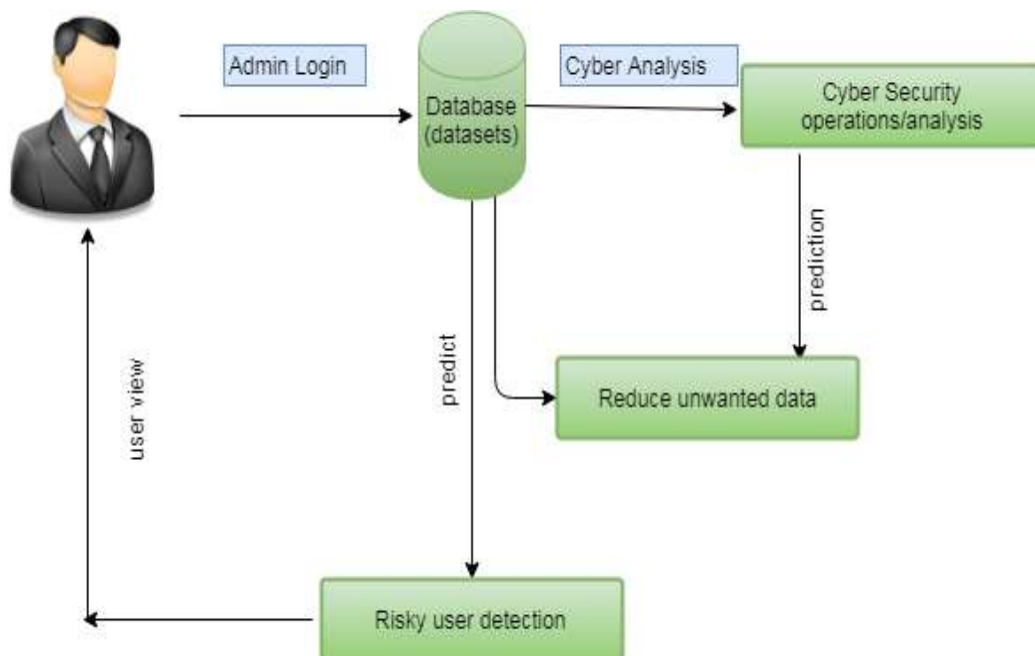


Fig 4.3.1 Module Design and Organization

4.4 CONCLUSION

The UML is a major step toward the standardization of software development. The standard has received widespread support from tool vendors and customers alike.

The UML includes specifications for nine different diagrams used to document different views of a software solution from project inception to installation and maintenance. The specifications define the elements of each model, how the models are assembled, and how they can be extended.

5. IMPLEMENTATION & RESULTS

5.1 INTRODUCTION

The implementation phase, the project plan is put into motion and the work of the project is performed. The project takes shape during the implementation phase. This phase involves the construction of the actual project result. Programmers are occupied with encoding, designers are involved in developing graphic material, contractors are building, the actual reorganization takes place.

5.2 EXPLANATION OF KEY FUNCTIONS

This module mainly focuses on improving an organization's threat detection, response and prevention capabilities by unifying and coordinating by using machine learning framework.

Initially, the most important thing to do after data collection is to understand the problem that we have to solve and extract relevant data and to remove unwanted information. To identify raw data we make use of segmentation and labelling techniques. We make use of transactional data to identify threat. The train and classify the data using classification algorithm i.e. SVM classifier.

- In this module, for training and testing data set, we make use of labelling for identifying raw data and adding one or more meaningful and informative labels to provide context so that a machine learning model can learn from it.
- Model evaluation is the process of using different evaluation metrics to evaluate a machine learning model's performance. Based on Model accuracy, we can choose one of the one best algorithms.
- In this module, based on Model accuracy we can select best (Support Vector Machine) ML algorithm and develop that to User Centric Machine Learning Framework for Cyber Security Operations Centre.

5.3 METHOD OF IMPLEMENTATION

Introduction to Python

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop

applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is *easy to learn* yet powerful and versatile scripting language, which makes it attractive for Application Development. Python's syntax and *dynamic typing* with its interpreted nature make it an ideal language for scripting and rapid application development. It supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.

We don't need to use data types to declare variable because it is *dynamically typed* so we can write `a=10` to assign an integer value in an integer variable. Python makes the development and debugging *fast* because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

What is Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

What can Python do

1. Python can be used on a server to create web applications.
2. Python can be used alongside software to create workflows.
3. Python can connect to database systems. It can also read and modify files.
4. Python can be used to handle big data and perform complex mathematics.
5. Python can be used for rapid prototyping, or for production-ready software development.

Why Python

1. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
2. Python has a simple syntax similar to the English language.
3. Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
4. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
5. Python can be treated in a procedural way, an object-orientated way or a functional way.

Python Features

1) Easy to Learn and Use

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

2) Expressive Language

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type `print("Hello World")`. It will take only one line to execute, while Java or C takes multiple lines.

3) Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

5) Free and Open Source

Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

6) Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

Application of Python

Mentioned domains we can find application of python:

- Web Development.
- Game Development.
- Machine Learning and Artificial Intelligence.
- Data Science and Data Visualization.
- Desktop GUI.
- Web Scraping Applications.
- Business Applications.

Introduction to Machine Learning

Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959.

With the help of sample historical data, which is known as training data, machine learning algorithms build a mathematical model that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

A machine has the ability to learn if it can improve its performance by gaining more data. It learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

There are 3 classifications of Machine learning includes Supervised learning, Unsupervised learning and Reinforcement learning. We implemented Supervised machine learning in our system.

Supervised Machine Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering.

Django

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) architecture. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

Django was design and developed by Lawrence journal world in 2003 and publicly released under BSD license in July 2005. Currently, DSF (Django Software Foundation) maintains its development and release cycle.

This framework uses a famous tag line: “The web framework for perfectionists with deadlines”. By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application.

MVT Structure has the following three parts –

Model: The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres).

View: The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/Javascript and Jinja files.

Template: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.

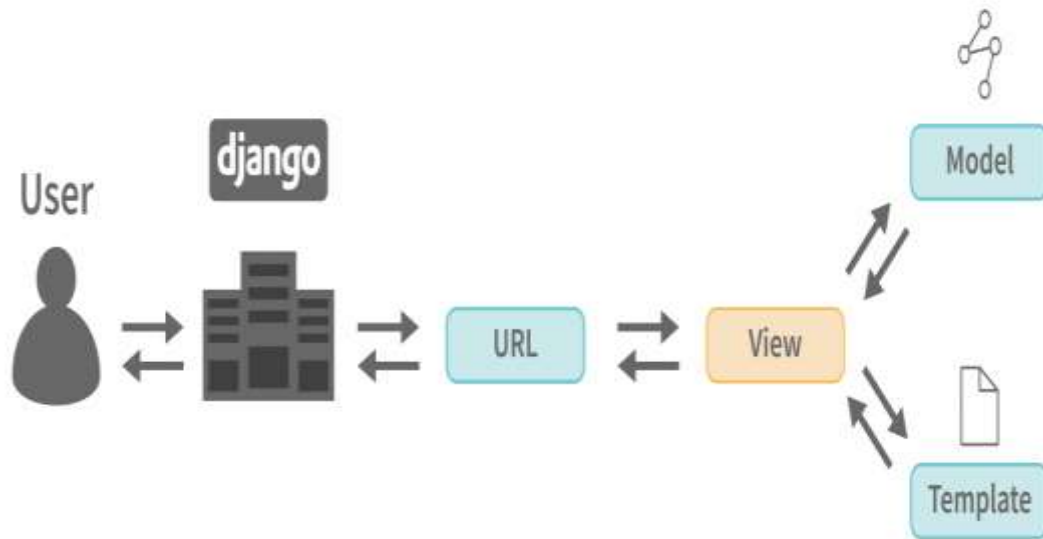


Fig 5.3.1 Django Architecture

MySQL

MySQL is a relational database management system, that stores data in separate tables rather than putting all the data in one big storeroom. The database structure is organized into physical files optimized for speed.

Python MySQL Connector is a Python driver that helps to integrate Python and MySQL. This Python MySQL library allows the conversion between Python and MySQL data types. MySQL Connector API is implemented using pure Python and does not require any third-party library.

XAMPP server

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the Apache Friends, and its native source code can be revised or modified by the audience. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself.

Standard libraries used for implementation:

➤ **NumPy**

NumPy stands for numeric python which is a python package for the computation and processing of the multidimensional and single dimensional array elements. It is an extension module of Python which is mostly written in C. It provides various functions which are capable of performing the numeric computations with a high speed. NumPy provides various powerful data structures, implementing multi-dimensional arrays and matrices. These data structures are used for the optimal computations regarding arrays and matrices.

➤ **Pandas**

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. Data analysis requires lots of processing, such as restructuring, cleaning or merging, etc. There are different tools available for fast data processing, such as Numpy, Scipy, Cython, and Panda. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools.

Pandas is built on top of the Numpy package, means Numpy is required for operating the Pandas. Before Pandas, Python was capable for data preparation, but it only provided limited support for data analysis. So, Pandas came into the picture and enhanced the capabilities of data analysis. It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., load, manipulate, prepare, model, and analyze.

➤ **Matplotlib**

Matplotlib is a Python library which is defined as a multi-platform data visualization library built on Numpy array. It can be used in python scripts, shell, web application, and other graphical user interface toolkit.

The plotting of numerical data is the responsibility of this library. It's for this reason that it's used in analysis of data. It's an open-source library that plots high-

definition figures such as pie charts, scatterplots, boxplots, and graphs, among other things.

➤ **Scikit-learn**

Scikit-learn (Sklarn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. Key concepts and features include: Algorithmic decision-making methods, including: Classification: identifying and categorizing data based on patterns.

5.4 CODE IMPLEMENTATION

5.4.1 SOURCE CODE

Urls.py (cyber_security_alert)

"""cyber_security_alert URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/1.11/topics/http/urls/>

Examples:

Function views

1. Add an import: from my_app import views
2. Add a URL to urlpatterns: url(r'^\$', views.home, name='home')

Class-based views

1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: url(r'^\$', Home.as_view(), name='home')

Including another URLconf

1. Import the include() function: from django.conf.urls import url, include
2. Add a URL to urlpatterns: url(r'^blog/', include('blog.urls'))

"""

from django.conf.urls import url

from django.contrib import admin

from cyber_alert import views as alert_view

from admins import views as admin_view

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', alert_view.admin_login, name="admin_login"),
    url(r'^admin_register/$', alert_view.admin_register, name="admin_register"),
    url(r'^giver_transaction/$', alert_view.giver_transaction, name="giver_transaction"),
    url(r'^analyze_page/$', alert_view.analyze_page, name="analyze_page"),
    url(r'^viewer/(?P<chart_type>\w+)', alert_view.viewer, name="viewer"),
    url(r'^update/$', alert_view.update, name="update"),
    url(r'^logout_page/$', alert_view.logout_page, name="logout_page"),
    url(r'^mydetails/$', alert_view.mydetails, name="mydetails"),
    url(r'^show/$', alert_view.show, name="show"),
    url(r'^receivealert/$', alert_view.receivealert, name="receivealert"),
    url(r'^admins/admin_page/$', admin_view.admin_page, name="admin_page"),
    url(r'^admins/analyze/$', admin_view.analyze, name="analyze"),
    url(r'^admins/adlogout/$', admin_view.adlogout, name="adlogout"),
    url(r'^admins/charts/(?P<chart_type>\w+)', admin_view.charts, name="charts"),
    url(r'^admins/riskuser/$', admin_view.riskuser, name="riskuser"),
    url(r'^admins/riskalert/(?P<tuser>\d+)$', admin_view.riskalert, name="riskalert"),
]
```

Settings.py(cyber_security_alert)

```
"""
```

Django settings for cyber_security_alert project.

Generated by 'django-admin startproject' using Django 1.11.5.

For more information on this file, see

<https://docs.djangoproject.com/en/1.11/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/1.11/ref/settings/>

```
"""
```

```
import os
```

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
```

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

```
# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/
```



```
# SECURITY WARNING: keep the secret key used in production secret!

SECRET_KEY = 'gn-jzi2u3%gw+olpxfrd%ye6210z3=$+(r@c5ly(%8j2$5)k77'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'cyber_alert',
    'admins'

]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'cyber_security_alert.urls'
```

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [((os.path.join(BASE_DIR, 'assests/templates')))],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [

                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

```

```
WSGI_APPLICATION = 'cyber_security_alert.wsgi.application'
```

```
# Database
```

```
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases
```

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'alarm',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    }
}

```

Password validation

<https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators>

```
AUTH_PASSWORD_VALIDATORS = [  
    {  
        'NAME':  
        'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',  
    },  
    {  
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',  
    },  
]
```

Internationalization

<https://docs.djangoproject.com/en/1.11/topics/i18n/>

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

Static files (CSS, JavaScript, Images)

<https://docs.djangoproject.com/en/1.11/howto/static-files/>

```

STATIC_URL = '/static/'
STATICFILES_DIRS= [os.path.join(BASE_DIR, 'assests/static')]
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'assests/media')

```

Recievealerts.html(User page):

```

<!DOCTYPE html>
{ % load staticfiles % }
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title</title>
<style>
body{
background: url("{ % static 'admin1.jpg' % }");
background-size: cover;
}
.menu table{
width:100%;
text-align:center;
}
.menu table td:hover{
background:rgb(0,0,0);
}
.menu table td{
background: #584b4f;
}
.menu table,.menu table th,.menu
table td {
border-collapse: collapse;
}
.menu table th,.menu table td {
padding: 15px;
}

```

```

.topic h1{
color:white;
padding:2px;
text-align:center;
border-style:none;
height:100px;
width:1330px;
float:left;
}
.giver
{
color: darkgrey;
font-family: cooper;
padding: 50px;
margin-right: 500px;
}
input:not([type]), input[type="email"i],
input[type="number" i],
input[type="password" i],
input[type="tel"i],
input[type="url" i],
input[type="text" i] {
padding: 5px 5px;
border-radius:10px;
margin-bottom:10px;
padding: 6px;
}
table1 {
border-collapse: collapse;
width: 100%;
}
th {
border: 1px solid #dddddd;
text-align: center;

```

```
padding: 8px;
}
.image1 {
background: url("{ % static'warning4.png' % }");
background-size: 100% 100%;
width: 700px;
height: 442px;
margin-top: -99px;
float: right;
}
</style>
</head>
<body style="background-color:#393e44">
<div class="topic"><h1 style="margin-
top:10px;margin-left:60px;
border-style:none;width:1300px;height:40px;
border-color:black;background:;>A User-Centric Machine Learning Framework
for Cyber Security Operations Center</h1></div>
<div class="menu">
<table>
<tr>
<td><a style="color:violet; font-family:cooper; text-decoration:none;"
href="{ % url 'mydetails' % } "> MY DETAILS </a></td>
<td><a style="color:violet;font-family:cooper; text-decoration: none;"
href="{ % url 'update' % } ">UPDATE DETAILS</a></td>
<td><a style="color:violet;font-family:cooper; text-decoration:none;" href="{ % url
'analyze_page' % } ">ANALYZE PAGE</a></td>
<td><a style="color:violet;font-family:cooper; text-decoration: none;"href="{ % url
'giver_transaction'% } ">TRANSACTION</a></td>
<td><a style="color:violet; font-family:cooper; text-decoration: none;" href="{ % url
'receivealert' % } ">RECEIVE ALERT</a></td>
<td><a style="color:violet; font-family:cooper; text-decoration: none;" href="{ %
url 'logout_page'% } ">LOGOUT</a></td>
</tr> </table> </div>
```

```

<div class="table1">
<table class="giver">
<tr style="border-style:groove">
<th>NAME</th>
<th>ALERT MESSAGE</th>
</tr>
{% for o in de %}
<tr>
<td class="td">{{ o.name }}</td>
<td class="td">{{ o.sendquery }}</td>
</tr>
{% endfor %}
</table>
</div>
<div class="image1">
</div>
</body>
</html>

```

5.4.2 OUTPUT SCREENS

User Output Screens :

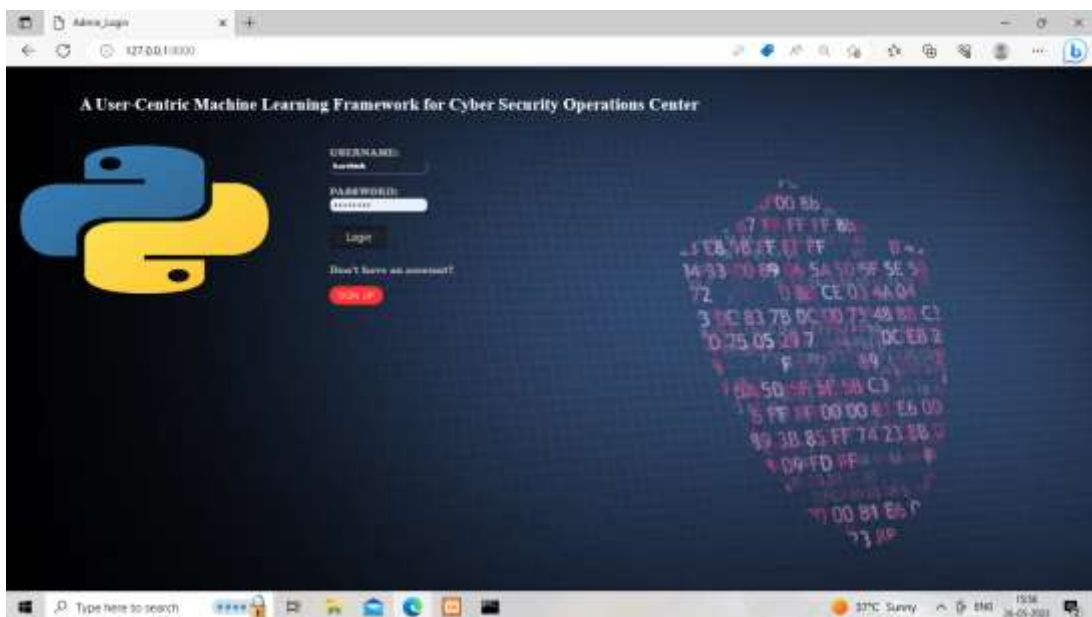
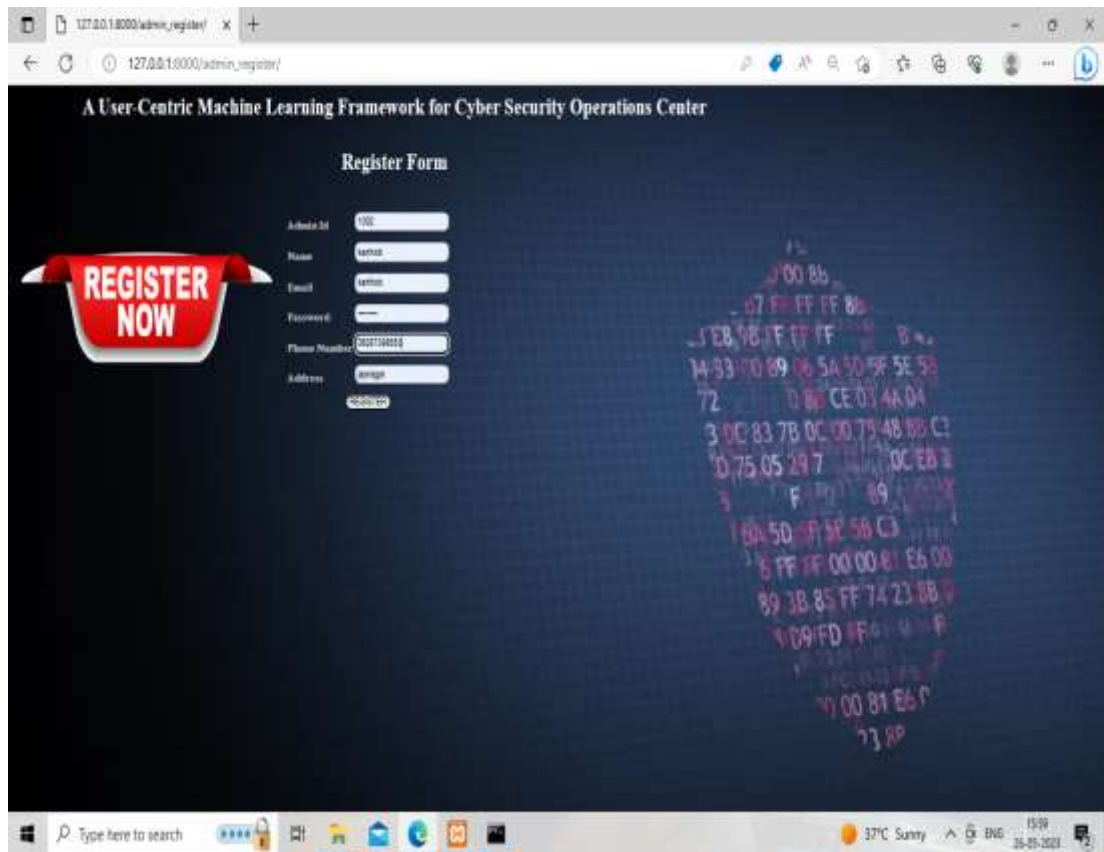


Fig 5.4.2.1 User Login Page



The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin_register/`. The page title is "A User-Centric Machine Learning Framework for Cyber Security Operations Center". The main content area has a dark blue background with a "REGISTER NOW" button on the left and a "Register Form" on the right. The form fields are: Admin Id (100), Name (admin), Email (admin), Password (admin), Phone Number (0012345678), and Address (Bangalore). A taskbar at the bottom shows the Windows search bar, task icons, and system tray with weather (37°C Sunny) and date (28-05-2023).

A User-Centric Machine Learning Framework for Cyber Security Operations Center

REGISTER NOW

Register Form

Admin Id:

Name:

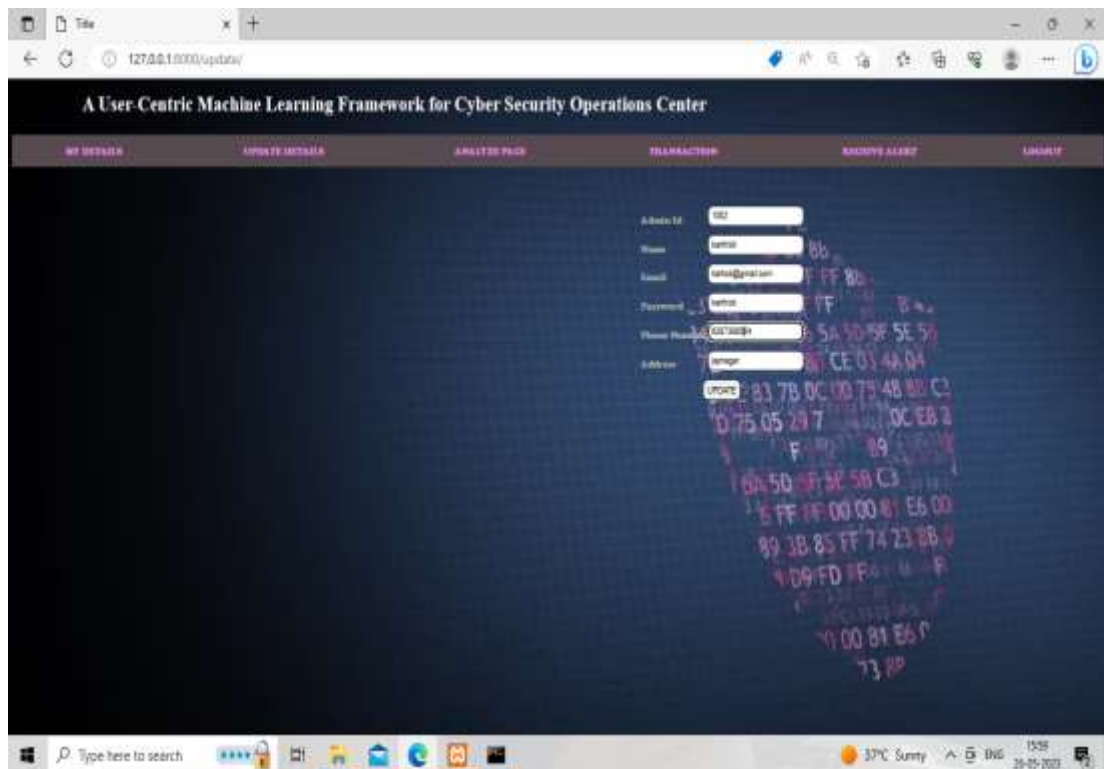
Email:

Password:

Phone Number:

Address:

Fig 5.4.2.2 User Register Page



The screenshot shows a web browser window with the URL `127.0.0.1:8000/update/`. The page title is "A User-Centric Machine Learning Framework for Cyber Security Operations Center". The main content area has a dark blue background with a navigation bar at the top containing links: MY DETAILS, ADMINISTRATION, ANALYSIS TOOL, TRANSACTION, SECURITY ALERT, and LOGIN. The "MY DETAILS" link is active. The form fields are: Admin Id (100), Name (admin), Email (admin@gmail.com), Password (admin), Phone Number (0012345678), and Address (Bangalore). There is an "UPDATE" button next to the Address field. A taskbar at the bottom shows the Windows search bar, task icons, and system tray with weather (37°C Sunny) and date (28-05-2023).

A User-Centric Machine Learning Framework for Cyber Security Operations Center

MY DETAILS | ADMINISTRATION | ANALYSIS TOOL | TRANSACTION | SECURITY ALERT | LOGIN

Admin Id:

Name:

Email:

Password:

Phone Number:

Address: **UPDATE**

Fig 5.4.2.3 User Update page

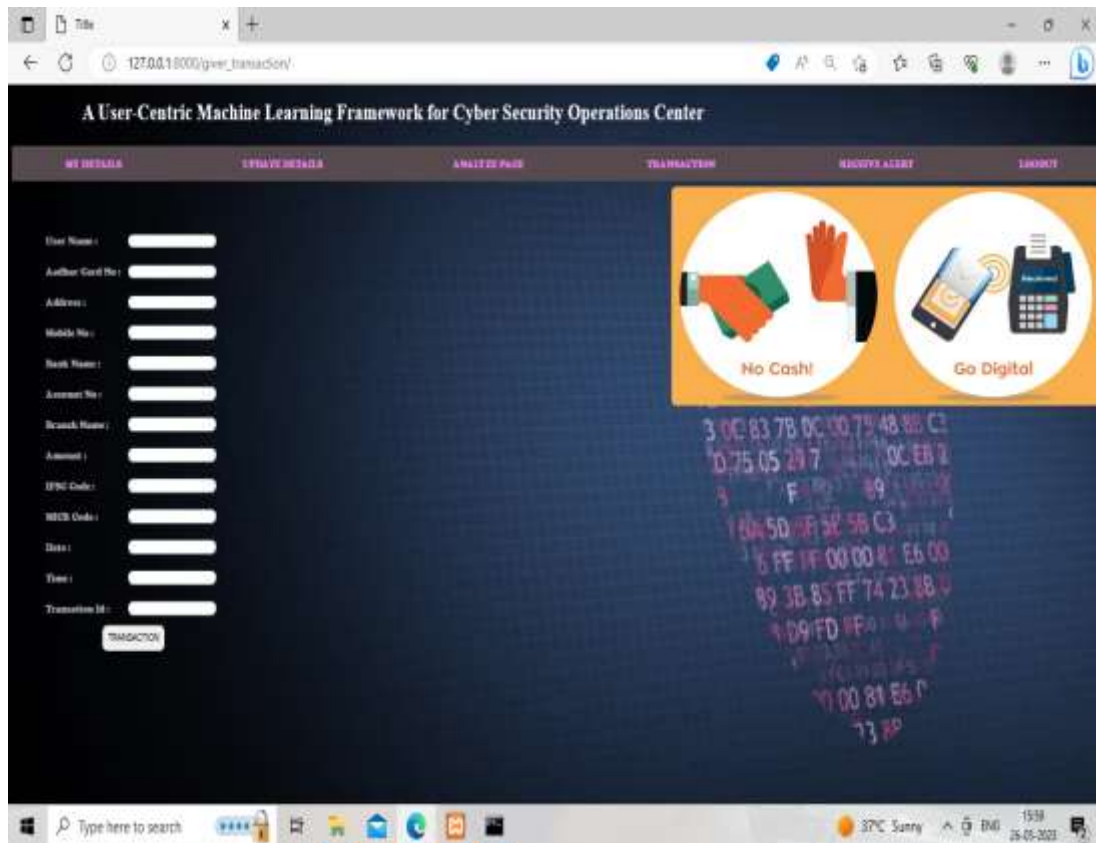


Fig 5.4.2.4 User Transaction Page

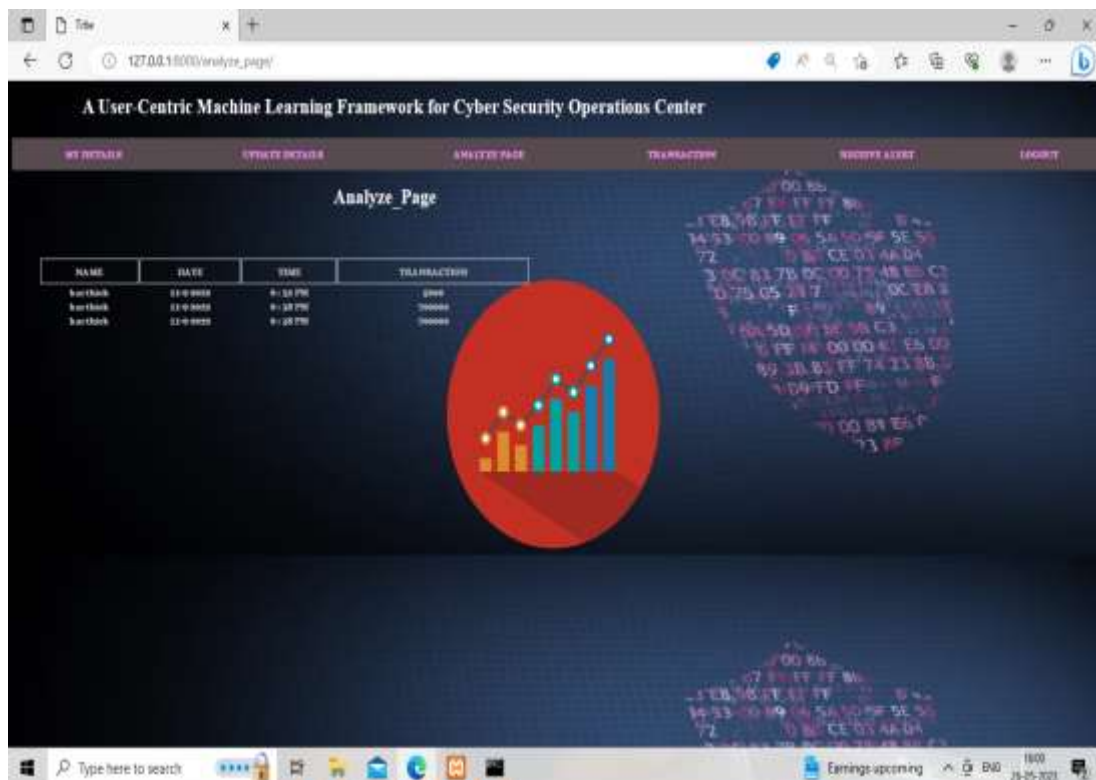


Fig 5.4.2.5 User Analyze Page

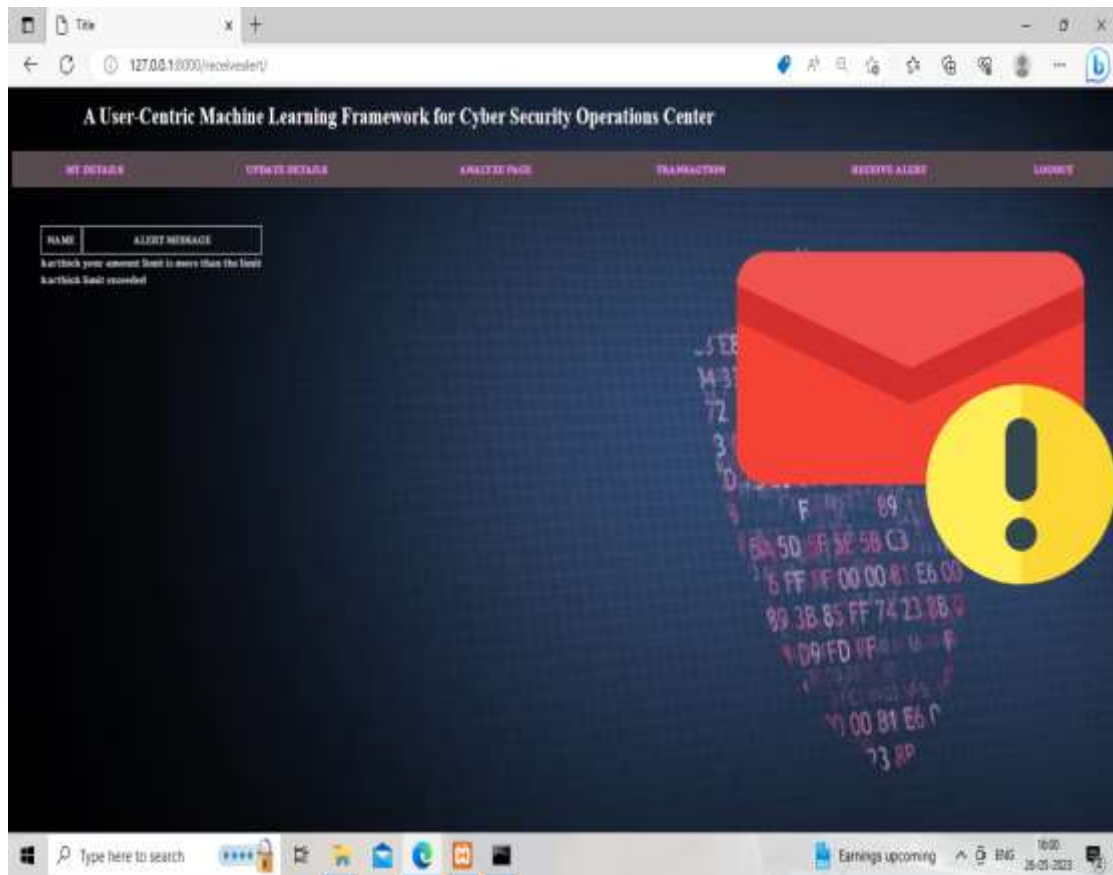


Fig 5.4.2.6 User Receive alert page

Admin Output Screens:

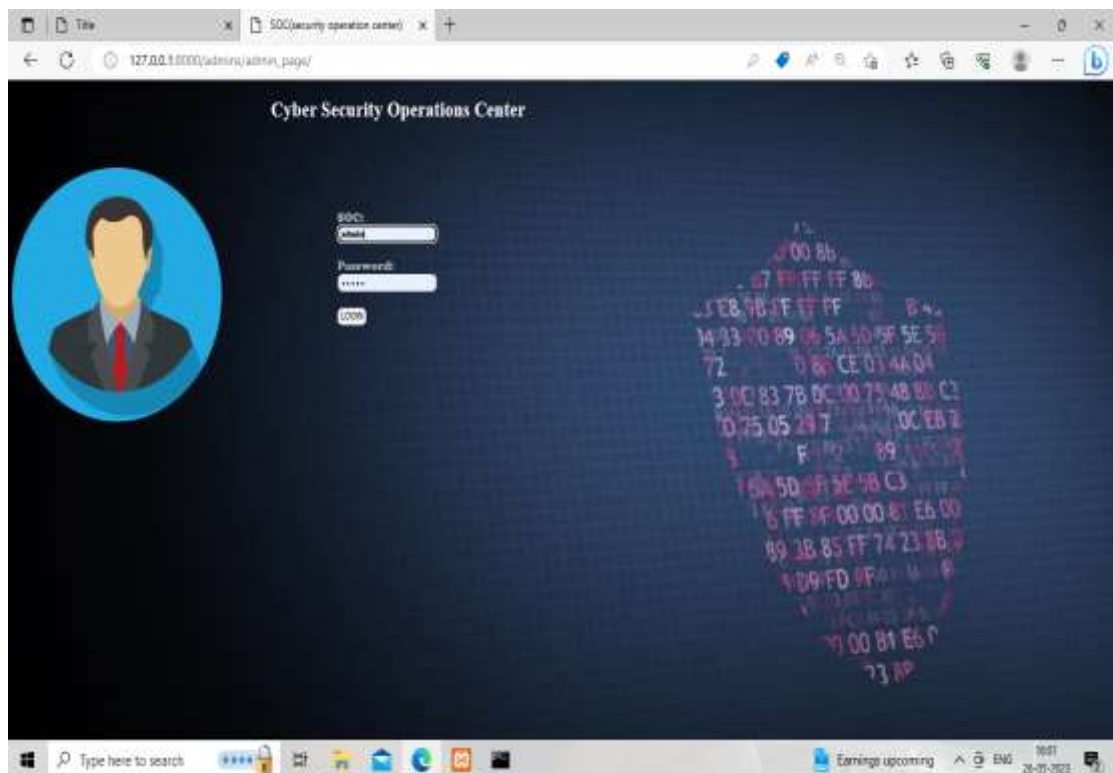


Fig 5.4.2.7 Admin Login page

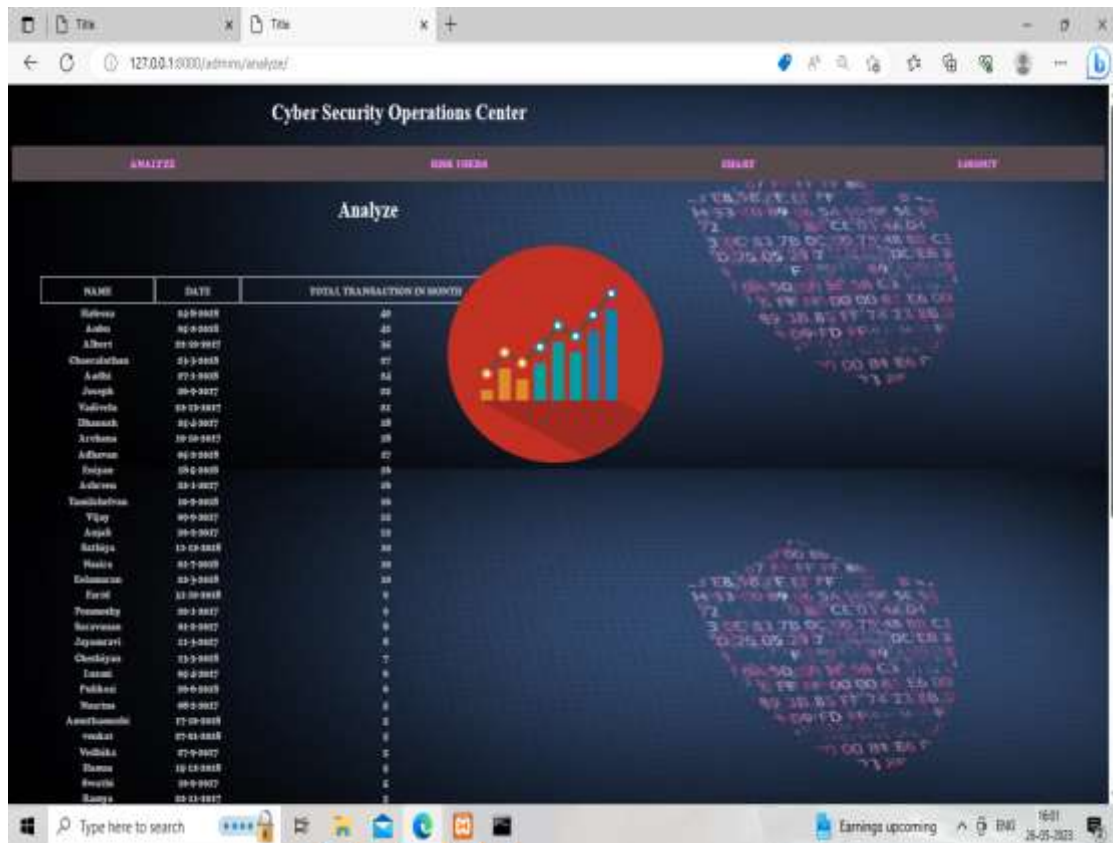


Fig 5.4.2.8 Admin analyze page

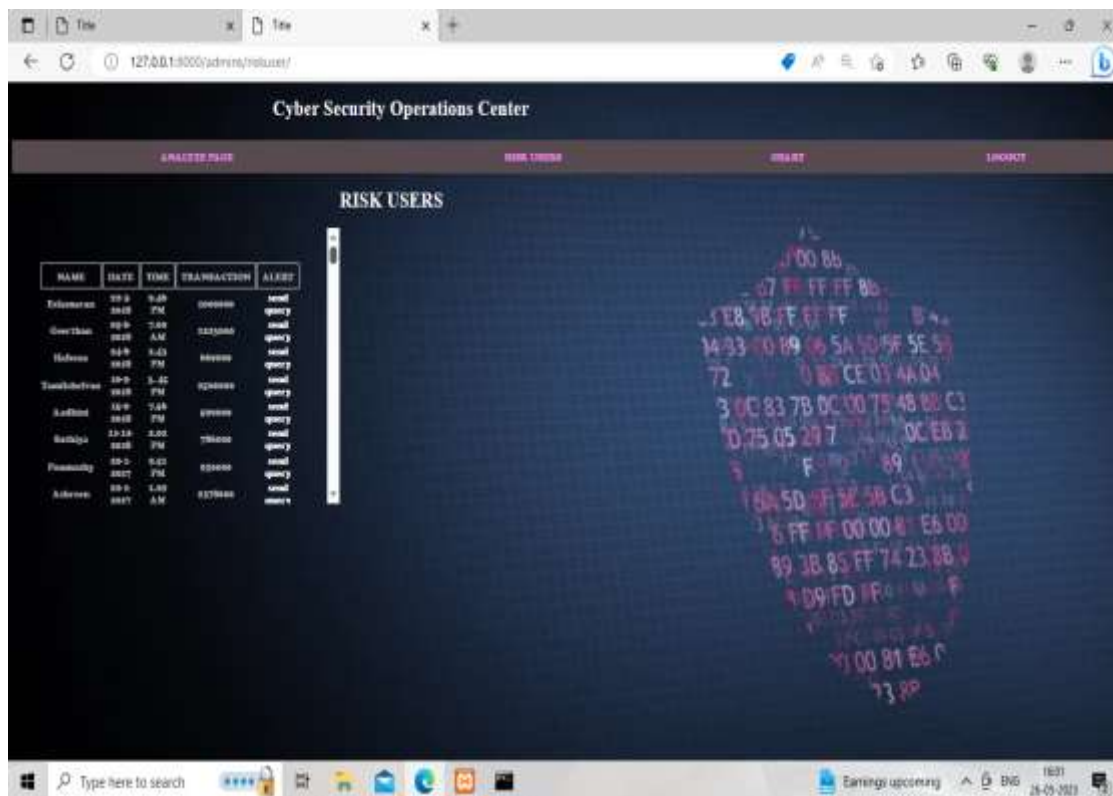


Fig 5.4.2.9 Admin Risk User Page

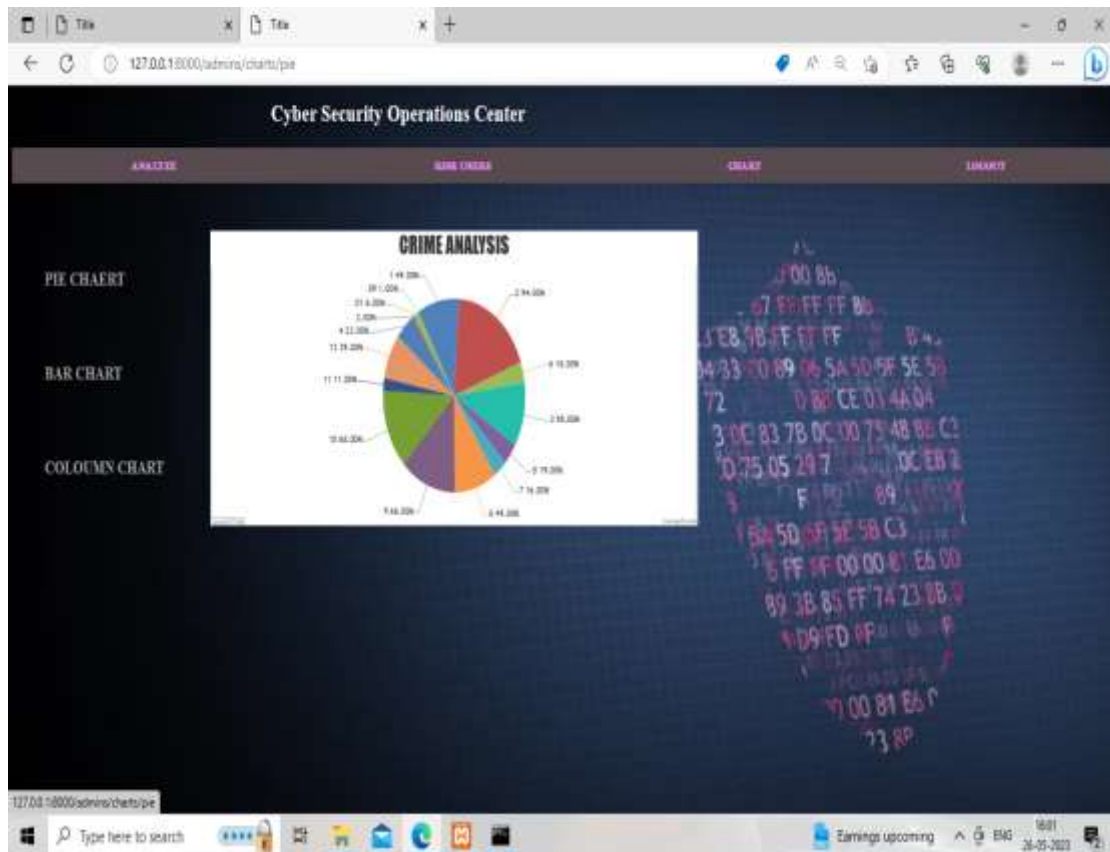


Fig 5.4.2.10 Admin pie chart Analysis page

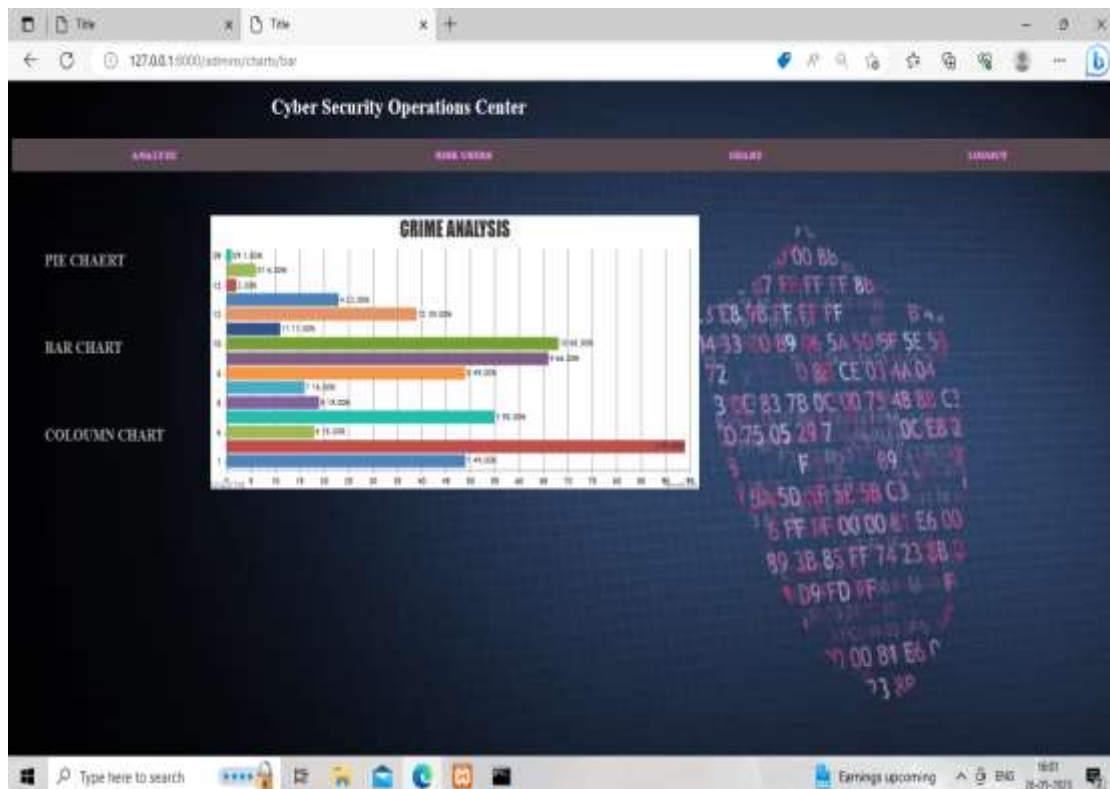


Fig 5.4.2.11 Admin bar chart Analysis page



Fig 5.4.2.12 Admin column chart Analysis page

5.4.3 RESULT ANALYSIS

Support Vector Machine gives the highest specificity of 80%, Random Forest and Linear Regression techniques give the accuracy of 78.33% and 76.67% respectively from the existing method and the proposed method improves 5% to 20% prediction. SVM uses kernel trick to solve non-linear problems whereas decision trees derive hyper-rectangles in input space to solve the problem. Decision trees are better for categorical data and it deals colinearity better than SVM.

SVM offers good accuracy compared to other algorithms in transactional data analysis. The accuracy is mainly high when the data is non-linearly and linearly separable. The accuracy is high in linear separable since all the variables are included efficiently with separating hyperplane.

5.5 CONCLUSION

We can conclude that the the process of execution is done using command prompt, when we run the code the IP address is provided by the server and the IP Address is used and which opens the website to take up the analyze and alert transactions above the limit.

At the time of the beginning of this project we had kept certain goals in the mind, and the system is meeting most of its requirements. In this project report we have mentioned all the details of the system, which includes all the systems of the system.

6. TESTING AND VALIDATION

6.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

6.2 DESIGN OF TEST CASES AND SCENARIOS

6.2.1 TYPES OF TESTING

The categorization of software testing is a part of diverse testing activities, such as test strategy, test deliverables, a defined test objective, etc. And software testing is the execution of the software to find defects.

The purpose of having a testing type is to confirm the AUT (Application Under Test). To start testing, we should have a requirement, application-ready, necessary resources available. To maintain accountability, we should assign a respective module to different test engineers.

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. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the 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.

White Box Testing

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form. We have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

Black Box Testing

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioural Testing.

- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested.

Conditional Testing

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

Data Flow Testing

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

Loop testing

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- Unstructured loops were resolved into nested loops or concatenated loops are tested

Each unit has been separately tested by the development team itself and all the input have been validated.

Code Testing

This strategy examines the logic of the program. To follow this method we developed some test data that resulted in executing every instruction in the program and module i.e. every path is tested. Systems are not designed as entire nor are they tested as single systems. To ensure that the coding is perfect two types of testing is performed or for that matter is performed or that matter is performed or for that matter is performed on all systems.

Integration Testing

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions. In this project integrating all the modules forms the main system. When integrating all the modules I have checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration.

➤ Top-Down Integration Testing

Top-down integration testing technique is used in order to simulate the behaviour of the lower-level modules that are not yet integrated. In this integration testing, testing takes place from top to bottom. First, high-level modules are tested and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended.

➤ **Bottom-Up Integration Testing**

In bottom-up testing, each module at lower levels is tested with higher modules until all modules are tested. The primary purpose of this integration testing is that each subsystem tests the interfaces among various modules making up the subsystem. This integration testing uses test drivers to drive and pass appropriate data to the lower level modules.

System Testing

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements. Here entire 'VOIP' has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not. Types of system testing include :

➤ **Recovery testing**

A recovery test is a system test that forces the software to fail in various ways, therefore verifying that the recovery is performed properly. If recovery is automatic, i.e., performed by the system itself, then re-initialization, data recovery, and system restarts are evaluated for correctness. If recovery requires human intervention, the time to repair is evaluated to determine whether it is within the acceptable limits.

➤ **Security testing**

Security testing attempts to verify that the production mechanism built into a system will protect it from improper penetration and other vulnerabilities. During security testing, potential threats to the system are uncovered so that appropriate measures can be taken. This may include threats such as system hijacking and information loss.

➤ **Performance testing**

Performance testing is designed to test the runtime performance of software within the context of an integrated system. Performance testing occurs throughout all steps in the testing process. It measures the scalability, responsiveness, and reliability of the software. Performance testing is often coupled with stress testing, and it may require both hardware and software instrumentation.

➤ **Regression testing**

Regression testing is a type of software testing. Test cases are re-executed to check the previous functionality of the application is working fine, and the new changes have not produced any bugs.

Regression testing can be performed on a new build when there is a significant change in the original functionality. It ensures that the code still works even when the changes are occurring. Regression means Re-test those parts of the application, which are unchanged. Regression tests are also known as the Verification Method. Test cases are often automated. Test cases are required to execute many times and running the same test case again and again manually, is time-consuming and tedious too.

Acceptance Testing

Acceptance Test is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behaviour of the system; Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied.

➤ **Alpha testing**

Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is one of the user acceptance testing. This is referred to as alpha testing only because it is done early on, near the end of the development of the software. Alpha testing is commonly performed by homestead software engineers or quality assurance staff. It is the last testing stage before the software is released into the real world.

➤ **Beta testing**

Beta testing is the live application of software in an environment that the developer cannot control. These tests are conducted by the end-users of the software, during the final stage of software testing. The customer records all problems they encountered during beta testing, and then they report these to the developer at regular interval.

6.2.2 TEST CASES

User Test Cases

S.no	Test Cases	Expected Output	Result	Remarks (if fails)
1	User Register	If user registered is successful	Pass	If already user exists
2	User Login	If user name and password is correct, it will validate user login.	Pass	Unknown users will not login
3	User details update	If user wants to change their personal details.	Pass	No such detail/No valid proof exists
4	User Transaction page	User should enter their details for transaction, with primary ID	Pass	Transaction fails if wrong details entered.
5	User Analyze page	User can view all the transaction list performed till date	Pass	Cannot Analyze with failed transactions.
6	User Alert Page	Alerts user about transactions exceeded the limit	Pass	Doesn't make alert for transactions within limit

Table 6.2.2.1 User Test Cases

Admin Test Cases

S.no	Test Cases	Expected Output	Result	Remarks (if fails)
1	Admin Login	If Admin name and password is correct, it will validate login.	Pass	No valid detail of admin exists
2	Admin Analyze Page	Show no. of transactions per person registered.	Pass	Fail to show list of no of transactions performed by each user.
3	Admin Alert Page	Shows details of users who have exceeded transaction limit.	Pass	Fail to view list of exceeded transaction limit users.
4	Send Query Page	Sends alert message for relevant user exceeding transaction limit.	Pass	Fails to send alert message
5	Admin Chart Page	Represents Analysis in diagrammatical format in the form of piecharts /bar graph / column graph	Pass	Fails to display Chart analysis

Table 6.2.2.2 Admin Test Cases

6.3 VALIDATION

Cross-validation is a technique for validating the model efficiency by training it on the subset of input data and testing on previously unseen subset of the input data. We can also say that it is a technique to check how a statistical model generalizes to an independent dataset.

In machine learning, there is always the need to test the stability of the model. It means based only on the training dataset; we can't fit our model on the training dataset. For this purpose, we reserve a particular sample of the dataset, which was not part of the training dataset. After that, we test our model on that sample before deployment, and this complete process comes under cross-validation.

Data can be useful to predict alert for transaction more than limit. Various techniques of Machine Learning can capable to do prediction, however its tough to choose best technique. Thus for this purpose we apply popular classification and ensemble methods on dataset for prediction.

6.4 CONCLUSION

Testing is an especially very important phase during the development of a project. It helps us find bugs and unwanted issues within the system. During this phase, we found some bugs in the system that we could easily fix. This helped us determine if the system was ready for real- world use. After rigorous testing, we could find that the system is ready for deployment.

7. CONCLUSION

7.1 PROJECT CONCLUSION

We provide a user-centered computer learning system that affects large data from various security logs, awareness information, and inspector intelligence. This method provides complete configuration and solution for dangerous user detection for the Enterprise System Operating Center. Select machine learning methods in the SOC product environment, evaluate efficiency, IO, host and users to create user-centric features. Even with simple mechanical learning algorithms, we prove that the learning system can understand more insights from the rankings with the most unbalanced and limited labels.

More than 20% of the neurological model of modeling is 5 times that of the current rule-based system. To improve the detection precision situation, we will examine other learning methods to improve the data acquisition, daily model renewal, real time estimate, fully enhance and organizational risk detection and management. As for future work, let's examine other learning methods to improve detection accuracy.

7.2 FUTURE ENHANCEMENT

As to the future work, we will research other learning algorithms to further improve the detection accuracy. We will increase the security with more new ways for securing the user operations. In SOC as an SOC Analyst we keep our efforts to find more new ways of finding an attack and implement changes needed to protect the organization from cyber threat.

The key areas which would include future scope of our system are Threat and vulnerability analysis, Investigating and reporting on any information security (InfoSec) issues as well as emerging trends, Analysing and responding to previously unknown hardware and software vulnerabilities and preparing disaster recovery plans accordingly.

8. REFERENCES

Referred Textbooks

- **Unified Modeling Language:** by Grady Booch, James Rumbaugh Ivar Jacobson
- **Python Programming** by Taneja Sheetal and Kumar Naveen
- **Web Development with Django** by Ben Shaw, Saurabh Badhwar, Bharat Chandra K S, Chris guest
- **Python Machine Learning, 2nd edition** by Sebastian Raschka Vahid Mirjalili

Referred Journals

- Yaswanth Sai Raj and J. Rene Beulah (2019). "Securing Identification Card Against Unauthorized Access", International Journal of Engineering and Advanced Technology, vol.8, Issue-3S, pp. 550-553.
- Nalini, M. and Anbu, S., "Anomaly Detection Via Eliminating Data Redundancy and Rectifying Data Error in Uncertain Data Streams", Published in International Journal of Applied Engineering Research (IJAER), Vol. 9, no. 24, 2014.
- M.M. Gamal, B. Hasan, and A.F. Hegazy, "A Security Analysis Framework Powered by an Expert System," International Journal of Computer Science and Security (IJCSS), Vol. 4, no. 6, pp. 505-527, Feb. 2011.
- NIKITA RANA, SHIVANI DHAR, PRIYANKA JAGDALE, NIKHIL JAVALKAR. Implementation of An Expert System for the Enhancement of E Commerce Security International Journal of Advances in Science Engineering and Technology, ISSN: 2321-9009 Volume-2, Issue-3, July-2014.
- S. Poonia, A. Bhardwaj, G. S. Dangayach, (2011) "Cyber Crime: Practices and Policies for Its Prevention", The First International Conference on Interdisciplinary Research and Development, Special No. of the International Journal of the Computer, the Internet and Management, Vol. 19, No. SP1.

Websites

- <https://www.w3schools.com/python/>
- <https://www.javatpoint.com/machine-learning>
- <https://www.tutorialspoint.com/django/index.htm>
- <https://towardsdatascience.com/support-vector-machines-svm>
- <https://www.ijraset.com/research-paper/cyber-security-operations-centre>