PREDUCTION DRUG-DRUG INTERACTION BASED ON INTEGRATED SIMILARITY AND SEMI-SUPERVISED LEARNING

The Project Report is submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications

Submitted by

KARUMURI JAYASRI

2185351061

Under the Esteemed Guidance of

Mr. G. Ramesh Kumar

Assistant Professor

B. V. Raju College

Vishnupur: Bhimavaram

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ADIKAVI NANNAYA UNIVERSITY

RAJAHMAHENDRAVRAM

2022-2023

B. V. RAJU COLLEGE

Accredited by NAAC B^{++} Grade

(Affiliated to Adikavi Nannaya University)

DEPARTMENT OF MCA

CERTIFICATE

This is to certify that this project entitled **"PREDUCTION DRUG-DRUG INTERACTION BASED ON INTEGRATED SIMILARITY AND SEMI-SUPERVISED LEARNING"** in partial fulfillment of the degree of MASTER OF COMPUTER APPLICATION to Adikavi Nannaya University through from TRAIN-A-TECH PVT.LTD, Hyderabad through B. V. Raju College, done by **Ms. K. Jaya Sri.** RegdNo:**2185351061** is an authentic work carried out by his during the Academic Year 2022-2023 under my guidance. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Internal Guide **Head of the Department**

External Examiner **Principal**

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I have now the opportunity to express my gratitude for all of them.

The first person I would like to thank **Dr. I.R. Krishnam Raju, Professor & Principal, Department of MCA, B V Raju College, Bhimavaram.** His wide knowledge and logical way of thinking have made a deep impression on me. His understanding, encouragement and personal guidance have provided the basis for this thesis. He is a source of inspiration for innovative ideas and his kind support is well known to all his students and colleagues.

I wish to thank **Dr. V. Bhaskara Murthy, Professor & Head of the department, Department of MCA, B. V. Raju College, Bhimavaram** for his support and valuable suggestions for the successful completion of this project.

I wish to thank my guide **Sri G. Ramesh Kumar, Assistant Professor, Department of MCA** for his support and valuable suggestions for the successful completion of this Project.

I would like to thanks to the faculty & staff members of MCA Department for successful completion of project.

K. Jayasri

2185351061

DECLARATION

This is to certify that the project report entitled "**PREDUCTION DRUG-DRUG INTERACTION BASED ON INTEGRATED SIMILARITY AND SEMI-SUPERVISED LEARNING**" is done by me is an authentic work carried out for the partial fulfillment of the requirements for the award of the degree of Master of Computer Applications under the guidance of **Mr. G. Ramesh Kumar**. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

Signature of the Student
K. Jayasri
2185351061
B. V. Raju college

INDEX

List of Figures

LIST OF SCREENS

INTRODUCTION

1.INTRODUCTION

The pharmacological effect of a drug is influenced by another drug, which usually appears when two or more drugs are administered simultaneously for a patient. These associations are also defined as drug-drug interactions (DDIs), and are either favorable efficacy or undesirable DDIs according to clinical results. Positive DDIs can provide more effective treatments and reduce the suffering of patients. However, undesirable DDIs are the major cause of adverse reaction events [1]. In serious cases, they can result in the drug withdrawal from the drug market and the death of a patient who is treated with multi-drugs [2], [3]. Currently, multi-drug therapies have been widely used in treating multiple illnesses or complex diseases, such as cancer [4], [5], [6]. The original purpose of multi-drugs treatment is to alleviate the patient suffering, improve the treatment effect and increase the overall survival rate [7]. However, undesirable DDIs have also been developed along with more and more drugs used in the synergistic treatment, and which also influence the treatment effect and even lead to serious complications as well as the financial burden. Therefore, in order to reduce the cost of drug development and improve the treatment effect, it is very urgent to identify DDIs in the drug development process.

Recently, many studies have proven that some commonly used drugs have high possibility to interact with each other, such as lipid lowering drugs, macrolides, oral antifungal agents, which are widely used to synergistic treatments [8], [9], [10]. Previous studies about DDIs can be divided into three categories: pharmaceutic, pharmacokinetic (PK) and pharmacodynamic (PD) [11], [12]. The pharmaceutic DDIs usually result from multi-drugs with the chemical incompatibility. A PK interaction is defined as the effects of a drug in the absorbed, distributed, or metabolized process of another drug in the patient body, which is usually related to adverse responses [4]. PD interactions often result from different drugs acting on the same receptor, site, or physiological system, and could have also either synergistic or harmful effects for patients [13]. Many PK and PD interactions have been used for inferring DDIs in previous studies [14], [15].

In silico, in vitro and in vivo experiments are the methods to discover DDIs among drugs, and the two latter methods are usually very time-consuming and labor-intensive cycles [16]. In addition, the side effects caused by DDIs are hard to be measured in vitro or in vivo experiments, which makes results that these methods hard to be executed [16]. As more and

more patients are simultaneously treated by multi-drugs, identifying DDIs has become an important issue of bioinformatics research and a very urgent need to drug developments. Moreover, compared to traditional biomedical experiment methods, the computational methods provide an opportunity to predict new DDIs with the low cost and high accuracy. Therefore, by considering its advancement to biological experiments, there exists a high demand for predicting DDIs via computational approaches [17]. In addition, the development of medical technologies and applications of multi-drug treatments also further imposes a very urgent demand to develop computational methods to predict potential DDIs.

Recently, based on machine learning models, many computational approaches have been developed to predict potential DDIs. Tatonetti *et al*. developed a signal discovery method to infer DDIs [18], main features of drugs used in this method are drug adverse event profiles. By combining drug chemical similarities, side effect similarities, protein protein interaction similarities and target sequence similarities, an INDI (Inferring Drug Interactions) framework was developed to predict DDIs, which used two types of drug interactions(potential CYP (Cytochrome P450)-related DDIs, and non-CYP-related DDIs (NCRDs)) [19]. By the combination of crizotinib with ketoconazole or rifampin, a PBPK (physiologically based pharmacokinetic) model was developed for predicting DDIs [20]. Based on properties of the drug metabolism, the text-mining and reasoning
approaches were also used to discover novel DDIs [21]. Vilar *et al*. computed the molecular fingerprint similarity and the molecular structure similarity of drugs to predict DDIs [22]. With 2D and 3D molecular structures, interaction profiles, target and side-effect similarities, Vilar *et al*. further developed a protocol applicable on a large scale data to infer novel DDIs [23]. Based on drug phenotypic, therapeutic, chemical, and genomic properties and machine learning
model, Cheng *et al*. proposed a computational method to predict DDIs [24]. Based on the drug molecular similarity and phenotypic similarity, Li *et al*. developed a computational method to discover the combination efficacy of drugs with a Bayesian network model [25]. Based on a random forest model, Liu *et al*. proposed a computational method to predict DDIs by integrating chemical interactions, protein protein interactions between targets of drugs and target enrichment of KEGG pathways [26]. This method adopted a feature selection technique to obtain the important features of drugs. Luo *et al*. developed a computational method to predict DDIs by implementing the chemical-protein interactome, which provided as a web server (called DDICPI) [27]. Based on the framework probabilistic soft logic, Sridhar *et al*. took a PSL (Probabilistic Soft Logic) method to predict novel DDIs by integrating networks of multiple drug similarities and known DDIs [13]. With 2D structural similarities of drugs, Takako *et al*. developed a logistic regression model to infer potential DDIs [28]. Its prediction

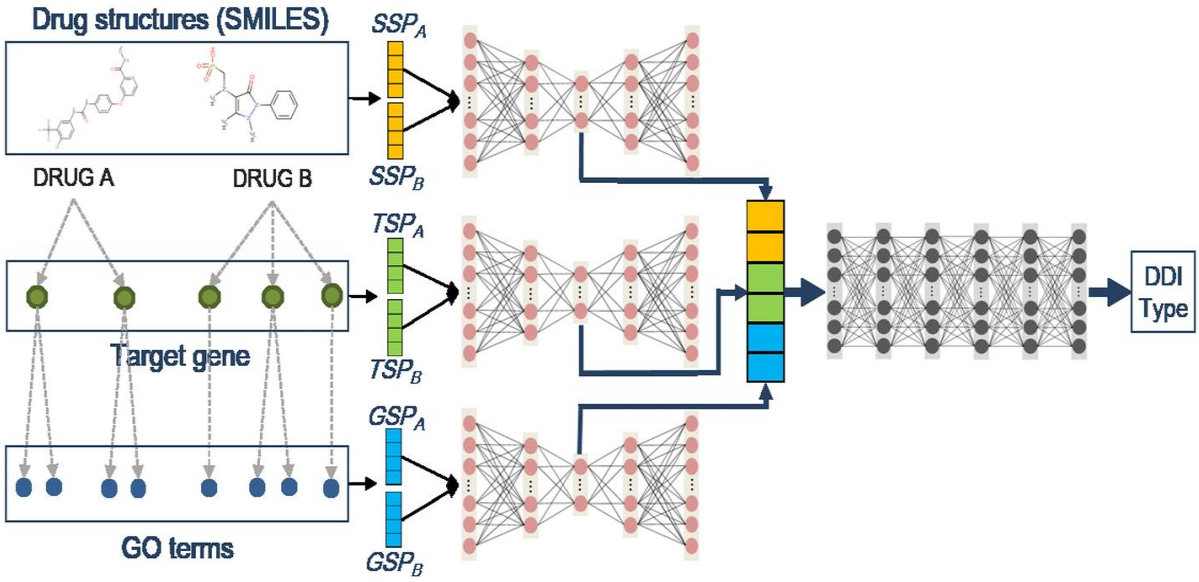
performance is further improved by combining target related and enzyme-related scores. Based on inner product based similarity measures (IPSMs), Ferdousi *et al*. provided an computational method to predict DDIs. This method also used the drug similarity constructed with key biological elements including carriers, transporters, enzymes
and targets of drugs. In addition, based on the assumption that synergistic effects with drugs are often similar and vice versa, NLLSS (Network-based Laplacian regularized Least Square Synergistic drug combination prediction) was
proposed to predict hidden synergistic drug combinations, but it can not predict DDIs for new drugs [29].

In addition, based on the network-based prediction models, many prediction methods have been developed for the DDI prediction and other closely related issues, such as the drug-disease prediction and the drug-target interaction prediction [30], [31]. The network-based DDI prediction methods can be divided into two types: one is constructing a similarity network of known DDIs to predict DDIs while another is performing the novel DDI prediction based on the structure of the DDI network. For example, the MEF (multiple evidence fusion) method was developed for predicting adverse drug reactions (ADRs), which is based on the network structural data of similarities of drugs and known drug-ADR interactions [32]. Cao *et al*. also provided a similarity network-based method to predict hidden DDIs by computed relational features of drugs [24]. The label propagation method is a typical network based method, which is also used to infer new DDIs by integrating drug chemical structures, drug side effects and off side effects [33]. Based on the drug-target interaction and protein-protein interaction networks, potential PK DDIs were predicted by analyzing the significant relation between ”S-score” and likelihood that DDIs occurs [34]. By computing the drug similarity via integrating the drug target interactions and protein-protein interactions, Park *et al*. adopted a random walk with restart model to calculate the DDI scores which was the likelihood of the occurrence of PD DDIs [35]. The DDI type is also an important issue of the DDI prediction, Jin *et al*. provided a computational method to infer the DDI type by applying a multi-task dyadic regression model [36]. Based on the Jaccard similarity index and interaction profile fingerprints (IPFs), Vilar *et al*. developed a computational method to predict new DDIs [37]. Furthermore, by integrating the chemical, biological, phenotype and known DDI network information, Zhang *et al*. developed an effective approach to predict potential DDIs, it explored three ensemble methods which include the weight average ensemble method, L1 ensemble (L1E) classifier and L2 ensemble (L2E) classifier.

However, although those above computational methods have achieved some effective results for predicting novel DDIs, some limits still should be addressed. For example,

some methods have not effectively integrated the known DDI network information and features of drugs by a reasonable model. Although there exist a large number of new drugs, these current computational methods have not paid enough attention to discover potential DDIs for them. With
the development of multi-drug treatments, the possibility of adverse effect occurrences is higher than before. Therefore, in order to improve the effect of multi-drug treatments and drug developments, it is very urgent to develop more effective computational methods to predict potential DDIs.

In this study, by integrating the chemical, biological and phenotype information of drugs, we develop a computational method (called DDI-IS-SL) to predict DDIs. These drug information includes drug chemical structures, drugtarget
interactions, drug enzymes, drug transports, drug pathways, drug indications, drug side effects, drug off side effects and known DDIs. First, based on these pieces of
drug information, we construct a high-dimensional binary vector to calculate the feature similarity of drugs via the cosine similarity method. Furthermore, we also compute the Gaussian Interaction Profile (GIP) kernel similarity [38] ofdrugs based on known DDIs. The final drug similarity is constructed by their feature similarity and GIP similarity. Then a Regularized Least Squares (RLS) classifier [39] is adapted to predict DDIs. For new drugs which do not have any interactions with other drugs, we also calculate their relational initial scores via performing the node-based drug network diffusion method. Therefore, our method can predict potential DDIs not only for known drugs but also for new drugs. The prediction performance of our method and other competing methods are systematically assessed by the 5-fold cross validation, the 10-fold cross validation and the de novo validation. The AUC (area under the ROC curve) is used as the metric to evaluate the performance of computational methods. In terms of AUC, our method is superior to other competing methods. Specifically, in the 5-fold cross validation, the AUC value of our method is 0.9691, which is larger than the AUC of 0.9570 from the state-of-the-art L1E. Furthermore, in the 10-fold cross validation, the AUC value of our method reaches 0.9745, which is also larger than the best result of L1E whose AUC value is 0.9599. Our method also obtain the best prediction performance in the de novo drug validation, its AUC value is 0.9292, which is also larger than the the best result of other methods (WAE (weighted average ensemble method) : 0.9073). In addition, the comparison of the average running time further improves that our method has the higher running efficiency than other competing methods. Finally, the verification results of case studies also prove the prediction ability of our method in practical applications and show that DDI-IS-SL is an effective computational method to predict new DDIs.


Drug structures (SMILES)
DRUG A
DRUG B
Target gene
GO terms
SSP_A
SSP_B
TSP_A
TSP_B
GSP_A
GSP_B
DDI Type

LITERATURE SURVEY

2. LITERATURE SURVEY

A drug-drug interaction (DDI) is defined as an association between two drugs where the pharmacological effects of a drug are influenced by another drug. Positive DDIs can usually improve the therapeutic effects of patients, but negative DDIs cause the major cause of adverse drug reactions and even result in the drug withdrawal from the market and the patient death. Therefore, identifying DDIs has become a key component of the drug development and disease treatment. In this study, we propose a novel method to predict DDIs based on the integrated similarity and semi-supervised learning (DDI-IS-SL). DDI-IS-SL integrates the drug chemical, biological and phenotype data to calculate the feature similarity of drugs with the cosine similarity method. The Gaussian Interaction Profile kernel similarity of drugs is also calculated based on known DDIs. A semi-supervised learning method (the Regularized Least Squares classifier) is used to calculate the interaction possibility scores of drug-drug pairs. In terms of the 5-fold cross validation, 10-fold cross validation and de novo drug validation, DDI-IS-SL can achieve the better prediction performance than other comparative methods. In addition, the average computation time of DDI-IS-SL is shorter than that of other comparative methods. Finally, case studies further demonstrate the performance of DDI-IS-SL in practical applications.

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Recently, based on machine learning models, many computational approaches have been developed to predict potential DDIs. Tatonetti *et al*. developed a signal discovery method to infer DDIs [18], main features of drugs used in this method are drug adverse event profiles. By combining drug chemical similarities, side effect similarities, proteinprotein interaction similarities and target sequence similarities, an INDI (INferring Drug Interactions) framework was developed to predict DDIs, which used two types of drug interactions(potential CYP (Cytochrome P450)-related
DDIs, and non-CYP-related DDIs (NCRDs)) [19].

With 2D and 3D molecular structures, interaction profiles, target and side-effect similarities, Vilar *et al*. further developed a protocol applicable on a large scale data to infer novel DDIs [23]. Based on drug phenotypic, therapeutic, chemical, and genomic properties and machine learning model, Cheng *et al*. proposed a computational method to predict DDIs [24]. Based on the drug molecular similarity
and phenotypic similarity, Li *et al*. developed a computational method to discover the combination efficacy of drugs with a Bayesian network model [25]. Based on a random forest model, Liu *et al*. proposed a computational method to predict DDIs by integrating chemical interactions, proteinprotein interactions between targets of drugs and target enrichment of KEGG pathways [26]. This method adopted a feature selection technique to obtain the important features of drugs.

Luo *et al*. developed a computational method to predict DDIs by implementing the chemical-protein interactome, which provided as a web server (called DDICPI) [27]. Based on the framework probabilistic soft logic, Sridhar *et al*. took a PSL (Probabilistic Soft Logic) method to predict novel DDIs by integrating networks of multiple drug similarities and known DDs [13]. With 2D structural similarities of drugs, Takako *et al*. developed a logistic regression model to infer potential DDIs [28]. Its prediction performance is further improved by combining.

Based on inner productbased similarity measures (IPSMs), Ferdousi *et al*. provided an computational method to predict DDIs. This method also used the drug similarity constructed with key biological elements including carriers, transporters, enzymes and targets of drugs. In addition, based on the assumption that synergistic effects with drugs are often similar and vice versa, NLLSS (Network-based Laplacian regularized Least Square

Synergistic drug combination prediction) was proposed to predict hidden synergistic drug combinations, but it can not predict DDIs for new drugs [29].

3.2 PROPOSED SYSTEM

In this study, by integrating the chemical, biological and phenotype information of drugs, we develop a computational method (called DDI-IS-SL) to predict DDIs. These drug information includes drug chemical structures, drugtarget interactions, drug enzymes, drug transports, drug pathways, drug indications, drug side effects, drug off side effects and known DDIs.

First, based on these pieces of drug information, we construct a high-dimensional binary vector to calculate the feature similarity of drugs via the cosine similarity method. Furthermore, we also compute the Gaussian Interaction Profile (GIP) kernel similarity [38] of drugs based on known DDIs.

The final drug similarity is constructed by their feature similarity and GIP similarity.

Then a Regularized Least Squares (RLS) classifier [39] is adapted to predict DDIs. For new drugs which do not have any interactions with other drugs, we also calculate their relational initial scores via performing the node-based drug network diffusion method. Therefore, our method can predict potential DDIs not only for known drugs but also for new drugs. The prediction performance of our method and other competing methods are systematically assessed by the 5-fold cross validation, the 10-fold cross validation and the de novo validation. The AUC (area under the ROC curve) is used as the metric to evaluate the performance of computational methods. In terms of AUC, our method is superior to other competing methods.

Specifically, in the 5-fold cross validation, the AUC value of our method is 0.9691, which is larger than the AUC of 0.9570 from the state-of-the-art L1E. Furthermore, in the 10-fold cross validation, the AUC value of our method raches 0.9745, which is also larger than the best result of L1E whose AUC value is 0.9599. Our method also obtain the best prediction performance in the de novo drug validation, its AUC value is 0.9292, which is also larger than the the best result of other methods (WAE (weighted average ensemble method) : 0.9073). In addition, the comparison of the average running time further improves that our method has the higher running efficiency than other competing methods. Finally, the verification results of case studies also prove the prediction ability of our method in practical applications and show that DDI-IS-SL is an effective computational method to predict

new DDIs.

3.3 FEASIBILITY STUDY

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

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

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

TECHNICAL FEASIBILITY

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

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some System.

3.4 REQUIREMENT ANALYSIS

HARDWARE REQUIREMENTS:

Processor	-	Pentium –IV
RAM	-	4 GB (min)
Hard Disk	-	20 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

SOFTWARE REQUIREMENTS:

Operating System	-	Windows 11.
Coding Language	-	Python.
Front-End	-	Python.
Back-End	-	Django-ORM
Designing	-	Html, css, javascript.
Data Base	-	MySQL (XAMP Server).

FUNCTIONAL REQUIREMENT:

SERVICE PROVIDER

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Login, Train and Test Drugs Data Sets, View Drugs Trained and Tested Accuracy in Bar Chart, View Drugs Trained and Tested Accuracy Results,

View Drug to Drug Interaction Predicted Details, Find Drug to Drug Interaction Predicted Ratio, Download Drug to Drug Interaction, View Drug to Drug Interaction Predicted Ratio Results, View All Remote Users.

VIEW AND AUTHORIZE USERS

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

REMOTE USER

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, PREDICT DRUG TO DRUG INTERACTION TYPE, VIEW YOUR PROFILE.

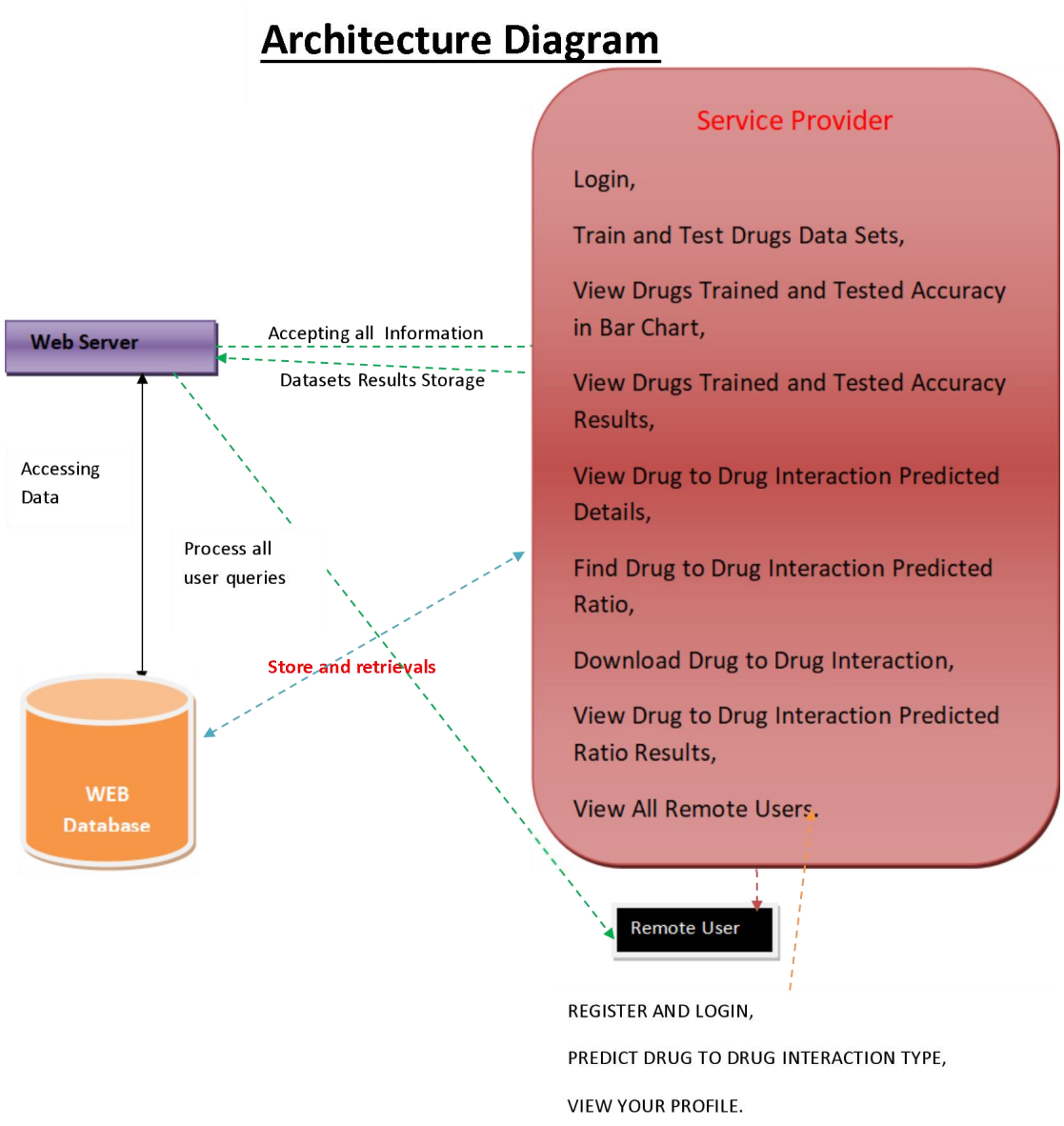
NON-FUNCTIONAL REQUIREMENTS

1.Accuracy: The algorithm should be able to predict crop yield accurately, with a minimal error rate. The accuracy of the model should be evaluated using appropriate metrics.
2.Scalability: The algorithm should be able to handle large datasets and scale up or down as needed. It should be able to process data from multiple sources and handle different types of data.
3.Performance: The algorithm should be efficient and fast, providing real-time or near real time predictions. The response time should be reasonable, and the system should be able to handle multiple requests simultaneously.
4.Security: The system should be secure, protecting sensitive data from unauthorized access, manipulation, or theft. Appropriate security measures should be implemented to ensure data privacy and confidentiality.
5.Reliability: The algorithm should be reliable, producing consistent results over time. The system should be able to handle errors and exceptions gracefully and recover from failures quickly.
6.Maintainability: The system should be easy to maintain, with clear documentation, well-structured code, and modular design. The system should be easy to update and upgrade, with minimal disruption to the users.
7.Usability: The system should be user-friendly, with a simple and intuitive interface. The system should be easy to use, even for non-technical users, and provide clear and concise feedback.

SYSTEM DESIGN

4. SYSTEM DESIGN

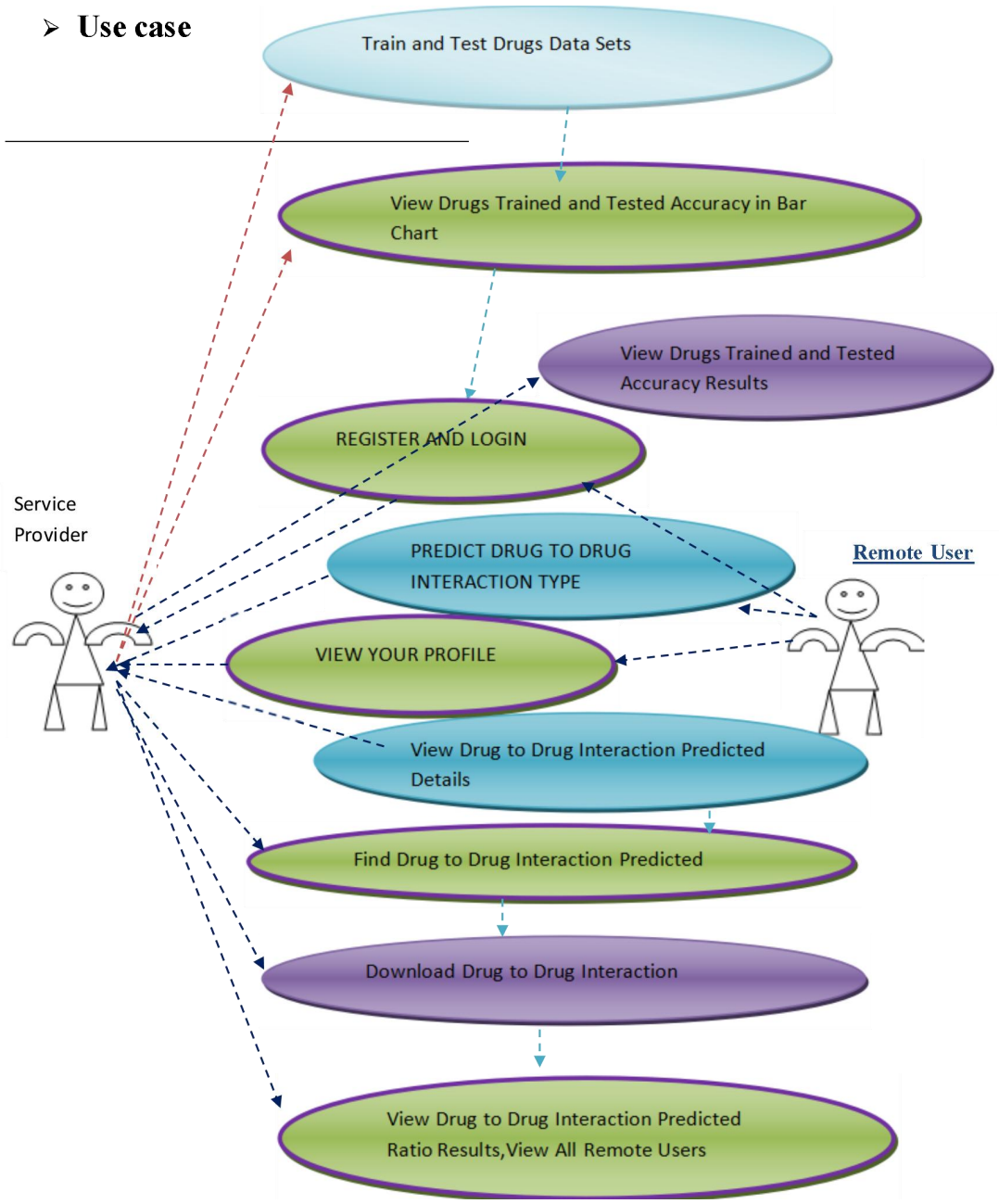
4.1 SYSTEM ARCHITECTURE



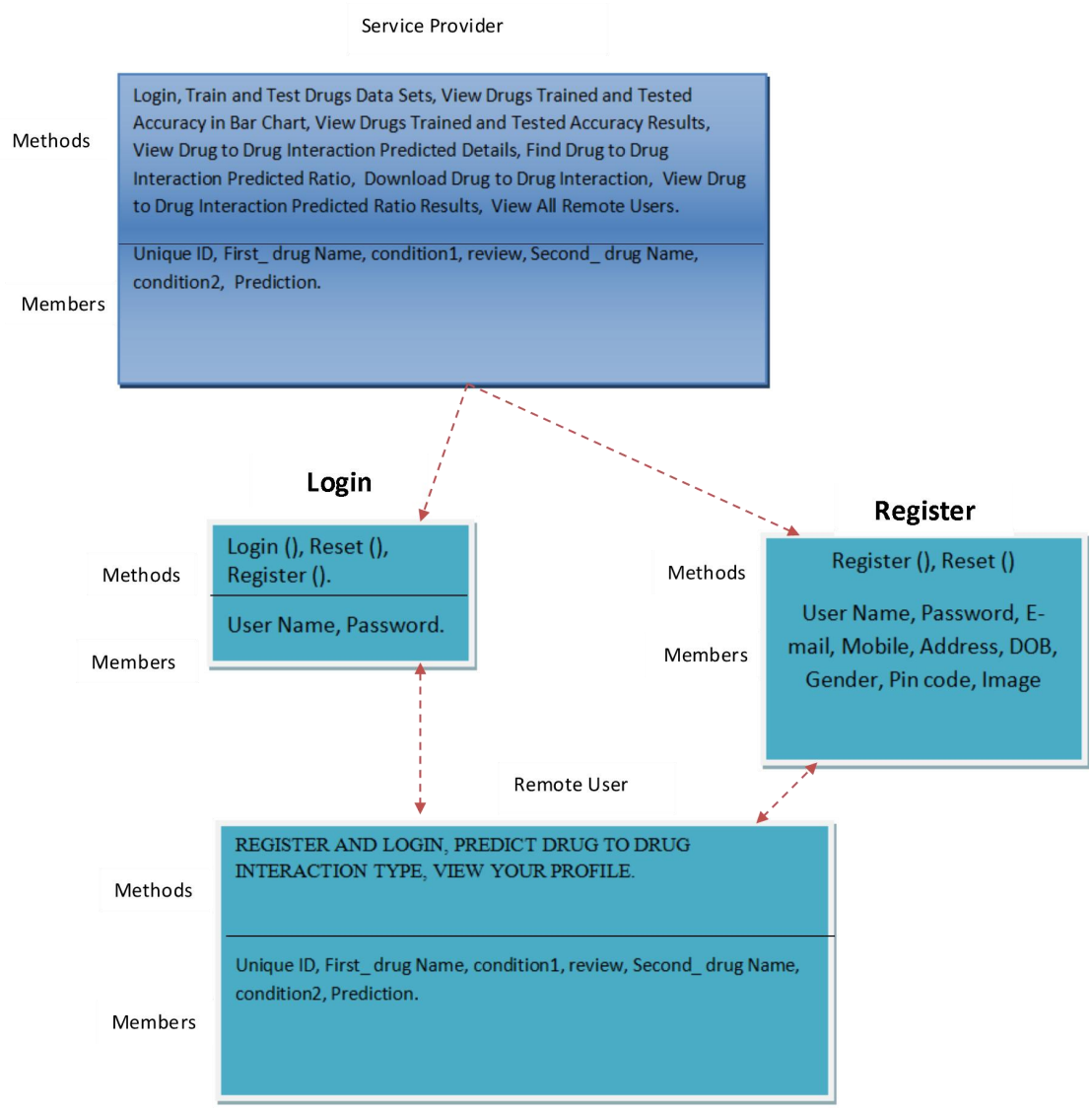
4.2 UML DIAGRAMS

4.2.1 USE CASE DIAGRAM

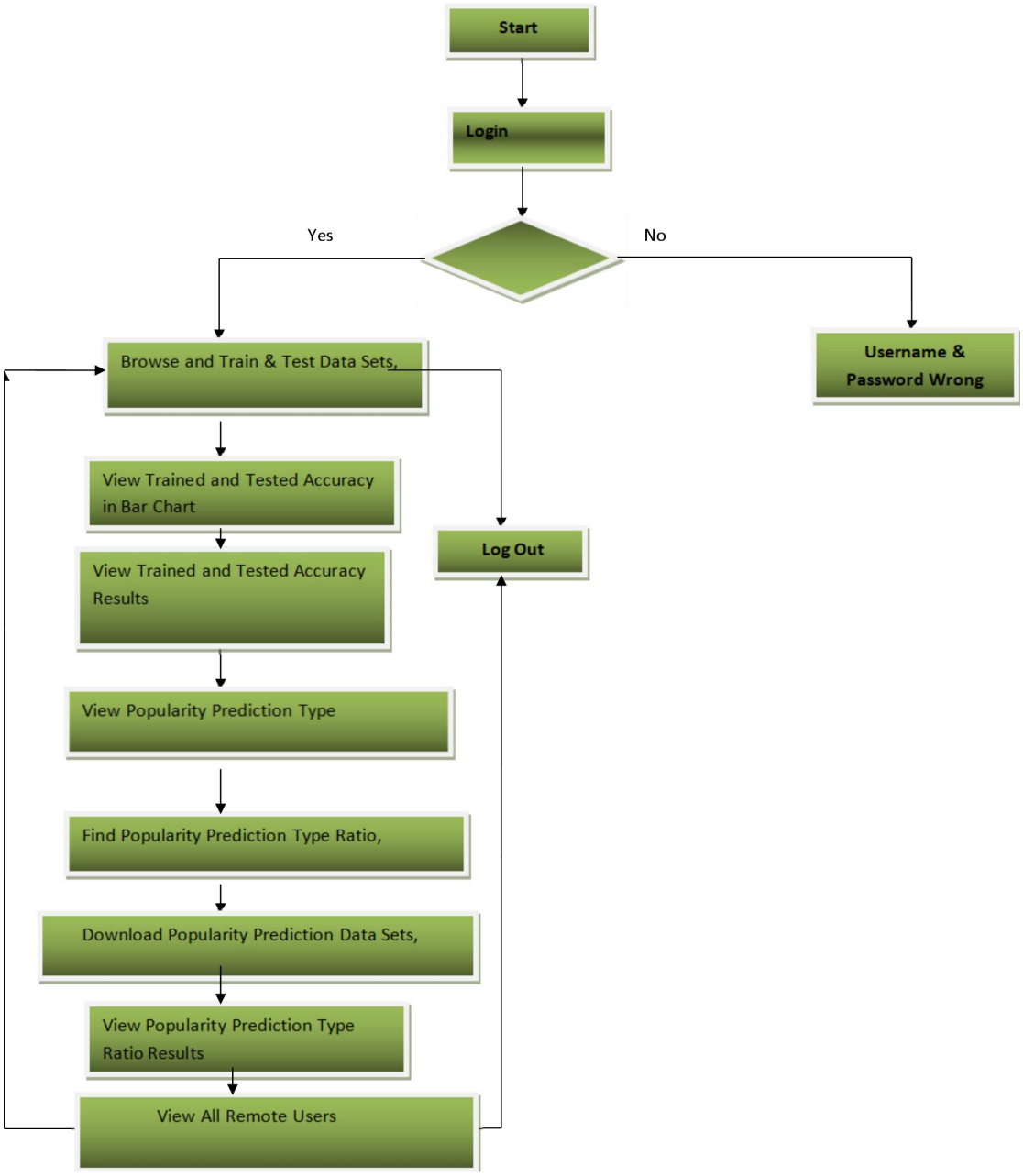
➢ **Use case**



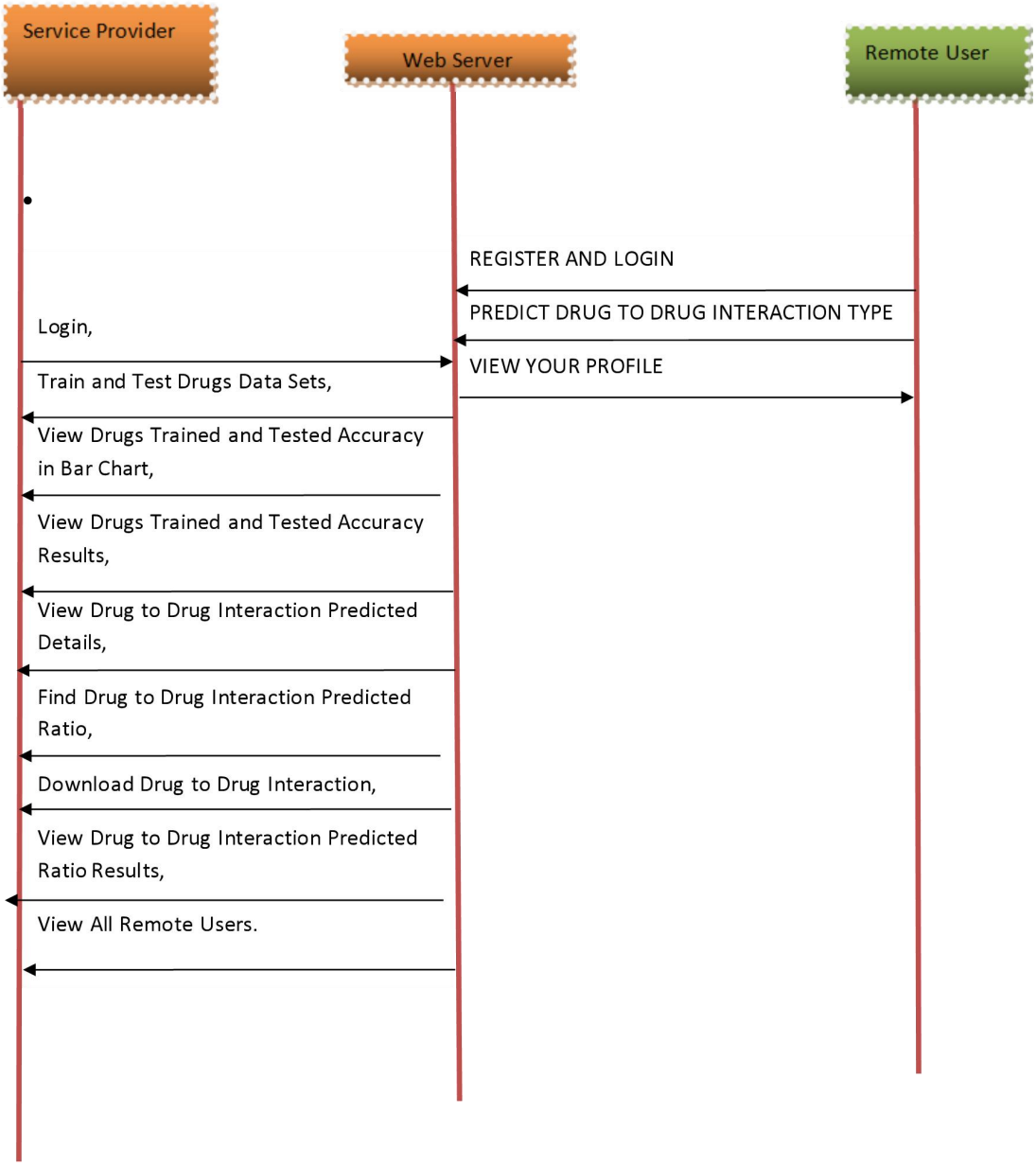
4.2.2 CLASS DIAGRAM



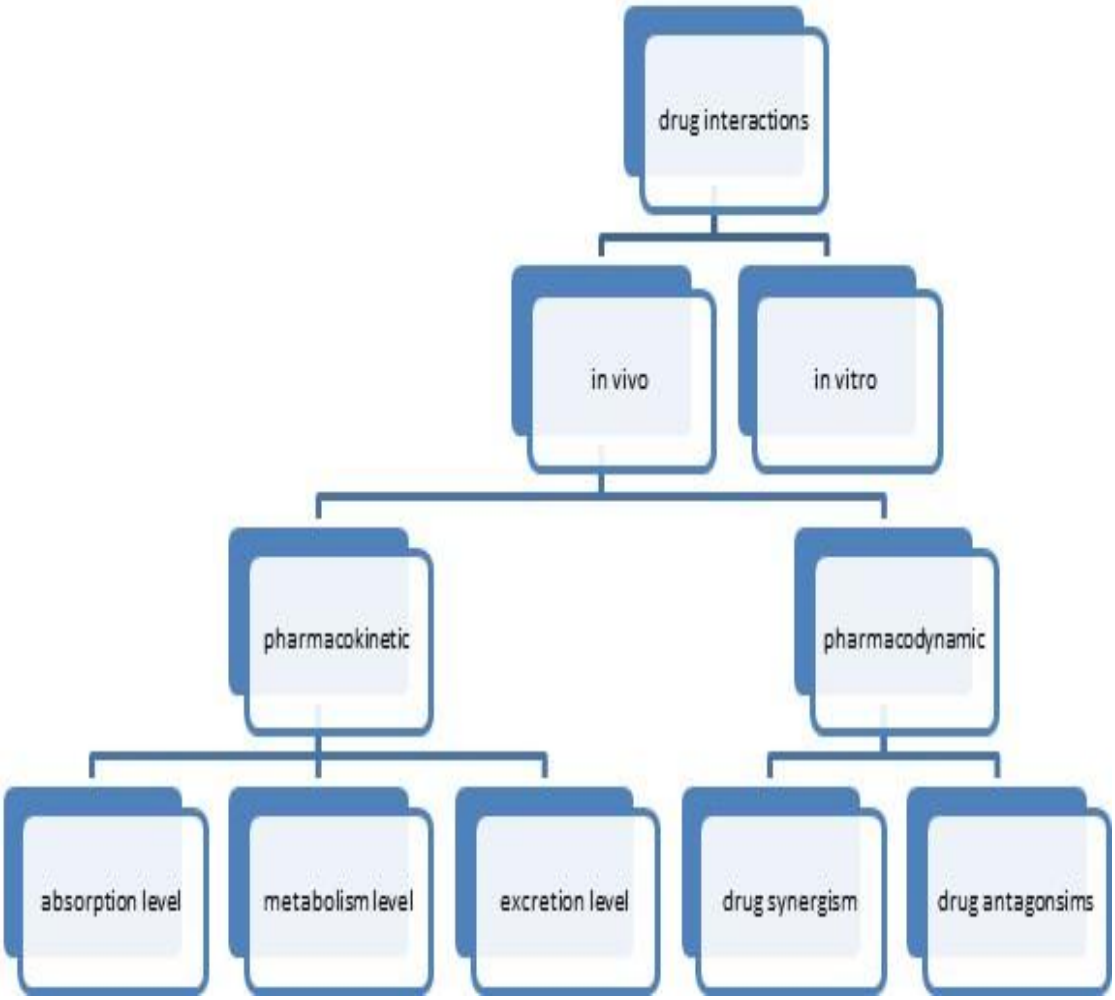
4.2.3 ACTIVITY DIAGRAM



4.2.4 SEQUENCE DIAGRAMS



4.2.5 COMPONENT DIAGRAM



4.3 INPUT AND OUTPUT DESIGN

INPUT DESIGN

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

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

OBJECTIVES

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

OUTPUT DESIGN

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

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

2.Select methods for presenting information.

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

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

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

SYSTEM IMPLEMENTATION

5. SYSTEM IMPLEMENTATION

5.1 Source Code

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys
def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'predicting_drug_drug_interactions.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
from django.db.models import  Count, Avg
from django.shortcuts import render, redirect
from django.db.models import Count
from django.db.models import Q
import datetime
import xlwt
from django.http import HttpResponse
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
from sklearn.ensemble import VotingClassifier

import warnings
warnings.filterwarnings("ignore")
plt.style.use('ggplot')
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score




# Create your views here.
from Remote_User.models import
ClientRegister_Model,drug_drug_interactions,detection_ratio,detection_accuracy



def serviceproviderlogin(request):
    if request.method  == "POST":
        admin = request.POST.get('username')
        password = request.POST.get('password')
        if admin == "Admin" and password =="Admin":
            detection_accuracy.objects.all().delete()
            return redirect('View_Remote_Users')

    return render(request,'SProvider/serviceproviderlogin.html')

def Find_Drug_To_Drug_Interact_Type_Ratio(request):
    detection_ratio.objects.all().delete()
```

```
ratio = ""
kword = 'Bad'
print(kword)
obj = drug_drug_interactions.objects.all().filter(Q(Prediction=kword))
obj1 = drug_drug_interactions.objects.all()
count = obj.count();
count1 = obj1.count();
ratio = (count / count1) * 100
if ratio != 0:
    detection_ratio.objects.create(names=kword, ratio=ratio)

ratio1 = ""
kword1 = 'Average'
print(kword1)
obj1 = drug_drug_interactions.objects.all().filter(Q(Prediction=kword1))
obj11 = drug_drug_interactions.objects.all()
count1 = obj1.count();
count11 = obj11.count();
ratio1 = (count1 / count11) * 100
if ratio1 != 0:
    detection_ratio.objects.create(names=kword1, ratio=ratio1)

ratio12 = ""
kword12 = 'Very Good'
print(kword12)
obj12 = drug_drug_interactions.objects.all().filter(Q(Prediction=kword12))
obj112 = drug_drug_interactions.objects.all()
count12 = obj12.count();
count112 = obj112.count();
ratio12 = (count12 / count112) * 100
if ratio12 != 0:
    detection_ratio.objects.create(names=kword12, ratio=ratio12)

obj = detection_ratio.objects.all()
```

```
    return render(request, 'SProvider/Find_Drug_To_Drug_Interact_Type_Ratio.html',
{'objs': obj})

def View_Remote_Users(request):
    obj=ClientRegister_Model.objects.all()
    return render(request,'SProvider/View_Remote_Users.html',{'objects':obj})

def ViewTrendings(request):
    topic =
drug_drug_interactions.objects.values('topics').annotate(dcount=Count('topics')).order
_by('-dcount')
    return  render(request,'SProvider/ViewTrendings.html',{'objects':topic})



def charts(request,chart_type):
    chart1 = detection_ratio.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts.html", {'form':chart1,
'chart_type':chart_type})

def charts1(request,chart_type):
    chart1 = detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/charts1.html", {'form':chart1,
'chart_type':chart_type})

def Predict_Drug_To_Drug_Interact_Type_Details(request):

    obj =drug_drug_interactions.objects.all()
    return render(request,
'SProvider/Predict_Drug_To_Drug_Interact_Type_Details.html', {'list_objects': obj})

def likeschart(request,like_chart):
    charts =detection_accuracy.objects.values('names').annotate(dcount=Avg('ratio'))
    return render(request,"SProvider/likeschart.html", {'form':charts,
'like_chart':like_chart})
```

```
def Download_Predicted_DataSets(request):

    response = HttpResponse(content_type='application/ms-excel')
    # decide file name
    response['Content-Disposition'] = 'attachment; filename="Predicted_Datasets.xls"'
    # creating workbook
    wb = xlwt.Workbook(encoding='utf-8')
    # adding sheet
    ws = wb.add_sheet("sheet1")
    # Sheet header, first row
    row_num = 0
    font_style = xlwt.XFStyle()
    # headers are bold
    font_style.font.bold = True
    # writer = csv.writer(response)
    obj = drug_drug_interactions.objects.all()
    data = obj  # dummy method to fetch data.
    for my_row in data:
        row_num = row_num + 1
        ws.write(row_num, 0, my_row.uniqueID, font_style)
        ws.write(row_num, 1, my_row.First_drugName, font_style)
        ws.write(row_num, 2, my_row.condition1, font_style)
        ws.write(row_num, 3, my_row.review, font_style)
        ws.write(row_num, 4, my_row.Second_drugName, font_style)
        ws.write(row_num, 5, my_row.condition2, font_style)
        ws.write(row_num, 6, my_row.Prediction, font_style)

    wb.save(response)
    return response

def train_model(request):
    detection_accuracy.objects.all().delete()
```

```
df = pd.read_csv('Drug_Drug_Interactions.csv',encoding='latin-1')

def apply_recommend(Rating):
    if (Rating <= 3):
        return 0  # Bad
    elif (Rating > 3 and Rating<=7):
        return 1  # Average
    elif (Rating >7 and Rating <= 10):
        return 2  # Very Good

df['Results'] = df['rating'].apply(apply_recommend)

#cv = CountVectorizer()
X = df['review']
y = df['Results']

cv = CountVectorizer(lowercase=False, strip_accents='unicode', ngram_range=(1,
1))

X = cv.fit_transform(X)

models = []
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
X_train.shape, X_test.shape, y_train.shape

print("Naive Bayes")
from sklearn.naive_bayes import MultinomialNB
NB = MultinomialNB()
NB.fit(X_train, y_train)
predict_nb = NB.predict(X_test)
naivebayes = accuracy_score(y_test, predict_nb) * 100
print(naivebayes)
```

```
    print(confusion_matrix(y_test, predict_nb))
    print(classification_report(y_test, predict_nb))
    models.append(('naive_bayes', NB))
    detection_accuracy.objects.create(names="Naive Bayes", ratio=naivebayes)

    # SVM Model
    print("SVM")
    from sklearn import svm
    lin_clf = svm.LinearSVC()
    lin_clf.fit(X_train, y_train)
    predict_svm = lin_clf.predict(X_test)
    svm_acc = accuracy_score(y_test, predict_svm) * 100
    print(svm_acc)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, predict_svm))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, predict_svm))
    models.append(('svm', lin_clf))
    detection_accuracy.objects.create(names="SVM", ratio=svm_acc)

    print("Logistic Regression")
    from sklearn.linear_model import LogisticRegression
    reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)
    y_pred = reg.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, y_pred) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, y_pred))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, y_pred))
    models.append(('logistic', reg))
    detection_accuracy.objects.create(names="Logistic Regression",
ratio=accuracy_score(y_test, y_pred) * 100)
```

```
    from sklearn.tree import DecisionTreeClassifier
    print("Decision Tree Classifier")
    dtc = DecisionTreeClassifier()
    dtc.fit(X_train, y_train)
    dtcpredict = dtc.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, dtcpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, dtcpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, dtcpredict))
    models.append(('DecisionTreeClassifier', dtc))
    detection_accuracy.objects.create(names="Decision Tree Classifier",
ratio=accuracy_score(y_test, dtcpredict) * 100)

    print("KNeighborsClassifier")
    from sklearn.neighbors import KNeighborsClassifier
    kn = KNeighborsClassifier()
    kn.fit(X_train, y_train)
    knpredict = kn.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, knpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, knpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, knpredict))
    models.append(('KNeighborsClassifier', kn))
    detection_accuracy.objects.create(names="KNeighborsClassifier",
ratio=accuracy_score(y_test, knpredict) * 100)

    classifier = VotingClassifier(models)
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)
```

```
predicts = 'Labeled_Data.csv'
# df['predict_nb'] = predict_text
df.to_csv(predicts, index=False)
df.to_markdown

obj = detection_accuracy.objects.all()
return render(request,'SProvider/train_model.html', {'objs': obj})
```

SYSTEM TESTING

6.SYSTEM TESTING

6.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to

ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

6.2 INTEGRATION TESTING

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

THE FOLLOWING ARE THE TYPES OF INTEGRATION TESTING:

1.TOP DOWN INTEGRATION

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. BOTTOM-UP INTEGRATION

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing

required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

6.2 ACCEPTANCE TESTING

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

6.3 INTEGRATING TESTING

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

VALIDATION CHECKING

Validation checks are performed on the following fields.

TEXT FIELD:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

NUMERIC FIELD:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

PREPARATION OF TEST DATA

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

USING LIVE TEST DATA:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

USING ARTIFICIAL TEST DATA:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the

requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

The generation of high-quality data to support drug discovery activities is of critical importance. No data analysis, regardless of its level of sophistication, can extract valuable insights from low quality data or inadequately designed experiments.

Drugs ranked according to total harm

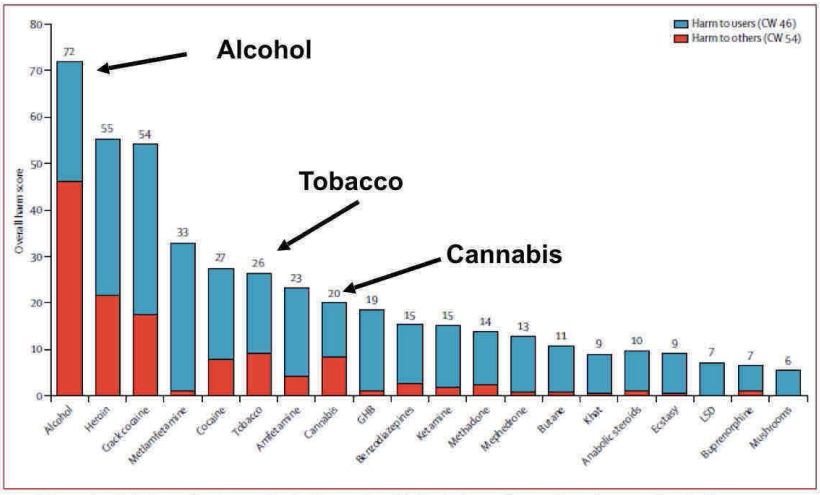


Figure 2: **Drugs ordered by their overall harm scores, showing the separate contributions to the overall scores of harms to users and harm to others**
The weights after normalisation (0–100) are shown in the key (cumulative in the sense of the sum of all the normalised weights for all the criteria to users, 46; and for all the criteria to others, 54). CW=cumulative weight. GHB=γ hydroxybutyric acid. LSD=lysergic acid diethylamide.

TESTING STRATEGY :

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly

for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is
achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this. The future holds a lot to offer to the development and refinement of this project.

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

All the test cases mentioned above passed successfully. No defects encountered.

METHODOLOGIES

USER ACCEPTANCE TESTING

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

USING LIVE TEST DATA:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

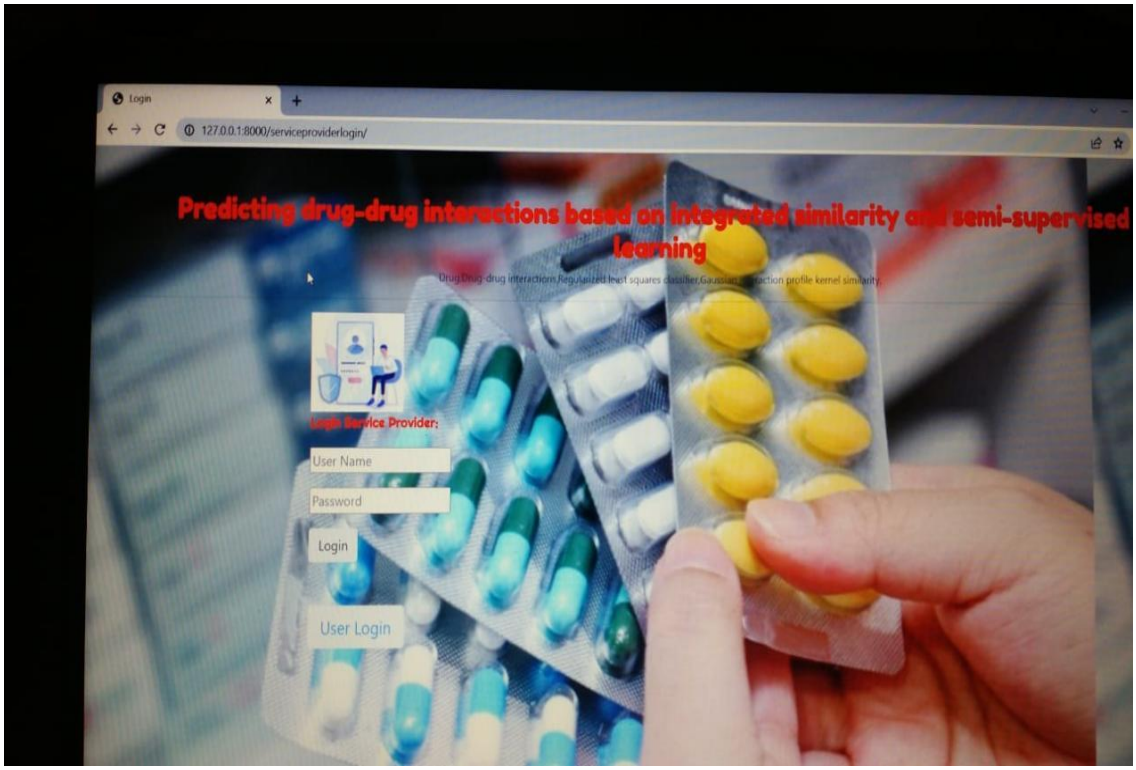
USING ARTIFICIAL TEST DATA:

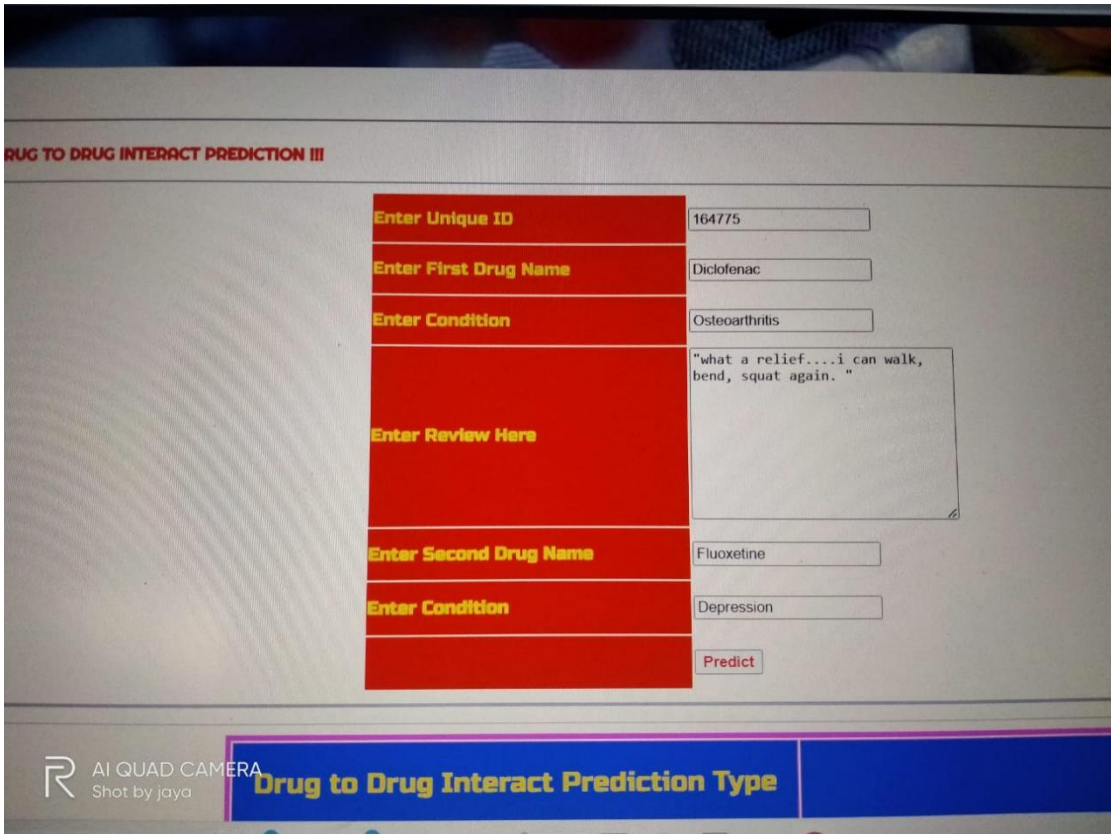
Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

SCREENS & REPORTS

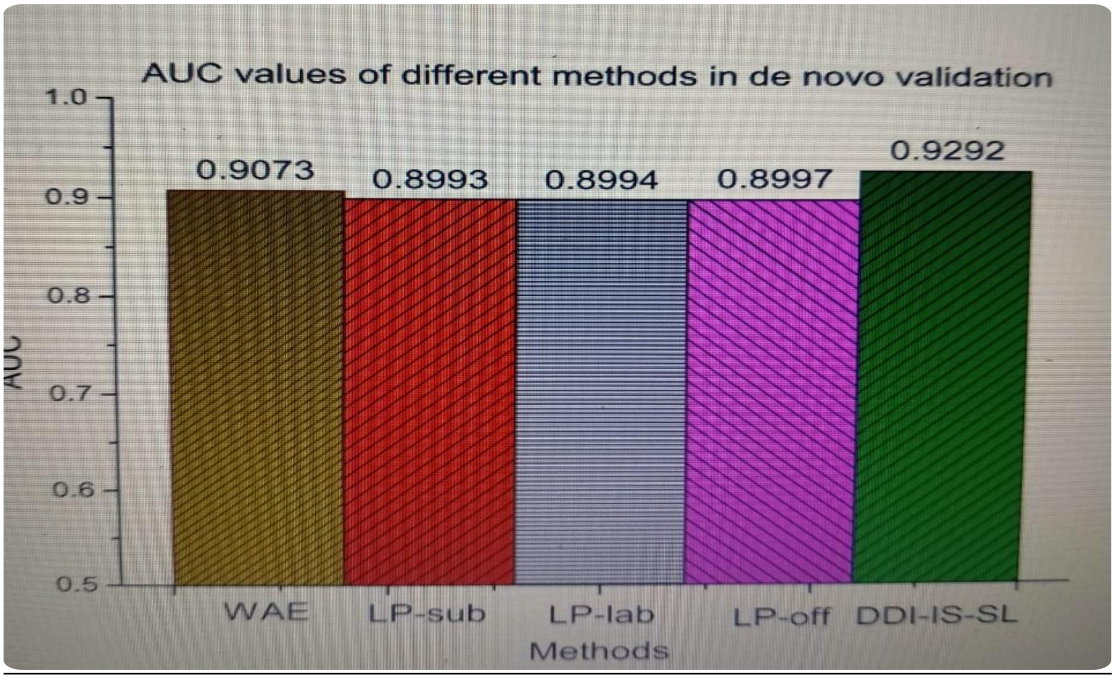
7.SCREENS & REPORTS

When we run the source code the below page will open

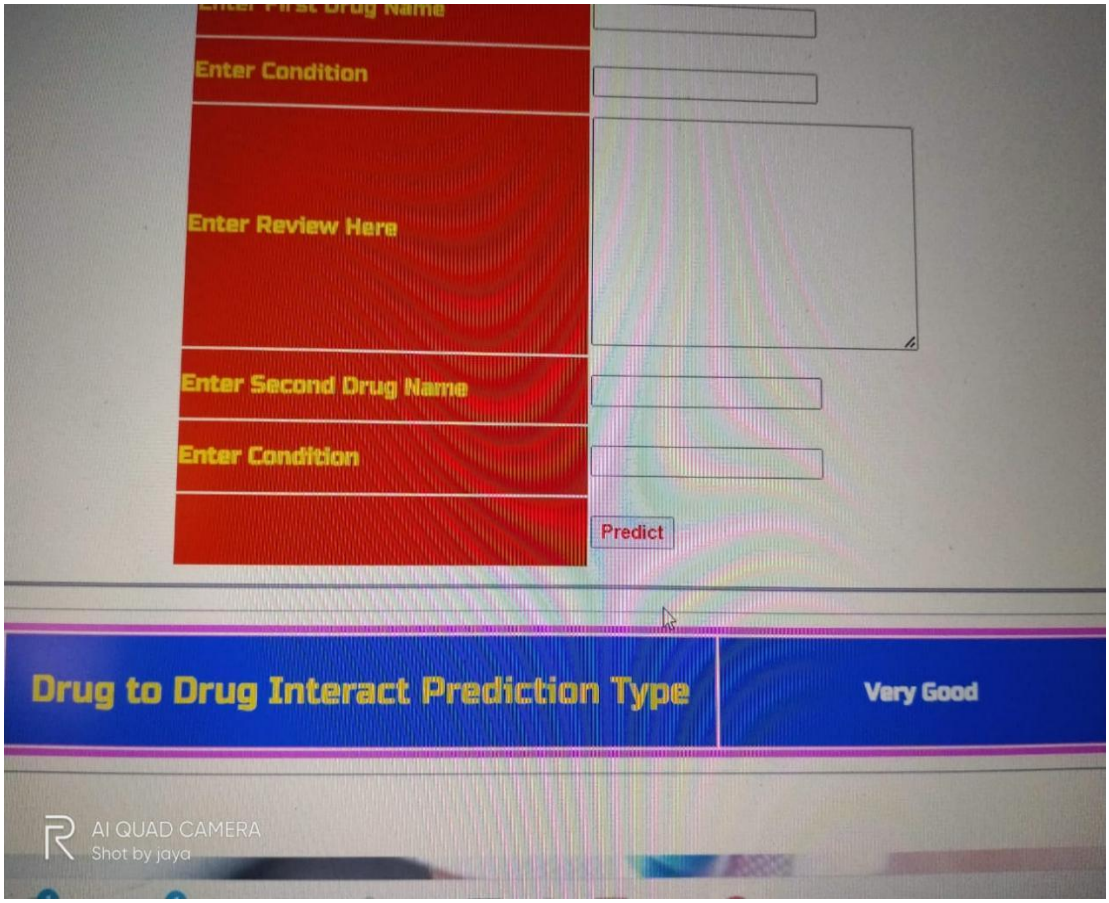




The above page shows the drug details to prediction.



The above fig shows the AUC values of different methods in de novo validation.



The above screen shows the result of the drug.

REPORTS:

Benchmark evaluation and evaluation indices :

In this study, we conduct the 5-fold cross validation, 10-fold cross validation and de novo drug validation to systematically assess the prediction performance of our method. The AUC is used as the metric. In addition, we also compare our method with other competing DDI prediction methods. In the 5-fold cross validation, the known DDIs are divided into 5 groups and then take turns to use one group as testing samples and the rest as the training samples. Similarly, in the 10-fold cross validation, we also divided the known DDIs into 10 groups and then take turns to use one group as testing samples and the rest as the training samples. The de novo drug validation is used to assess the prediction ability of computational methods for new drugs.

One drug is chosen as the test set and the other drugs as the training set in each time of the de novo drug validation, and we conduct the de novo drug validation for all drugs. The AUC value was widely used as the metric to assess the prediction performance of methods. The AUC value of 1 represents the perfect prediction performance of method. The AUC value of less than 0.5 represents the inability of the prediction. Furthermore, we also compare the average computation times of these methods with the 5-fold cross validation.

TABLE 1
The description of datasets

Data type	Data	Database	dimensionality
chemical	Chemical substructures	PubChem	881
Biological	Drug-targets	DrugBank	780
	Drug transporters	DrugBank	18
	Drug enzymes	DrugBank	129
	Drug pathways	KEGG	253
Phenotypic	Drug indications	SIDER	4,897
	Drug side effects	SIDER	4,897
	Drug off side effects	OFFSIDES	9,496
Interaction	Drug-drug interactions	TWOSIDES	DDIs:48,584

Known DDIs are downloaded from TWOSIDES [40]. After projecting drugs of TWOSIDES to PubChem, DrugBank, KEGG, SIDER and OFFSIDER, we obtain a benchmark dataset of known DDIs. It includes 548 drugs and 48,584 DDIs among them. The basic description about data type, data source and dimension of these datasets are demonstrated in Table 1. Furthermore, these datasets also can be
downloaded from previous literature [47]

TABLE 2
The prediction performances of different methods in 5-fold cross validation, the best result is in the bold face.

Method	Feature	AUC
WAE	Chemical data, biological data, phenotypic data	0.9502
L1E	Chemical data, biological data, phenotypic data	0.9570
L2E	Chemical data, biological data, phenotypic data	0.9561
LP	Drug-sub	0.9356
	Drug-Label	0.9364
	Drug-Off Label	0.9374
DDI-IS-SL	Chemical data, biological data, phenotypic data	**0.9691**

TABLE 3
The prediction performances of different methods in 10-fold cross validation, the best result is in the bold face.

Method	Feature	AUC
WAE	Chemical data, biological data, phenotypic data	0.9530
L1E	Chemical data, biological data, phenotypic data	0.9599
L2E	Chemical data, biological data, phenotypic data	0.9594
LP	Drug-sub	0.9359
	Drug-Label	0.9368
	Drug-Off Label	0.9378
DDI-IS-SL	Chemical data, biological data, phenotypic data	**0.9745**

LP was a network-based method to predict DDIs, which only used the drug substructures of chemical data, and drug side effects and drug off side effects of phenotypic data. WAE was a weighted average ensemble model by applying the genetic algorithm (GA) to determine optimal weights, which integrates with the neighbor recommend method, random walk method and matrix perturbation method. L1E and L2E were the classifiifier ensemble models, which adopt the classifiifier ensemble rule by a logistic regression classifier with L1 regularization and L2 regularization , respectively.

5-FOLD CROSS VALIDATION

We obtain the the prediction performances of different methods in the 5-fold cross validation by 10 repeats. Table 2 shows the experiment results of different methods. We can see from Table 2 that our method is superior to other competing methods in terms of AUC values (DDI-IS-SL: 0.9691, WAE: 0.9502, L1E: 0.9570, L2E:0.9561, LP (max:DrugOff Label): 0.9374).

10-FOLD CROSS VALIDATION

We also conduct 10 repeats of the 10-fold cross validation to obtain the prediction performances of our method and other competing methods. Table 3 shows that our method also outperforms other competing methods in terms of AUC.

Specifically, the AUC value of our method reaches 0.9745, which is larger than other competing methods (WAE: 0.9530, L1E: 0.9599, L2E: 0.9594 and LP(max:Drug-Off Label): 0.9378)

Parameter analysis for α:

For new drugs, we take the node-based drug network diffusion model to compute the initial interaction profifiles and also use the parameter α to control the expanded possibility scores of the maximum value. Therefore, we set the value of parameter α based on experiment results of the de novo drug validation.

ITERATION TIMES OF NODE-BASED DRUG NETWORK DIFFUSSION MODEL:

In the node-based drug network diffusion model, the drugsubstructure network and drug-target interaction network are used in the fifirst transfer process, and the drug-drug interaction network is used in the second transfer process. Therefore, in order to full use the drug-substructure, drug-target interaction network and drug-drug interaction network information, we set that the node-based drug networks diffusion model includes two resource transfer processes. However, we can also compute the initial interaction scores for new drugs when using the fifirst transfer process, and the AUC value is 0.7954. In addition, we can also compute the initial interaction scores for new drugs when running the node-based drug networks diffusion model multiple times. Table shows the experiment results of DDI-IS-SL when conducting the node-based drug network diffusion model K times. We can see from

Table 7 that DDI-IS-SL obtains the best prediction performance when $K = 1$. Therefore, we running the node-based drug network diffusion one time in DDI-IS-SL.

FEATURE COMBINATION ANALYSIS :

In this section, we analyze the prediction performances of different drug features in our method. We take the 10-fold cross validation and the de novo drug validation to choose final drug features to predict DDIs. Table 6 demonstrates that the prediction performances of DDI-IS-SL with different drug feature combinations in terms of sensitivity. We can see from Table 6 that DDI-ISSL can obtain the reliable prediction performance when any drug feature is combined with GIP. However, the AUC value of DDI-IS-SL is 0.9617 when only using GIP. The prediction performance is the best (AUC:0.9745) when all drug features and GIP are combined. In addition, our method also can obtain the best prediction performance when drug off side effects and GIP are combined. We think that the combination of drug features and GIP can improve the prediction performance of DDI-IS-SL. Therefore, we choose the combination of all drug features and GIP to predict DDIs.

CONCLUSION & FUTURE WORK

8. CONCLUSION & FUTURE WORK

Multi-drug therapies have widely been used to treat diseases, especially complex diseases such as cancer to improve the treatment effect and reduce the burden of patients. However, the adverse effects resulted from multi-drug therapies have also been observed, which may caused some serious complications and even the patient death. Therefore, identifying drug-drug interactions is helpful in contributing to improved treatment of diseases and reducing the difficulty of drug developments. Especially, it is very necessary to develop new computational methods for identifying DDIs.

In this study, we propose a new computational method (DDI-IS SL) to infer DDIs. DDI-IS-SL integrates the drug chemical, drug biological and drug phenotypic data. The used chemical substructure information of drugs is Pub- Chem substructure which is the 2D binary fingerprints (0 and 1). The biological features of drugs contain drug target interactions, drug enzymes, drug transports and drug pathways. The phenotypic data of drugs include drugindications, drug side effects and drug-off side effects. For each drug, a high-dimensional binary feature vector is constructed with these data. Then we calculate the feature similarit of drugs with the cosine measure. We also compute the GIP similarity of drugs by known DDIs. The final similarity of drugs is calculated as the mean of drug feature similarity and drug GIP similarity. Then we use a semi-supervised learning model (RLS) to compute the probability scores of drug pairs. In the 5-fold cross validation and 10-fold cross validation, DDI-IS-SL achieves the better prediction performance than other competing methods. Furthermore, for new drugs, we also calculate the relational initial interaction scores by using the node-based drug network diffusion method. Our method also achieves the better prediction performance in de novo validation than competing methods.

Although the DDI-IS SL is an effective approach to predict the potential DDIs, there are still some areas for the improvement. For example, we can also consider other more sophisticated methods to integrate the chemical, biological and phenotypic data of drugs. In addition, other prediction models such as deep learning method and identify DDIs in the future.

BIBILIOGRAPHY

9.BILIOGRAPHY

[1] D. Quinn and R. Day, “Drug interactions of clinical importance,” *Drug safety*, vol. 12, no. 6, pp. 393–452, 1995.

[2] T. Prueksaritanont, X. Chu, C. Gibson, D. Cui, K. L. Yee, J. Ballard, T. Cabalu, and J. Hochman, “Drug–drug interaction studies: Regulatory guidance and an industry perspective,” *The AAPS journal*, vol. 15, no. 3, pp. 629–645, 2013.

[3] H. Kusuhara, “How far should we go? perspective of drug-drug interaction studies in drug development,” *Drug metabolism and pharmacokinetics*, vol. 29, no. 3, pp. 227–228, 2014.

[4] N. R. Crowther, A. M. Holbrook, R. Kenwright, and M. Kenwright, “Drug interactions among commonly used medications. chart simplifies data from critical literature review.” *Canadian Family Physician*, vol. 43, p. 1972, 1997.

[5] R. Nahta, M.-C. Hung, and F. J. Esteva, “The her-2-targeting antibodies trastuzumab and pertuzumab synergistically inhibit the survival of breast cancer cells,” *Cancer research*, vol. 64, no. 7,
pp. 2343–2346, 2004.

[6] T.-C. Chou, “Drug combination studies and their synergy quantifification using the chou-talalay method,” *Cancer research*, vol. 70, no. 2, pp. 440–446, 2010.

[7] K. Venkatakrishnan, L. L. von Moltke, R. Obach, and D. J. Greenblatt, “Drug metabolism and drug interactions: application and clinical value of in vitro models,” *Current drug metabolism*, vol. 4,
no. 5, pp. 423–459, 2003.

[8] P. J. Neuvonen, M. Niemi, and J. T. Backman, “Drug interactions with lipid-lowering drugs: mechanisms and clinical relevance,” *Clinical Pharmacology & Therapeutics*, vol. 80, no. 6, pp. 565–581,
2006.

[9] Y. B¨ottiger, K. Laine, M. L. Andersson, T. Korhonen, B. Molin, M.- L. Ovesj¨o, T. Tirkkonen, A. Rane, L. L. Gustafsson, and B. Eiermann, “Sfifinxła drug-drug interaction database designed for clinical de
cision support systems,” *European journal of clinical pharmacology*, vol. 65, no. 6, pp. 627–633, 2009.

[10] M. P. Pai, D. M. Graci, and G. W. Amsden, “Macrolide drug interactions: an update,” *Annals of Pharmacotherapy*, vol. 34, no. 4, pp. 495–513, 2000.

[11] J. Kuhlmann and W. M¨uck, “Clinical-pharmacological strategies to assess drug interaction potential during drug development,” *Drug safety*, vol. 24, no. 10, pp. 715–725, 2001.
[12] S. Preskorn and S. Werder, “Detrimental antidepressant drug– drug interactions: Are they clinically relevant?” *Neuropsychopharmacology*, vol. 31, no. 8, pp. 1605–1612, 2006.
[13] D. Sridhar, S. Fakhraei, and L. Getoor, “A probabilistic approach for collective similarity-based drug–drug interaction prediction,” *Bioinformatics*, vol. 32, no. 20, pp. 3175–3182, 2016.
[14] S. Ekins and S. A. Wrighton, “Application of in silico approaches to predicting drug–drug interactions,” *Journal of pharmacological and toxicological methods*, vol. 45, no. 1, pp. 65–69, 2001.
[15] G. Jin, H. Zhao, X. Zhou, and S. T. Wong, “An enhanced petri-net model to predict synergistic effects of pairwise drug combinations from gene microarray data,” *Bioinformatics*, vol. 27, no. 13, pp.
i310–i316, 2011.
[16] R. Ferdousi, R. Safdari, and Y. Omidi, “Computational prediction of drug-drug interactions based on drugs functional similarities,” *Journal of Biomedical Informatics*, vol. 70, pp. 54–64, 2017.
[17] R. Safdari, R. Ferdousi, K. Aziziheris, S. R. Niakan-Kalhori, and Y. Omidi, “Computerized techniques pave the way for drug-drug interaction prediction and interpretation,” *BioImpacts: BI*, vol. 6,
no. 2, p. 71, 2016.
[18] N. P. Tatonetti, G. H. Fernald, and R. B. Altman, “A novel signal detection algorithm for identifying hidden drug-drug interactions IEEE/ACM Transactions on Computational Biology and Bioinformatics,Volume:19,Issue:1,Issue Date:01.Jan-Feb.202211 in adverse event reports,” *Journal of the American Medical Informatics Association*, vol. 19, no. 1, pp. 79–85, 2011.
[19] A. Gottlieb, G. Y. Stein, Y. Oron, E. Ruppin, and R. Sharan, “Indi: a computational framework for inferring drug interactions and their associated recommendations,” *Molecular systems biology*, vol. 8,
no. 1, p. 592, 2012.
[20] S. Yamazaki, T. R. Johnson, and B. J. Smith, “Prediction of drugdrug interactions with crizotinib as the cyp3a substrate using a physiologically based pharmacokinetic model,” *Drug Metabolism*
and Disposition, vol. 43, no. 10, pp. 1417–1429, 2015.
[21] L. Tari, S. Anwar, S. Liang, J. Cai, and C. Baral, “Discovering drug– drug interactions: a text-mining and reasoning approach based on properties of drug metabolism,” *Bioinformatics*, vol. 26, no. 18, pp.

i547–i553, 2010.
[22] S. Vilar, R. Harpaz, E. Uriarte, L. Santana, R. Rabadan, and C. Friedman, “Drugłdrug interaction through molecular structure similarity analysis,” *Journal of the American Medical Informatics*
Association, vol. 19, no. 6, pp. 1066–1074, 2012.
[23] S. Vilar, E. Uriarte, L. Santana, T. Lorberbaum, G. Hripcsak, C. Friedman, and N. P. Tatonetti, “Similarity-based modeling in large-scale prediction of drug-drug interactions,” *Nature protocols*,
vol. 9, no. 9, pp. 2147–2163, 2014.
[24] F. Cheng and Z. Zhao, “Machine learning-based prediction of drug–drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties,” *Journal of the American*
Medical Informatics Association, vol. 21, no. e2, pp. e278–e286, 2014.
[25] P. Li, C. Huang, Y. Fu, J. Wang, Z. Wu, J. Ru, C. Zheng, Z. Guo, X. Chen, W. Zhou *et al.*, “Large-scale exploration and analysis of drug combinations,” *Bioinformatics*, vol. 31, no. 12, pp. 2007–2016,
2015.
[26] L. Liu, L. Chen, Y.-H. Zhang, L. Wei, S. Cheng, X. Kong, M. Zheng, T. Huang, and Y.-D. Cai, “Analysis and prediction of drug–drug interaction by minimum redundancy maximum relevance and
incremental feature selection,” *Journal of Biomolecular Structure and Dynamics*, vol. 35, no. 2, pp. 312–329, 2017.
[27] H. Luo, P. Zhang, H. Huang, J. Huang, E. Kao, L. Shi, L. He, and L. Yang, “Ddi-cpi, a server that predicts drug–drug interactions through implementing the chemical–protein interactome,” *Nucleic*
acids research, vol. 42, no. W1, pp. W46–W52, 2014.
[28] T. Takeda, M. Hao, T. Cheng, S. H. Bryant, and Y. Wang, “Predicting drug–drug interactions through drug structural similarities and interaction networks incorporating pharmacokinetics and
pharmacodynamics knowledge,” *Journal of Cheminformatics*, vol. 9, no. 1, p. 16, 2017.
[29] X. Chen, B. Ren, M. Chen, Q. Wang, L. Zhang, and G. Yan, “Nllss: predicting synergistic drug combinations based on semisupervised learning,” *PLoS computational biology*, vol. 12, no. 7, p.
e1004975, 2016.
[30] H. Luo, J. Wang, M. Li, J. Luo, X. Peng, F.-X. Wu, and Y. Pan,
“Drug repositioning based on comprehensive similarity measures and bi-random walk algorithm,” *Bioinformatics*, vol. 32, no. 17, pp. 2664–2671, 2016.

[31] C. Yan, J. Wang, W. Lan, F.-X. Wu, and Y. Pan, “Sdtrls: Predicting drug-target interactions for complex diseases based on chemical substructures,” *Complexity*, vol. 2017, 2017.
[32] D.-S. Cao, N. Xiao, Y.-J. Li, W.-B. Zeng, Y.-Z. Liang, A.-P. Lu, Q.-S. Xu, and A. Chen, “Integrating multiple evidence sources to predict adverse drug reactions based on a systems pharmacology model,”
CPT: pharmacometrics & systems pharmacology, vol. 4, no. 9, pp. 498– 506, 2015.
[33] P. Zhang, F. Wang, J. Hu, and R. Sorrentino, “Label propagation prediction of drug-drug interactions based on clinical side effects,” *Scientifific reports*, vol. 5, 2015.
[34] J. Huang, C. Niu, C. D. Green, L. Yang, H. Mei, and J.-D. J. Han, “Systematic prediction of pharmacodynamic drug-drug interactions through protein-protein-interaction network,” *PLoS computational biology*, vol. 9, no. 3, p. e1002998, 2013.
[35] K. Park, D. Kim, S. Ha, and D. Lee, “Predicting pharmacodynamic drug-drug interactions through signaling propagation interference on protein-protein interaction networks,” *PloS one*, vol. 10, no. 10, p. e0140816, 2015.
[36] B. Jin, H. Yang, C. Xiao, P. Zhang, X. Wei, and F. Wang, “Multitask dyadic prediction and its application in prediction of adverse drug-drug interaction,” in *AAAI*, pp. 1367–1373, 2017.
[37] S. Vilar, E. Uriarte, L. Santana, N. P. Tatonetti, and C. Friedman, “Detection of drug-drug interactions by modeling interaction pro- fifile fifingerprints,” *PloS one*, vol. 8, no. 3, p. e58321, 2013.
[38] T. van Laarhoven, S. B. Nabuurs, and E. Marchiori, “Gaussian interaction profifile kernels for predicting drug–target interaction,” *Bioinformatics*, vol. 27, no. 21, pp. 3036–3043, 2011.
[39] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled
examples,” *Journal of machine learning research*, vol. 7, no. Nov, pp.
2399–2434, 2006.
[40] N. P. Tatonetti, P. Y. Patrick, R. Daneshjou, and R. B. Altman, “Data-driven prediction of drug effects and interactions,” *Science translational medicine*, vol. 4, no. 125, pp. 125ra31–125ra31, 2012.
[41] Y. Wang, J. Xiao, T. O. Suzek, J. Zhang, J. Wang, and S. H. Bryant, “Pubchem: a public information system for analyzing bioactivities of small molecules,” *Nucleic acids research*, vol. 37, no. suppl 2, pp.
W623–W633, 2009.

[42] D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey, “Drugbank: a comprehensive resource for in silico drug discovery and exploration,” *Nucleic acids research*, vol. 34, no. suppl 1, pp. D668–D672, 2006.

[43] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu *et al.*, “Drugbank 3.0: a comprehensive resource for omics research on drugs,” *Nucleic acids research*, vol. 39, no. suppl 1, pp. D1035–D1041, 2010.

[44] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa, “Kegg for representation and analysis of molecular networks involving diseases and drugs,” *Nucleic acids research*, vol. 38, no. suppl 1, pp. D355–D360, 2009.

[45] V. Law, C. Knox, Y. Djoumbou, T. Jewison, A. C. Guo, Y. Liu, A. Maciejewski, D. Arndt, M. Wilson, V. Neveu *et al.*, “Drugbank 4.0: shedding new light on drug metabolism,” *Nucleic acids research*, vol. 42, no. D1, pp. D1091–D1097, 2013.

[46] H. Luo, J. Wang, C. Yan, M. Li, W. Fangxiang, and P. Yi, “A novel drug repositioning approach based on collaborative metric learning,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, to be published. DOI: 10.1109/TCBB.2019.2926453.

[47] W. Zhang, Y. Chen, F. Liu, F. Luo, G. Tian, and X. Li, “Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data,” *BMC bioinformatics*, vol. 18, no. 1, p. 18, 2017.

[48] C. Yan, J. Wang, and F.-X. Wu, “Dwnn-rls: regularized least squares method for predicting circrna-disease associations,” *BMC bioinformatics*, vol. 19, no. 19, p. 520, 2018.

[49] C. Yan, G. Duan, F. Wu, Y. Pan, and J. Wang, “Brwmda: Predicting microbe-disease associations based on similarities and bi-random walk on disease and microbe networks,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, to be published. DOI: 10.1109/TCBB.2019.2907626.

[50] W. Lan, M. Li, K. Zhao, J. Liu, F.-X. Wu, Y. Pan, and J. Wang, “Ldap: a web server for lncrna-disease association prediction,” *Bioinformatics*, vol. 33, no. 3, pp. 458–460, 2016.

[51] C. Lu, M. Yang, F. Luo, F.-X. Wu, M. Li, Y. Pan, Y. Li, and J. Wang, “Prediction of lncrna-disease associations based on inductive matrix completion,” *Bioinformatics*, vol. 34, no. 19, pp. 3357–3364,2018.

TEXT BOOKS:

"Jab, Jab, Jab, Right Hook: How to Tell Your Story in a Noisy Social World" by Gary Vaynerchuk

"Crushing It!: How Great Entrepreneurs Build Their Business and Influence-

and How You Can, Too" by Gary Vaynerchuk

"The Social Media Bible: Tactics, Tools, and Strategies for Business Success"

by Lon Safko and David K. Brake

"Social Media Marketing All-in-One For Dummies" by Jan Zimmerman and Deborah Ng

"Contagious: Why Things Catch On" by Jonah Berger

"Youtility: Why Smart Marketing Is About Help, Not Hype" by Jay Baer

"Permission Marketing: Turning Strangers into Friends and Friends into Customers" by Seth Godin

"Likeable Social Media: How to Delight Your Customers, Create an Irresistible Brand, and Be

Amazing on Facebook, Twitter, LinkedIn, Instagram, Pinterest, and More" by Dave Kerpen

"Epic Content Marketing: How to Tell a Different Story, Break through the Clutter, and Win More

Customers by Marketing Less" by Joe Pulizzi

"The Power of Visual Storytelling: How to Use Visuals, Videos, and Social Media to Market Your

Brand" by Ekaterina Walter and Jessica Gioglio

Remember, ethical marketing is about providing value to your audience and building genuine relationships rather than spamming or bombarding them with irrelevant content. Always follow best practices and guidelines set by the platforms you use for marketing purposes.