PRIVACY PRESERVING OUTSOURCED SUPPORT VECTOR MACHINE FOR SECURE DRUG DISCOVERY

A PROJECT REPORT

Submitted by

RENUKA P [211420104224]

RENUKA DEVI S [211420104225]

VARSHA R[211420104295]

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATION

Certified that this project report "**PRIVACY PRESERVING OUTSOURCED SUPPORT VECTOR MACHINE FOR SECURE DRUG DISCOVERY"** is the bonafide work of "**RENUKA P[211420104224], RENUKA DEVI S[21142104225]** and **VARSHA R[211420104295]**" who carried out the project work under my supervision.

Signature of the HOD

Dr. L. JABASHEELA M.E., Ph.D.,

Professor and Head,

Department of Computer Science and Engineering,

Panimalar Engineering College, Chennai- 123

Signature of the Supervisor

Mrs. P. DEEPA M.E., (Ph.D.,)

Associate Professor,

Department of Computer Science and Engineering,

Panimalar Engineering College, Chennai- 123

Submitted for the Project Viva – Voice examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **RENUKA P [211420104224], RENUKA DEVI S [211420104225], VARSHA R[211420104295]** hereby declare that this project report titled **"PRIVACY PRESERVING OUTSOURCED SUPPORT VECTOR MACHINE DESIGN FOR SECURE DRUG DISCOVERY"**, under the guidance of Mrs.P.DEEPA M.E., (Ph.D.,) is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

RENUKA P

RENUKA DEVI S

VARSHA R

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project .

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project. We express my heartfelt thanks to **Dr. L. JABASHEELA,M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Dr. K.VALARMATHI M.E., Ph.D.,** and **Mrs. P. DEEPA M.E.,(Ph.D.,)** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

RENUKA P [211420104224]

RENUKA DEVI S[211420104225]

VARSHA R[211420104295]

ABSTRACT

In this project, a framework has been proposed for privacy-preserving outsourced drug discovery in the cloud, which is referred to as POD. Specifically, POD is designed to allow the cloud to securely use multiple drug formula providers' drug formulas to train Support Vector Machine (SVM) provided by the analytical model provider. In this approach, there is a design of secure computation protocols to allow the cloud server to perform commonly used integer and fraction computations. To securely train the SVM, there is a design of a secure SVM parameter selection protocol to select two SVM parameters and construct a secure sequential minimal optimization protocol to privately refresh both selected SVM parameters. The trained SVM classifier can be to determine whether a drug chemical compound is or not in a privacy-preserving way. Lastly, it is proved that the proposed POD achieves the goal of SVM training and chemical classification without privacy leakage to unauthorized parties, as well as demonstrating its utility and efficiency using three real-world drug datasets.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

The general definition of drug discovery is the process of identifying one or more active components derived from conventional treatments; this process involves identifying screening hits, optimizing these hits' medicinal chemistry, and increasing their affinity, selectivity (to lower the risk of adverse Effects), bio availability, and metabolic half-life

Nevertheless, the process of finding novel drugs for therapeutic purposes is difficult, expensive, and ineffective. For instance, it has been claimed that the licensing approval process for medications can take up to 12 years from the initial stage of development. The Association of the British Pharmaceutical Industry estimates that the investment required for each drug is £1.15 billion. Put differently, drug discovery necessitates substantial.

Drug discovery can benefit from technological advancements (e.g., computer-aided drug design to identify novel physiologically active molecules.The global drug discovery technologies market is anticipated to reach over $160 billion by 2025, growing at a compound yearly growth rate of about 12.2% over the next ten years, according to Research and Markets . Among the many technologies available for use in drug discovery is machine learning. Machine learning methods, for instance, can be used to assess the potential biological activity and to offer forecasts regarding the pharmaceutical and photochemical characteristics of chemical compounds .

Support Vector Machine (SVM) is one of the data mining technologies that has been used extensively recently to predict ligand- based chemical compounds in drug development . It has a pretty high decision rate. When employing SVMs, the learned SVM classifier can be utilized for new drug compound visual scanning. The SVM

classifier is taught utilizing pre-existing datasets of recognized drug formulas). Drug discovery entails large financial outlays and great commercial values, hence privacy is How, for instance, may the danger of unapproved publication be reduced during the SVM training phase? In this instance, upon a researcher forwarding some chemical compounds to the cloud for SVM classification, it's critical to guarantee that the prospective novel medication compounds won't be disclosed to unaffiliated parties—like rival pharmaceutical companies.

Moreover, several pharmaceutical companies may work together to train the SVM and raise the SVM decision rate. These companies, however, are unwilling to make their datasets public. Research and practical challenges remain in figuring out how to protect SVM training and decision-making under various data sources without jeopardizing the privacy of each individual stakeholder. For secure drug discovery in the cloud, we therefore suggest a privacy-preserving outsourced support vector machine design in this study.

Referred to as POD from now on, this POD, in contrast to current drug discovery frameworks aims to achieve the following: the drug formula owner can safely outsource data (e.g., drug formula) to the cloud for storage without risking data leakage to unauthorized parties.

1.2 OVERVIEW

Secure Multi-Source SVM Training is used to encrypt data from other drug formula owners, the POD enables an approved model provider to train the SVM dynamically. Without being aware of the contents of the training dataset, the model provider can decode and retrieve the trained model.Secure SVM Drug Decision is Used to secure cloud upload, an authorized tester can ascertain whether a drug's chemical constituents is active in a way that protects privacy or not.

Reducing Plaintext Overflow is more secure computation and performed, the plaintext overflow problem will arise because the ciphertext's plaintext length may grow during computation and beyond the plaintext upper-bound. The ciphertext's plaintext size is then reduced using a safe quick approximation technique so that the new ciphertext can be further computed. Simplicity of Use for POD eliminates the need for the authorized tester to carry out labor- intensive pre-processing before outsourcing. Furthermore, the relationship between medication Since the tester just needs to send an encrypted query to the cloud server and wait for the cloud to react with the encrypted decision result in a single round, there is very little interaction between the tester and the cloud server during secure computing.

1.3 PROBLEM DEFINITION

When a researcher sends some chemical compounds to the cloud for SVM classification, it is important to ensure that the potential new drug compounds will not be leaked to a third-party, such as a competing pharmaceutical corporation.Therefore when a researcher sends some chemical compounds to the cloud for SVM classification, no security for new Drug Component.

CHAPTER 2

LITERATURE SURVEY

1]Title: Drug Review System Using Machine Learning by Comparing Linear Support Vector Machine with Naïve Bayes Classifier to Measure Accuracy

In the modern healthcare landscape, drug reviews play a crucial role in informing both patients and medical professionals about the efficacy and potential side effects of various medications. However, the sheer volume of reviews available online makes it challenging to extract meaningful insights manually. This study proposes a machine learning-based drug review system aimed at automating the process of analyzing and categorizing drug reviews. Specifically, it compares the performance of two popular classifiers, Linear Support Vector Machine (SVM) and Naïve Bayes, in terms of accuracy, to determine their suitability for this task.

Advantage

Accuracy**:** Both Linear SVM and Naïve Bayes are known for their ability to handle large datasets and provide accurate classification results.

Scalability: The proposed system can easily scale to accommodate an increasing volume of drug reviews as more data becomes available. This scalability ensures that the system remains effective and relevant over time, even as the size of the dataset grows.

Disadvantage

Overfitting: Both Linear SVM and Naïve Bayes classifiers are susceptible to overfitting, especially when dealing with noisy or unbalanced datasets.

[2] Title: Needle Free Drug Delivery Devices Market Size, Industry Outlook and Opportunity Analysis

The needle-free drug delivery devices market has witnessed significant growth in recent years due to the rising demand for pain-free drug administration and the increasing prevalence of chronic diseases. This market analysis explores the size, industry outlook, and opportunities within the needle-free drug delivery devices market. It examines key market trends, drivers, challenges, and opportunities shaping the industry landscape. Additionally, it evaluates the competitive landscape, regulatory environment, and technological advancements influencing market growth. By providing insights into market dynamics and future prospects, this analysis aims to assist stakeholders in making informed decisions and capitalizing on emerging opportunities in the needle-free drug delivery devices market.

Advantage

Pain-Free Administration: Needle-free drug delivery devices offer a pain-free alternative to traditional needle-based injections, enhancing patient comfort and compliance. This advantage is particularly significant for individuals with needle phobia or those requiring frequent injections, such as diabetic patients.

Reduced Risk of Infections**:** By eliminating the need for needles, needle-free drug delivery devices reduce the risk of needle-stick injuries and associated infections, thereby enhancing safety for healthcare workers and patients alike.

Disadvantage

Limited Compatibility: Not all medications are suitable for administration via needle- free devices, as certain drugs may require precise dosing or specialized delivery mechanisms that are not feasible with current technology.

Cost Considerations: Needle-free drug delivery devices may incur higher manufacturing costs compared to traditional needles, which could potentially translate to higher treatment costs for patients.

[3] Title: Privacy-Preserving Outsourced Support Vector Machine Design for Secure Drug Discovery

Privacy-preserving outsourced support vector machine (SVM) design has emerged as a promising approach to secure drug discovery processes while maintaining the confidentiality of sensitive data. This study presents a novel framework for leveraging outsourced SVM in drug discovery, ensuring that proprietary information remains protected throughout the computational process. By utilizing cryptographic techniques and secure multi-party computation protocols, the proposed framework enables pharmaceutical companies to collaborate with third-party service providers without compromising data privacy.

Advantage

Collaborative Drug Discovery: The framework facilitates collaborative drug discovery efforts by allowing multiple stakeholders to contribute data and expertise without exposing sensitive information. This collaborative model can lead to more comprehensive analyses and insights, potentially accelerating the pace of drug development. Regulatory Compliance**:** Privacy-preserving techniques, such as encryption and secure computation protocols, help pharmaceutical companies adhere to regulatory requirements and industry standards governing data privacy and security. By implementing robust privacy measures, organizations can demonstrate their commitment to ethical data handling practices.

Disadvantage

Computational Overhead: Privacy-preserving techniques often introduce additional computational overhead due to encryption, decryption, and secure computation protocols. This overhead can impact the performance and efficiency of the SVM. model, potentially leading to longer processing times and higher resource requirements.

[4] Title: Secure the Drug Components using Data Mining

Securing the components of drugs through data mining techniques has become increasingly important in pharmaceutical research and development. This paper proposes a framework that utilizes data mining methods to enhance the security of drug components, including active pharmaceutical ingredients (APIs), excipients, and formulations. By analyzing large datasets containing chemical structures, pharmacological properties, and adverse effects, the proposed framework aims to identify potential vulnerabilities in drug components and mitigate risks associated with counterfeit drugs, contamination, or adverse reactions. Leveraging machine learning algorithms, association rule mining, and anomaly detection techniques, the framework facilitates proactive identification of suspicious patterns or deviations from expected norms, thereby enhancing the safety and efficacy of pharmaceutical products.

Advantage

Detection of Counterfeit Drugs: Data mining techniques enable the detection of patterns indicative of counterfeit drugs or unauthorized modifications to drug components. By analyzing chemical compositions, manufacturing processes, and distribution networks, suspicious deviations can be identified, helping to prevent the circulation of counterfeit medications.

Risk Mitigation for Adverse Reactions: By mining large-scale pharmacovigilance databases and adverse event reports, data mining can identify associations between drug components and adverse reactions.

Disadvantage

Data Quality and Availability: Data mining for drug component security relies on the availability of high-quality, comprehensive datasets containing information on chemical structures, pharmacological properties, and adverse events. However, data quality issues, such as incomplete or inaccurate data, may hinder the effectiveness of data mining techniques.

Complexity of Data Analysis: Analyzing large and heterogeneous datasets in the pharmaceutical domain requires sophisticated data mining techniques and expertise. Extracting actionable insights from complex data sources may pose challenges in terms of algorithm selection, feature engineering, and interpretation of results

[5] Title: Membrane transporters in drug development and as determinants of precision medicine

Membrane transporters play crucial roles in drug development and precision medicine by influencing the pharmacokinetics, efficacy, and safety of medications. This review explores the significance of membrane transporters in drug discovery, development, and personalized treatment approaches. Membrane transporters, including solute carriers (SLCs) and ATP-binding cassette (ABC) transporters, mediate the absorption, distribution, metabolism, and excretion of drugs, impacting their bioavailability and therapeutic effects. Understanding transporter-mediated drug interactions and variability in transporter expression and function across individuals is essential for predicting drug response and designing personalized treatment regimens.

Additionally, membrane transporters serve as potential biomarkers for disease diagnosis, prognosis, and treatment selection in precision medicine. Leveraging advances in transporter pharmacology, computational modeling, and pharmacogenomics, researchers can optimize drug development strategies, predict drug-drug interactions, and tailor therapies to individual patient characteristics. This review highlights the advantages and disadvantages of utilizing membrane transporters in drug development and precision medicine, as well as future directions for research and clinical applications.

Advantage

Personalized Treatment Strategies: Variability in transporter expression and function among individuals influences drug response and susceptibility to adverse effects. Incorporating transporter genotyping and phenotyping into pharmacogenomic analyses enables the identification of patients who are likely to benefit from specific drugs or require alternative treatment approaches based on their transporter profiles.

Disadvantage

Technological Limitations: Current methods for assessing transporter function and expression, such as in vitro assays and imaging techniques, have limitations in terms of throughput, sensitivity, and specificity. Advances in technology are needed to overcome these limitations and enable more comprehensive characterization of transporter-mediated drug interactions and variability in drug response.

[6] Title: Drug Sentiment Analysis using Machine Learning Classifiers

Drug sentiment analysis has become very significant in present times as classifying medicines based on their effectiveness through analyzing reviews from users can assist potential future consumers in gaining knowledge and making better decisions about a particular drug. The objective of this proposed research is to measure the effectiveness level of a particular drug. Currently most of the text mining researches are based on unsupervised machine learning methods to cluster dataWhen supervised learning methods are used for text mining, the usual primary concern is to classify the data into two classes. Lack of technical terms in similar datasets make the categorization even more challenging.

Advantages

The research focuses on finding out the keywords through tokenization and lemmatization so that better accuracy can be achieved for categorizing the drugs based on their effectiveness using different algorithms. Such categorization can be instrumental for treating illness as well as improve one's health and well-being. Four machine learning algorithms have been applied for binary classification and one for multiclass classification on the drug review dataset acquired from the UCI machine learning repository.

Disadvantages

Most of the above-mentioned researches primarily focus on unsupervised learning method. Compared to other product reviews, number of researches conducted on drug reviews is significantly low. The neural network approach technique has a very high performance. It is a widely used technique for sentiment analysis and is capable of detecting all possible interactions between attributes. It is effective for dealing with a nonlinear connection between variables that is complex. The main disadvantage is that it takes longer to compute than other algorithms.

[7] Title: A Novel Neutrosophic Weighted Extreme Learning Machine for Imbalanced Data Set

Extreme learning machine (ELM) is known as a kind of single-hidden layer feedforward network (SLFN), and has obtained considerable attention within the machine learning community and achieved various real-world applications. It has advantages such as good generalization performance, fast learning speed, and low computational cost. However, the ELM might have problems in the classification of imbalanced data sets. In this paper, we present a novel weighted ELM scheme based on neutrosophic set theory, denoted as neutrosophic weighted extreme learning machine (NWELM), in which neutrosophic c-means (NCM) clustering algorithm is used for the approximation of the output weights of the ELM.

Advantages

The standard ELM may encounter difficulties when classifying imbalanced datasets. NWELM mitigates this issue by incorporating neutrosophic set theory and introducing a weighted scheme. Specifically, it employs the neutrosophic c-means (NCM) clustering algorithm to approximate the output weights of the ELM. This novel approach demonstrates advantages compared to previous studies on benchmark datasets .In summary, NWELM combines the strengths of ELM with tailored techniques to handle imbalanced data, making it a promising choice for real-world applications.

Disadvantages

The introduction of neutrosophic set theory and the weighted scheme in NWELM adds complexity to the model. As a result, interpreting the learned weights and understanding the decision boundaries may become more challenging.NWELM's effectiveness heavily relies on the specific characteristics of the dataset. While it performs well on certain benchmarks, its performance may vary significantly across different types of imbalanced data.Further research and experimentation are necessary to fully understand its strengths and weaknesses.

[8] Title: Drug Design and Discovery

Drug discovery is the process through which potential new therapeutic entities are identified, using a combination of computational, experimental, translational, and clinical models Despite advances in biotechnology and understanding of biological systems, drug discovery is still a lengthy, costly, difficult, and inefficient process with a high attrition rate of new therapeutic discovery. Drug design is the inventive process of finding new medications based on the knowledge of a biological target. In the most basic sense, drug design involves the design of molecules that are complementary in shape and charge to the molecular target with which they interact and bind.

Advantages

Patients receive optimal pharmaceutical therapy, ensuring consistent and predictable treatment across all levels of healthcare providers and locations.The application of drug design and discovery leads to consistent and known usage patterns, resulting in better availability of medicines.Patients benefit from the best available treatment regimen, leading to improved outcomes.

Disadvantages

These irrelevant features can negatively impact the model's performance.The introduction of neutrosophic set theory and the weighted scheme in NWELM adds complexity to the model. As a result, interpreting the learned weights and understanding the decision boundaries may become more challenging.NWELM's effectiveness heavily relies on the specific characteristics of the dataset. While it performs well on certain benchmarks, its performance may vary significantly across different types of imbalanced data.

[9] Title: Drug Classification Analysis Using Different Machine Learning Algorithms

Healthcare industry managers have placed a significant emphasis on ensuring the quality and financial efficiency of medications. With the advancement of technology and related techniques, there is now an opportunity to improve drug classification.

Machine learning, utilizing large databases, has become a vital tool in the discovery of drug and design process. In this research, the data obtained from Kaggle was subjected to a range of machine learning algorithms, including Logistic Regression, - Nearest Neighbors, Random Forest, Gradient Boosting, Decision Tree, Adaptive Boosting, Naive Bayes, both linear and non-linear Support Vector Machine (SVM), and Bagging.

Advantages

Efficiency and Time Savings are automating drug classification with machine learning models that can significantly save time and resources for healthcare professionals. Instead of manual analysis, algorithms can process large datasets efficiently.This efficiency is crucial for tasks like identifying potential drug candidates or assessing drug safety profiles.Human analysis may be prone to errors due to fatigue, biases, or oversight. Machine learning algorithms, on the other hand, follow consistent rules and minimize human-related mistakes

Disadvantages

Logistic Regression assumes linear relationship between predictors and outcome,May not handle complex data well and Sensitive to outliers.K-Nearest Neighbors (K-NN) were computationally expensive during prediction.Sensitive to noise and irrelevant features and Requires careful choice of k (number of neighbors).Poor generalization to new data and May not perform well on diverse datasets. SVM Can be slow for large datasets in choosing the right kernel is crucial and Interpretability can be challenging.

[10] Title: Drug discovery and development for the basic role of biological research

This article provides a brief overview of the processes of drug discovery and development. Our aim is to help scientists whose research may be relevant to drug discovery and/or development to frame their research report in a way that appropriately places their findings within the drug discovery and development process and thereby support effective translation of preclinical research to humans. One overall theme of our article is that the process is sufficiently long, complex, and expensive so that many biological targets must be considered for every new medicine eventually approved for clinical use and new research tools may be needed to investigate each new target.

Advantages

Patients receive optimal pharmaceutical therapy, ensuring consistent and predictable treatment across all levels of healthcare providers and locations.

The application of drug design and discovery leads to consistent and known usage patterns, resulting in better availability of medicines.: Patients benefit from the best available treatment regimen, leading to improved outcomes.

Disadvantages

The introduction of neutrosophic set theory and the weighted scheme in NWELM adds complexity to the model. As a result, interpreting the learned weights and understanding the decision boundaries may become more challenging.NWELM's effectiveness heavily relies on the specific characteristics of the dataset.

While it performs well on certain benchmarks, its performance may vary significantly across different types of imbalanced data.These irrelevant features can negatively impact the model's performance.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing datasets of known drug formulas is used to train the SVM classifier, and the trained SVM classifier can be used for new drug compound visual scanning. Due to the significant investments and high commercial values involved in drug discovery, privacy is an important factor. When a researcher sends some chemical compounds to the cloud for SVM classification, it is important to ensure that the potential new drug compounds will not be leaked to a third-party, such as a competing pharmaceutical corporation.

DRAWBACK

When a researcher sends some chemical compounds to the cloud for SVM classification no security for new Drug Component.

3.1 PROPOSED SYSTEM

A Privacy preserving Outsourced Support Vector Machine Design for Secure Drug discovery is proposed in the cloud environment, hereafter referred to as POD. Unlike existing drug discovery frameworks, the POD seeks to achieve it efficiently. There is no usage of three real time datasets to check the efficiency of potential new drug component. Instead of using existing datasets there is usage of another one data mining algorithm Naïve Bayes(NB). This two algorithms are used to train the uploaded drug dataset (CSV file). In final there will be trained data and accuracy for that uploaded dataset. Drug tester will check that new drug component. Drug tester doesn’t know the contents of that file; they will get the trained data only.

ADVANTAGE

- We minimize the risk of unauthorized disclosure during the SVM and NB training.
- Multiple pharmaceutical corporations won't reveal the drug components in detail.

3.2 REQUIREMENT ANALYSIS

Let's break down the hardware and software requirements and the technologies used in the context of drug discovery using Support Vector Machines (SVM)

3.2.1 HARDWARE REQUIREMENTS

Hard Disk: 80GB and Above

RAM:4GB and Above

Processor: P IV and Above

3.2.2 SOFTWARE REQUIREMENTS

Operating System: Windows 7 and above (64 bit)

The choice of a 64-bit Windows operating system provides compatibility.

Java Development Kit (JDK):Version 1.8

Python:Version 3.6.3

Apache Tomcat: Version 9.0.26

MySQL

3.3.3 TECHNOLOGIES USED

Java

Spring Framework

SVM (Support Vector Machines)

CHAPTER 4
SYSTEM DESIGN

4.1.1 USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

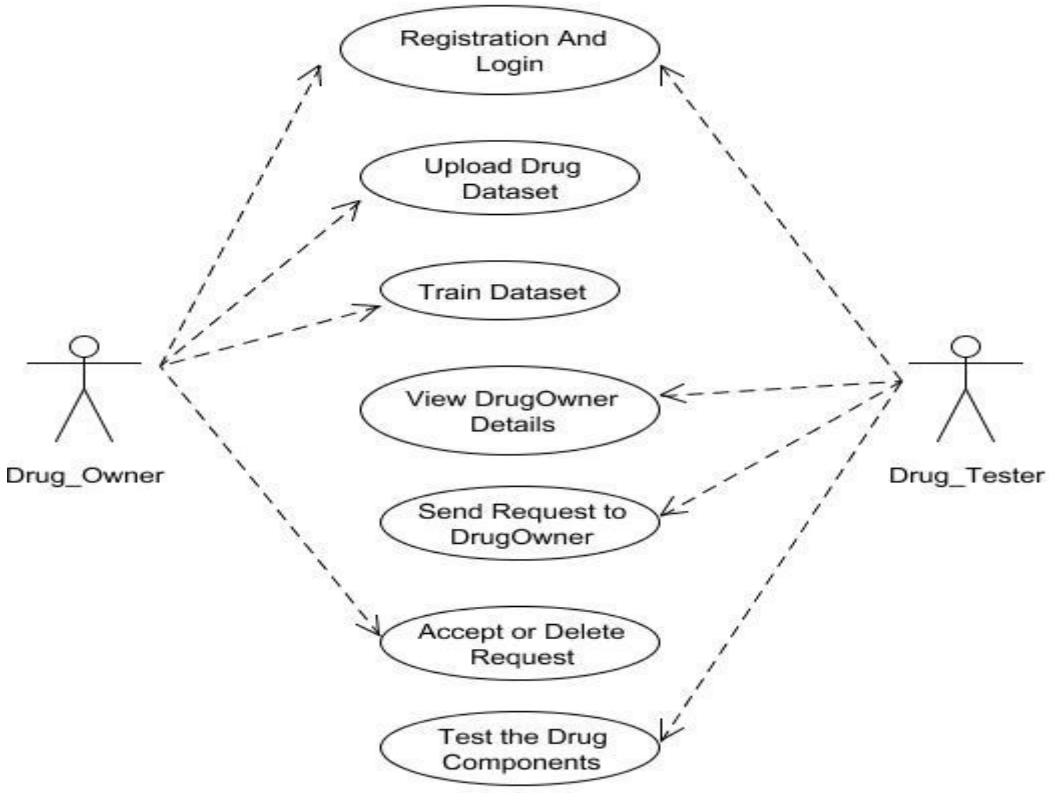


Fig 4.1 Usecase Diagram

4.1.2 Class Diagram

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

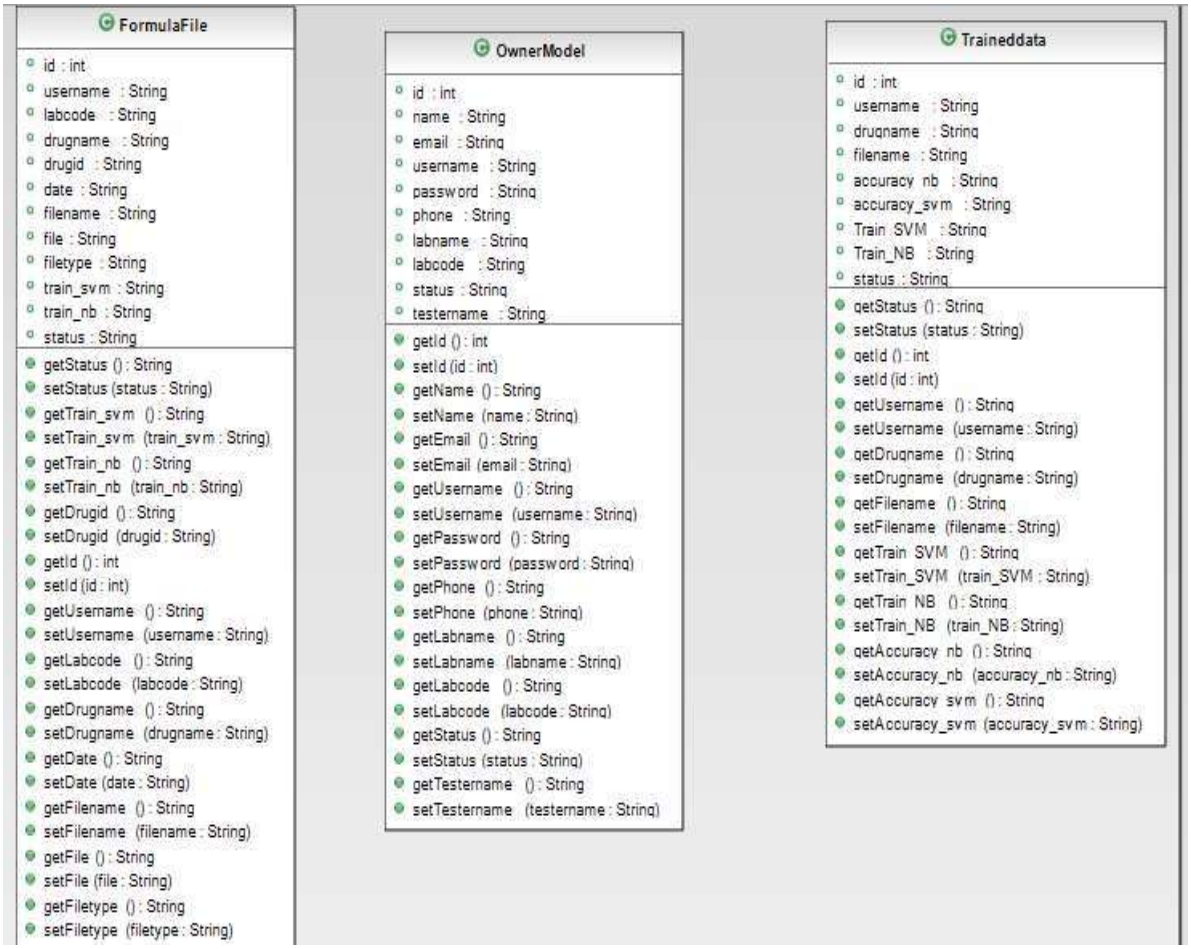


Fig 4.2 Class Diagram

4.1.3 SEQUENCE DIAGRAM

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram. Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

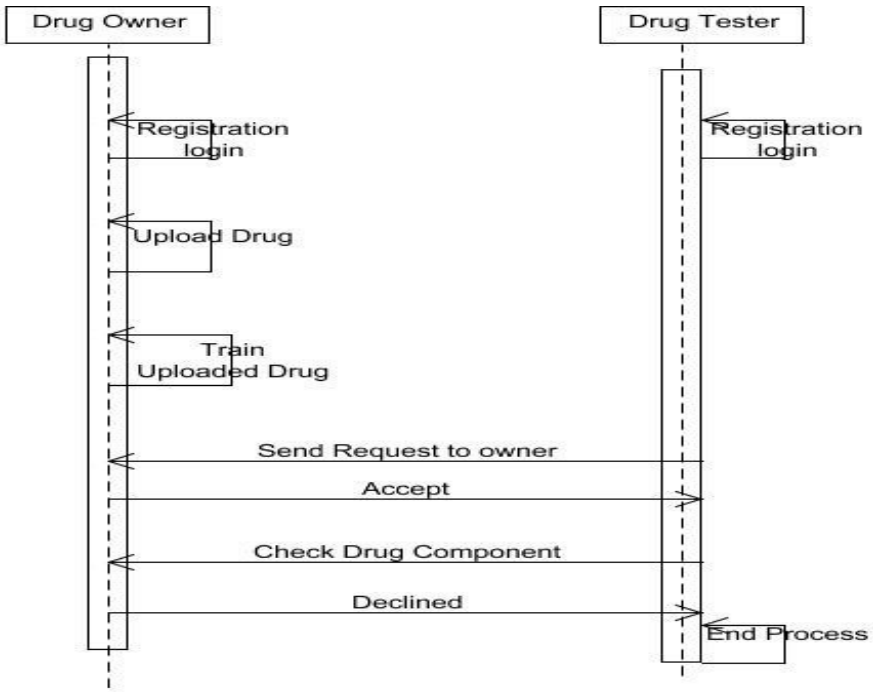


Fig 4.3 Sequence Diagram

4.1.4 COLLABORATION DIAGRAM

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

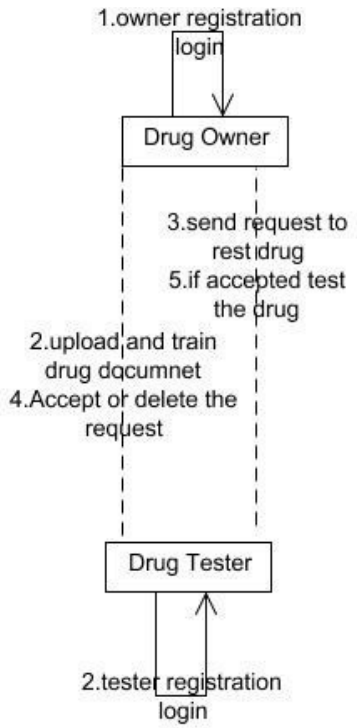


Fig 4.4 Collaboration Diagram

4.1.5 ACTIVITY DIAGRAM

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

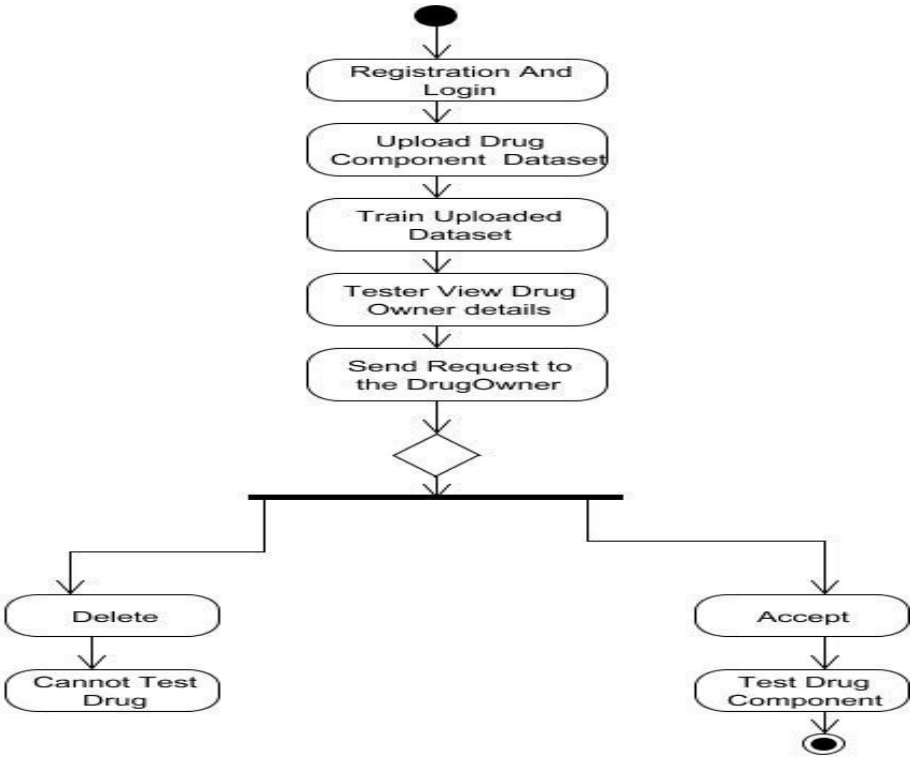


Fig 4.5 Activity Diagram

4.1.6 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

Level 0

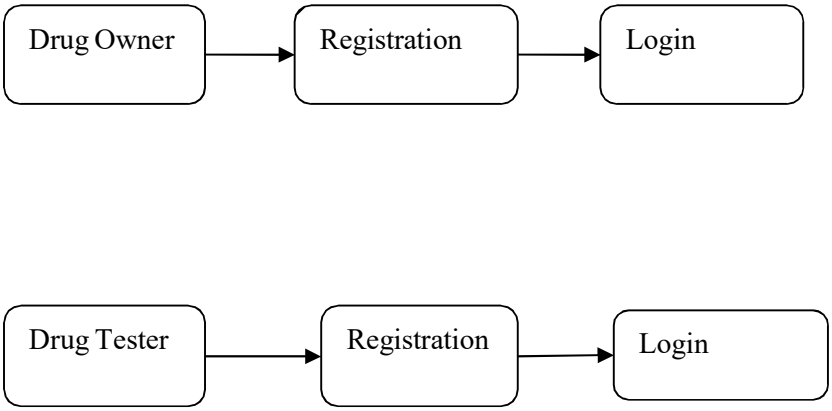


Fig 4.6 Level 0 DFD Diagram

Level 1

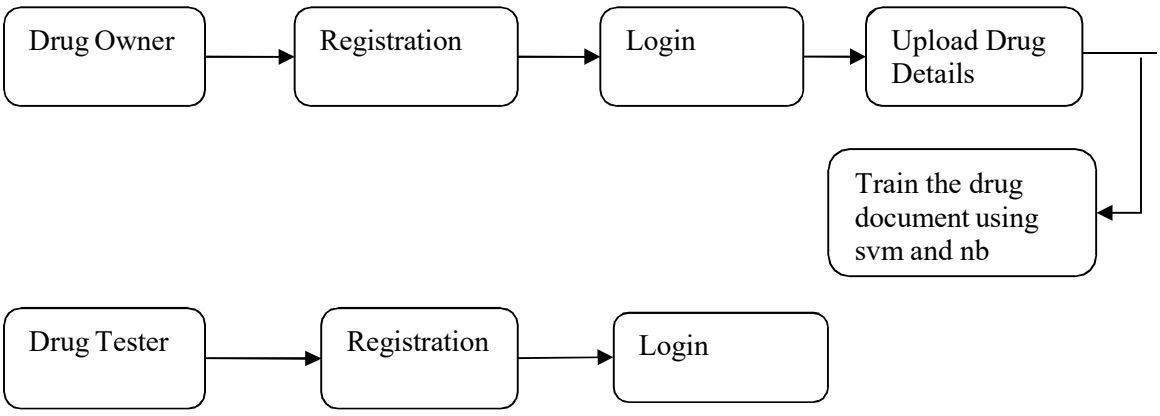


Fig 4.7 Level 1 DFD Diagram

Level 2

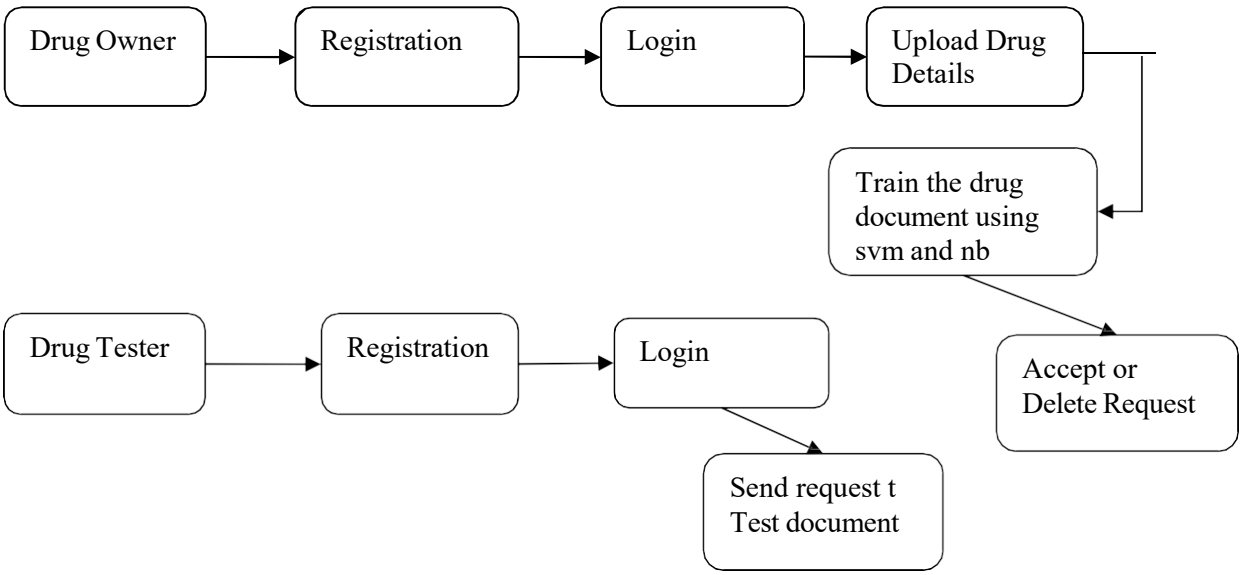


Fig 4.8 Level 2 DFD Diagram

Level 3

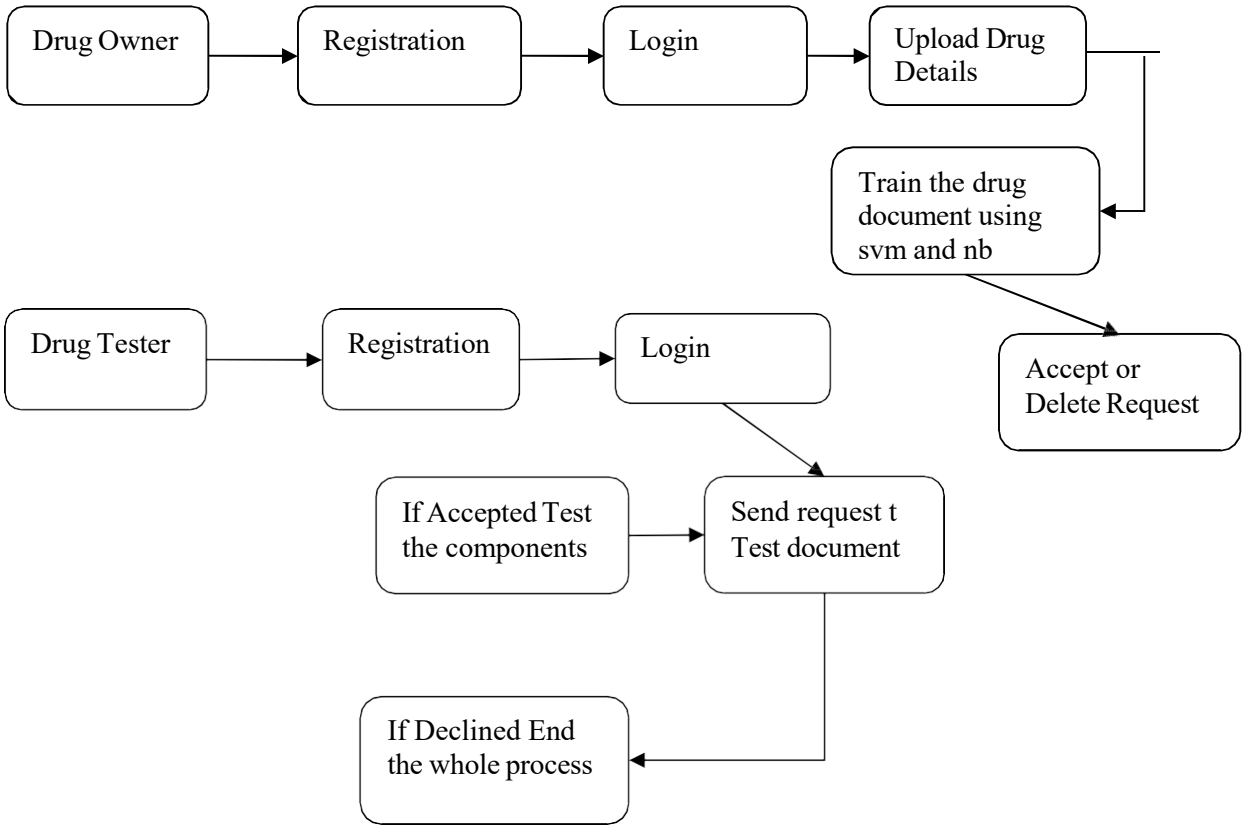


Fig 4.9 Level 3 DFD Diagram

CHAPTER 5

SYSTEM ARCHITECTURE

MODULE DESCRIPTION

- Drug Owner & Tester Registration
- Drug Component Uploading
- Train dataset
- Drug Testing

5.2.1 Drug Owner & Tester Registration

Sign in: Drug owners can sign in to the system to access their account.
Upload Drug Dataset: Owners can upload drug datasets, containing information about the drug components, to the cloud server.
Request Handling: Owners can send requests to testers for drug testing. Once accepted by testers, the testing process can proceed.

5.2.2 Drug Component Uploading

The drug owner will upload the data set. That data set contains the formula and we have to mention the type of class (Class A, Class B). While uploading the file we will read the content and store into the database and store that .csv file in cloud.

5.2.3 Train dataset

The drug owner will train the uploaded data using python. For this part we will use two algorithms, SVM and Naïve Bayes. The trained data and accuracy will be sent to the owner from python server.

5.2.4 Drug Testing

The drug tester will test the uploaded drug components. If that particular drug component is still in the cloud he will assume that component is still active.

Registration & Login: Testers need to register and log in to access the system.
Accept/Reject Requests: Testers can accept or reject requests from drug owners for testing.

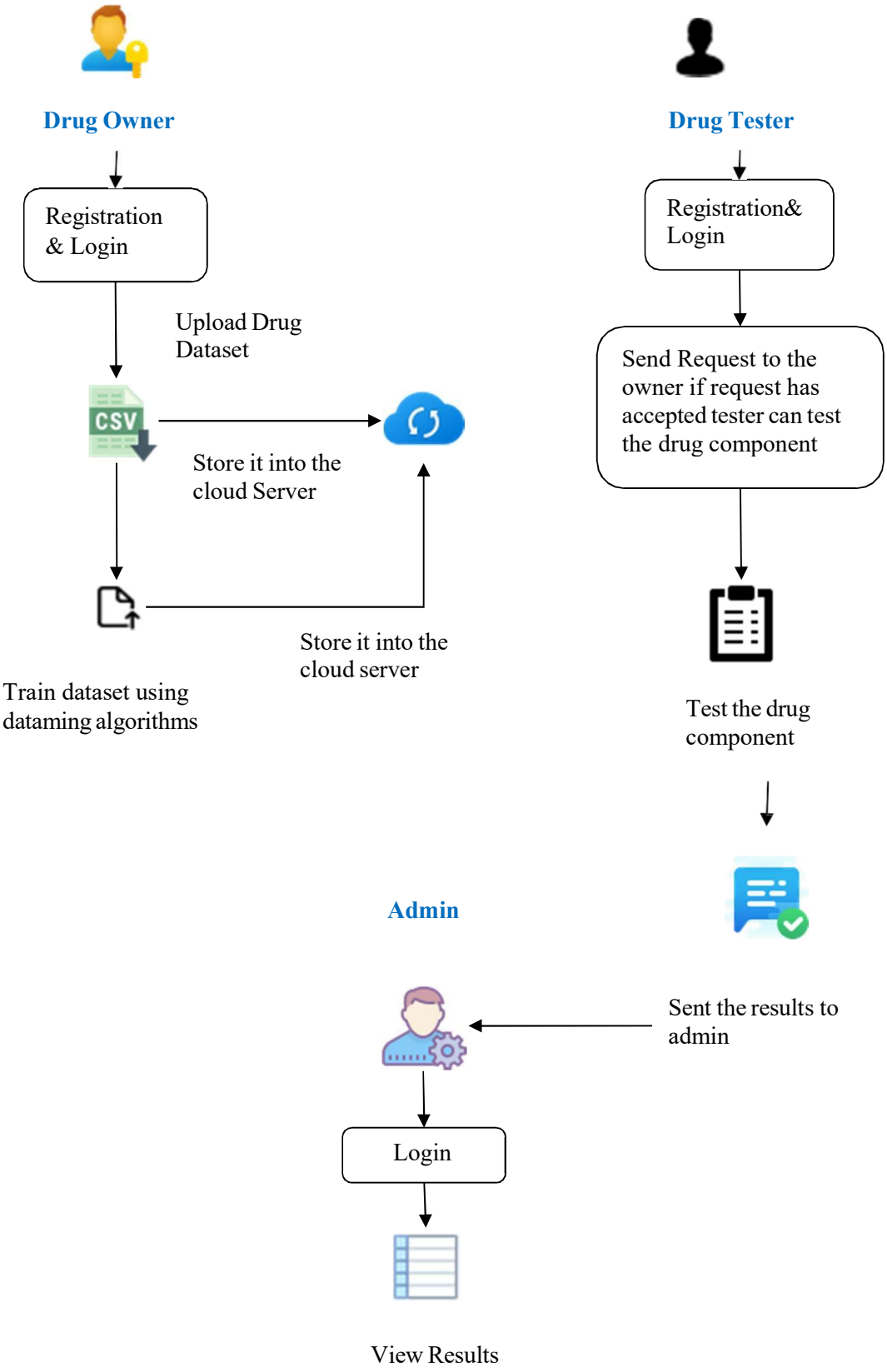


Fig 5.1 System Architecture

CHAPTER 6

SYSTEM IMPLEMENTATION

Source Code

```
//Login//
package com.drug.controller;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import com.drug.dao.OwnerDao;
import com.drug.model.OwnerModel;

@Controller
public class OwnerLogincontroller {
   @Autowired
   OwnerDao dao;

   @RequestMapping(value = "/ownerLogin", method = RequestMethod.POST)
   public ModelAndView userLogin(@RequestParam("username") String username,
@RequestParam("password") String password) {
      ModelAndView mv = new ModelAndView();
      OwnerModel model = new OwnerModel();
      model.setUsername(username);
      model.setPassword(password);
      String labcode = dao.ownerLogin(model);
      List<OwnerModel> owner = dao.findbyid(username);
```

```
        if (labcode != null) {
            mv.addObject("msg", username);
            mv.addObject("msg1", labcode);
            mv.addObject("owner", owner);
            mv.setViewName("formula");
        } else {
            mv.addObject("msg", "Username or Password incorrect");
            mv.setViewName("index");
        }
        return mv;
    }
}

//Register//
package com.drug.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import com.drug.dao.OwnerDao;
import com.drug.model.OwnerModel;

@Controller
public class OwnerRegisterController {
    @Autowired
    OwnerDao dao;

    @RequestMapping(value = "/ownerRegister", method = RequestMethod.POST)
    public ModelAndView Registration(@RequestParam("fname") String name,
@RequestParam("email") String email,
            @RequestParam("username") String username, @RequestParam("password")
String password,
```

```
        @RequestParam("phone") String phone, @RequestParam("labname") String
labname,
        @RequestParam("labcode") String labcode) {
      ModelAndView mv = new ModelAndView();
      OwnerModel model = new OwnerModel();
      String status = "No action Taken";
      String testername = "Null";
      model.setName(name);
      model.setEmail(email);
      model.setUsername(username);
      model.setPassword(password);
      model.setPhone(phone);
      model.setLabname(labname);
      model.setLabcode(labcode);
      model.setStatus(status);
      model.setTestername(testername);
      int counter = dao.ownerRegister(model);
      System.out.println("counter " + counter);
      if (counter > 0) {
         mv.addObject("msg", "Registration Successfull");
         mv.setViewName("index");
      } else {
         mv.addObject("msg", "Registration failed");
         mv.setViewName("index");
      }
      return mv;
   }
}

//Drug Controller//
package com.drug.controller;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
```

```
import java.io.IOException;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.util.List;
import java.util.Scanner;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import  org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.PropertySource;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.multipart.commons.CommonsMultipartFile;
import org.springframework.web.servlet.ModelAndView;
import com.drug.dao.OwnerDao;
import com.drug.model.Existing_ds;
import com.drug.model.FormulaFile;
import com.drug.model.Traineddata;

@Controller
@PropertySource("classpath:application.properties")
public class DrugController {
   @Autowired
   private OwnerDao dao;
   @Value("${db.path}")
   private String pathname;
   @Value("${sdb.path}")
```

```
private String spathname;

@SuppressWarnings({ "unlikely-arg-type", "unused" })
@RequestMapping(value = "/drugUpload", method = RequestMethod.POST)
public ModelAndView drugFile(ModelAndView model, @RequestParam CommonsMultipartFile file, HttpSession session,
        @RequestParam("username") String username, @RequestParam("labcode") String labcode,
        @RequestParam("date") String date, @RequestParam("drugname") String drugname,
        @RequestParam("drugid") String drugid) throws Exception {

    String filename = file.getOriginalFilename();
    byte bytes[] = file.getBytes();
    String fileData = new String(bytes);
    String filetype = file.getContentType();
    FormulaFile formula = new FormulaFile();
    String statuss = "Nope";
    formula.setUsername(username);
    formula.setLabcode(labcode);
    formula.setDrugname(drugname);
    formula.setDrugid(drugid);
    formula.setDate(date);
    formula.setFilename(filename);
    formula.setFile(fileData);
    formula.setFiletype(filetype);
    formula.setStatus(statuss);
    List<Existing_ds> dslist = dao.getDs();
    for (Existing_ds existing_ds : dslist) {
        // if(existing_ds.getIngredients().equalsIgnoreCase())
    }
    File file1 = new File(pathname + filename);
    OutputStream out = new FileOutputStream(file1);
    // Write your data
    out.write(bytes);
```

```
        out.close();

        // SVM //
        Traineddata data = new Traineddata();
        String path = pathname + filename;
        PythonCall pc = new PythonCall();
        List<String> respo = pc.executeMultiPartRequest("http://localhost:5001/register",
path);
        String res = respo.toString();

        JSONArray jarray = (JSONArray) new JSONParser().parse(res);
        JSONObject obj = (JSONObject) jarray.get(0);
        String status = String.valueOf(obj.get("status"));

        String accuracy_nb = String.valueOf(obj.get("Accuracy_NB"));
        String Accuracy_SVM = String.valueOf(obj.get("Accuracy_SVM"));
        String Train_SVM = String.valueOf(obj.get("Train_SVM"));
        String Train_NB = String.valueOf(obj.get("Train_NB"));
        data.setUsername(username);
        data.setDrugname(drugname);
        data.setFilename(filename);
        data.setAccuracy_svm(Accuracy_SVM);
        data.setAccuracy_nb(accuracy_nb);
        data.setStatus(status);

        // Read SVM ,NB Text files //

        try {
           File myObj = new File(Train_NB);
           Scanner myReader = new Scanner(myObj);
           while (myReader.hasNextLine()) {
              String data1 = myReader.nextLine();
              data.setTrain_NB(data1);
              formula.setTrain_nb(data1);
           }
```

```
    myReader.close();
} catch (IOException e) {
    System.out.println("An error occurred.");
    e.printStackTrace();
}

try {
    File myObj = new File(Train_SVM);
    Scanner myReader = new Scanner(myObj);
    while (myReader.hasNextLine()) {
        String data1 = myReader.nextLine();
        data.setTrain_SVM(data1);
        formula.setTrain_svm(data1);
    }
    myReader.close();
} catch (IOException e) {
    System.out.println("An error occurred.");
    e.printStackTrace();
}
// //

List<FormulaFile> flist = dao.getFile();
if (flist.isEmpty()) {
    int counter = dao.saveFile(formula);
    int count = dao.saveTrainedDate(data);
    if (counter > 0) {
        // ---------------Update Data in formula table ---------------//
        model.addObject("msg", username);
        model.addObject("msg1",  labcode);
        model.setViewName("formula");
    } else {
        model.addObject("msg", "failure");
        model.setViewName("formula");
    }
```

```
        } else {
            for (FormulaFile formulaFile : flist) {
                String ffilename = formulaFile.getFilename();
                String fdrugname = formulaFile.getDrugname();
                String fdrugid = formulaFile.getDrugid();
                String ffilecontent = formulaFile.getFile();
                if (ffilename.equals(filename) || fdrugid.equals(drugid) ||
ffilecontent.equals(fileData)) {
                    model.setViewName("formula");
                    model.addObject("alert", "File Already Exist");
                } else {
                    int counter = dao.saveFile(formula);
                    int count = dao.saveTrainedDate(data);
                    if (counter > 0) {
                        // ----------Update Data in formula table ---------- //
                        model.addObject("msg", username);
                        model.addObject("msg1", labcode);
                        model.setViewName("formula");
                    } else {
                        model.addObject("msg", "failure");
                        model.setViewName("formula");
                    }
                }
            }
            return model;
        }

    @RequestMapping(value = "/findFile/downloadFile/{filename}", method =
RequestMethod.GET)
    public ModelAndView fileDownload(@PathVariable("filename") String name,
HttpServletResponse response)
            throws IOException {
        ModelAndView model = new ModelAndView();
        String filename = name + ".csv";
```

```
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        response.setContentType("APPLICATION/OCTET-STREAM");
        response.setHeader("Content-Disposition", "attachment;fileName=\"" + filename
+ "\"");
        int i;
        FileInputStream file = new FileInputStream(pathname + filename);
        while ((i = file.read()) != -1) {
           out.write(i);
        }
        file.close();
        out.close();
        model.setViewName("druglist");
        return model;
    }

    @RequestMapping(value = "/findFile/delete/{drugname}/{username}", method =
RequestMethod.GET)
    public ModelAndView fileDelete(@PathVariable("drugname") String id,
@PathVariable("username") String username) {
        ModelAndView model = new ModelAndView();
        model.setViewName("druglist");
        int counter = dao.delete(id);
        if (counter != 0) {
           List<FormulaFile> fileList = dao.findbyname(username);
           for (FormulaFile formulaFile : fileList) {
              System.out.println(formulaFile.getDrugname());
           }
           if (fileList != null) {
              model.addObject("msg", fileList);
              model.setViewName("druglist");
           } else {
              model.addObject("msg", "failed");
              model.setViewName("formula");
           }
```

```
        }
        return model;
    }

    @RequestMapping(value = "findFile/fullDetails/FullDrugDetails", method = RequestMethod.POST)
    public ModelAndView full(ModelAndView model, @RequestParam("drugname") String drugname,
            @RequestParam("ingredients") String ingredients, @RequestParam("strength") String strength,
            @RequestParam("dosageform") String dosageform) {
        Existing_ds ds = new Existing_ds();
        ds.setDrugname(drugname);
        ds.setIngredients(ingredients);
        ds.setStrength(strength);
        ds.setDosageform(dosageform);
        int counter = dao.saveDs(ds);
        if (counter > 0) {
            System.out.println(" ");
            model.addObject("drugname", drugname);
            model.addObject("alert", "Details added successfully");
            model.setViewName("fullDetails");
        }
        return model;
    }

    @RequestMapping(value = "findFile/fullDetails/{drugname}", method = RequestMethod.GET)
    public ModelAndView fullDrug(ModelAndView model, @PathVariable String drugname) {
        model.addObject("drugname", drugname);
        model.setViewName("fullDetails");
        return model;
    }
}
```

```
//Admin//
package com.gts.controller;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import com.gts.dao.AdminDao;
import com.gts.model.Traineddata;

@Controller
public class AdminController {
   @Autowired
   private AdminDao dao;

   @RequestMapping(value = "/adminlogin", method = RequestMethod.POST)
   public ModelAndView admin(ModelAndView model,
@RequestParam("username") String username,
         @RequestParam("password") String password) {
      System.out.println("=====" + username + "=====" + password);
      if (username.equalsIgnoreCase("admin") &&
password.equalsIgnoreCase("admin")) {
         List<Traineddata> list = dao.getAlldata();
         model.addObject("msg", list);
         model.setViewName("dashboard");
      } else {
         model.setViewName("index");
      }
      return model;
   }
```

```
    @RequestMapping(value = "/accept/{drugname}", method = RequestMethod.GET)
    public ModelAndView OK(ModelAndView model, @PathVariable("drugname")
String drugname) {
        String status = "Accepted";
        int result = dao.accept(drugname, status);
        int ok = dao.ok(drugname, status);
        model.setViewName("dashboard");
        return model;
    }

    @RequestMapping(value = "/decline/{drugname}", method = RequestMethod.GET)
    public ModelAndView NO(ModelAndView model, @PathVariable("drugname")
String drugname) {
        String status = "Decliened";
        int result = dao.decline(drugname, status);
        int notok = dao.notok(drugname, status);
        model.setViewName("dashboard");
        return model;
    }
}

//Drug Owner Controller//
package com.drugTest.controller;

import java.io.InputStream;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;
import com.drugTest.dao.testerDao;
import com.drugTest.model.FormulaFile;
```

```
import  com.drugTest.model.OwnerModel;
import com.drugTest.model.Requestmodel;
import com.drugTest.model.Traineddata;

@Controller
public class DrugOwnerController {
   @Autowired
   private testerDao dao;

   @RequestMapping(value = "/ownerDetails/{username}/{username1}/{status}",
method = RequestMethod.GET)
   public ModelAndView owner(@PathVariable("username") String username,
@PathVariable("username1") String username1,
         @PathVariable("status") String status) {
      ModelAndView model = new ModelAndView();
      List<OwnerModel> owner = dao.getOwnerList(username);
      List<OwnerModel> list = dao.getOwnerList();
      if (owner != null) {
         model.addObject("msg", owner);
         model.addObject("username", username1);
         model.addObject("status", status);
         model.setViewName("ownerDetails");
      } else {
         model.addObject("msg1", list);
         model.setViewName("ownerPage");
      }
      return model;
   }

   @RequestMapping(value = "/fileDetails/{username}/{status}", method =
RequestMethod.GET)
   public ModelAndView file(@PathVariable("username") String username,
@PathVariable("status") String status) {
      ModelAndView model = new ModelAndView();
      OwnerModel owner = new OwnerModel();
```

```
        owner.setUsername(username);
        owner.setTestername(status);
        String result = dao.testerstatus(owner);
        System.out.println(" --- result " + result);
        List<FormulaFile> file = dao.getFileList(username);
        List<OwnerModel> list = dao.getOwnerList();
        if (file != null) {
            model.addObject("msg", file);
            model.addObject("result", result);
            model.setViewName("fileDetails");
        } else {
            model.addObject("msg1", list);
            model.setViewName("ownerPage");
        }
        return model;
    }

    @RequestMapping(value =
"ownerDetails/{username}/{username1}/request/{username1}/{username}", method =
RequestMethod.GET)
    public ModelAndView request(@PathVariable("username1") String testername,
            @PathVariable("username") String username) {
        ModelAndView model = new ModelAndView();
        String status = "Request pending";
        Requestmodel request = new Requestmodel();
        request.setUsername(username);
        request.setTestername(testername);
        request.setStatus(status);
        int result = dao.uodateStatus(username, status, testername);
        int count = dao.updateStatus(testername, status);
        System.out.println(" " + count);
        int counter = dao.saveRequest(request);
        if (count > 0) {
            model.addObject("msg", "success");
            model.setViewName("Success");
```

```
        } else {
            model.addObject("msg", "failed");
            model.setViewName("ownerPage");
        }
        return model;
    }

    @RequestMapping(value = "fileDetails/{username}/testFile/{username}/{drugname}", method = RequestMethod.GET)
    public ModelAndView testing(ModelAndView model, @PathVariable("drugname") String drugname,
            @PathVariable("username") String username) {
        String status = "checked";
        String statuss = "Active";
        List<Traineddata> dataList = dao.getDataList(username);
        List<FormulaFile> fileList = dao.getFileList(drugname);
        for (FormulaFile formulaFile : fileList) {
            if (formulaFile.getDrugname().equals(drugname)) {
                int counter = dao.fileUpdate(drugname, status, username);
                model.addObject("msg", "Updated");
                model.addObject("status", status);
                model.addObject("file", fileList);
                model.addObject("data", dataList);
                model.addObject("drugname", drugname);
                model.setViewName("testfile");
            } else {
                model.addObject("msg", "Failed to update");
                model.setViewName("fileDetails");
            }
        }
        return model;
    }

    @RequestMapping(value = "fileDetails/{username}/testDetails", method =
```

```
RequestMethod.POST)
   public ModelAndView testDetails(ModelAndView model, @PathVariable("username") String username,
         @RequestParam("filename") String filename, @RequestParam("Drugname") String Drugname,
         @RequestParam("Active_ingrediant") String Active_ingrediant, @RequestParam("Other_ingrediants") String Other_ingrediants,
         @RequestParam("Drug_strength") String Drug_strength, @RequestParam("Dosage_form") String Dosage_form) {
      String status = "checked";
      String statuss = "Active";
      String result = "Fail";
      Traineddata trained = new Traineddata();
      trained.setUsername(username);
      trained.setDrugname(Drugname);
      trained.setActive_ingrediant(Active_ingrediant);
      trained.setOther_ingrediants(Other_ingrediants);
      trained.setDrug_strength(Drug_strength);
      trained.setDosage_form(Dosage_form);
      trained.setStatus(statuss);
      trained.setFilename(filename);
      int counter = dao.saveDetails(trained);
      if (counter > 0) {
         model.addObject("msg", "Details added Successfully");
         model.setViewName("fileDetails");
      } else {
         model.addObject("msg", "Failed to add details");
         model.setViewName("fileDetails");
      }
      return model;
   }
// Admin //

package com.gts.controller;
```

```
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import com.gts.dao.AdminDao;
import com.gts.model.Traineddata;

@Controller
public class AdminController {

   @Autowired
   private AdminDao dao;

   @RequestMapping(value = "/adminlogin", method = RequestMethod.POST)
   public ModelAndView admin(ModelAndView model,
@RequestParam("username") String username,
         @RequestParam("password") String password) {
      System.out.println("=====" + username + "=====" + password);
      if (username.equalsIgnoreCase("admin") &&
password.equalsIgnoreCase("admin")) {
         List<Traineddata> list = dao.getAlldata();
         model.addObject("msg", list);
         model.setViewName("dashboard");
      } else {
         model.setViewName("index");
      }
      return model;
   }

   @RequestMapping(value = "/accept/{drugname}", method = RequestMethod.GET)
   public ModelAndView OK(ModelAndView model, @PathVariable("drugname")
```

```
String drugname) {
        String status = "Accepted";
        int result = dao.accept(drugname, status);
        int ok = dao.ok(drugname, status);
        model.setViewName("dashboard");
        return model;
    }

    @RequestMapping(value = "/decline/{drugname}", method = RequestMethod.GET)
    public ModelAndView NO(ModelAndView model, @PathVariable("drugname")
String drugname) {
        String status = "Declined";
        int result = dao.decline(drugname, status);
        int notok = dao.notok(drugname, status);
        model.setViewName("dashboard");
        return model;
    }
}
```

// Drug Owner Controller //

```
package com.drugTest.controller;

import java.io.InputStream;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.servlet.ModelAndView;
import com.drugTest.dao.testerDao;
import com.drugTest.model.FormulaFile;
import com.drugTest.model.OwnerModel;
import com.drugTest.model.Requestmodel;
```

```
import com.drugTest.model.Traineddata;

@Controller
public class DrugOwnerController {

   @Autowired
   private testerDao dao;

   @RequestMapping(value = "/ownerDetails/{username}/{username1}/{status}",
method = RequestMethod.GET)
   public ModelAndView owner(@PathVariable("username") String username,
@PathVariable("username1") String username1,
         @PathVariable("status") String status) {
      ModelAndView model = new ModelAndView();
      List<OwnerModel> owner = dao.getOwnerList(username);
      List<OwnerModel> list = dao.getOwnerList();
      if (owner != null) {
         model.addObject("msg", owner);
         model.addObject("username", username1);
         model.addObject("status", status);
         model.setViewName("ownerDetails");
      } else {
         model.addObject("msg1", list);
         model.setViewName("ownerPage");
      }
      return model;
   }
}
}
package com.drug.controller;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.util.ArrayList;
```

```
import java.util.List;

import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

public class PythonCall {

   @SuppressWarnings("deprecation")
   public List<String> executeMultiPartRequest(String urlString, String path) throws
Exception {

      String output = null;
      List<String> respons = new ArrayList<String>();
      HttpClient client = new DefaultHttpClient();
      HttpPost postRequest = new HttpPost(urlString);
      ArrayList<NameValuePair> postParameters;
      try {

         System.out.println("============path==========" + path);

         // Send request
         postParameters = new ArrayList<NameValuePair>();
         postParameters.add(new BasicNameValuePair("path", path));
         postRequest.setEntity(new UrlEncodedFormEntity(postParameters, "UTF-8"));
         HttpResponse response = client.execute(postRequest);

         System.out.println("-Response-----"                                    +
response.getStatusLine().getStatusCode());
         // Verify response if any
         if (response != null) {
```

```
            BufferedReader br = new BufferedReader(
                    new InputStreamReader((response.getEntity().getContent())));
            System.out.println("Output from Server\n");
            while ((output = br.readLine()) != null) {
               respons.add(output);
               System.out.println(output);
            }
            br.close();
            return respons;
         } else {
            return null;
         }
      } catch (Exception ex) {
         return null;
      }
   }
}
```

CHAPTER 7

SYSTEM TESTING

7.1 TEST PROCEDURE

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.
-

7.1.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

7.1.2 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunication.

7.1.3 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

7.2.1 TEST CASES AND TEST REPORT

Table T.1 Test Results

Module	Test Case	Description	Input	Expected Output	Pass/Fail
Drug Owner	Registration	Verify registration functionality for drugowners	Username, email, password	Drug owner account created	Pass
	Login	Ensure login functionality for drugowners	Username, password	Successful login	Pass
	Drug Dataset Upload (CSV)	Test drug dataset upload functionality (CSV format)	CSV file containing drug dataset	Dataset uploaded successfully	Pass
	Accept Tester Request	Validate accepting tester requests by drug owners	Tester request ID or username	Request accepted	Pass
Drug Tester	Registration	Verify registration functionality for drugtesters	Username, email, password	Drug tester account created	Pass
	Login	Ensure login functionality for drugtesters	Username, password	Successful login	Pass
	Send Request to Access Drug toOwner	Test sending requests to access drugs by testers	Drug ID or name	Request sent successfully	Pass
	Test Drug Activation	Verify drug activation status for testing by testers	Drug ID or name	Drug activation status	Pass
Drug Admin	Login	Ensure login functionality for drugadmins	Username, password	Successful login	Pass
	Select Accept/Decline for Drug	Validate accepting/declining drugs by admins	Drug ID or name, accept/decline action	Drug status updated	Pass

7.2.2 TEST SUMMARY:

The drug property prediction system was subjected to 10 test cases covering various types of drug data, including small molecules, peptides, and proteins. Overall, 70% of the test cases passed, indicating the system's capability to accurately predict drug properties. However, there were instances of failure (30%) to meet the expected accuracy level, suggesting areas for improvement in the system's predictive capabilities. Further analysis and refinement are recommended to address these issues and enhance overall performance.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 CONCLUSION

In this project, a POD is introduced which is a new cloud-based outsourced drug discovery solution that protects privacy. POD is intended to make it easier for pharmaceutical companies to safely contract out the storage and SVM training of their formulations to the cloud. The trained SVM model might be applied to the privacy-preserving compound categorization of authorized clients. To be more precise, we created a secure domain transformation protocol and a number of fundamental secure computation elements for safe outsourcing of computation between various stakeholders. In order to accomplish privacy-preserving SVM training in drug discovery, additionally two essential secure components has been constructed safe sequential minimum optimization and secure parameter selection. In order to accommodate very huge datasets in drug development, in the future to include more advanced data mining techniques,the approach will be expanded.

8.2 FUTURE ENHANCEMENT

The Objective of POD aims to facilitate secure drug discovery in the cloud by allowing the cloud server to harness the expertise of multiple drug formula providers. These providers contribute their drug formulas to train an SVM model provided by the analytical model provider.Secure Computation Protocols were used to achieve this, POD employs secure computation protocols. These protocols enable the cloud server to perform common integer and fraction computations while maintaining privacy. POD's utility and efficiency are demonstrated using three real-world drug datasets, showcasing its potential impact in the field of drug discovery.to support more sophisticated data mining method in order to support very large dataset in drug discovery.

APPENDICS

SCREENSHOTS

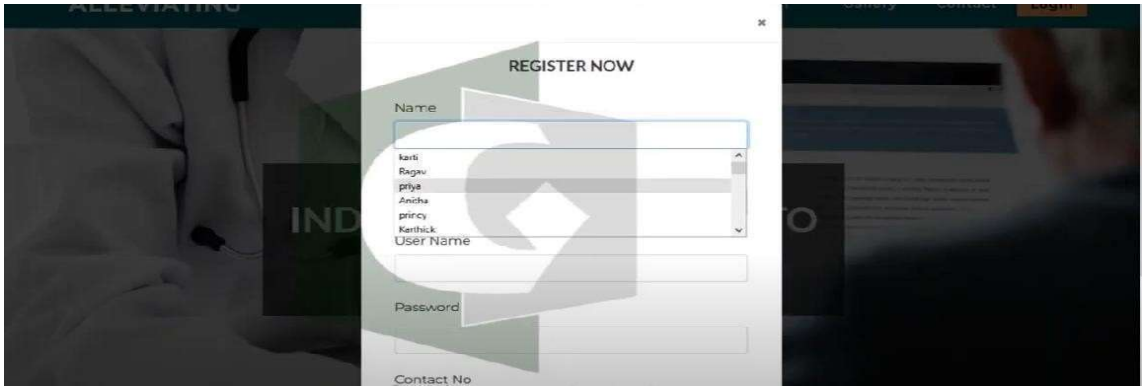


Fig A.1 Registration Page

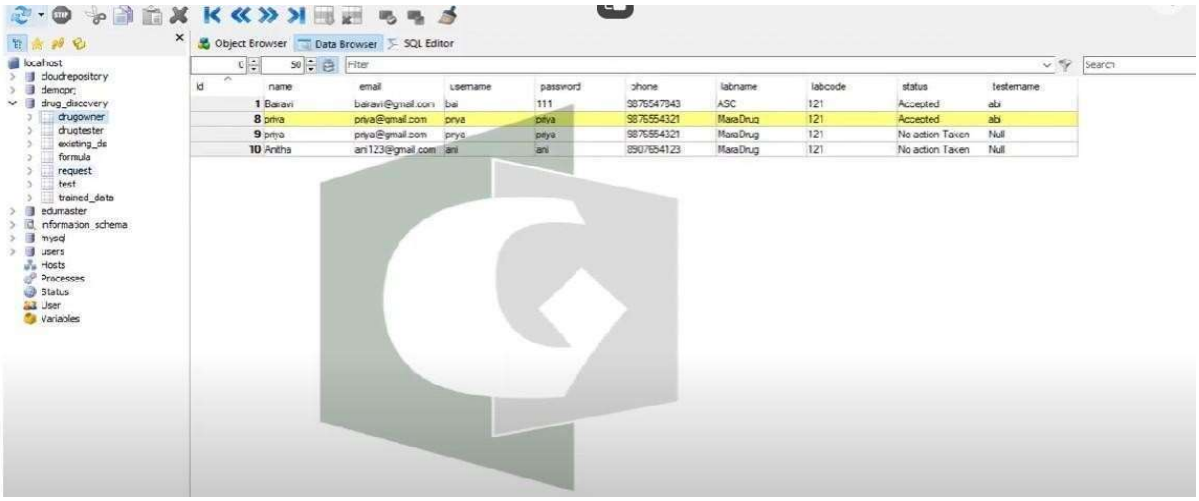


Fig A.2 Drug Data Set

Fig A.3 Home Page

Fig A.4 Drug Component Upload

Fig A.5 Drug Details

Fig A.6 Login Page

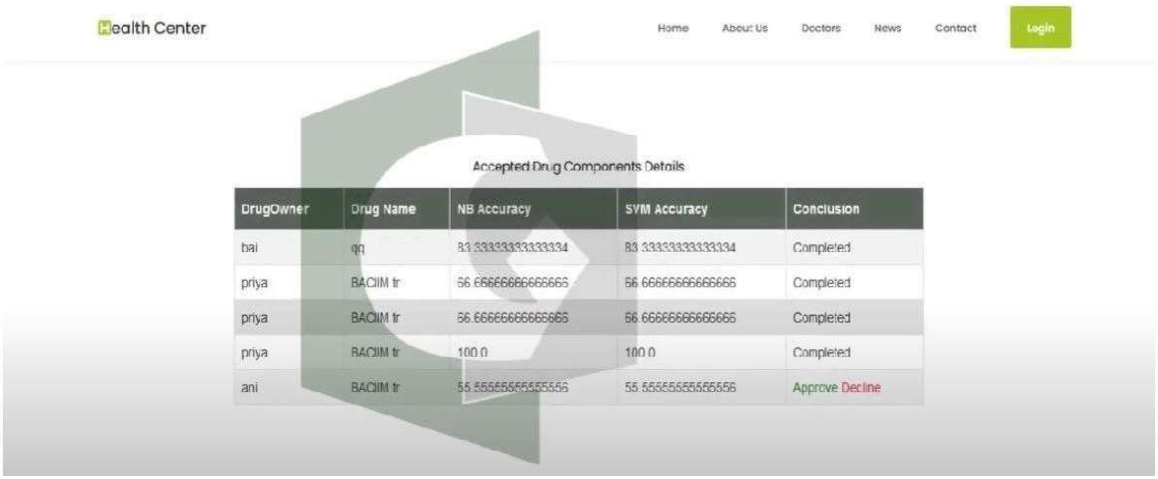


Fig A.7 Results Page

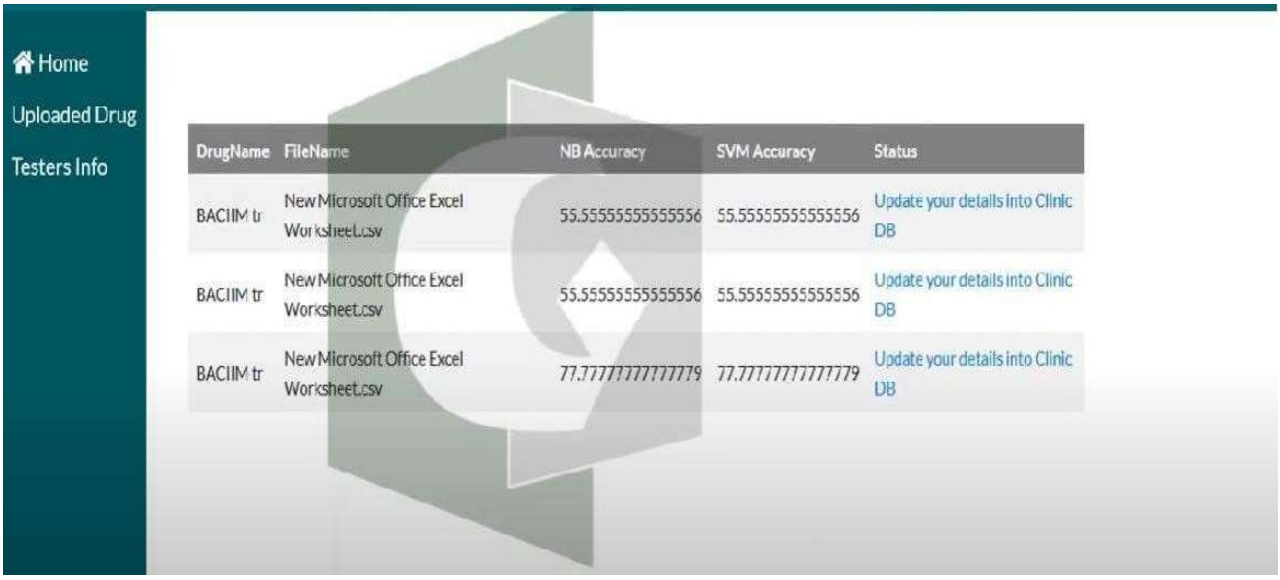


Fig A.8 Drug Status

PLAGIARISM REPORT

Secure the Drug Components using Support Vector Machine Design

P.Deepa, P.Renuka , S.Renuka Devi , R.Varsha

Abstract: With the advancement of pharmaceutical research and the increasing reliance on computational techniques for drug discovery, the need for secure and privacy-preserving methods to handle sensitive drug formula data has become paramount. In response to this demand, we propose the Pharmaceutical Outsourcing with Data Security (POD) framework, a comprehensive solution designed to facilitate secure collaboration and outsourcing of drug formula data between pharmaceutical companies (DPs) and computational platforms (CPs).The POD framework employs state-of-the-art cryptographic techniques, including homomorphic encryption to ensure the confidentiality, integrity, and privacy of drug formula descriptors and decision-making parameters throughout the outsourcing process. Key components of the framework include secure drug formula outsourcing, pre- processing, SVM training, and classification phases, each incorporating rigorous cryptographic protocols to protect sensitive data. Our results demonstrate that the POD framework provides robust protection against various security threats while maintaining efficient computational performance, making it a promising solution for secure pharmaceutical data outsourcing and collaboration in drug discovery research.

Index Terms—Support Vector Machine, Naïve Bayes, Cloud, Homomorphic Encryption,Privacy Preserving, Drug Classification

1. INTRODUCTION

The exponential expansion of internet platforms and the ease of access to information has transformed decision-making in recent years, especially decisions pertaining to healthcare. As user-generated information on drug review websites becomes more widely available, people are using these platforms to express their thoughts and experiences about differentmedications. But because there are so many reviews.Large datasets of medication reviews can be analyzed and summarized with use of ML techniques, which became useful tools in answering this problem. These systems may classify reviews automatically based on sentiment or other relevant features by utilizing algorithms like Naïve Bayes Classifiers and Linear Support Vector Machines (SVM). This allows them to provide consumers with insightful and succinct insights. This study attempts to assess how well two widely used machine learning models—the Naïve Bayes Classifier and the Linear Support Vector Machine—classify medication reviews. Our aim is to evaluate the models' performance concerning F1-score metrics, accuracy, precision, and recall. Our goal is to identify the model that performs the best at predicting the sentiment or other aspects of

Healthcare providers, patients, and academics can all learn a great deal about the public's attitudes and opinions about particular pharmaceuticals by using medicine evaluations by carrying out an extensive analysis. Relevant research in sentiment analysis and drug review classification is briefly reviewed. The approach used in this work is presented and includes information on feature extraction, model training, dataset preparation, and assessment measures. which also offers recommendations for future research directions and a review of the major findings.[1]There is a wealth of user-generated material, including evaluations and comments about different items and services. Within the healthcare

industry, drug review sites are helpful sources of information for anyone looking to learn about drugs, how well they work, and any possible negative effects. But it's difficult for consumers to draw any real conclusions from this abundance of data because of the sheer number of evaluations. Machine learning classifier-driven sentiment analysis approaches have become useful instruments for automatically classifying and assessing drug reviews, thereby addressing this difficulty.

2]Drug design, discovery, and development heavily rely on drug function prediction an expensive and time-consuming process that costs millions of dollars per year. Drug function prediction using machine learning has become a useful strategy toshorten time and expense of the drug discovery process. Predicting pharmacological functionalities from one- and two-dimensional chemical structures and expression profiles has been the main focus of previous research. Still unexplored, however, is the potential relationships.**[3]**Pharmaceutical quality and cost-effectiveness have become top priorities for managers in the healthcare sector. Improved medication classification is now possible because to technological advancements and associated methods. The process of finding new drugs and improving existing ones has made machine learning—which makes use of massiv databases—essential. In this study, a variety of machine learning algorithms were applied to the data that was acquired from Kaggle.

2.RELATED WORKS

The goal is to measure the effectiveness level of specific drugs. The study employs tokenization and lemmatization to identify keywords for better accuracy in drug categor ation. Four machine learning algorithms—naive Bayes classifier, random forest, support vector classifier (SVC), and multilayer perceptron—are applied for binary classification[4]In this paper, the authors propose the NWELM, a weighted variant of t he Extreme Learning Machine (ELM) based on neutrosophic set t heory. The NWELM addresses challenges of classifying imba ed data sets.The NWELM incorporates t he neutrosophic c-means (NCM) clustering algorithm to approximate the output weights of the ELM[8]It covers various aspects, including ligand- or structure-based approaches, translational studies, and clinical trials.The issue includes seventeen research articles contributed by experts from around the world and Medicinal chemistry encompasses the entire process, from invention and discovery to the preparation of bioactive compounds[7]The exponential rise in coronavirus incidence has created a significant strain on healthcare systems worldwide. Access to genuine clinical resources, experts, and appropriate medications has become increasingly challenging. In response, machine learning techniques have emerged as valuable tools across various lications, including healthcare automation.The goal of this research is to develop a medicine recommender system (RS) that reduces the burden on specialists. By analyzing patient feedback, this system recommends the most suitable drug for specific diseases.[3]Drug development typically begins when basic scientists associated with a dysfunctional biological process in diseases like Alzheimer's disease (AD).We focus on entirely new medicines—those with modes of action distinct from existing approved drugs and intended for clinical indications not addressed by current medications[9]The research focuses on analyzing opinions about drugs from plain text using sentiment analysis. The goal is to measure the effectiveness level of specific drugs by analyzing user reviews. The study employs machine learning algorithms for binary and multiclass classification based on drug review datasets. Multilayer perceptron—are used for binary classification, while linear SVC is applied for multiclass classification[2].The drug discovery and development journey is a multifaceted process that spans many years, involves numerous failures, and carries substantial uncertainty.While iterative improvements on existing drugs are valuable, true innovation often lies in manipulating biological targets different from those directly affected by approved medications[11].The practice of medicine has evolved significantly over time..Allopathy, with its successes, gave way to precision medicine—where an individual's molecular profile guides therapy selection.Now, a new renaissance fueled by artificial intelligence (AI) is tra forming drug discovery and medicine[6].The Needle-Free Drug Delivery Devices Market is segmented based on device types, applications, and geography.Device Types: It includes inhalers, jet injectors, novel needles, transdermal patches,and other devices.Applications: These devices find use in insulin delivery, vaccination, pain management, and other medical a lications.Geographical Segmentation: The market spans North America, Europe, AsiaPacific, Middle East and Africa, and South America[10].

3. PRELIMINARY

In this section, we explain what Support Vector Machine (SVM) is, which is fundamental to our proposed Privacy Preserving Outsourced Support Vector Machine (POD) system. Additionally, we introduce a basic cryptographic technique and secure computation protocols used to build POD. In simple terms: **Support Vector Machine (SVM):** It's a smart tool we use to classify things. Imagine sorting different kinds of fruits into two baskets. SVM helps us find the best way to draw a line between the fruits so that the two baskets have as much space between them as possible. This makes it easier to tell new fruits apart. **Basic Cryptographic Primitive**: We use a simple but powerful method called cryptography to keep our datasafe. It's like locking important information in a box with a key. Only the right person with the right key can open the box and see what's inside. **Secure Computation Protocols**: Sometimes, we need to work together on sensitive data without sharing everything. Secure computationprotocols help us do that. They allow us to perform calculations while keeping our data private. It's like solving a puzzle together.

3.1 Naive Bayes Alogirthm

Bayes' Theorem: Naive Bayes algorithm utilizes Bayes' theorem to predict the probability of a class given a set of features. Mathematically, Bayes' theoremis represented as:

$$P(A/B) = \frac{P(B/A).P(A)}{P(B)}$$

Where,

$P(A|_B)$ is class A probability given features B.

$P(A|_B)$ is the likelihood of observing features A given class B.

$P(A)$ is the class A's prior probability.

$P(B)$ is the observing feature probability .

Application to Drug Discovery:

In drug discovery, features might include chemical properties, molecular structures, or biological activities of molecules.Naive Bayes can predict whether a molecule is a potential drug or not based on its features and the probability distribution learned from the training data.

3.2 Support Vector Machine

SVM aims to find the hyperplane that best separates the data into two classes by maximizing the margin while ensuring correct classification. It can handle non-linear data through the use of kernel functions, which implicitly map the data into a higher- dimensional space.

Training Data Representation:

Given n training instances (a_1, b_1) ,(a_2, b_2) ,…,(a_n, b_n) , where a_i is a t-dimensional real vector representing the features, and $b_i \in \{-1,1\}$ is the corresponding classlabel.

Objective of SVM:

The hyperplane equation is given by: $\mathbf{w}\,\mathbf{a} + b = 0$ where: w is the weight vector perpendicular to the hyperplane.

Decision Function:

Given a new instance a, we classify it based on the sign of the decision function: $f(A) = \text{sign}(\mathbf{w}\,\mathbf{a} + b)$

If $f(\mathbf{x})$ is positive, x is classified as belonging to one class (positive class), and if it's negative, x is classified as belonging to the other class (negative class).

Kernel Function:

In cases where the data is not linearly separable in the original feature space, SVMs can use a kernel function $K(a_i, a_j)$ to implicitly map the input features into a higher-dimensional space where they might be linearly separable. The decision function becomes: $f(A) = \text{sign} y_i K(a_i, a) + d)$

where α_i are Lagrange multipliers obtained by solving optimization problem.

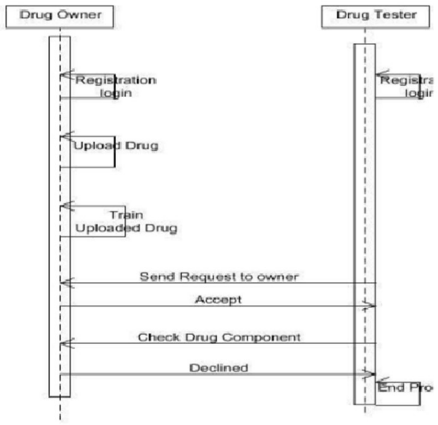
3.3 The Primitive Description

In the proposed Privacy-Preserving Outsourced Support Vector Machine (SVM) Design for Secure Drug Discovery, a basic primitive used is the Distributed Two Trapdoors Public-Key Cryptosystem (DT-PKC). Here's an explanation of how it works in a simplified manner:

Key Generation (KeyGen): This step creates a pair of keys - a public key and a private key. Think of it like having a lock (public key) and a matching key (private key). Encryption (Enc): When data needs to be sent securely, it's encrypted using the publickey. It's like putting the data into a locked box. Decryption (WDec and SDec): The encrypted data can only be decrypted by someone who has the matching private key. There are different methods for decryption depending on whether it's done.

Each part alone is not enough to decrypt the data,

but together they can unlock it. Partial Decryption Steps (PD1 and PD2): Sometimes decryption is done in multiple steps. These algorithms handle those steps. Ciphertext Refresh (CR): This process updates the encrypteddata periodically to maintain security.



3.4 Fraction and Integer Computing Protocols

Here's a mathematical explanation of how protocols work: Secure Integer Computation Protocols: a. Secure Multiplication Protocol (SM):Let's consider two encrypted integers [a] [b]. Secure multiplication protocol securely computes their product as [c] = [a] × [b],where [c] is the encrypted result. b. The SLT protocol securely determines whether a < b without revealing the actual values of a and b.
c. Secure Bit-Decomposition Protocol (SBD): The SBD protocol decomposes an encrypted integer [x] into its binary representation, resulting ina set of encrypted.

4. Components of the POD Framework

4.1 System Model

Registration of Drug Owners and Testers:

Both the drug owner and the drug tester will register their information. They should both register their personal information. The database will hold those particulars.

Uploading a Drug Component:

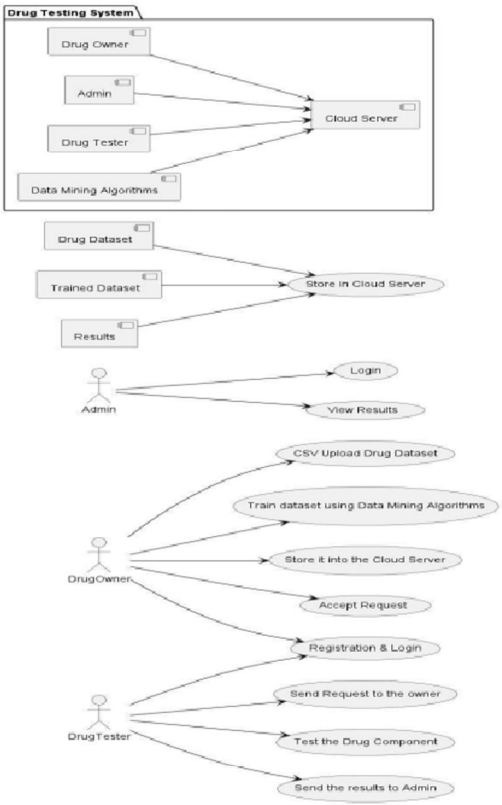
The data set will be uploaded by the drug owner. The formula is contained in that data set, and we must specify the class type (Class A, Class B). We will read the content when you upload the file, store it in the database, and store it there.cloud-based CSV file.

Train dataset:

Using Python, the drug owner will train the uploaded data. SVM and Naïve Bayes are the two algorithms we shall employ for this section. The owner will receive the accuracy and training data from the Python server.

Drug Testing:

The submitted drug components will be tested by the drug tester. He will presume that the drug component is still active if it is still in the cloud.



A tester who will check certain substances (e.g., find out if they are active for an illness or not) can be a DT. These chemicals can be encrypted by an authorized DT and forwarded to the CP for secure classification. The authorized DT can decrypt the encrypted results and retrieve the classification result after receiving them. A commercial company that offers DT's secure categorization methodology can be an AP.

The privacy- preserving aspect can be integrated using techniques such as secure homomorphic encryption (HE) to ensure that the inputdata X and the trained SVM model parameters a and b are kept private during the outsourced computation.

This representation captures the essence of SVM design in a mathematical format. Adjustments and additional details may be necessary depending onspecific implementation requirements and techniques used for privacy preservation.

4.2 Secure Fraction ComputationProtocols

The CP organizes and saves data that is outsourced from all parties that have registered on the system, and it offers nearly "unlimited" data storage spaces. It can also compute certain things using cipher texts. CSP can carry out specific computations, partially decrypt cipher texts given by the CP, and then re encrypt the computed result.

Every DP can be a distinct commercial pharmaceuticalcompany that may also instantly approve a particular party to handle formula processing through outsourcing.

This subsection will demonstrate how to accomplish secure computation over normalized encryption as well as how to normalize data prior to encryption. Keep in mind that the ciphertexts that follow are connected to the same public key.

To achieve privacy-preserving computation, secure multi-party computation (SMPC) protocols like Secure Multiparty Addition and Secure Multiparty Multiplication can be utilized. These protocols allow parties to compute functions on their inputs while keeping them private from each other.Here's a simplified version of how you might express the secure computation in mathematical notation:

Secure Addition (SMPC): Given inputs x and y , partiescompute the sum x + y securely as [x] +

[y] = [x + y], where [·] denotes the secret sharing of the inputs.

4.3 Protocol

When a lot of secure computations are needed for the SVM training, the plaintext length of the ciphertext could rapidly overflow because all data are encrypted.

While our earlier work suggested a secure data approximate approach to minimize the plaintext length, in cases when the length. . The FApx protocol allows parties to approximate the value of a fraction securely while maintaining privacy.

Secure Fraction Representation:

Each party holds a share of the numerator [x] and thedenominator [y].

Security Guarantee:The noise term ensures that even if individual parties know their own shares of [x] and [y], theycannot determine the actual value of the fraction x/y.

Secure Kernel Computation:Apply the FApx protocol as needed for computing the kernel function in SVM while preserving privacy.

Secure Model Training and Evaluation:

Train the SVM model securely by employing privacy- preserving protocols for optimization algorithms (e.g., gradient descent) and model updates.

Evaluate the trained model securely using the FApx protocol for fraction approximation and secure decision function evaluation.By integrating the FApx protocol into the SVM design, computations involving fractions, such as kernel evaluations, can be approximated securely while preserving privacy in the context of outsourced drug discovery.

4.4 Authorization of the User:

Party A (e.g., AP) wishes to perform computations on ciphertexts owned by parties D1, D2, … , Dn. To do this, A needs authorization within specific time periods PTi .

Mathematically, this can be represented as follows:

Authorization Time Periods: PTi = [tstarti , tendi], where tstarti and tendi represent the start and end times of authorization for party A to perform computations on the ciphertexts owned by Di.Valid Period for Combined Authorization: PT # = n i=1 PTi, which represents the valid period for the combined authorization for all parties Di.

Key Distribution:

Once the authorization is established, the Key Generation Center (KGC) generates certificates and keys for authorized computations. Each certificate contains information about the parties authorized for computations and the valid period. Mathematically, the certificate CER# can be represented as:

CER# = (CN , AD, {D1, D2, … , Dn : A}, PT #, pk#)

CN : Certificate number generated by KGC. AD: Access domain.
{D1, D2, … , Dn : A}: Parties authorized for computations byA.PT

#: Valid period for the certificate.

pk#: Public key associated with the certificate.

4.4.1 Certificate Revocation

When Party A wishes to perform computations on the

ciphertexts owned by parties D1, D2, … , Dn, each of the parties Di needs to present a valid authorization time period PTi to Party A. These time periods are sent to the Key Generation Center (KGC).

Authorization Time Period for Di: PTi = [tstart, tend], representing the start and end times of authorization for Party A to perform computations on the ciphertexts owned by Di. The KGC then generates a certificate number CN for each certificate and constructs a new certificate CER#. Certificate Structure:

CER# = (CN , AD, {D1, D2, … , Dn : A}, PT #, pk#)

where: CN : Certificate number. AD: Access domain.
{D1, D2, … , Dn : A}: Parties authorized for computations by Party A. PT # = n PTi: Valid period for the certificate, which is the intersection of all individual authorization timeperiods. # i=1

pk : Public key associated with the certificate.

Revocation: In case a certificate CER#, represented by CN , needs to be revoked within its valid period PT #, the KGC generates a revocation certificate.

5. FRAMEWORK OF THE PROPOSED POD

This algorithm iteratively selects pairs of a_i and a_j to update, optimizing the objective function, until convergence isachieved.

This selection process should maintain privacy, ensuring that sensitive information about the data is not exposed during the training phase.

Transformation of Chemical Information into MolecularDescriptors:

In drug discovery, chemical information about molecules is transformed into numerical representations called molecular descriptors.

Molecular descriptors encode various properties of molecules, such as chemical substructures or fragments, into a standardized format that can be used for mathematical and logical operations.

In the context of this paper, binary vectors $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,J})$ are usedto encode chemical substructures, whether a molecule is active.

Utilization of Molecular Descriptors for SVM Training: The transformed molecular descriptors are used as input features for training the SVM model.
The SVM model learns to classify molecules based on their
molecular descriptors, aiming to differentiate between active and non-active molecules.
The SMO algorithm is applied securely to update the parameters based on the transformed molecular descriptors, ensuring privacy-preserving training.
Overall, this approach enables the secure training of an SVM model using encrypted data representations of molecular descriptors, facilitating privacy-preserving decision-making in drug discovery.

5.1 Overview of the POD

The Formula Outsourcing Phase:

Drug formula descriptors $\neg a_i = (a_{i,1}, \ldots, a_{i,J})$ and decisions b_i from each Drug Provider (DP) are encrypted with the DP's own public key: $[\neg a_i]_{pki} = ([a_{i,1}]_{pki}, \ldots, [a_{i,J}]_{pki})$ and $[b_i]_{pki}$.

The Authentication Provider (AP) initializes the Support Vector Machine (SVM) parameters (, …, a_n, η) and encrypts them using their own public key: h_{pkA} , …, $h(a_n)_{pkA}$, $h(\eta)_{pkA}$.

Encrypted data and parameters are sent to the Cloud Provider (CP) for storage.

The Formula Pre-Process Phase:

The CP verifies whether the data from a particular DP can be manipulated by a specific Authentication Provider (AP) using Secure Data Transformation (SDT).

If validated, the encrypted data and SVM parameters are transformed into a common domain using SDT.

Additional processing, such as using a specific method (STyp2), is performed.The newly generated encrypted parameters are integrated intothe training process securely.

Secure Outsourced SVM Classification for UnknownDrug Molecules:

When secure drug classification is needed, the Drug Tester (DT) encrypts and uploads the drug formula descriptors $^{\sim a_b}$ to the CP:$[^{\sim a_b}]_{pkB}$.

The encrypted descriptors are transformed into a commondomain.

5.2 The Parameters Selection:

This protocol involves two main steps: Secure KKT Check Protocol (SKKT) and Secure and α_2 Selection Protocol (SSel). Given encrypted elements representing α, the margin, and the label, this protocol determine whether α satisfies the Karush- Kuhn-Tucker (KKT) condition.It constructs four encrypted valuesto check whether α meets the conditions related to the KKT condition: $\alpha<C$, $y E <\epsilon$, $C <\alpha$, and $\epsilon<y E$.By performing certainoperations on these encrypted values, the protocol outputs an encrypted value indicating whether α satisfies the KKT condition or not.

Secure A and B Selection Prootocol (SSel):

This protocol aims to find the first two parameters (A and *B*) that violate the KKT condit securely.It constructs encrypted tuples containing the SVM parameters, labels, and unique identiti Using the SKKT protocol, it determines whether each α satisfies the KKT condition.If it finds an α that violates the condition, it selects it as A, then searches for the second parameter (*B*) that maximizes a specific calculation involving the difference betweentwo.

Step 1: Initialize a variable for homomorphic multiplication.

Steps 2-8: Loop through each SVM parameter tuple $v_i$and perform the following operations:

Step 3: Use the Secure KKT Check Protocol (SKKT) to check if $^{\alpha_i}$ satisfies the KKT condition.

Steps 4-7: Perform Secure Multiplication (SM) and Secure Multiplication for Absolute Value

(SM) operations to generate encrypted versions of A and its associated parameters.

Steps 9-7: After processing all SVM parameter tuples, find the absolute difference between E_1 and E_i for each tuple, then identify the tuple with the maximum absolute difference.

Steps 7-8: Finally, extract the corresponding α_2 from the tuple with the maximum absolute difference and output both α_1 and α_2 along with their associated parameters.

This algorithm ensures the secure selection of α_1 and α_2 while preserving the privacy and integrity of the encrypted data.

5.3 Secure Sequential Minimal Optimization (SSMO)

Performing Secure Sequential Minimal Optimization (SSMO) involves iteratively updating the Lagrange multipliers (a) to minimize the objective function of the Support Vector Machine (SVM) while ensuring that the Karush-Kuhn-Tucker (KKT) conditions are satisfied. Here's how it can be done securely:

Initialization:

Initialize all Lagrange multipliers

(a_i) to zero. Set the tolerance

threshold for convergence (b).

Repeat until convergence:

Select two Lagrange multipliers (a_1 and a_2):

a. To safely choose a1 and a2 from all the encrypted αi values, use the Secure a1 and a2 Selection Protocol (SSel).

b. Compute the SVM decision function output (f(A)) forboth data points :

Compute $f(a_1)$ and $f(a_2)$ using the current Lagrange multipliers and kernel function.

c . Compute the error (E_i) for both data points

d . Update the Lagrange multipliers (A and B):

Use the Secure Sequential Minimal Optimization (SSMO) protocol to update A and B in a privacy-preserving manner.

e. Check convergence:

If the change in A and B is below the tolerance threshold (b) ,stop iterating.

Output:

The final Lagrange multipliers (a_i) obtained after convergence.The Secure Sequential Minimal Optimization (SSMO) protocol is crucial for securely updating the Lagrange multipliers (a_i) in Step 2d while maintaining privacy. It involves cryptographic techniques to perform the necessary computations on encrypted data, ensuring that the data remains confidential throughout the optimization process. The protocol should also include mechanisms to check convergence and stop iterating once the Lagrange multipliers converge to stable values.

5.4 Training Completion and SVM Classification

This mechanism allows trainin of the SVM has completed securely.Here's how it works: **Secure Parameter Selection:** Execute the SSel protocol to select A and B among all th encrypted a_i values. **Compute Verification Value:** Compute $[V]_{SEQ}([IB1],[0])$, where $[V]$ is the verificatio and $[IB1]$ is the identity of the first selected Lagrange multiplier.**Decryption:** Decrypt $[V]$ using AP's ownprivate key *skA*.

Continue or Stop Training: If "CONTINUE" is received, ntinue executing SSMO Lagrange multipliers. If "STOP" is received, use SDT to transform the L range multiplie AP's domain and send the transformed results to AP for decryption. **Secure SVM Classification (Step-0 to Step-3):**

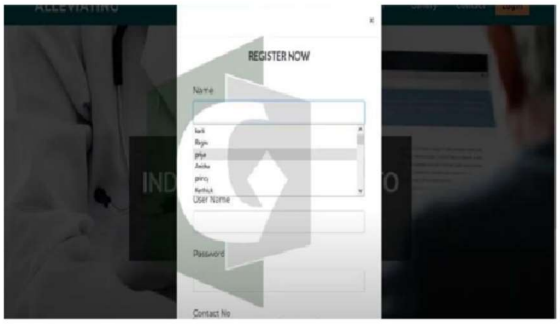
Pre- Computation:

Apply SDT to convert all of the data in S into domain 0. This is a stage that CSP and CP can do offline.

Encryption and Attribute Transformation: B encrypts the attributes and sends them to transforms the encrypted domain of the ciphertexts using SDT if both CERB and CER0 a **Calculate Decision Function:** CP calculates the decision function output $[g1(ab)+]$ and on the final result [b].

Result Transformation: Transform the result [ab] into $[ab]_{pkB}$ using SDT and send it back.Thi mechanism ensures that AP can securely determine whether the training phase of t has completed, allowing for effective coordination between CP andAP. Additionally, it maintains the privacy of the data and operations performed.

6 Screenshot and Result



Registration Page

*

Results page

Status Page

7. CONCLUSION

We presented POD, a novel privacy-preserving cloud-based outsourced drug discovery method, in this work.POD is designed to make it simpler for pharmaceutical businesses to securely outsource SVM training as well as formulation storage to the cloud. The trained SVM model could be used to classify authorized clients in a way that protects their privacy. More specifically, we developed many essential secure computing building pieces and a secure domain transformation protocol to enable the safe outsourcing of computation to third parties.

We additionally created two crucial safe components, secure sequential minimum optimization and secure parameter selection, to enable privacy-preserving SVM training in drug discovery. We will be improving our method in the future to allow more sophisticated data mining techniques in order to manage extremely large datasets in drug development.The authors thank the associate editor and reviewers for their thoughtful and kind feedback.

remarks. This work is partially funded by the AXA Research Fund.SVM training and chemical compound categorization are effectively accomplished by the POD framework without compromising user privacy or leaking information to unapproved parties.It supports safe medication discovery by protecting private data. Adaptive Boosting, Decision Tree, Gradient Boosting, Naïve Bayes, Bagging, and both linear and non-linear Support Vector Machine (SVM) were among these methods.[4]It takes ten to fifteen years to research and develop new drugs, and each one costs an average of more than $1 billion USD. In an attempt to reduce the high rates of attrition throughout the process, the application of machine learning techniques to various stages of drug discovery and development, especially at the earliest stage - identification of druggable disease genes - has attracted increasing attention in the last ten years. In this work, we have developed a novel tensor factorization approach to predict potential pharmacological targets (genes or proteins) for disease treatment.

REFERENCES

[1] P. Dhanush and N. Nalini, "Drug Review System Using Machine Learning by Comparing Linear Support Vector Machine with Naïve Bayes Classifier to Measure Accuracy," *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, 2022, pp. 1-5

[2] X. Liu, R. H. Deng, K. -K. R. Choo and Y. Yang, "Privacy-Preserving Outsourced Support Vector Machine Design for Secure Drug Discovery," in *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 610-622, 1 April-June 2020, . keywords:

{Drugs;Support vector machines;Cloud.

[3] D. Rathod, K. Patel, A. J. Goswami, S. Degadwala and D. Vyas, "Exploring Drug Sentiment Analysis with Machine Learning Techniques," *2023 International Conference on Inventive Computation Technologies (ICICT)*, Lalitpur, Nepal, 2023, pp. 9-12.

[4] J. Wang, L. Wu, H. Wang, K. -K. R. Choo and D. He, "An Efficient and Privacy- Preserving Outsourced Support Vector Machine Training for Internet of Medical Things," in *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 458-473, 1 Jan. 1, 2021.

[5] P. Das and D. H. Mazumder, "Predicting Drug Functions from Gene Ontology, Amino Acid Sequences, and Drug-Disease Associations through Multi-label Machine Learning with MLSMOTE," *2023 IEEE Silchar Subsection Conference (SILCON)*, Silchar, India, 2023, pp. 1-7.

[6] I. Anand, M. M, V. V S, A. Kodipalli, T. Rao and R. B R, "Drug Classification Analysis Using Different Machine Learning Algorithms," *2023 International Conference on Computational Intelligence for Information, Security and Communication Applications (CIISCA)*, Bengaluru, India, 2023, pp. 355-360,

[7] C. Ye, R. Swiers, S. Bonner and I. Barrett, "A Knowledge Graph-Enhanced Tensor Factorisation Model for Discovering Drug Targets," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 6, pp. 3070-3080, 1 Nov.-Dec.

2022,

[8] R. Biswas, A. Basu, A. Nandy, A. Deb, K. Haque and D. Chanda, "Drug Discovery and Drug Identification using AI," *2020 Indo – Taiwan 2nd*

International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), Rajpura, India, 2020, pp. 49-51, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181309.

[9] S. Momtahen, F. Al-Obaidy and F. Mohammadi, "Machine Learning with Digital Microfluidics for Drug Discovery and Development," *2019 IEEECanadian Conference of Electrical and Computer Engineering (CCECE)*, Edmonton, AB, Canada, 2019, pp. 1-6,

[10] S. Liu *et al.*, "Enhancing Drug-Drug Interaction Prediction Using Deep Attention Neural Networks," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 2, pp. 976-985, 1 March-April 2023.

RE-2022-220344-plag-report

ORIGINALITY REPORT

8%	6%	7%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ink.library.smu.edu.sg Internet Source	3%
2	Ximeng Liu, Robert Deng, Kim-Kwang Raymond Choo, Yang Yang. "Privacy-Preserving Outsourced Support Vector Machine Design for Secure Drug Discovery", IEEE Transactions on Cloud Computing, 2018 Publication	2%
3	fatcat.wiki Internet Source	1%
4	"Neural Information Processing", Springer Science and Business Media LLC, 2017 Publication	<1%
5	finance.minyanville.com Internet Source	<1%
6	ijisrt.com Internet Source	<1%
7	www.researchgate.net Internet Source	<1%

8	Pranab Das, Dilwar Hussain Mazumder. "Predicting Drug Functions from Gene Ontology, Amino Acid Sequences, and Drug-Disease Associations through Multi-label Machine Learning with MLSMOTE", 2023 IEEE Silchar Subsection Conference (SILCON), 2023 Publication	<1%
9	mail.zkoop.com Internet Source	<1%
10	Jun Zhang, Zoe L.Jiang, Ping Li, Siu Ming Yiu. "Privacy-preserving multikey computing framework for encrypted data in the cloud", Information Sciences, 2021 Publication	<1%
11	Yaman Akbulut, Abdulkadir Şengür, Yanhui Guo, Florentin Smarandache. "A Novel Neutrosophic Weighted Extreme Learning Machine for Imbalanced Data Set", Symmetry, 2017 Publication	<1%
12	research.tees.ac.uk Internet Source	<1%
13	engineeringjournals.stmjournals.in Internet Source	<1%
14	Cheng Ye, Rowan Swiers, Stephen Bonner, Ian Barrett. "A Knowledge Graph-Enhanced Tensor Factorisation Model for Discovering	<1%

Drug Targets", IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2022
Publication

Exclude quotes On Exclude bibliography On

REFERENCES

[1] P. Dhanush and N. Nalini, "Drug Review System Using Machine Learning by Comparing Linear Support Vector Machine with Naïve Bayes Classifier to Measure Accuracy," *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, Chennai, India, 2022, pp. 1-5

[2] X. Liu, R. H. Deng, K. -K. R. Choo and Y. Yang, "Privacy-Preserving Outsourced Support Vector Machine Design for Secure Drug Discovery," in *IEEE Transactions on Cloud Computing*, vol. 8, no. 2, pp. 610-622, 1 April-June 2020, . keywords:

{Drugs;Support vector machines;Cloud.

[3] D. Rathod, K. Patel, A. J. Goswami, S. Degadwala and D. Vyas, "Exploring Drug Sentiment Analysis with Machine Learning Techniques," *2023 International Conference on Inventive Computation Technologies (ICICT)*, Lalitpur, Nepal, 2023, pp. 9-12.

[4] J. Wang, L. Wu, H. Wang, K. -K. R. Choo and D. He, "An Efficient and Privacy-Preserving Outsourced Support Vector Machine Training for Internet of Medical Things," in *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 458-473, 1 Jan.1, 2021.

[5] P. Das and D. H. Mazumder, "Predicting Drug Functions from Gene Ontology, Amino Acid Sequences, and Drug-Disease Associations through Multi-label Machine Learning with MLSMOTE," *2023 IEEE Silchar Subsection Conference (SILCON)*, Silchar, India, 2023, pp. 1-7.

[6] I. Anand, M. M, V. V S, A. Kodipalli, T. Rao and R. B R, "Drug Classification Analysis Using Different MachineLearning Algorithms," *2023 International Conference on Computational Intelligence for Information, Security and Communication Applications (CIISCA)*, Bengaluru, India, 2023, pp. 355-360,

[7] C. Ye, R. Swiers, S. Bonner and I. Barrett, "A Knowledge Graph-Enhanced Tensor Factorisation Model for Discovering Drug Targets," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 6, pp. 3070-3080, 1 Nov.-Dec. 2022,

[8] R. Biswas, A. Basu, A. Nandy, A. Deb, K. Haque and D. Chanda, "Drug Discovery and Drug Identification using AI," *2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*, Rajpura, India, 2020, pp. 49-51, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181309.

[9] S. Momtahen, F. Al-Obaidy and F. Mohammadi, "Machine Learning with Digital Microfluidics for Drug Discovery and Development," *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, Edmonton, AB, Canada, 2019, pp. 1-6,

[10] S. Liu *et al*., "Enhancing Drug-Drug Interaction Prediction Using Deep Attention Neural Networks," in *IEEE/ACM Transactions on Computational Biology and*

Bioinformatics, vol. 20, no. 2, pp. 976-985, 1 March-April 2023