Cairo University

Faculty of Computers and Information Department of Computer Sciences

Prediction of drug–target interactions using machine learning

Supervised by

Dr. Khaled Bahnasy

Dr. Samar Hesham

Implemented by

20178027	**Abdelrahman Moustafa**
20178055	**Khaled Waheed**
20178022	**Zyad Ayman**
20178007	**Ahmed Alaa**

Graduation Project

Academic Year 2020-2021

Documentation

Table of Contents

List of tables

List of figures

List of Abbreviations

Abbreviation	Meaning
DTI	Drug Target Interaction
DDR	Design and Development Research
ADR	Adverse Drug Reaction
FDA	Food and Drug Administration
SDF	Structure Data File
H-Softmax	Hierarchical Softmax
SVM	Support Vector Machine
RBF	Radial Basis Function
ATC	Anatomical Therapeutic Chemical Classification System
CADD	Computer Aided Drug Design
KNN	K Nearest Neighbor
ML	Machine Learning
NRWRH	Network-based Random Walk with Restart on the Heterogeneous Network
GPCR	G Protein-Coupled Receptors
GIP	Gaussian Interaction Profile
RLS	Regularized Least Squares
ERD	Entity Relationship Diagram

Acknowledgment

The proceeding words are written with love We want to express our gratitude and appreciation for all your exerted efforts throughout the past four academic years .Your efforts resulted in what we are today, citizens yearning to serve their country with all the knowledge and experience we acquired from our most efficient, beloved and parent like professors, and special thanks for our graduation project professor and the biggest supporter DR. Khaled Bahnasy, we are glad to work with you along the year and learn a lot from your knowledge.

Abstract

Computational drug repurposing aims at finding new medical uses for existing drugs. The identification of novel drug target interactions (DTIs) can be a useful part of such a task. Computational determination of DTIs is a convenient strategy for systematic screening of a large number of drugs in the attempt to identify new DTIs at low cost and with reasonable accuracy. This necessitates development of accurate computational methods that can help focus on the follow-up experimental validation on a smaller number of highly likely targets for a drug. Although many methods have been proposed for computational DTI prediction, they suffer the high false-positive prediction rate or they do not predict the effect that drugs exert on targets in DTIs. In this report, first, we present a comprehensive review of the recent progress in the field of DTI prediction from data-centric and algorithm-centric perspectives. The aim is to provide a comprehensive review of computational methods for identifying DTIs, which could help in constructing more reliable methods. Then, we present DDR, an efficient method to predict the existence of DTIs.

Chapter 1

1. Introduction

1.1 Motivation

In the past most drugs have been discovered either by identifying the active ingredient from traditional remedies or by serendipitous discovery. A new approach of Drug discovery has been to understand how disease and infection are controlled at the molecular and physiological level and to target specific entities based on this knowledge.

Although some (ADRs) are not very serious, others cause the death of more than 2 million people in the United States each year, including more than 100,000 fatalities. Drug development time frame can range from 3–20 years and costs can range between several billion to tens of billions of dollars

1.2 Problem Definition

- The number of newly approved drugs by the FDA is decreasing due to in the unacceptable toxicity and adverse side effects for those drug candidates.
- Recent research definitely indicates that harmful side effects is due to namely off-target effects in addition to the primary therapeutic targets.
- Studies also showed that most of the FDA-approved drugs can have interaction with multiple targets (proteins). Aiming to reduce the spent cost and time of bringing a new drug to the market. The purpose of drug repositioning is the detection for new clinical uses for existing drugs that have already been strictly verified for their safety and bioavailability and narrow down the scope of search of candidate medications .so drug repositioning, is another important part in drug discovery. As The —multi-target, multi-drugǁ in place of —one target, one drugǁ model has been widely accepted in order to speed up the drug development process.
- Serendipity is one of the many factors that may contribute to drug discovery, it means finding of one thing while looking for something else.
- The known drug-target interactions based on wet-lab experiments are limited to a very small number due to cost and time.

Biological Background

Protein Structure and Function

The building blocks of proteins are amino acids; the defining feature of an amino acid is its side chain. When connected together by a series of peptide bonds, amino acids form a polypeptide, another word for protein. The polypeptide will then fold into a specific conformation depending on the interactions forming ionic bonds, hydrogen bonds, van der Waals interactions, covalent bonds (By Cysteines) between its amino acid side chains. **Assigning the same function to things that look similar is innate to human nature. The same process can be used for protein sequences:** This is called **homology annotation** and the principle that enables scientists to use similarity to infer function is based on the conservation of a given sequence or slight variations of it throughout evolution. In general terms, the more similar two sequences are, the more likely they are to be related. Consequently, homology annotation is based on the comparison of DNA or proteins at the sequence level - that is, by comparing the similarity of nucleotides or amino acids sequences between related sequences. Protein sequences that confer function are often found in blocks of conservation called protein domains. These regions have a defined three-dimensional structure or motif (shape) that can function and evolve independently from the rest of the protein sequence. These blocks of conservation are found in proteins throughout nature, and any given protein sequence can have more than one protein domain. The key to using motif similarity to infer function relies on the principle that when two proteins have a conserved function, although their sequence similarity at the amino acid level can be lost, their protein domain conservation must remain.

Drug Structure and Function

All drugs are chemical compounds, prepared from chemical reactions like condensation reaction, cyclization reaction, elimination reaction, substitution reaction. The chemical structure for drugs determines the shape of the drug and how to interact with the binding sites of the protein by chemical bonds like ionic, hydrogen, and covalent bonds, and van der Waals forces. Hydrogen and ionic are the most common types of drugs and proteins bonds require little energy and are made and broken easily the chemical structure of the drug determines these bonds

Pharmaceutical drugs are often classified into drug classes—groups of related drugs that have similar chemical structures, the same mechanism of action (binding to the same biological target), a related mode of action, and that are used to treat the same disease. The Anatomical Therapeutic Chemical Classification System (ATC), the most widely used drug classification system, assigns drugs a unique ATC code, which is an alphanumeric code that assigns it to specific drug classes within the ATC system. Another major classification system is the Biopharmaceutics Classification System. This classifies drugs according to their solubility and permeability or absorption properties.

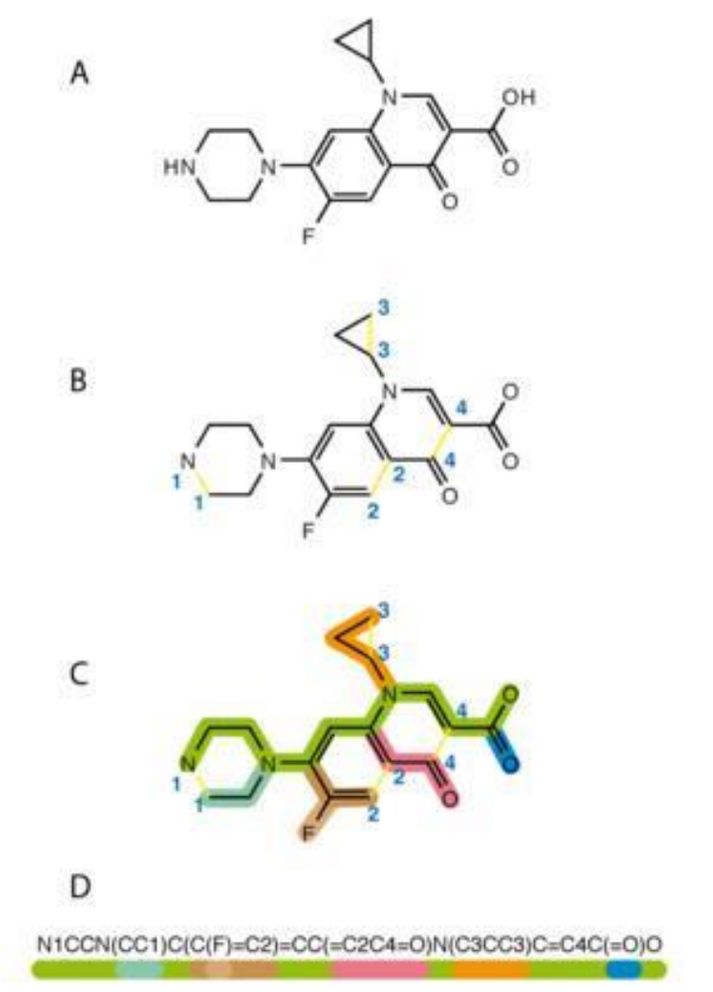


Figure 1- Smile Format

Drug design

Drug Design:

In the most basic sense, drug design involves the design of small molecules that are complementary in shape and charge to the biomolecular target with which they interact and therefore will bind to it. Drug design frequently but not necessarily relies on computer modeling techniques. This type of modeling is often referred to as **computer-aided drug design (LB-CADD)**. Finally, drug design that relies on the knowledge of the three-dimensional structure of the biomolecular target is known as **structure-based drug design**. The phrase —drug design‖ is to some extent a misnomer. What is really meant by drug design **is ligand design** (i.e., design of a small molecule that will bind tightly to its target). Drugs may be designed that bind to the active region and inhibit this key molecule. Another approach may be to enhance the normal pathway by promoting specific molecules in the normal pathways that may have been affected in the diseased state.

Drug Development Challenges

- Drug development is a lengthy, complex, and costly process, entrenched with a high degree of uncertainty that a drug will actually succeed.
- The unknown pathophysiology for many nervous system disorders makes target identification challenging. • Animal models often cannot recapitulate an entire disorder or disease.
- Challenges related to heterogeneity of the patient population might be alleviated with increased clinical phenotyping and endotyping.
- Greater emphasis on human data might lead to improved target identification and validation.
- FDA organization doesn’t approve on new drugs as it need long time till it makes sure that these drugs have no dangerous effects on people As a result, to these challenges we need fast and safe way to discover if there are drugs already made and can target more than protein and this can

be made by DDR method.

Common Approaches of Targeted Drug Delivery

(smart drug delivery)

1) Controlling the distribution of drug by incorporating it in a carrier system.
2) Altering the structure of the drug at molecular level.
3) Controlling the input of the drug into biological environment to ensure a programmed and desirable bio distribution.

Properties of Ideal Targeted Drug Delivery

1) Nontoxic, biocompatible and physicochemical stable in vivo and in vitro.
2) Restrict drug distribution to target cells or tissue or organ or should have uniform capillary distribution.
3) Controllable and predictable rate of drug release.
4) Minimal drug leakage during transit.
5) Carrier used must be biodegradable or readily eliminated from the body without any problem. (decomposed)
6) Its preparation should be easy or reasonably simple, reproductive and cost effective.

Computational Approaches for DTI prediction

There are different approaches that have been proposed for addressing the problem of predicting new DTIs; some major prediction approaches are docking simulation, ligand-based approaches, machine learning, network inference and text mining approaches. Although many methods have been proposed for computational DTI prediction, they suffer the high false-positive prediction rate. In the following sections we characterize the commonly used computational techniques and how they tackled the problem of predicting new DTIs.

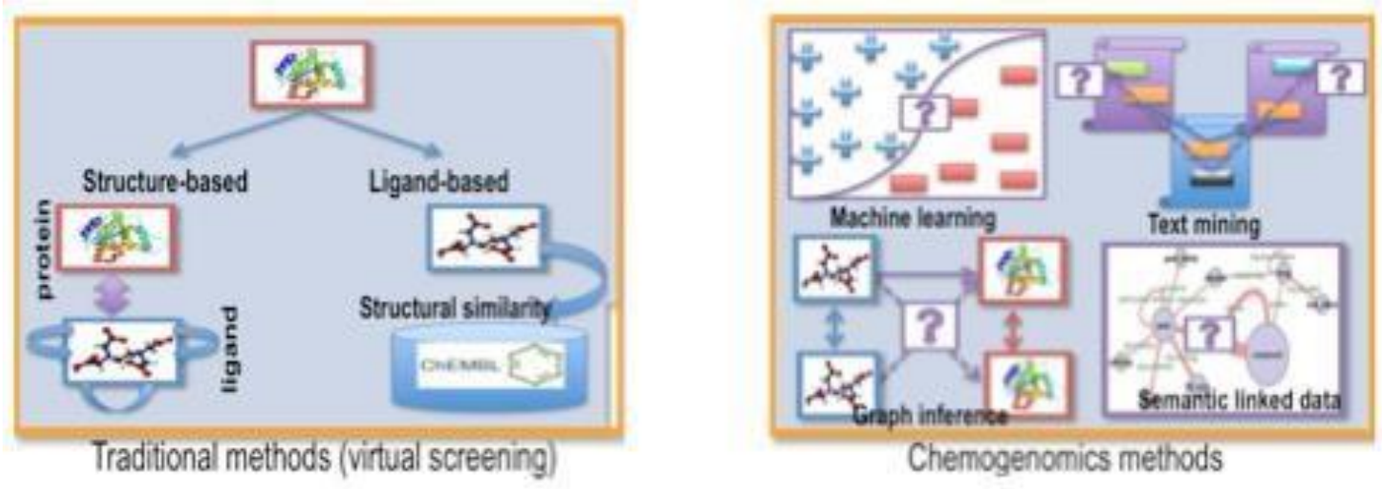


Figure 2- Summary of the computational approaches for the DTI prediction

Early attempts in the DTI prediction

Early attempts in computational prediction of DTIs can be categorized into two main groups and include docking simulations and ligand-based approaches. These approaches typically focus on one particular target of interest (i.e., the 3D structure of the target (or the compound). Docking methods consider the three-dimensional structure of target proteins. However, this approach is extensively time-consuming and the structural information of targets is not available for all target proteins. Ligand-based methods compare a query ligand with a set of known ligands with target proteins. However, it may not perform well in cases the number of known ligands with target proteins is small.

The use of network topology in the DTI prediction

The uses of the topology of DTI network, as the only source of information for the prediction of new DTIs links is capable of predicting true DTIs with reasonable accuracy. Some of the DTI prediction methods are based on only considering known DTIs using techniques based on graph theory and network analysis. For instance, the prediction model built using only interaction profiles from known DTIs, compared to the model that uses additional information about the chemical structure and genomic sequence similarities, can be used as an accurate tool for prediction of DTIs. The relevance of using DTI topology network as a source of information for predicting new DTI is based on the assumption that drugs exhibiting a similar pattern of interaction and non-interaction with the targets of a DTI network are likely to show similar interaction behavior with respect to new targets. (homology annotation)

DDR Method

The heterogeneous DTI graph is a weighted graph that is constructed with m nodes from the drug set and n nodes from the set of target proteins. The edge between two drug nodes or two target protein nodes represents the similarity between them and is weighted by the similarity value obtained from the similarity calculation. The edge between a drug and a target protein represents a known DTI and is weighted by 1.

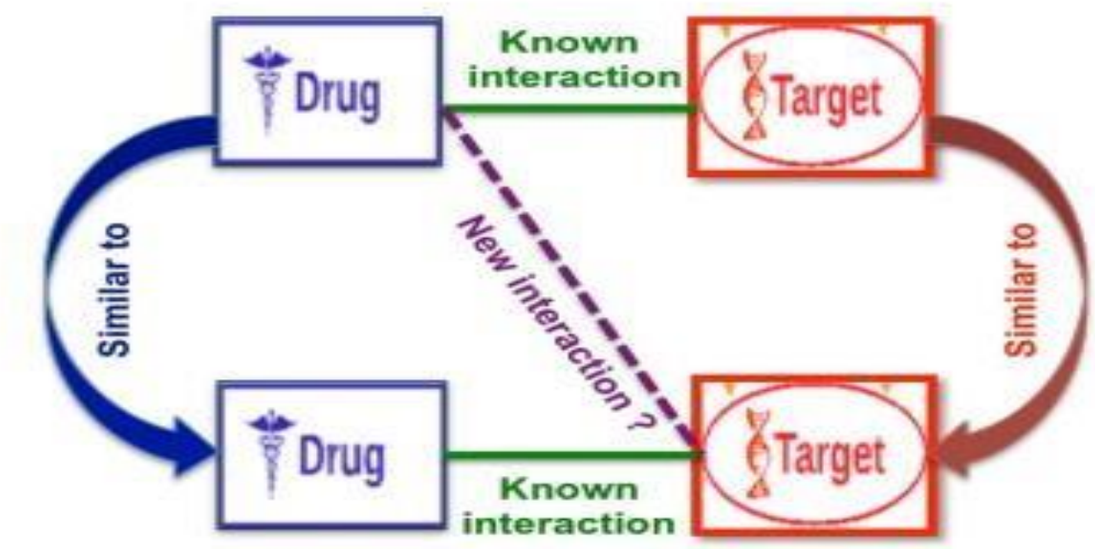


Figure 3- DTI Graph

Similarity Measures

-Similarity Matrix between drugs and each other, its size is m*m, where m is number of drugs. ranges between [0,1].

-Similarity matrix between proteins and each other, its size n*n where n is the number of targets, ranges between [0,1]. the closer to zero the less similar.

Inferring interaction profile

Drugs (or target proteins) with high similarities to a new drug (or a new target protein) are said to be the neighbors of the drug (the target protein), the inferred value of interaction for a new drug with a specific target protein is represented as the ratio of the sum of similarity values for drug. neighbors interacting with this target protein relative to the total sum of all neighbors‟ similarity values.

Graph Mining

Is the set of tools and techniques used to (a) analyze the properties of real-world graphs, (b) predict how the structure and properties of a given graph might affect some application, and (c) develop models that can generate realistic graphs that match the patterns found in real-world graphs of?
interest.

Graph Mining has played an important role in a range or real world applications:

- Medicines: structure of molecules.
- Bioinformatics: biological networks.
- Social Science: social networks.

We formalize data mining and machine learning challenges as graph **problems.**

Graph mining, which has gained much attention in the last few decades, is one of the novel approaches for mining the dataset represented by graph
structure.

Graph mining finds its applications in various problem domains, including:
bioinformatics, chemical reactions, Program flow structures, computer
networks, social networks.

Different data mining approaches are used for mining the graph-based data and performing useful analysis on these mined data.

Various graph mining approaches have been proposed. Each of these approaches is based on either classification; clustering or decision trees
data mining techniques.

Path-category-based features

A path structure of a path that starts at a D node and ends up at a T node describes a subgraph that sequentially links drug and target protein nodes. For example, a path Drug1-Drug2-Target1 connects the Drug1 node with the Target1 node through the similarity edge between Drug1 and Drug2 and via the interaction edge between Drug2 and Target1. The path structure of this path is D-D-T. All paths with more than one edge and without loops, starting at a D node and ending at a T node, and having the same path structure define a path-category on the heterogeneous DTI graph.

Predicting drug–target interactions

We will consider these path-categories through which drug nodes could connect to target protein nodes. To do this we start with a given drug di to reach a given target protein tj through a specific path-category. We restrict traversing the graph to retrieve all paths passing only through the K-nearest neighbors of drugs to di and only through the K-nearest neighbors of target proteins to tj. Next, for each path we calculate an edge-weight product value

obtained by multiplying all weights of edges of these paths.

Machine Learning

- **KNN:**

  The k-nearest neighbors (KNN) algorithm is a simple, easy-to implement **supervised machine learning algorithm** that can be used to solve both classification and regression problems.

  The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

- **Decision Tree**

  Covering both **classification and regression as supervised machine learning algorithm**. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in **data mining** for deriving a strategy to reach a particular goal.

- **SVM (Support-vector machine)**

  More formally, a support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.

- **Random Forest**

  Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both

  Classification and Regression problems in ML. **"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

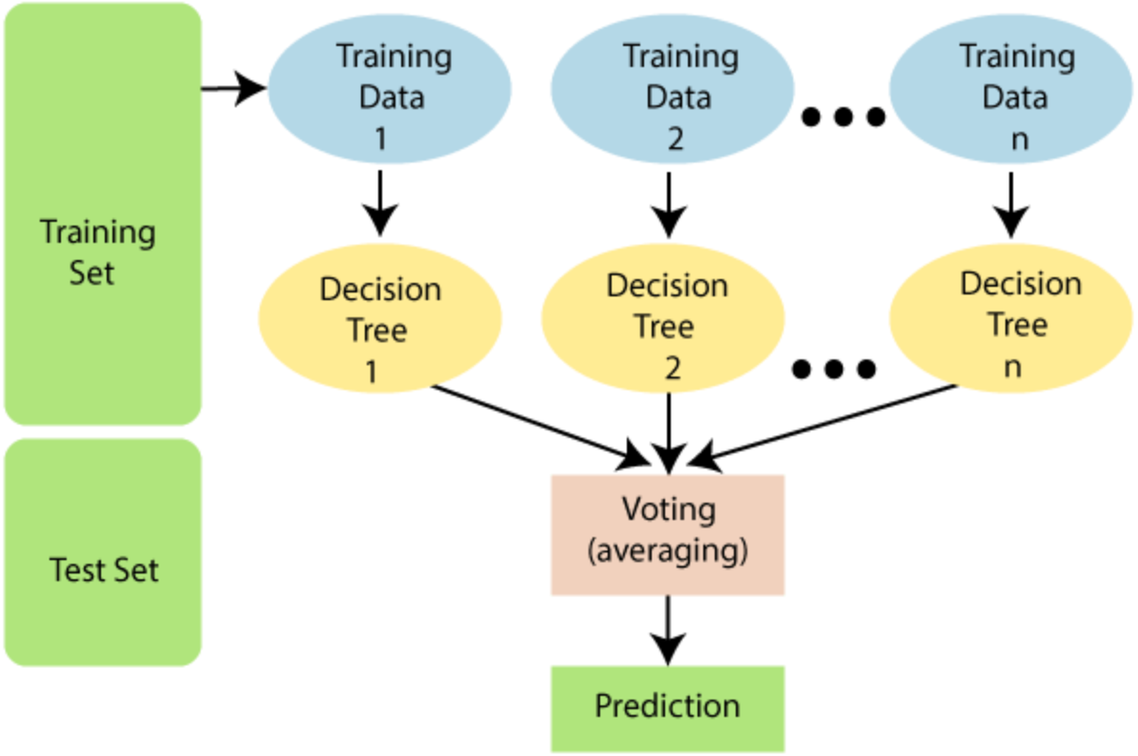


Figure 4- Random Forest

1.3 Objectives

- Motivated by the limitations of the existing methods for the prediction of the potential DTIs, the types of the effects a drug exert on the target, and aiming to further improve their prediction accuracy, we present a method that utilize a heterogeneous drug-target graph that contains information about DTIs, effect types of DTIs, as well as multiple similarities between drugs and multiple similarities between target proteins.
- The main objective of our idea is to find computational strategy to identify drug–target interactions (DTIs) types at low cost with reasonable accuracy in order to speed the drug development process.
- Avoid the side effects of a drug and Link the newly identified DataTarget interactions of a known drug to the treatment of diseases that are different from diseases for which the drug has been originally developed.

1.4 Time Plan

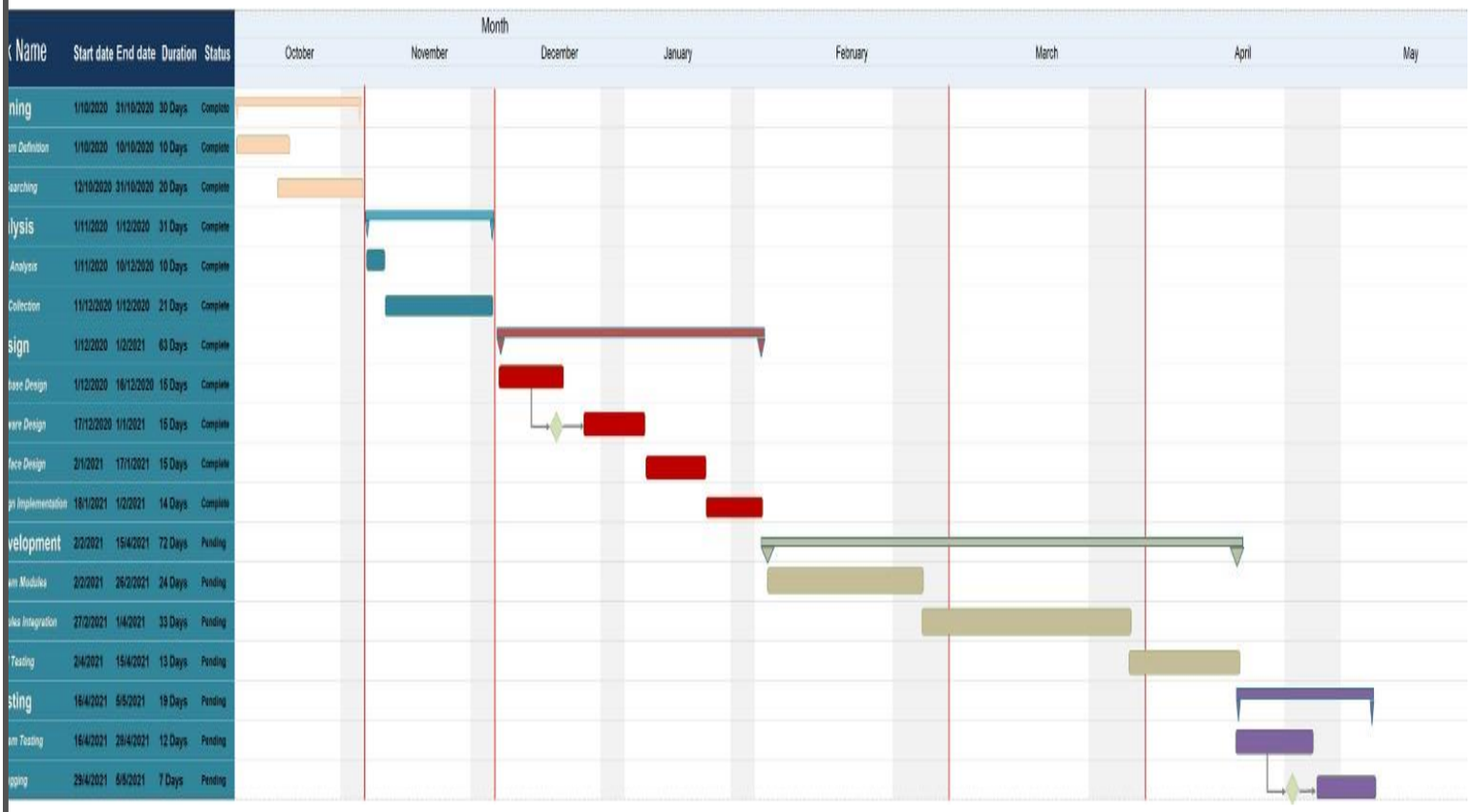


Figure 5- Time Plan

1.5 Project development methodology

Iterative Planning is the process to adapt as the project unfolds by changing the plans. Plans are changed based on feedback from the monitoring process or changes in the project assumptions. It is important to involve the team in the planning process. The people doing the work should be actively involved in planning the project. When they get involved in the decision, they become motivated to get it right. After all, they were hired and they have the skills to understand the dependencies. Once they complete the plans, they will own it and will accept the schedule. That's why everyone in the team worked on a certain thing during the whole project period and exchanged many tasks.

We used this developmental approach as we monitored the performance and changed some algorithms and methods for delivering some valuable improvements or additional features in each increment.

First of all, we went through the process of breaking down big problems into smaller tasks, where we divided the project into 3 ways which are: - features, similarities and interactions. At the first iteration, we collected dataset from DrugBank which was CSV file containing 13,581 drugs with their features which were: type, state, groups, kingdom, superclass, subclass, direct parent and class. At the second iteration we started calculating Euclidean distance and Manhattan distance between the 13,581 drugs. Unfortunately, we found that at this iteration that the features we extracted didn't show true similarity for the collected drugs. At the third iteration, we downloaded another whole new dataset from DrugBank which was SDF file containing 11,160 drugs and we extracted some features which were: ID and chemical structure (for the fingerprint to be calculated) and started calculating the fingerprint for each molecule by using Tanimoto similarity which is the most popular similarity measure for comparing chemical structures represented by means of fingerprints**,** two structures are usually considered similar if T > 0.85, but some pairs of compounds didn't get a high score, even they look very similar with each other. Below is one example: 6037 (Pubchem CID) ------ DB00988 (the similarity is only 0.61). At the fourth iteration we tried another fingerprint method which was the Morgan fingerprint; Morgan fingerprints are circular fingerprints, which are also topological fingerprints. They are obtained by modifying the standard Morgan algorithm. It can be roughly equivalent to Extended-Connectivity Fingerprints (ECFPs). This type of fingerprint has many advantages, such as fast calculation speed, no predefined (which can represent an infinite number of different molecular characteristics), can contain chiral information, each element in the fingerprint represents a specific substructure, and can be easily Carry out analysis and interpretation, and make corresponding modifications according to different needs.

The initial purpose of this type of fingerprint design is to search for molecular features related to activity, not substructure search. In addition, it can also be used in similarity search, clustering, virtual screening and other directions.

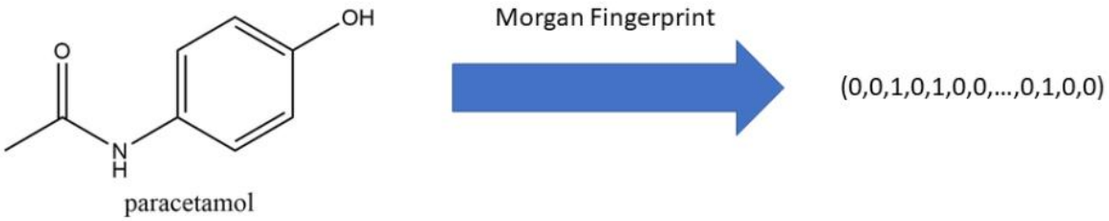


Figure 6- Morgan Fingerprint

So we found at this iteration better results for similarity than the past one as the Morgan fingerprint is well suited to capture structural differences (atom type, bond type, connectivity etc).

So we used the Dice coefficient is the number of features in common to both molecules relative to the average size of the total number of features present, The equation for this concept is: $2 * |X \cap Y| / (|X| + |Y|)$.

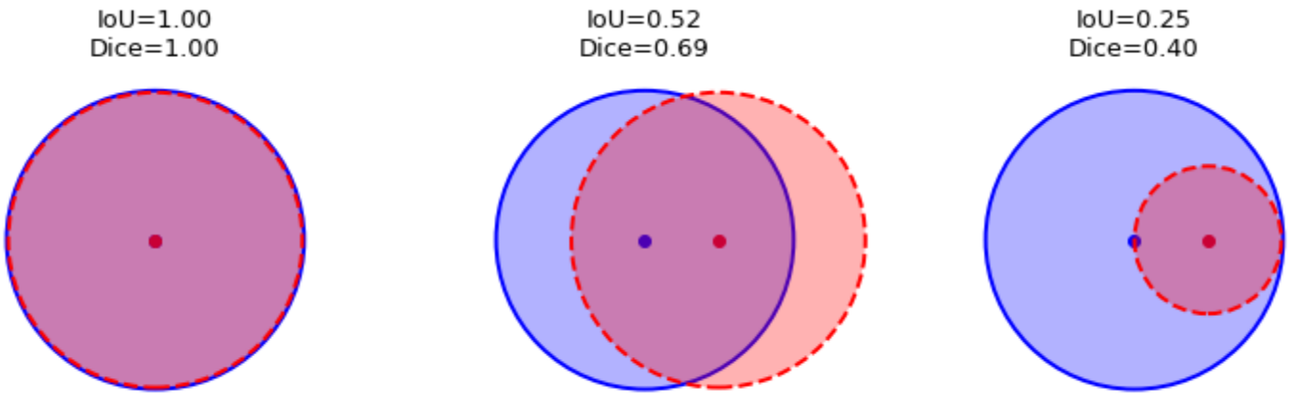


Figure 7- Dice Coefficient

After finishing the drug part we went through collecting dataset for protein from Uniprot as a FASTA file which had 563975 records and extracted these records by using BioPython to deal only with protein id and sequence, then we tried to get the similarity between these proteins by using Pairwise Sequence Alignment which is used to identify regions of similarity that may indicate functional, structural and/or evolutionary relationships between two biological sequences (protein or nucleic acid).

After finishing the similarities part, we collected a new dataset containing interactions between drugs and proteins which held 11,060 record in a CSV file and we used only the drug id and the protein id in the data.

At the end of the fourth iteration we managed to get the best score for the drug similarity using the Dice similarity and the best score for protein similarity using Blast similarity.

At the fifth iteration we went through a new process which is the graph node embedding. Node embedding algorithms compute low-dimensional vector representations of nodes in a graph. These vectors, also called embeddings, can be used for machine learning. In our project, we created a graph using NetworkX library in python and used DeepWalk algorithm on the graph we created.

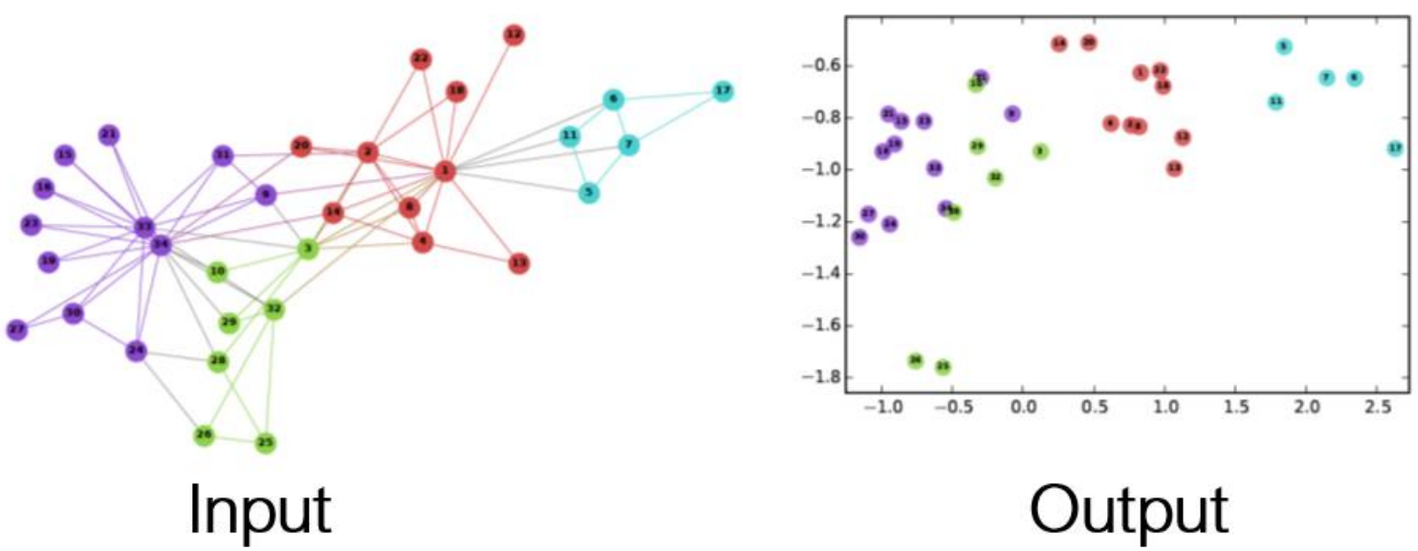


Figure 8- Node Embedding

DeepWalk is an algorithm that is used to create embeddings of the nodes in a graph. The embeddings are meant to encode the community structure of the graph. To obtain node embeddings, we first need to arrange for sequences of nodes from the graph. How do we get these sequences from a graph? Well, there is a technique for this task called Random Walk.

Random Walk is a technique to extract sequences from a graph. We can use these sequences to train a skip-gram model to learn node embeddings. So we extracted sequences from the nodes in this graph.

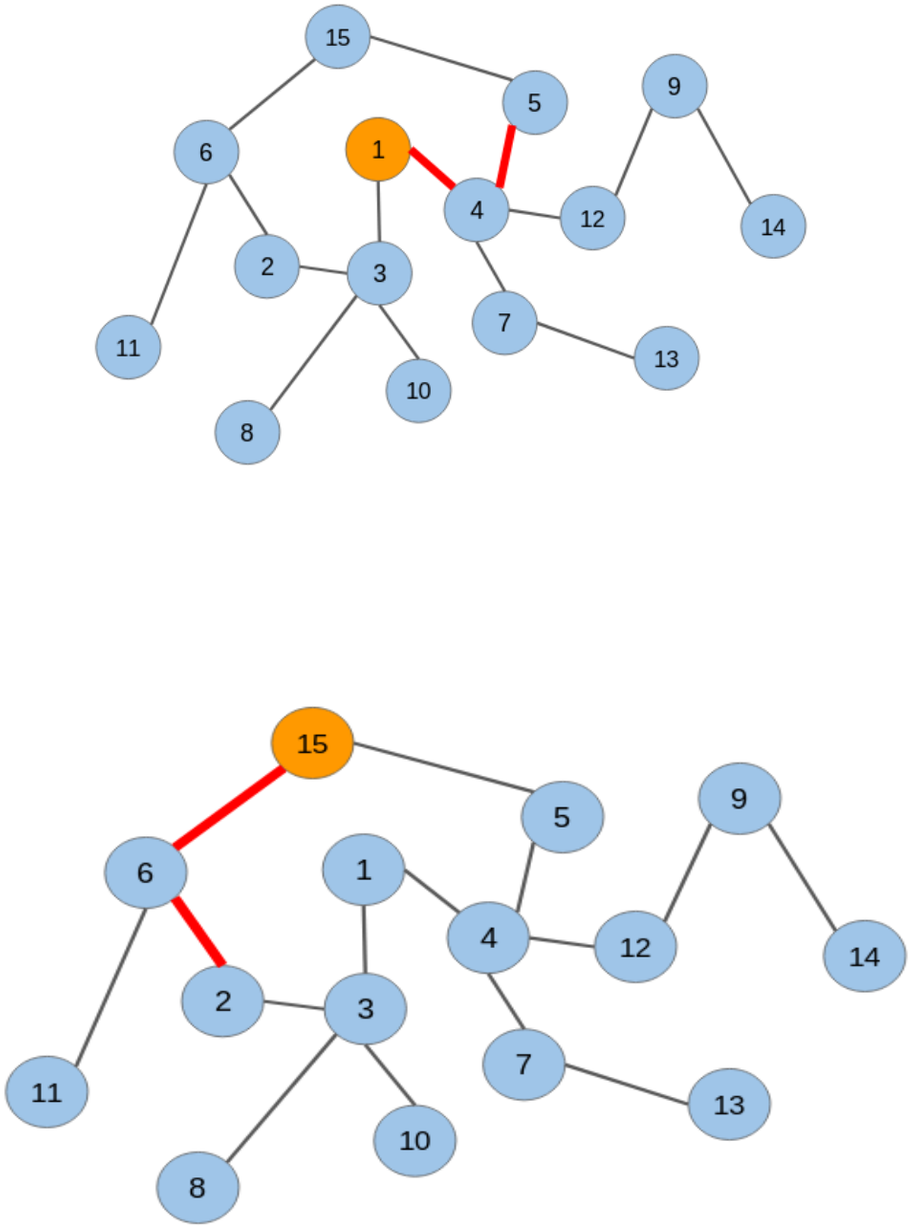


Figure 9 - DeepWalk Algorithm

After generating node-sequences, we have to feed them to a skip-gram model to get node embeddings. That entire process is known as DeepWalk.

This leads us to another new concept which is Word2vec, Word2vec is a two-layer neural net that processes text by “vectorizing” words. Its input is a text corpus and its output is a set of vectors: feature vectors that represent words in that corpus. While Word2vec is not a deep neural network, it turns text into a numerical form that deep neural networks can understand. The output of the Word2vec neural net is a vocabulary in which each item has a vector attached to it, which can be fed into a deep-learning net or simply queried to detect relationships between words. The Word2vec model is dependent on many parameters which are: skip gram, word-size and hierarchical softmax.

Skip gram; the aim of skip-gram is to predict the context given a word. In our project we used skip gram in order to predict the next or previous similar protein or drug. Two separate errors are calculated with respect to the two target variables and the two error vectors obtained are added element-wise to obtain a final error vector which is propagated back to update the weights. The weights between the input and the hidden layer are taken as the word vector representation after training.

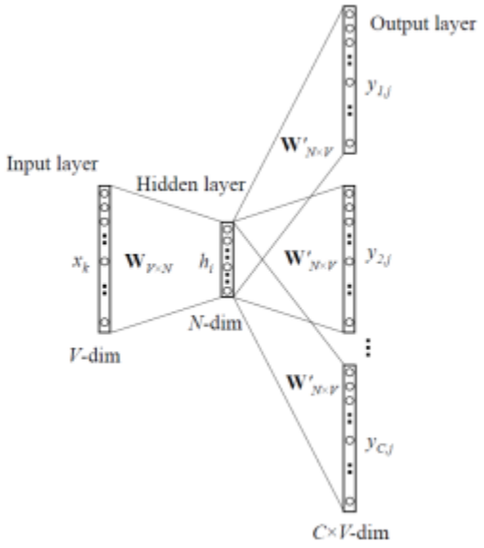


Figure 10- Skip Gram Architecture

The idea for using skip gram model:

-Skip-gram model can capture two semantics for a single word

-Skip-gram with negative sub-sampling outperforms every other method generally.

The softmax layer is a core part of many current neural network architectures. When the number of output classes is very large, such as in the case of language modeling, computing the softmax becomes very expensive. This post explores approximations to make the computation more efficient. H-Softmax essentially replaces the flat softmax layer with a hierarchical layer that has the words as leaves. This allows us to decompose calculating the probability of one word into a sequence of probability calculations, which saves us from having to calculate the expensive normalization over all words. Replacing a softmax layer with H-Softmax can yield speedups for word prediction tasks of at least 50× and is thus critical for low-latency tasks.

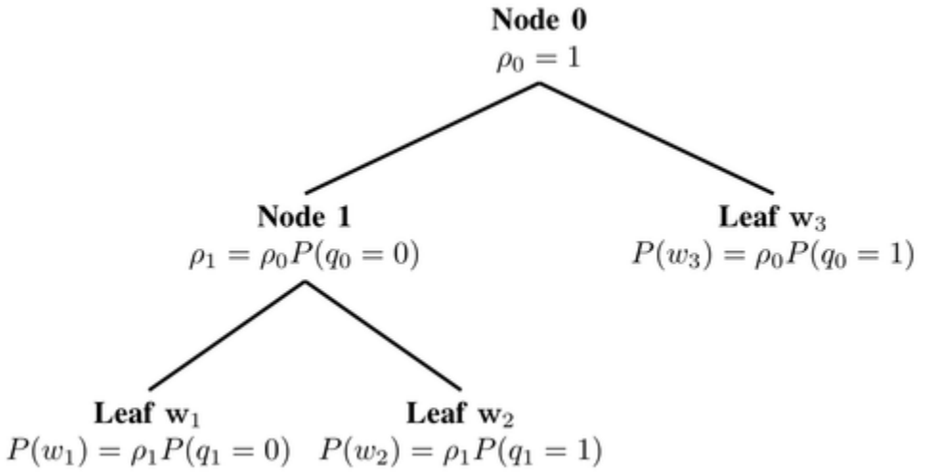


Figure 11- Hierarchical Softmax

After finishing the whole Word2vec process, the result is a new dataset containing all the node embeddings with their features and holding a record of 4864 rows x 64 columns holding features.

Using our drug-target interactions dataset, we extract the vectors representing each interaction between a drug and protein whether it's expressing a positive or negative interaction. After this process, we get a new dataset which is interaction embedding dataset.

At the sixth iteration we'll go through our machine learning phase, in this phase we tried 29 different models and we'll refer to these models in the testing chapter. The best model we figured was the SVM model.

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text. Compared to newer algorithms like neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands). This makes the algorithm very suitable for text classification problems, where it's common to have access to a dataset of at most a couple of thousands of tagged samples.

A support vector machine takes the data points and outputs the hyperplane (which in two dimensions it's simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.

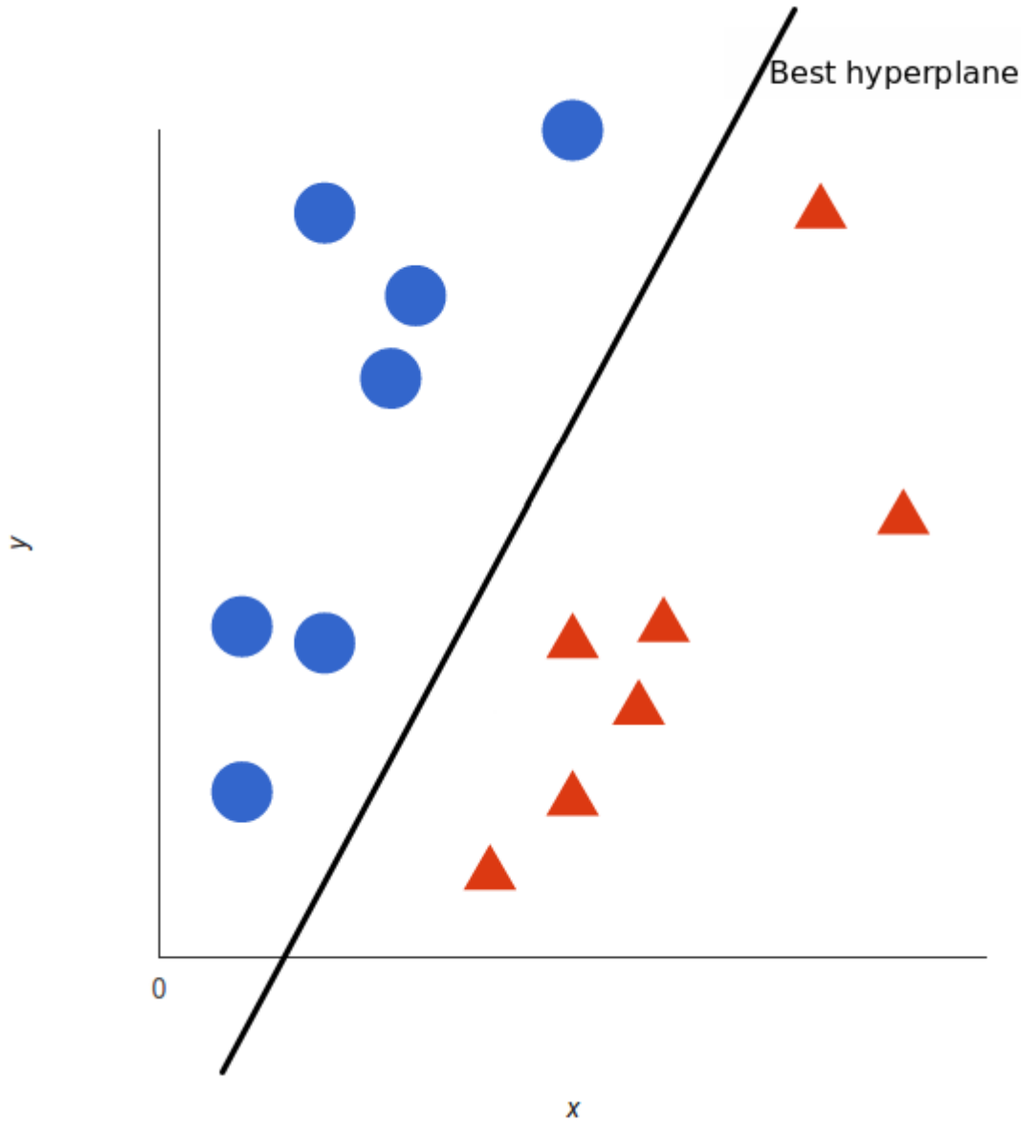


Figure 12- SVM Model and Hyperplane

A kernel is a method of placing a two-dimensional plane into a higher-dimensional space, so that it is curved in the higher-dimensional space. (In simple terms, a kernel is a function from the low-dimensional space into a higher-dimensional space.)

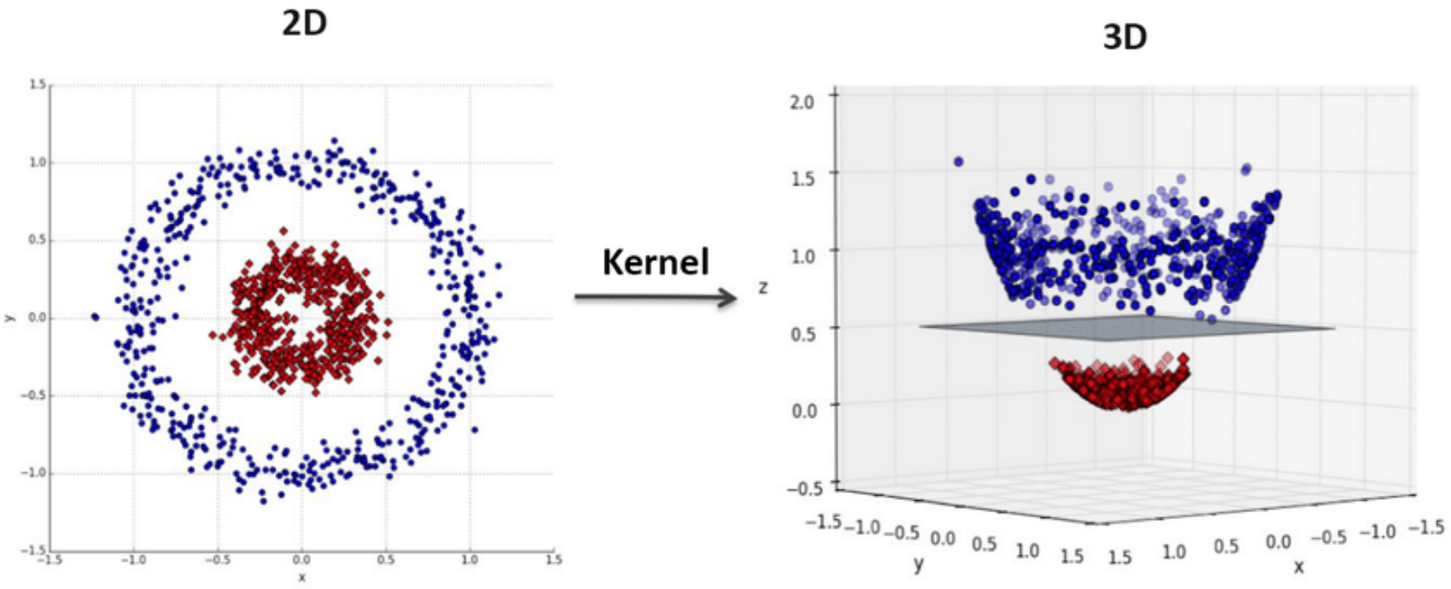


Figure 13- Kernel Trick

One of the powerful kernel tricks is the RBF kernel, RBF kernel is a function whose value depends on the distance from the origin or from some point.

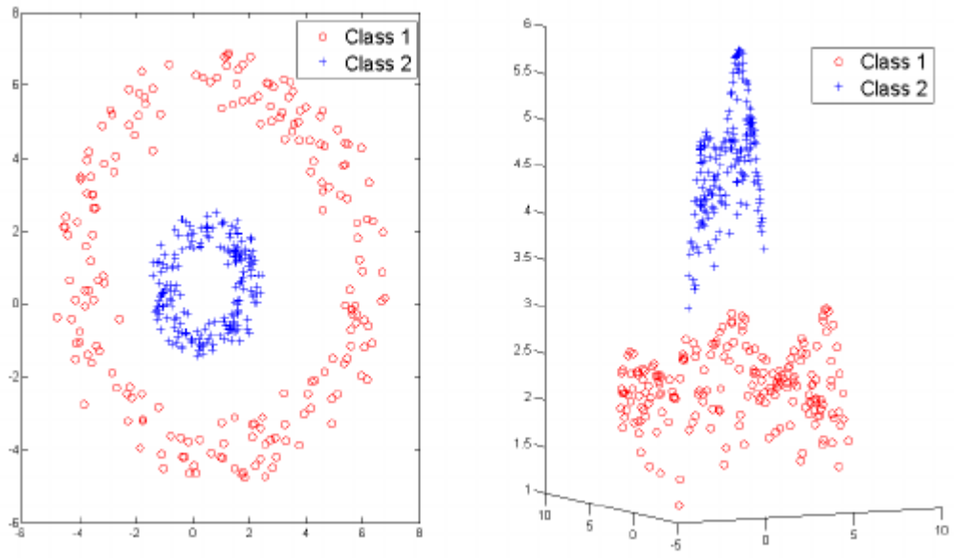


Figure 14- RBF Kernel method

The seventh and the last iteration is the GUI part, in this part we used streamlit framework, Streamlit is an open-source app framework for Machine Learning and Data Science teams. We created a front-end demo using this framework to show the whole functionalities for our project like drug-drug similarity, protein-protein similarity and prediction of new drug or new protein using our built SVM model.

And we'll refer to this at Chapter 5.

1.6 Tools

• Streamlit
• Lucidchart
• Creately
• Microsoft (gantt chart template)
• Git and Github
• Google collab
• Anaconda

Table 1- Used Tools

1.7 Report Organization

This document divided into six chapters describing all aspects of our project, and the description of how each chapter is written in the document comes as follows:

Chapter 1 (Introduction):

In this chapter, we define the motivation, the problem, project objective, Gantt chart for our time plan, project development methodology and used tools.

Chapter 2 (Related Work):

We got the previous ideas done for our project with their methods and results, after this we compare these previous ideas to our project.

Chapter 3 (System Analysis):

In This chapter we specify the project, describe the functional and non-functional requirements and describe our use case diagram.

Chapter 4 (System Design):

In this chapter we describe all the diagrams of the system including component diagram, class diagram, sequence diagrams and the ERD model and after this we go through our demo with a simple front-end GUI made by streamlit framework.

Chapter 5 (Implementation and Testing):

In This Chapter describe in details the dataset of drugs, dataset of proteins, dataset of already known interactions, algorithms and the techniques used to implement this project starting with drug-drug similarity, protein-protein similarity, forming graph, applying DeepWalk on this graph, extracting node embeddings and we finished by applying SVM ML model on our embeddings.

Chapter 2

Related Work

Title	Author	DataSet	Method	Accuracy	Comment
PREDICT: a method for inferring novel drug indications with application to personalized medicine	Assaf Gottlieb, Gideon Y Stein,Eytan Ruppin and Roded Sharan	OMIM,DrugBank,DailyMed,Drugs.com	Predict (predicting drug indications)	92%	Compared with two other methods (GBA method / Cmap method) getting that **PREDICT** method has higher AUC.
Drug–target interaction prediction by random walk on the heterogeneous network	Xing Chen,Ming-Xi Liu and Gui-Ying Yan	KEGG,DrugBank,SuperTarget	Network-based Random Walk with Restart on the Heterogeneous network	90%	
Gaussian interaction profile kernels for predicting drug–target interaction	Twan van Laarhoven ,Sander B.Nabuurs and Elena Marchiori	KEGG ,DrugBank,BRENDA	Gaussian Interaction Profile Kernel	98%	developing computational methods that predict true interaction pairs with high accuracy.

Table 2- Related Work

1-method for the large-scale prediction of drug indications (PREDICT) that can handle both approved drugs and novel molecules

This method is based on the observation that similar drugs are indicated for similar diseases, and utilizes multiple drug–drug and disease–disease similarity measures for the prediction task on cross-validation, We validate our predictions by their overlap with drug indications that are currently under clinical trials, and by their agreement with tissue-specific expression information on the drug targets we further show that disease-specific genetic signatures can be used to accurately predict drug indications for new diseases.

2-Network-based Random Walk with Restart on the Heterogeneous network

(NRWRH) is developed to predict potential drug–target interactions on a large scale under the hypothesis that similar drugs often target similar target proteins and the framework of Random Walk. Compared with traditional supervised or semi-supervised methods, NRWRH makes full use of the tool of the network for data integration to predict drug–target associations.

It integrates three different networks (protein–protein similarity network, drug–drug similarity network, and known drug–target interaction networks) into a heterogeneous network by known drug–target interactions and implements the random walk on this heterogeneous network.

When applied to four classes of important drug–target interactions including enzymes, ion channels, GPCRs and nuclear receptors, NRWRH significantly improves previous methods in terms of cross-validation and potential drug–target interaction prediction.

3-A simple machine learning method that uses the drug-target network as the only source of information is capable of predicting true interaction pairs with high accuracy we introduce interaction profiles of drugs (and of targets) in a network, which are binary vectors specifying the presence or absence of interaction with every target (drug) in that network. We define a kernel on these profiles, called the Gaussian Interaction Profile (GIP) kernel, and use a simple classifier, (kernel) Regularized Least Squares (RLS), for prediction drug-target interactions.

Chapter 3

3. System Analysis

3.1 System Architecture

Figure 15- System Architecture

-Data access layer

1-Proteins Database: (UniProt)

Which is online database containing proteins features such as Protein's id, name, sequence which will be used to find similarity between proteins

2-Drugs Database: (DrugBank)

Which is online database containing drugs features that we will use to find the similarity between them such as group, status,
targeted proteins, class, sub-class.

3-Drugs and Targets positive (known) interactions Database: (DrugBank)

Which is online database containing the already known interactions between the drug database and the proteins database.

-Application Layer

- Calculate the similarities between drugs and each other.
- Calculate the similarities between proteins and each other.
- Based upon these similarities make the prediction.

-Stakeholders

- Pharmacists
- Drug companies
- Researchers
- Bioinformaticians

3.1.1 Functional Requirements

FUNCTIONAL REQUIREMENTS
Show the 3D structure of any target the user asks for
Find drug-drug similarity for the user
Find target-target similarity for the user
Find the drug-target interaction for the user

Figure 16- Functional Requirements

3.1.2 Non-Functional Requirements

NON-FUNCTIONAL REQUIREMENTS
Data Usability (User ability to get useful info from data)
Performance
Data Availability (Data is available always at a certain level of performance)
Data Integrity (Data is trusted and validated and also its accuracy is measured)
Data Reusability (User able to reuse data)

Figure 17- Non-Functional Requirements

3.2 Use-case Diagram

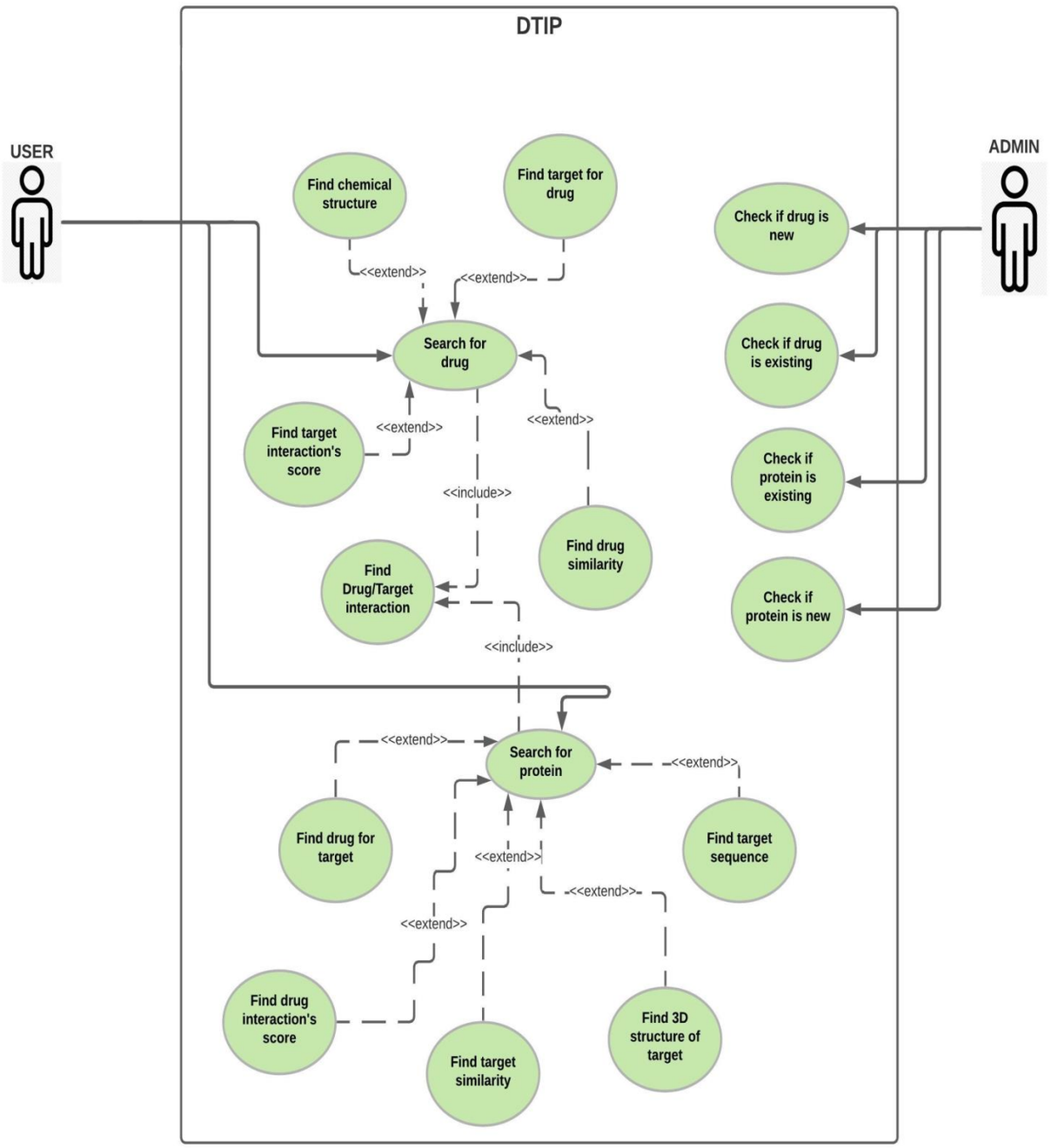


Figure 18- Use Case Diagram

Chapter 4

4. System Design

- Component Diagram

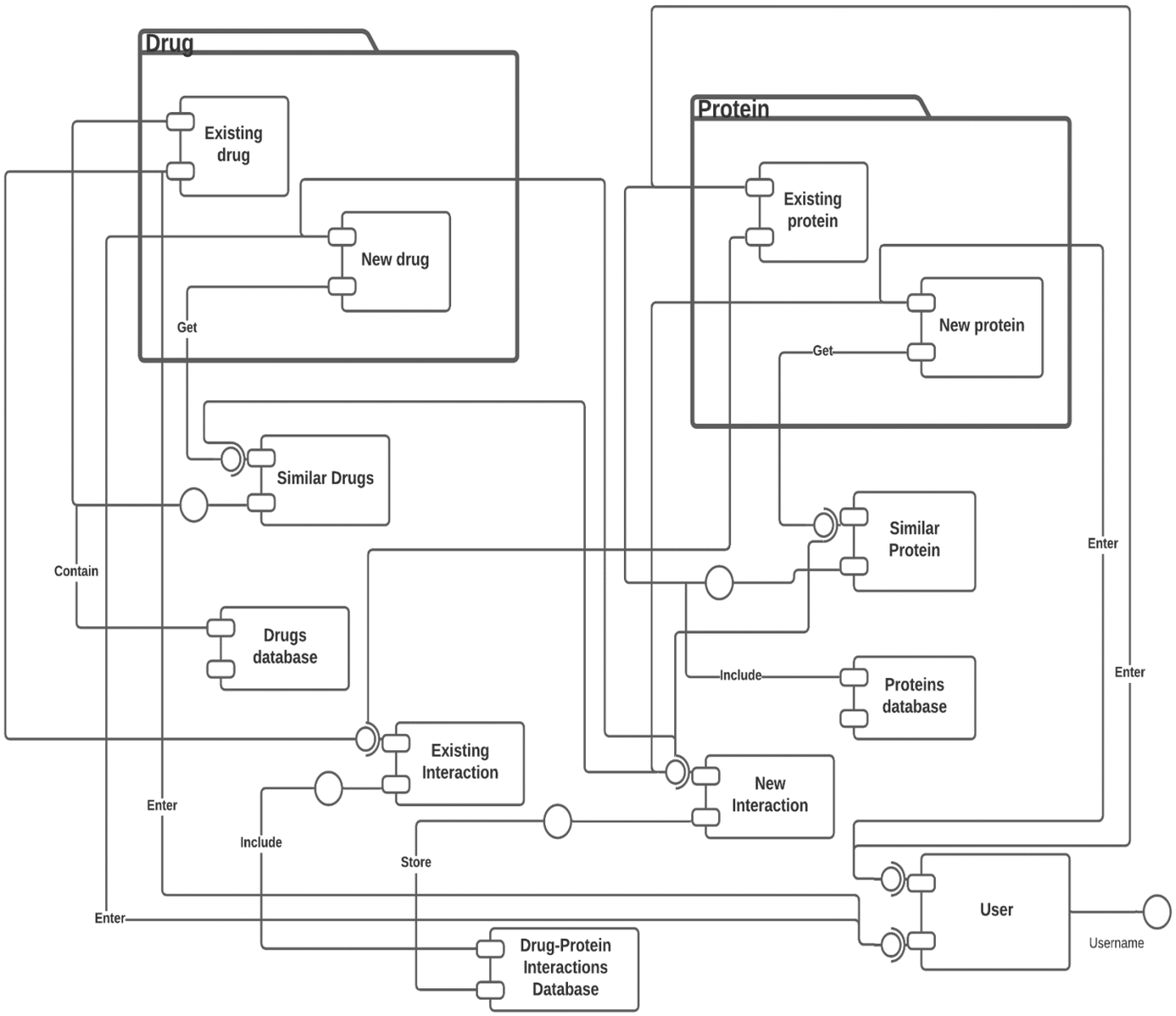


Figure 19- Component Diagram

- Class Diagram

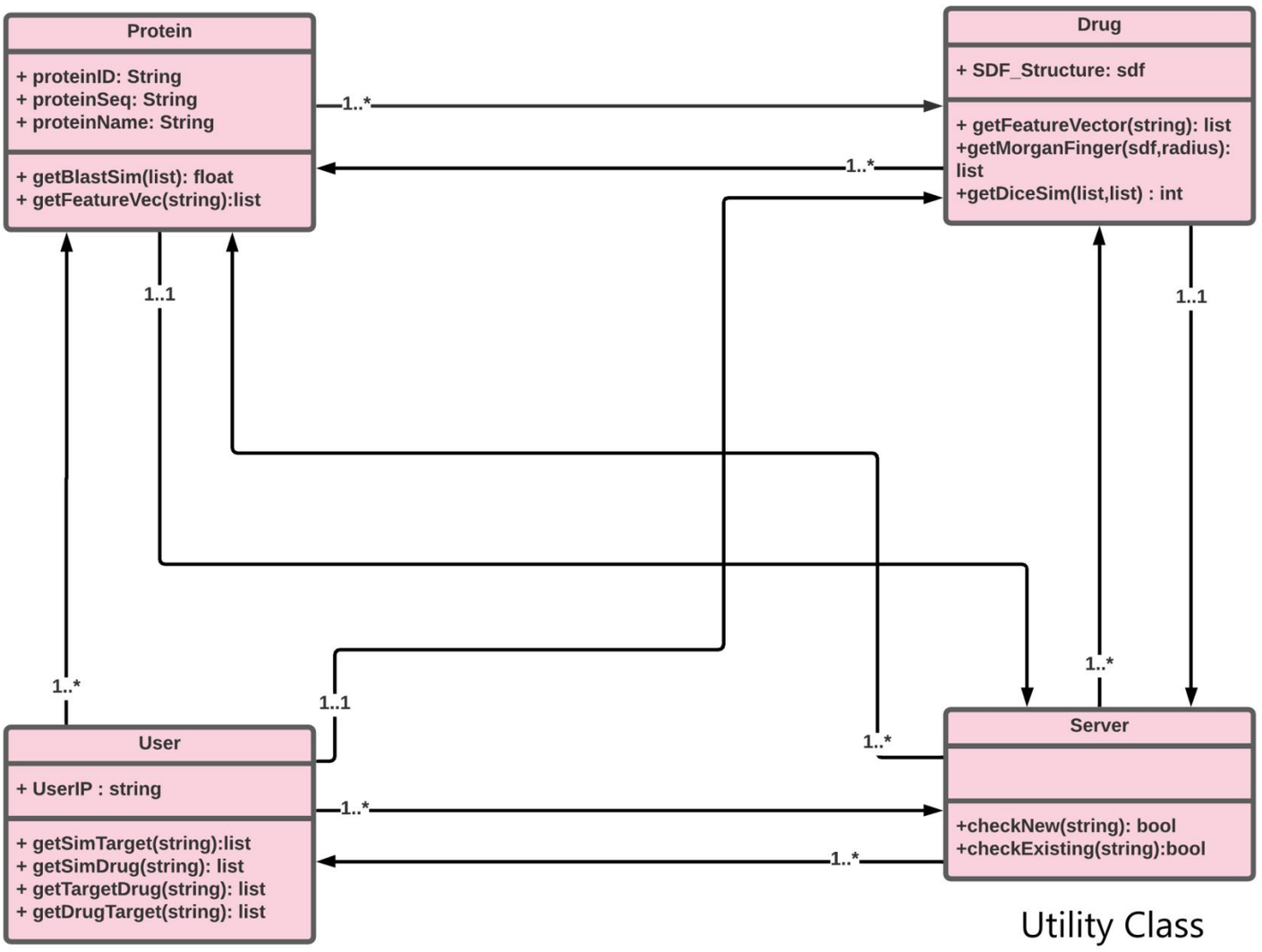


Figure 20- Class Diagram

- Sequence Diagram

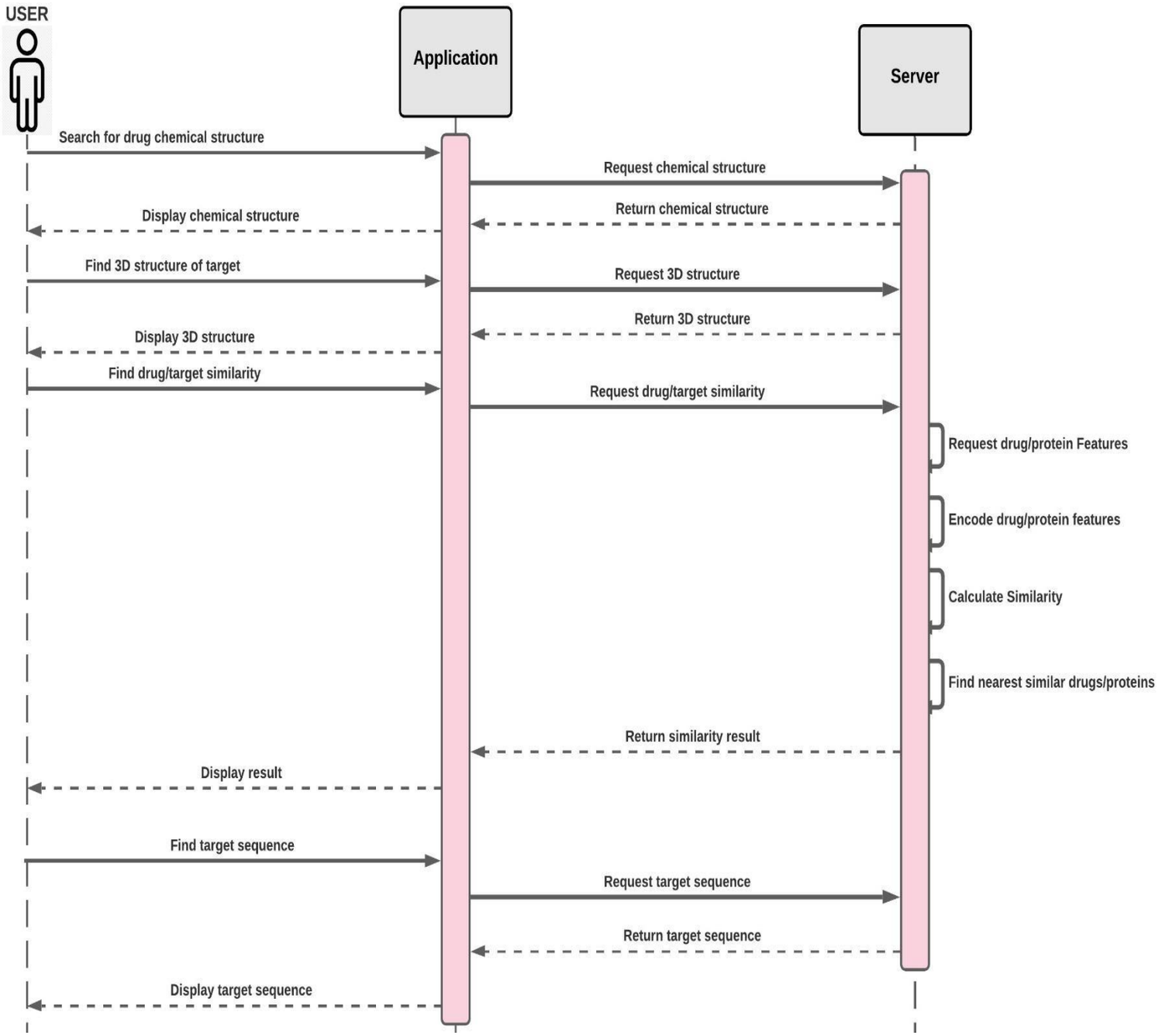


Figure 21- Drug-Drug similarity / Target-Target similarity

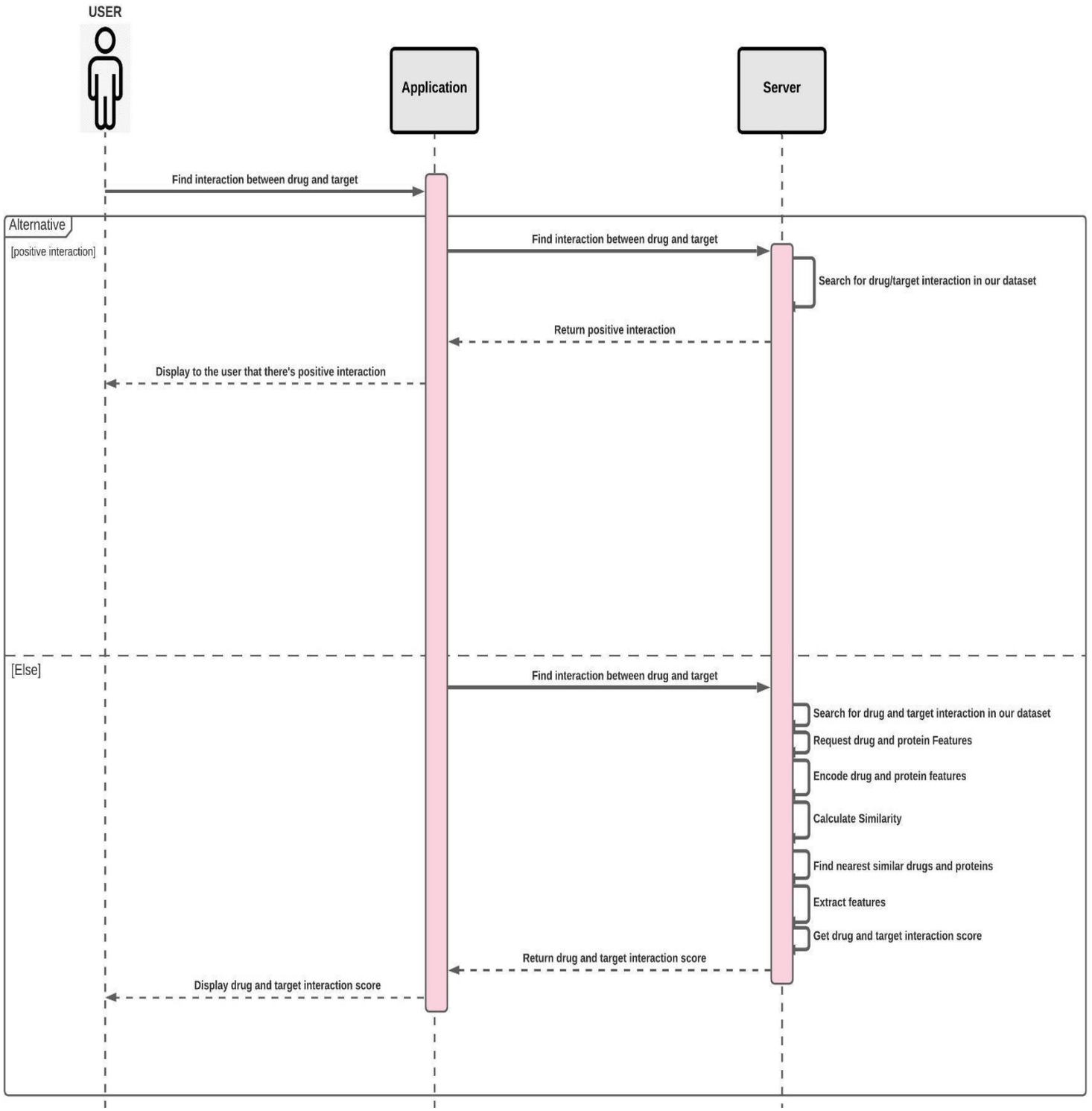


Figure 22- Drug target interaction

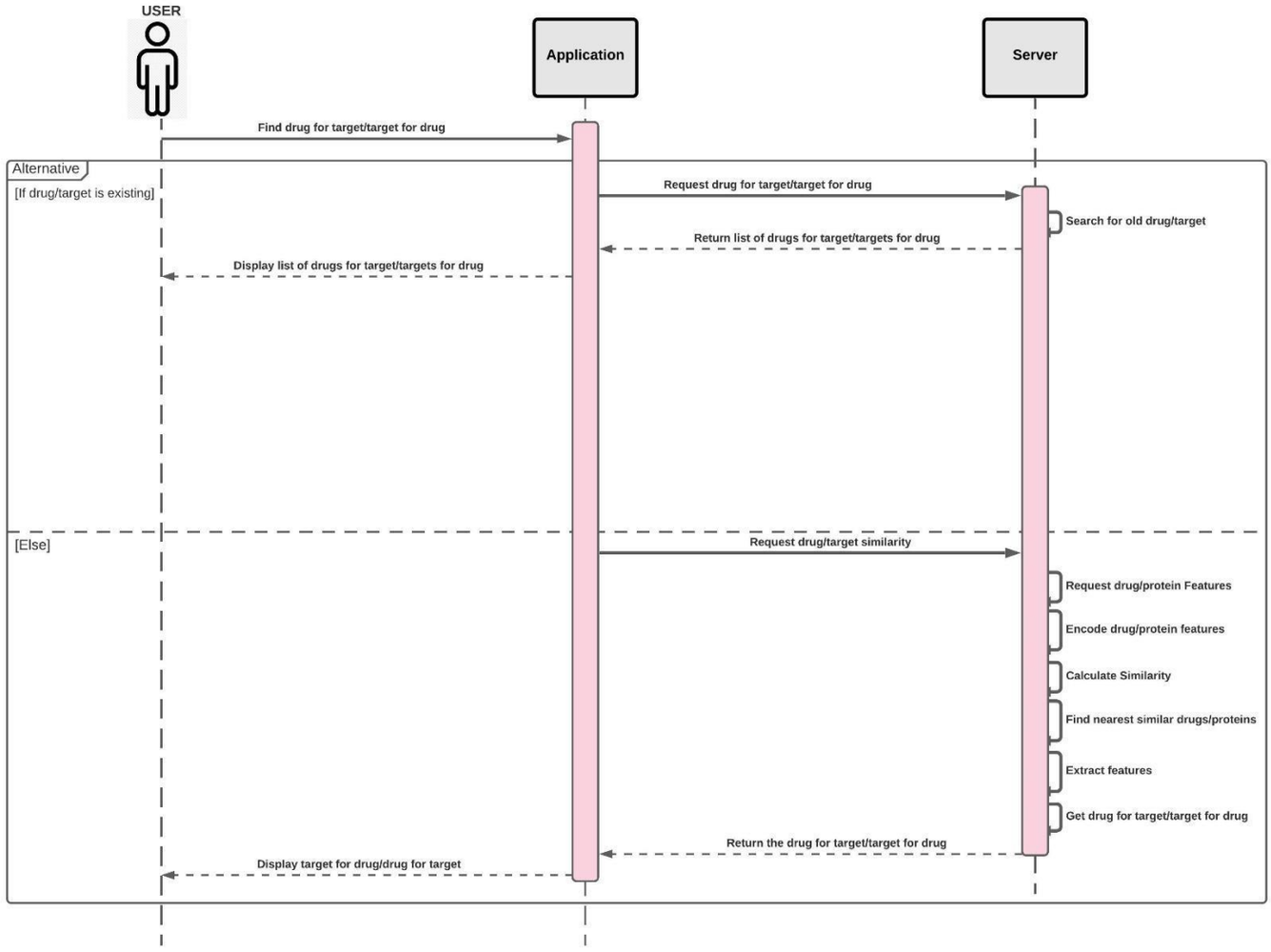


Figure 23- Drug for target / Target for drug

- ERD Diagram

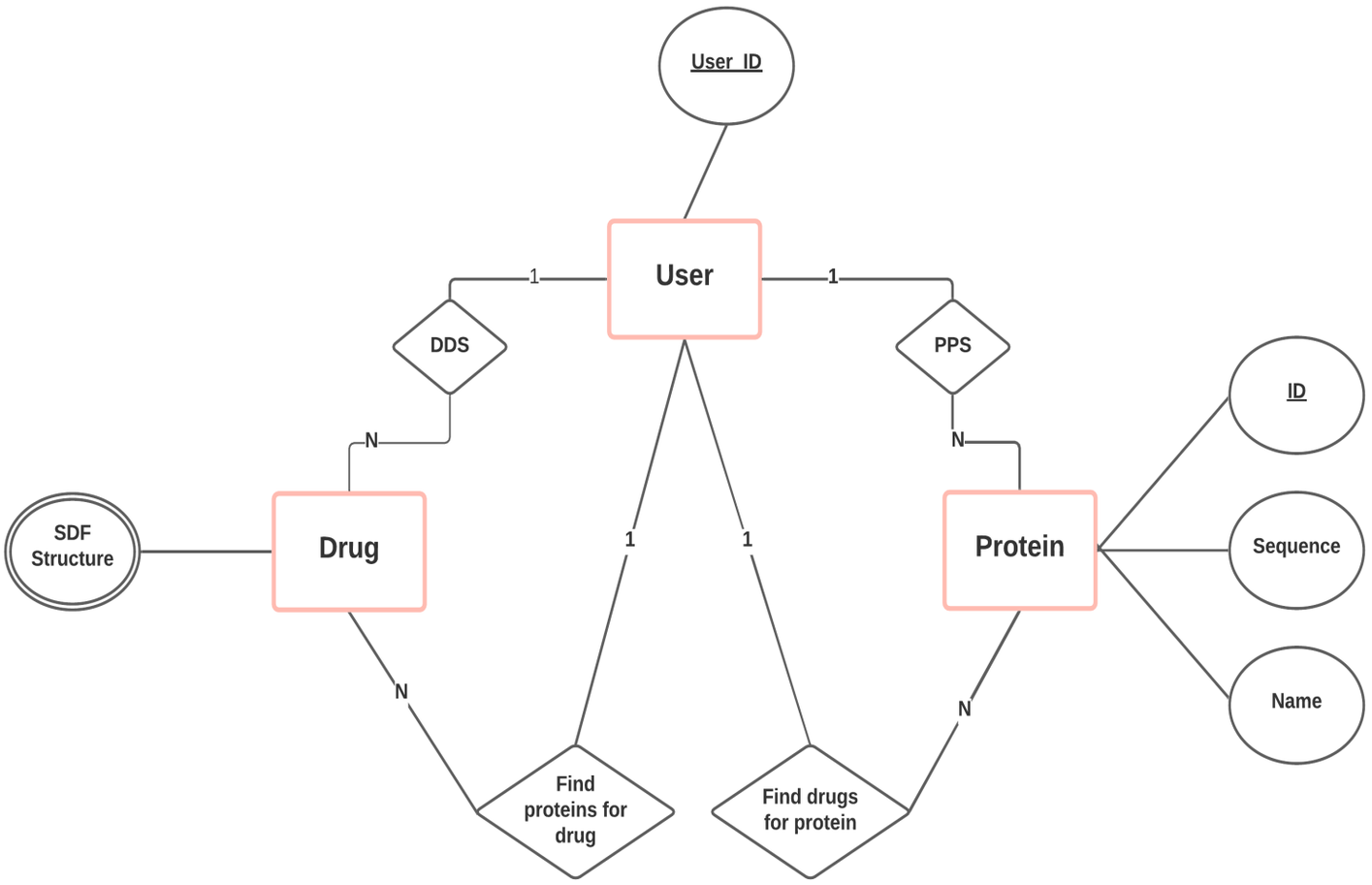


Figure 24- ERD Diagram

- Sequence Diagrams

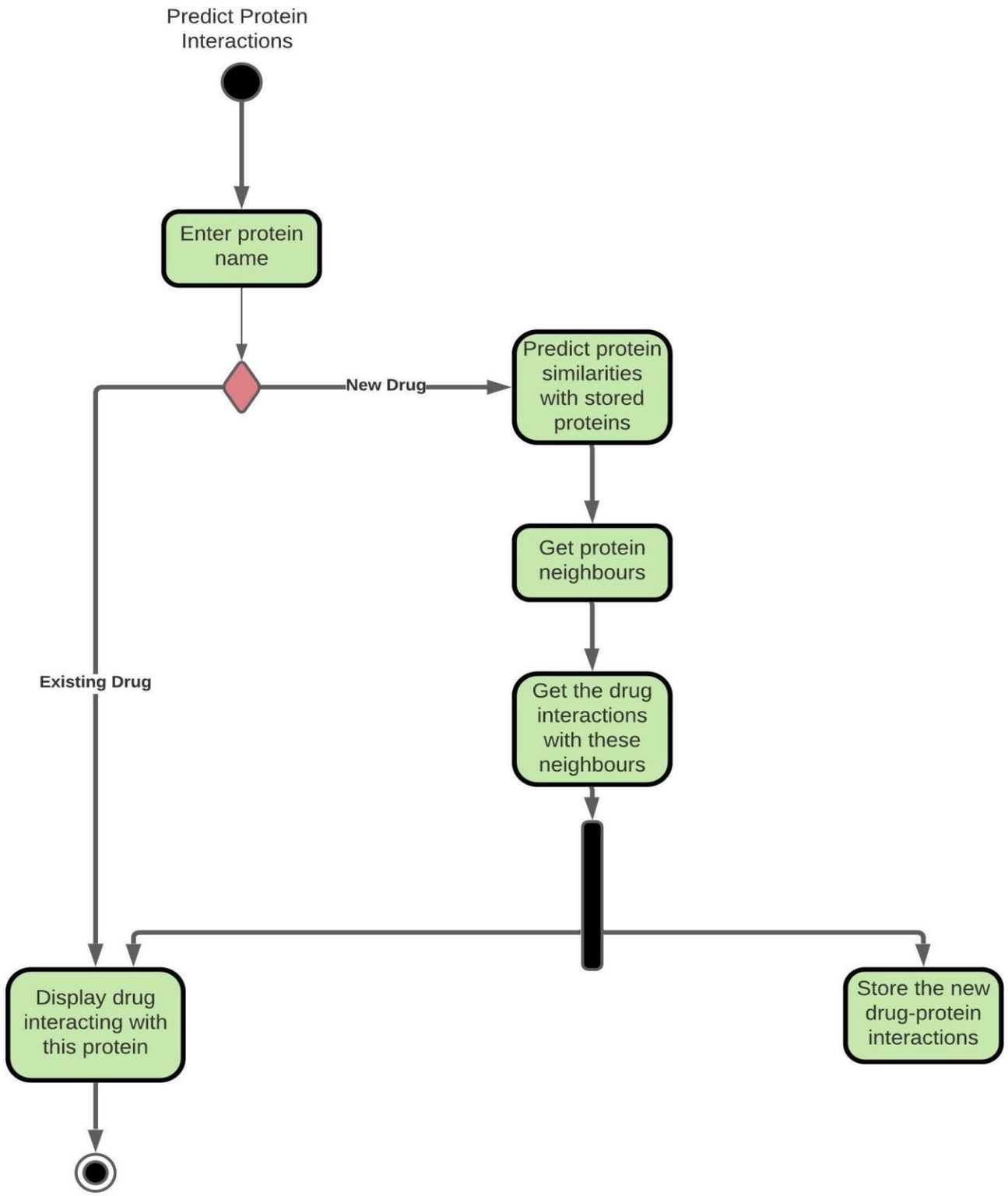


Figure 25- Predict proteins interaction activity diagram

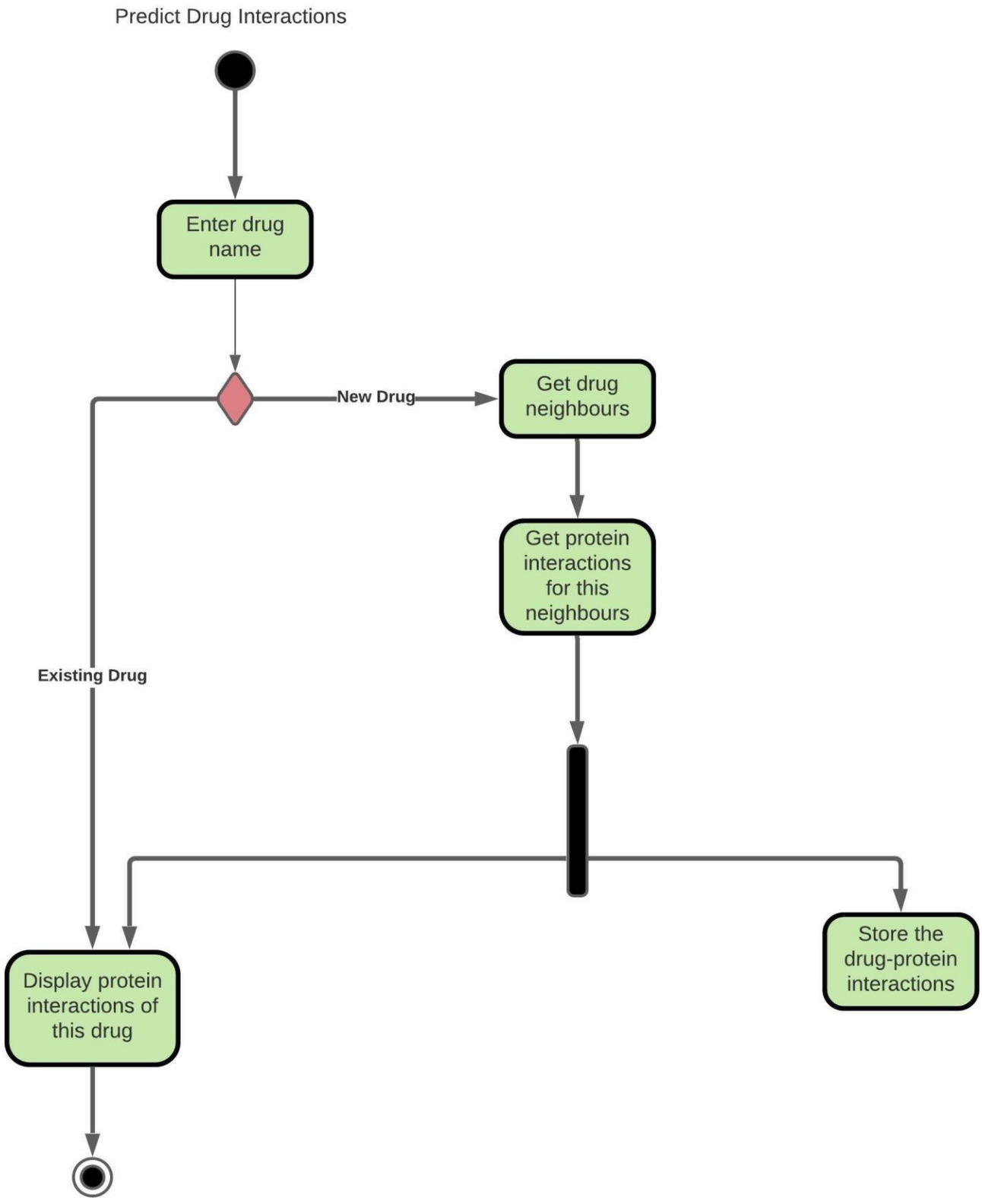


Figure 26- Predict drug interactions activity diagram

- System GUI design

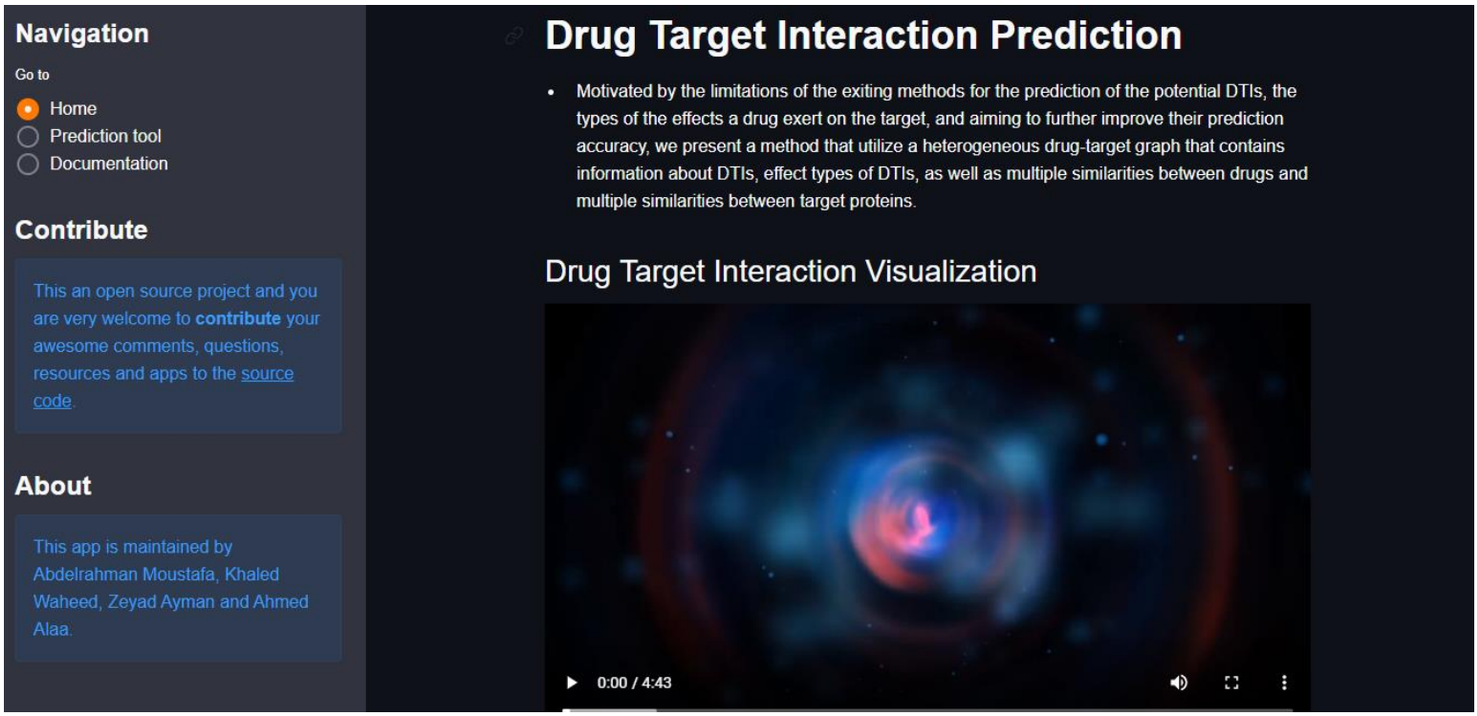


Figure 27- Home Page

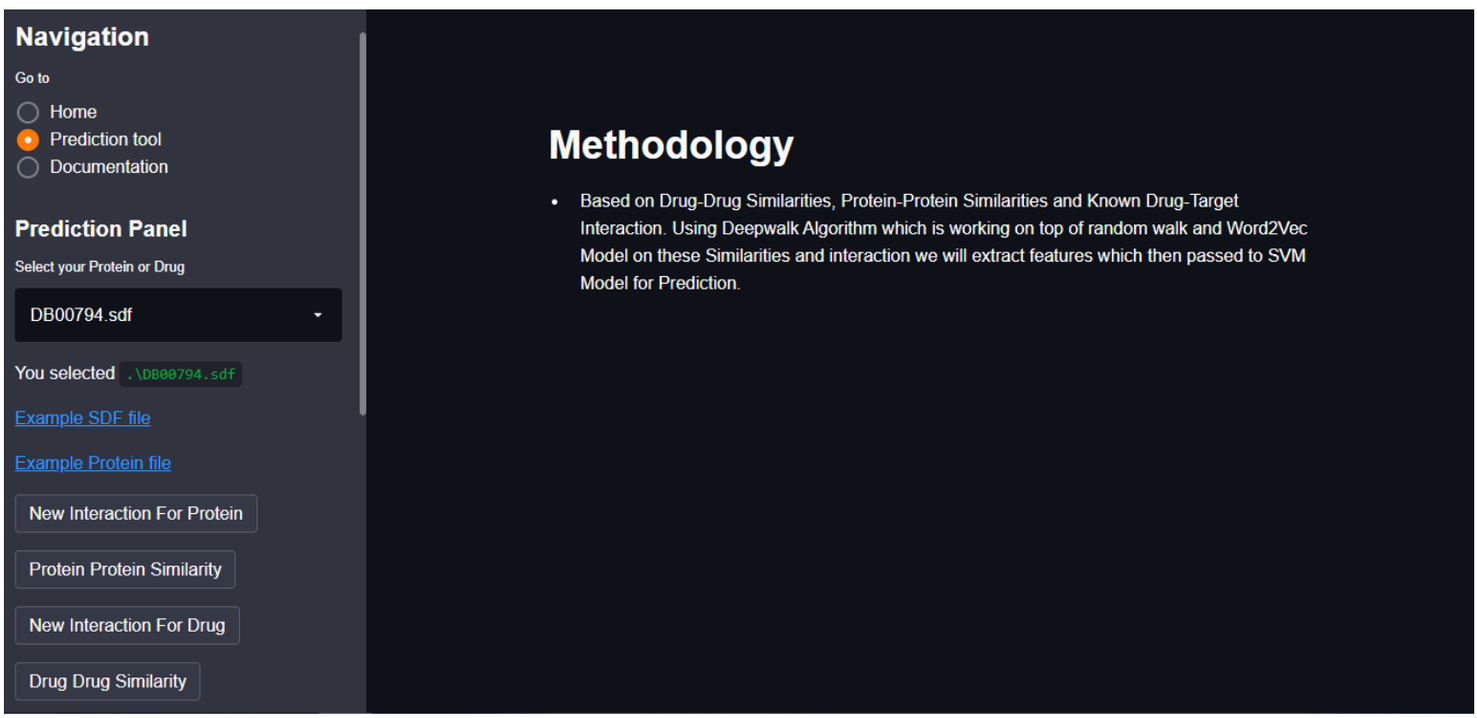


Figure 28- Prediction Tool

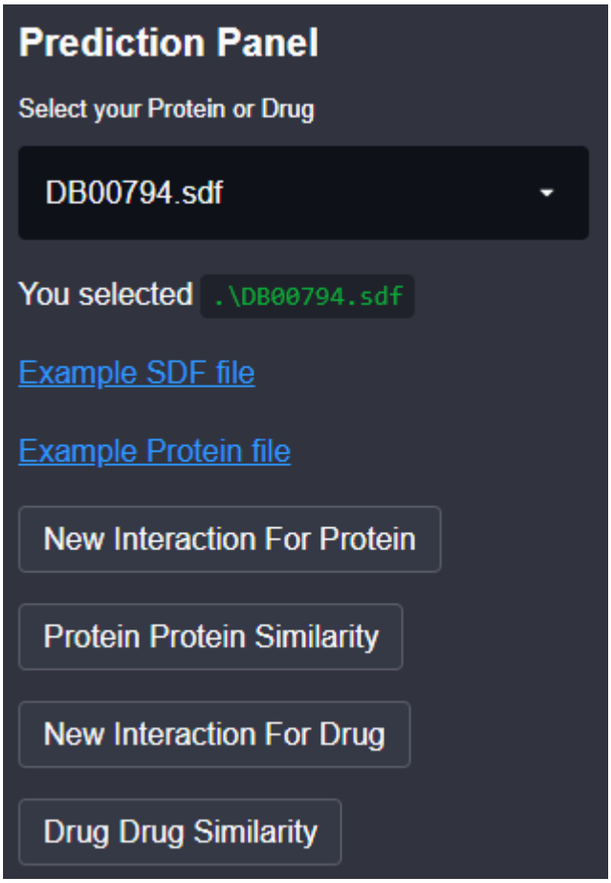
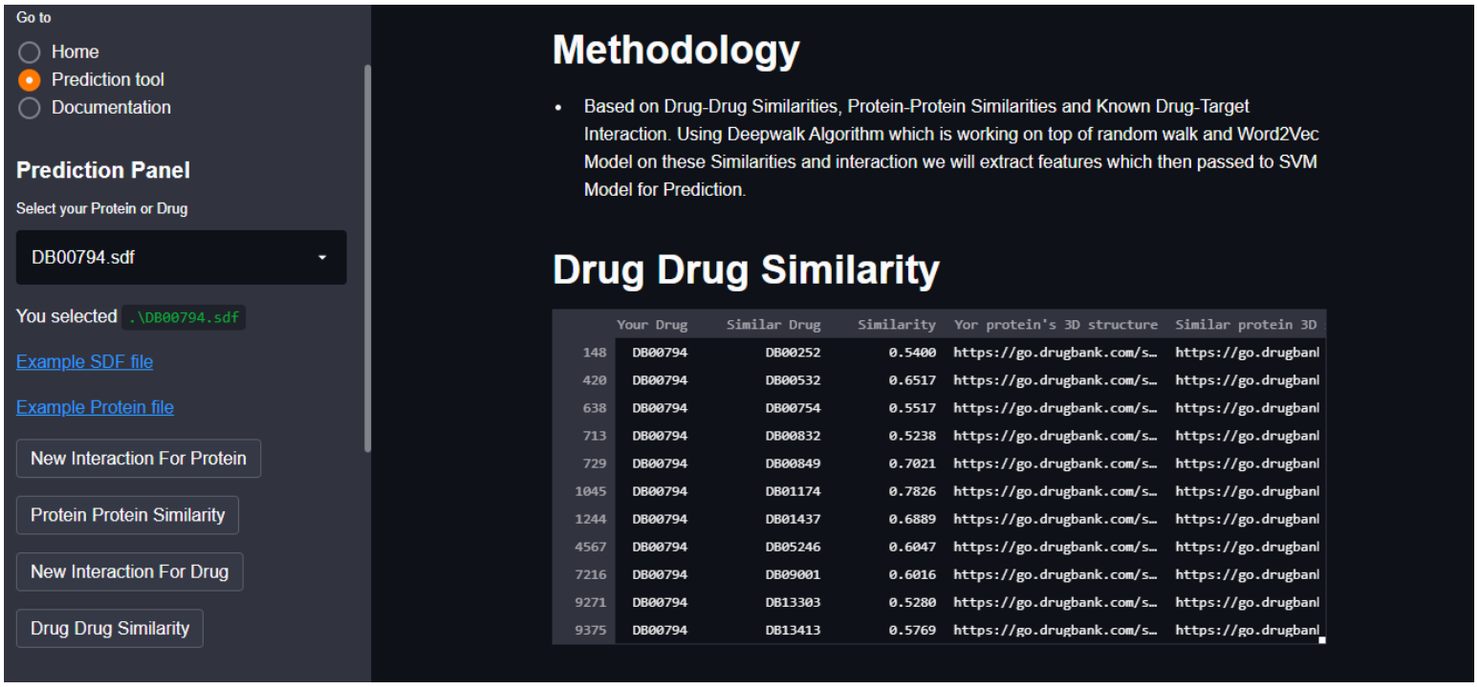


Figure 29-Prediction Panel



	Your Drug	Similar Drug	Similarity	Yor protein's 3D structure	Similar protein 3D
148	DB00794	DB00252	0.5400	https://go.drugbank.com/s…	https://go.drugbanl
420	DB00794	DB00532	0.6517	https://go.drugbank.com/s…	https://go.drugbanl
638	DB00794	DB00754	0.5517	https://go.drugbank.com/s…	https://go.drugbanl
713	DB00794	DB00832	0.5238	https://go.drugbank.com/s…	https://go.drugbanl
729	DB00794	DB00849	0.7021	https://go.drugbank.com/s…	https://go.drugbanl
1045	DB00794	DB01174	0.7826	https://go.drugbank.com/s…	https://go.drugbanl
1244	DB00794	DB01437	0.6889	https://go.drugbank.com/s…	https://go.drugbanl
4567	DB00794	DB05246	0.6047	https://go.drugbank.com/s…	https://go.drugbanl
7216	DB00794	DB09001	0.6016	https://go.drugbank.com/s…	https://go.drugbanl
9271	DB00794	DB13303	0.5280	https://go.drugbank.com/s…	https://go.drugbanl
9375	DB00794	DB13413	0.5769	https://go.drugbank.com/s…	https://go.drugbanl

Figure 30- Drug-Drug test case

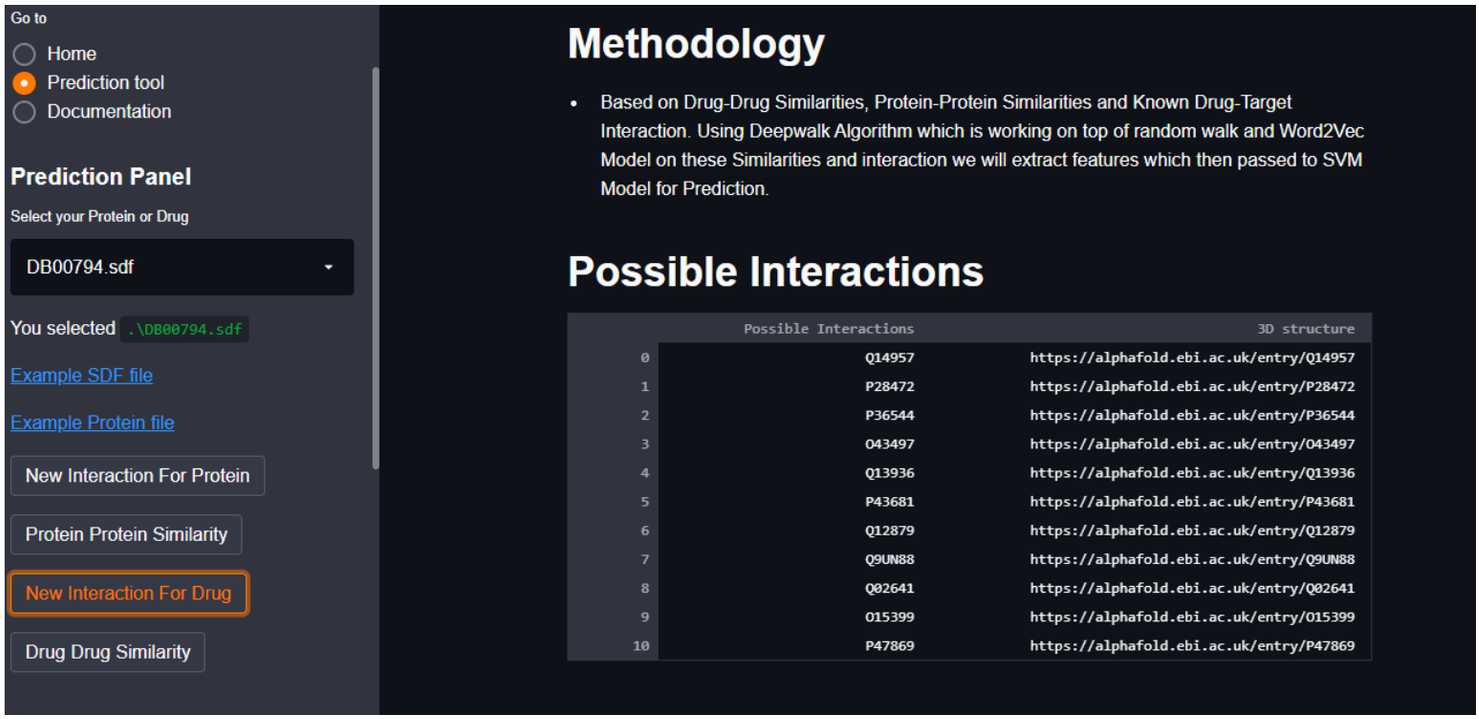


Figure 31- Drug possible interactions test case

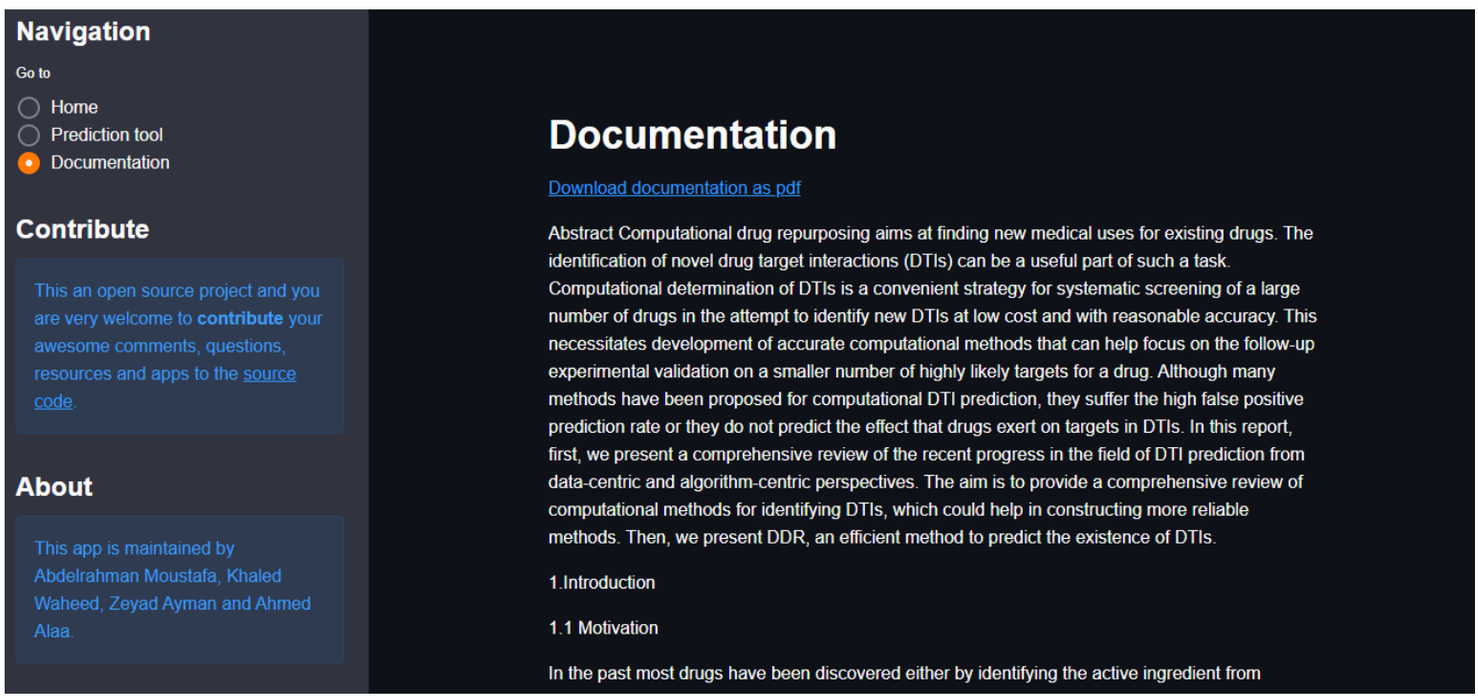


Figure 32- Documentation

Chapter 5

Implementation

In the implementation phase we had 6 main modules which are:

1- Drug module:

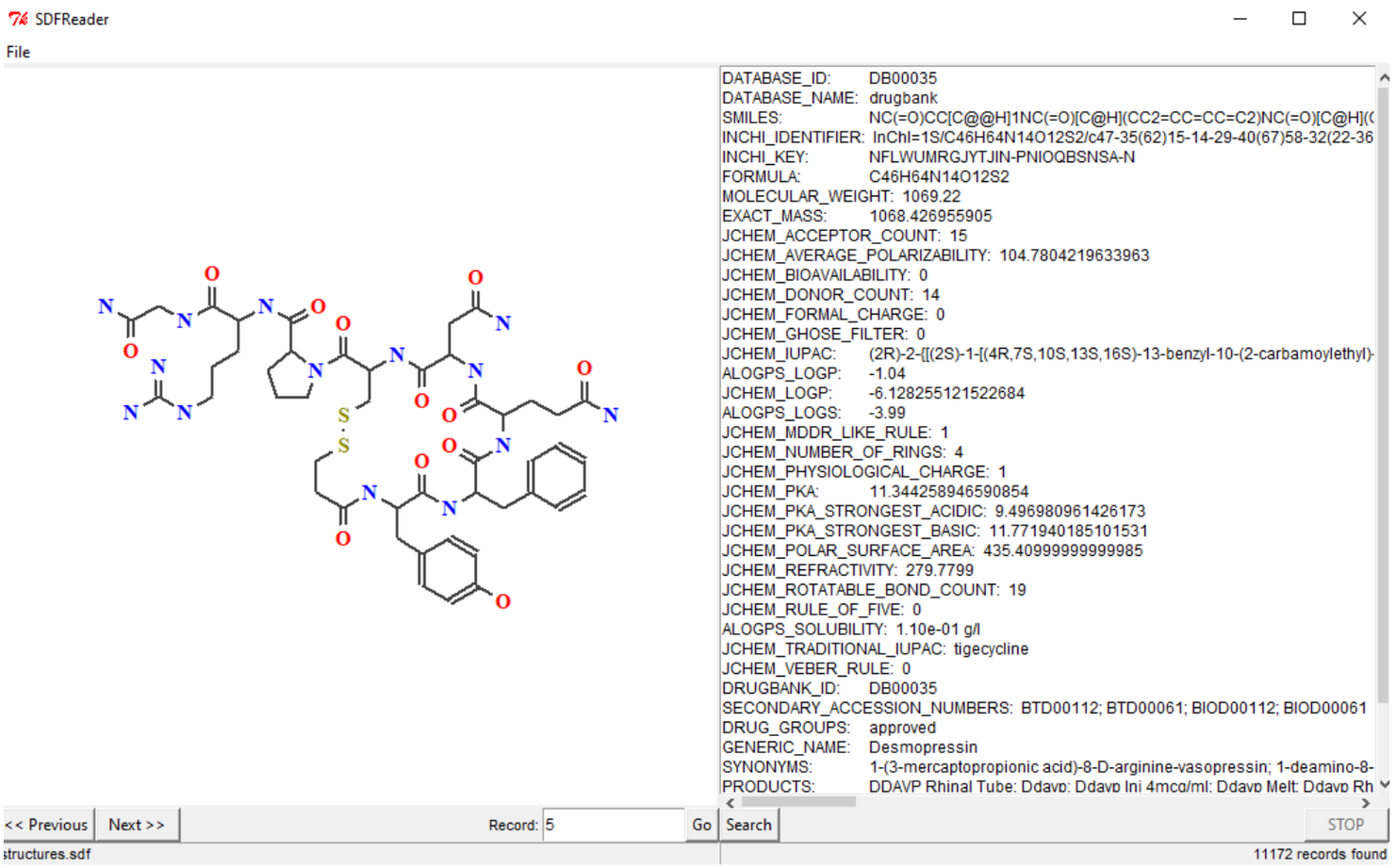


Figure 33- SDF Dataset

We collected dataset for the drugs from DrugBank which was in the SDF format and held 11172 records, we calculated Morgan fingerprint for each molecule in this dataset, then we calculated Dice similarity between each fingerprint, after applying threshold of 0.5 this was the shape of the data after refining :

	compound0	compound1	similarity
467	DB01421	DB04465	0.6207
525	DB01421	DB00479	0.7000
600	DB01421	DB01172	0.7761
718	DB01421	DB15617	0.5098
720	DB01421	DB00581	0.5287
...	...	...	...
1693063	DB00333	DB11071	0.5091
1693358	DB00279	DB01235	0.5859
1693572	DB01179	DB06174	0.5294
1694164	DB01185	DB00769	0.5570
1694609	DB00393	DB00401	0.7179

8324 rows × 3 columns

Figure 34- Drug-Drug Similarity

2- Protein Module:

We collected the protein data from UniProt which was in FASTA file and then converted this data to a CSV file holding 1356 records after applying threshold of 0.5 and this was the shape of FASTA file:

```
>sp|Q6GZX4|001R_FRG3G Putative transcription factor 001R OS=Frog virus 3 (isolate Goorha) OX=654924 GN=FV3-001R PE=4 SV=1
MAFSAEDVLKEYDRRRRMEALLLSLYYPNDRKLLDYKEWSPPRVQVECPKAPVEWNNPPS
EKGLIVGHFSGIKYKGEKAQASEVDVNKMCCWVSKFKDAMRRYQGIQTCKIPGKVLSDLD
AKIKAYNLTVEGVEGFVRYSRVTKQHVAAFLKELRHSKQYENVNLIHYILTDKRVDIQHL
EKDLVKDFKALVESAHRMRQGHMINVKYILYQLLKKHGHGPDGPDILTVKTGSKGVLYDD
SFRKIYTDLGWKFTPL
>sp|Q6GZX3|002L_FRG3G Uncharacterized protein 002L OS=Frog virus 3 (isolate Goorha) OX=654924 GN=FV3-002L PE=4 SV=1
MSIIGATRLQNDKSDTYSAGPCYAGGCSAFTPRGTCGKDWDLGEQTCASGFCTSQPLCAR
IKKTQVCGLRYSSKGKDPLVSAEWDSRGAPYVRCTYDADLIDTQAQVDQFVSMFGESPSL
AERYCMRGVKNTAGELVSRVSSDADPAGGWCRKWYSAHRGPDQDAALGSFCIKNPGAADC
KCINRASDPVYQKVKTLHAYPDQCWYVPCAADVGELKMGTQRDTPTNCPTQVCQIVFNML
DDGSVTMDDVKNTINCDFSKYVPPPPPPKPTPPTPPTPPTPPTPPTPPTPRPVHNRK
VMFFVAGAVLVAILISTVRW
>sp|Q197F8|002R_IIV3 Uncharacterized protein 002R OS=Invertebrate iridescent virus 3 OX=345201 GN=IIV3-002R PE=4 SV=1
MASNTVSAQGGSNRPVRDFSNIQDVAQFLLFDPIWNEQPGSIVPWKMNREQALAERYPEL
QTSEPSEDYSGPVESLELLPLEIKLDIMQYLSWEQISWCKHPWLWTRWYKDNVVRVSAIT
FEDFQREYAFPEKIQEIHFTDTRAEEIKAILETTPNVTRLVIRRIDDMNYNTHGDLGLDD
LEFLTHLMVEDACGFTDFWAPSLTHLTIKNLDMHPRWFGPVMDGIKSMQSTLKYLYIFET
YGVNKPFVQWCTDNIETFYCTNSYRYENVPRPIYVWVLFQEDEWHGYRVEDNKFHRRYMY
STILHKRDTDWVENNPLKTPAQVEMYKFLLRISQLNRDGTGYESDSDPENEHFDDESFSS
GEEDSSDEDDPTWAPDSDDSDWETETEEEPSVAARILEKGKLTITNLMKSLGFKPKPKKI
QSIDRYFCSLDSNYNSEDEDFEYDSDSEDDDSDSEDDC
>sp|Q197F7|003L_IIV3 Uncharacterized protein 003L OS=Invertebrate iridescent virus 3 OX=345201 GN=IIV3-003L PE=4 SV=1
MYQAINPCPQSWYGSPQLEREIVCKMSGAPHYPNYYPVHPNALGGAWFDTSLNARSLTTT
PSLTTCTPPSLAACTPPTSLGMVDSPPHINPPRRIGTLCFDFGSAKSPQRCECVASDRPS
TTSNTAPDTYRLLITNSKTRKNNYGTCRLEPLTYGI
>sp|Q6GZX2|003R_FRG3G Uncharacterized protein 3R OS=Frog virus 3 (isolate Goorha) OX=654924 GN=FV3-003R PE=3 SV=1
MARPLLGKTSSVRRRLESLSACSIFFFLRKFCQKMASLVFLNSPVYQMSNILLTERRQVD
RAMGGSDDDGVMVVALSPSDFKTVLGSALLAVERDMVHVVPKYLQTPGILHDMLVLLTPI
FGEALSVDMSGATDVMVQQIATAGFVDVDPLHSSVSWKDNVSCPVALLAVSNAVRTMMGQ
PCQVTLIIDVGTQNILRDLVNLPVEMSGDLQVMAYTKDPLGKVPAVGVSVFDSGSVQKGD
AHSVGAPDGLVSFHTHPVSSAVELNYHAGWPSNVDMSSLLTMKNLMHVVVAEEGLWTMAR
TLSMQRLTKVLTDAEKDVMRAAAFNLFLPLNELRVMGTKDSNNKSLKTYFEVFETFTIGA
LMKHSGVTPTAFVDRRWLDNTIYHMGFIPWGRDMRFVVEYDLDGTNPFLNTVPTLMSVKR
KAKIQEMFDNMVSRMVTS
>sp|Q6GZX1|004R_FRG3G Uncharacterized protein 004R OS=Frog virus 3 (isolate Goorha) OX=654924 GN=FV3-004R PE=4 SV=1
MNAKYDTDQGVGRMLFLGTIGLAVVVGGLMAYGYYYDGKTPSSGTSFHTASPSFSSRYRY
>sp|Q197F5|005L_IIV3 Uncharacterized protein 005L OS=Invertebrate iridescent virus 3 OX=345201 GN=IIV3-005L PE=3 SV=1
MRYTVLIALQGALLLLLLIDDGQGQSPYPYPGMPCNSSRQCGLGTCVHSRCAHCSSDGTL
CSPEDPTMVWPCCPESSCQLVVGLPSLVNHYNCLPNQCTDSSQCPGGFGCMTRRSKCELC
KADGEACNSPYLDWRKDKECCSGYCHTEARGLEGVCIDPKKIFCTPKNPWQLAPYPPSYH
QPTTLRPPTSLYDSWLMSGFLVKSTTAPSTQEEEDDY
```

Figure 35- FASTA File

And this was the shape of the CSV file after refining:

0	P0A7S9	MARIAGINIPDHKHAVIALTSIYGVGKTRSKAILAAAGIAEDVKIS...
1	P17612	MGNAAAAKKGSEQESVKEFLAKAKEDFLKKWESPAQNTAHLDQFER...
2	Q9BQB4	MQLPLALCLVCLLVHTAFRVVEGQGWQAFKNDATEIIPELGEYPEP...
3	P0CG04	GQPKANPTVTLFPPSSEELQANKATLVCLISDFYPGAVTVAWKADG...
4	Q13938	MQCHRDLALSQALWGWQLSKQSGWAHPSLPHSPLPSTVHSCSWAPP...
...	...	...
2549	Q96LZ3	MGNEASYPAEMCSHFDNDEIKRLGRRFKKLDLDKSGSLSVEEFMSL...
2550	Q9NRX3	MAGASLGARFYRQIKRHPGIIPMIGLICLGMGSAALYLLRLALRSP...
2551	Q13470	MLPEAGSLWLLKLLRDIQLAQFYWPILEELNVTRPEHFDFVKPEDL...
2552	P12259	MFPGCPRLWVLVVLGTSWVGWGSQGTEAAQLRQFYVAAQGISWSYR...
2553	Q99707	MSPALQDLSQPEGLKKTLRDEINAILQKRIMVLDGGMGTMIQREKL...

Figure 36- Protein Dataset

And this was the shape of data after getting all the similarities:

	compound0	compound1	similarity
0	P13647	P02538	0.805085
1	O00469	O60568	0.617886
2	P29323	P54760	0.589573
3	Q8IWU9	P17752	0.677551
4	Q9HB55	P08684	0.765408
...	...	...	...
1351	O00763	Q13085	0.736371
1352	Q9UI33	Q15858	0.556338
1353	Q9UI33	Q99250	0.555112
1354	Q9UI33	Q14524	0.564980
1355	Q9UI33	P35499	0.580610

1356 rows × 3 columns

Figure 37- Protein-Protein Similarity

3- Known DTI Module: We extracted the dataset from DrugBank holding positive-negative interactions and it held 11058 records in this shape:

	DrugBank ID	UniProt ID
0	DB00001	P00734
1	DB00002	P00533
2	DB00002	O75015
3	DB00002	P02745
4	DB00002	P02746
...	...	...
11053	DB15900	Q05320
11054	DB15935	Q9UJM8
11055	DB15982	P03952
11056	DB16019	P07288
11057	DB16385	Q05320

11058 rows × 2 columns

Figure 38- DTI Dataset

4- Graph node embedding Module: From the 3 previous modules, we had 3 edge lists that formed a graph, this graph contained 4864 node with 20450 edge, then we applied DeepWalk algorithm on this graph, DeepWalk was divided into 2 steps which are random walk and word2vec.

This was an example for random walk on our graph:

```
[['DB00001',
  'P00734',
  'DB13152',
  'P03951',
  'DB00100',
  'P38435',
  'DB00142',
  'P43003',
  'O00341',
  'P43004'],
 ['DB00001', 'P00734', 'DB06695'],
 ['DB00001',
  'P00734',
  'DB01593',
  'P01308',
  'DB14487',
  'P02647',
  'DB14548',
  'P22792',
  'DB14533',
  'P78330'],
```

Figure 39- Random Walk

This our word2vec model:

```
INFO - 21:53:38: collected 4864 word types from a corpus of 190699 raw words and 24320 sentences
INFO - 21:53:38: Loading a fresh vocabulary
INFO - 21:53:38: effective_min_count=5 retains 4864 unique words (100% of original 4864, drops 0)
INFO - 21:53:38: effective_min_count=5 leaves 190699 word corpus (100% of original 190699, drops 0)
INFO - 21:53:38: deleting the raw counts dictionary of 4864 items
INFO - 21:53:38: sample=0.001 downsamples 7 most-common words
INFO - 21:53:38: downsampling leaves estimated 188691 word corpus (98.9% of prior 190699)
INFO - 21:53:38: constructing a huffman tree from 4864 words
INFO - 21:53:39: built huffman tree with maximum node depth 15
INFO - 21:53:39: estimated required memory for 4864 words and 64 dimensions: 5895168 bytes
INFO - 21:53:39: resetting layer weights
```

Figure 40- Word2Vec Model

This was the word2vec model training:

```
INFO - 21:55:58: EPOCH - 20 : training on 190699 raw words (188699 effective words) took 0.6s, 322328 effective words/s
INFO - 21:55:58: training on a 3813980 raw words (3773767 effective words) took 11.8s, 318922 effective words/s
```

Figure 41- Word2Vec Training

Most similar proteins for drug or drugs for protein:

```
sims = model.wv.most_similar('O00238', topn=10)  # get other similar words
print(sims)

INFO - 17:32:24: precomputing L2-norms of word weight vectors
[('P36896', 0.900877833366394), ('Q04771', 0.8839342594146729), ('P36894', 0.874199628829956), ('P36897', 0.8690418601036072)
```

Figure 42- Most similar from the graph

From our word2vec model we extracted the following node embedding:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
DB00001	-0.540512	0.092158	0.207670	0.862200	0.027477	0.264244	-0.318120	0.782379	0.528775	-0.555405	-0.564396	-0.962679	0.436289	0.458397	-0.726142
P00734	-0.677274	0.286098	0.491007	0.889328	-0.065398	0.150813	-0.167208	0.490411	0.063003	-0.333332	-0.462789	-1.024350	0.528921	0.857329	-0.593675
DB00002	0.083410	0.493798	0.228755	-0.018127	-0.483175	0.166077	0.235328	1.069189	-0.201625	1.030034	-0.594906	0.946047	1.185776	0.574978	-0.071930
P00533	0.564742	-0.102836	-0.006209	0.391906	-0.673503	0.193609	-0.161899	0.226796	-1.257882	0.453504	-0.093287	1.251748	-0.443452	0.223182	-0.502012
O75015	0.274760	0.589039	0.282539	-0.334547	-0.833501	0.174490	-0.532150	1.200008	0.628423	0.485096	-0.437084	0.142635	0.755268	0.677057	-0.262400
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
DB15900	-0.324031	0.534862	-0.250508	0.259918	0.533255	0.484646	-0.038060	0.074509	0.520602	-0.207474	-0.590616	-0.041843	-0.029671	0.365439	0.159257
DB15935	0.423406	0.329281	-0.452357	-1.471913	-0.177848	0.251424	0.326018	-0.493647	0.302025	-0.705841	0.105838	0.342397	0.264238	0.155618	0.756848
DB15982	-0.223717	0.137481	0.468581	0.185484	0.049291	0.438450	0.272807	0.277051	-0.188923	0.384691	-0.619241	-1.185018	0.967229	-0.231137	0.135561
DB16019	0.224390	-0.127087	-0.308249	0.424206	0.619211	0.067568	-0.931612	-0.107029	-0.575410	0.275161	0.561138	-0.168382	0.856733	0.789602	0.973417
DB16385	-0.510681	0.145890	-0.360575	0.525539	0.341176	0.357840	-0.004002	0.110893	0.492429	0.125604	-0.858085	-0.171388	-0.077886	0.420243	0.024833

4864 rows × 64 columns

Figure 43- Node Embedding

5- Machine learning Module: We joined positive-negative interactions with node embedding to give us the following shape:

	Drug ID	Target ID	Label	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
0	DB00001	P00734	1	-1.217786	0.378256	0.698677	1.751529	-0.037921	0.415056	-0.485328	1.272790	0.591778	-0.888737	-1.027185	-1.987029	0.965209
1	DB00002	P00533	1	0.648152	0.390961	0.222547	0.373779	-1.156678	0.359685	0.073429	1.295985	-1.459507	1.483538	-0.688193	2.197794	0.742324
2	DB00002	O75015	1	0.358170	1.082837	0.511295	-0.352674	-1.316676	0.340566	-0.296822	2.269197	0.426798	1.515130	-1.031990	1.088681	1.941044
3	DB00002	P02745	1	0.293993	1.047876	0.473940	-0.961737	-1.112729	0.094836	0.011162	2.130513	-0.026086	1.914564	-1.080309	1.669392	2.333219
4	DB00002	P02746	1	0.095389	0.923703	0.418403	-0.299447	-0.490171	0.464269	0.482566	1.852560	-0.189535	1.932169	-0.971075	1.177750	1.947225
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
22048	DB01621	P24024	-1	-0.850574	-1.850052	-0.969218	0.115276	0.090584	-0.432322	-0.309347	0.208837	0.227392	0.233711	-0.265583	1.037599	-0.645470
22049	DB00334	P80192	-1	-0.310029	0.367153	0.283497	0.138202	-0.652884	0.122107	0.622828	-0.612768	-0.475770	0.565235	-0.094234	2.191473	-0.228976
22050	DB00600	P07332	-1	0.519640	0.686901	1.051235	0.279311	0.003030	-0.332314	0.342641	-0.742446	-0.185600	-0.050082	-0.577091	1.807565	-1.014483
22051	DB00794	Q9P289	-1	0.708255	0.652213	0.339616	0.405748	-0.525282	-0.553748	0.299562	0.178145	-0.141515	0.467173	-0.516608	1.128142	0.159517
22052	DB00891	P62955	-1	-1.410860	0.193427	0.457322	-0.736913	-0.183554	0.922351	0.261430	-0.599523	-2.004297	0.151853	-0.652541	0.282468	-0.279254

22053 rows × 67 columns

Figure 44- Node Embedding with interactions

After applying SVM model to our final data we got the following training model:

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 45- SVM Model

After testing the SVM on our model we got the following accuracy :

	precision	recall	f1-score	support
-1	0.96	0.93	0.95	3266
1	0.94	0.96	0.95	3352
accuracy			0.95	6618
macro avg	0.95	0.95	0.95	6618
weighted avg	0.95	0.95	0.95	6618

Figure 46- SVM Model Accuracy

6- The final module and also the test cases module is streamlit in which we tried our possible drugs, proteins, protein-protein similarity and drug-drug similarity as follows:

Drug Drug Similarity

	Your Drug	Similar Drug	Similarity	Yor protein's 3D structure	Similar protein
148	DB00794	DB00252	0.5400	https://go.drugbank.com/s…	https://go.drug
420	DB00794	DB00532	0.6517	https://go.drugbank.com/s…	https://go.drug
638	DB00794	DB00754	0.5517	https://go.drugbank.com/s…	https://go.drug
713	DB00794	DB00832	0.5238	https://go.drugbank.com/s…	https://go.drug
729	DB00794	DB00849	0.7021	https://go.drugbank.com/s…	https://go.drug
1045	DB00794	DB01174	0.7826	https://go.drugbank.com/s…	https://go.drug
1244	DB00794	DB01437	0.6889	https://go.drugbank.com/s…	https://go.drug
4567	DB00794	DB05246	0.6047	https://go.drugbank.com/s…	https://go.drug
7216	DB00794	DB09001	0.6016	https://go.drugbank.com/s…	https://go.drug
9271	DB00794	DB13303	0.5280	https://go.drugbank.com/s…	https://go.drug
9375	DB00794	DB13413	0.5769	https://go.drugbank.com/s…	https://go.drug

Figure 47- Test Case 1

Possible Drugs

	Drugs	3D structure
0	DB14487	https://go.drugbank.com/structures/DB14487/image.svg
1	DB14533	https://go.drugbank.com/structures/DB14533/image.svg
2	DB06404	https://go.drugbank.com/structures/DB06404/image.svg
3	DB06689	https://go.drugbank.com/structures/DB06689/image.svg
4	DB01088	https://go.drugbank.com/structures/DB01088/image.svg
5	DB09213	https://go.drugbank.com/structures/DB09213/image.svg
6	DB08818	https://go.drugbank.com/structures/DB08818/image.svg
7	DB14548	https://go.drugbank.com/structures/DB14548/image.svg
8	DB05311	https://go.drugbank.com/structures/DB05311/image.svg
9	DB14597	https://go.drugbank.com/structures/DB14597/image.svg
10	DB01593	https://go.drugbank.com/structures/DB01593/image.svg

Figure 48- Test Case 2

Protein Protein Similarity

	Your Protein	Similar Protein	Similarity	Yor drug's 3D structure	Similar dru
0	P00734	P03952	0.4044	https://alphafold.ebi.ac.…	https://alph
1	P00734	P00750	0.4035	https://alphafold.ebi.ac.…	https://alph
2	P00734	P00748	0.4019	https://alphafold.ebi.ac.…	https://alph
3	P00734	Q14520	0.4035	https://alphafold.ebi.ac.…	https://alph

Figure 49- Test Case 3

Possible Proteins

	Possible Interactions	3D structure
0	P35498	https://alphafold.ebi.ac.uk/entry/P35498
1	P54284	https://alphafold.ebi.ac.uk/entry/P54284
2	O14764	https://alphafold.ebi.ac.uk/entry/O14764
3	P34903	https://alphafold.ebi.ac.uk/entry/P34903
4	P28472	https://alphafold.ebi.ac.uk/entry/P28472
5	Q14957	https://alphafold.ebi.ac.uk/entry/Q14957
6	P31644	https://alphafold.ebi.ac.uk/entry/P31644
7	Q9UN88	https://alphafold.ebi.ac.uk/entry/Q9UN88
8	Q8TCU5	https://alphafold.ebi.ac.uk/entry/Q8TCU5
9	O60840	https://alphafold.ebi.ac.uk/entry/O60840
10	O43497	https://alphafold.ebi.ac.uk/entry/O43497

Figure 50- Test Case 4

Testing

We tried 29 different machine learning model to find that the SVM model gave us the best accuracy from all of these models and here's our 29 models with their accuracies:

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	matthews_corrcoef	Time Taken
SVC	0.95	0.95	0.95	0.95	0.91	18.53
ExtraTreesClassifier	0.94	0.94	0.94	0.93	0.87	3.00
RandomForestClassifier	0.93	0.93	0.93	0.93	0.87	16.15
LGBMClassifier	0.93	0.93	0.93	0.93	0.85	2.40
XGBClassifier	0.92	0.92	0.92	0.92	0.85	12.50
NuSVC	0.91	0.91	0.91	0.91	0.83	41.16
KNeighborsClassifier	0.88	0.88	0.88	0.88	0.77	17.66
BaggingClassifier	0.88	0.88	0.88	0.88	0.75	14.35
DecisionTreeClassifier	0.79	0.79	0.79	0.79	0.58	2.59
AdaBoostClassifier	0.77	0.77	0.77	0.77	0.55	7.76
ExtraTreeClassifier	0.76	0.76	0.76	0.76	0.53	0.10
QuadraticDiscriminantAnalysis	0.75	0.75	0.75	0.75	0.51	0.21
GaussianNB	0.71	0.71	0.71	0.71	0.43	0.09
LinearSVC	0.65	0.65	0.65	0.65	0.30	5.92
LogisticRegression	0.65	0.65	0.65	0.65	0.30	0.14
LinearDiscriminantAnalysis	0.65	0.65	0.65	0.65	0.30	0.29
RidgeClassifier	0.65	0.65	0.65	0.65	0.30	0.15
RidgeClassifierCV	0.65	0.65	0.65	0.65	0.30	0.23
CalibratedClassifierCV	0.65	0.65	0.65	0.65	0.30	22.51
SGDClassifier	0.64	0.64	0.64	0.64	0.28	0.48
BernoulliNB	0.62	0.62	0.62	0.61	0.24	0.12
PassiveAggressiveClassifier	0.61	0.61	0.61	0.61	0.21	0.17
NearestCentroid	0.60	0.60	0.60	0.60	0.21	0.10

Figure 51- Trial Models

And here’s a bar plot showing differences between different models:

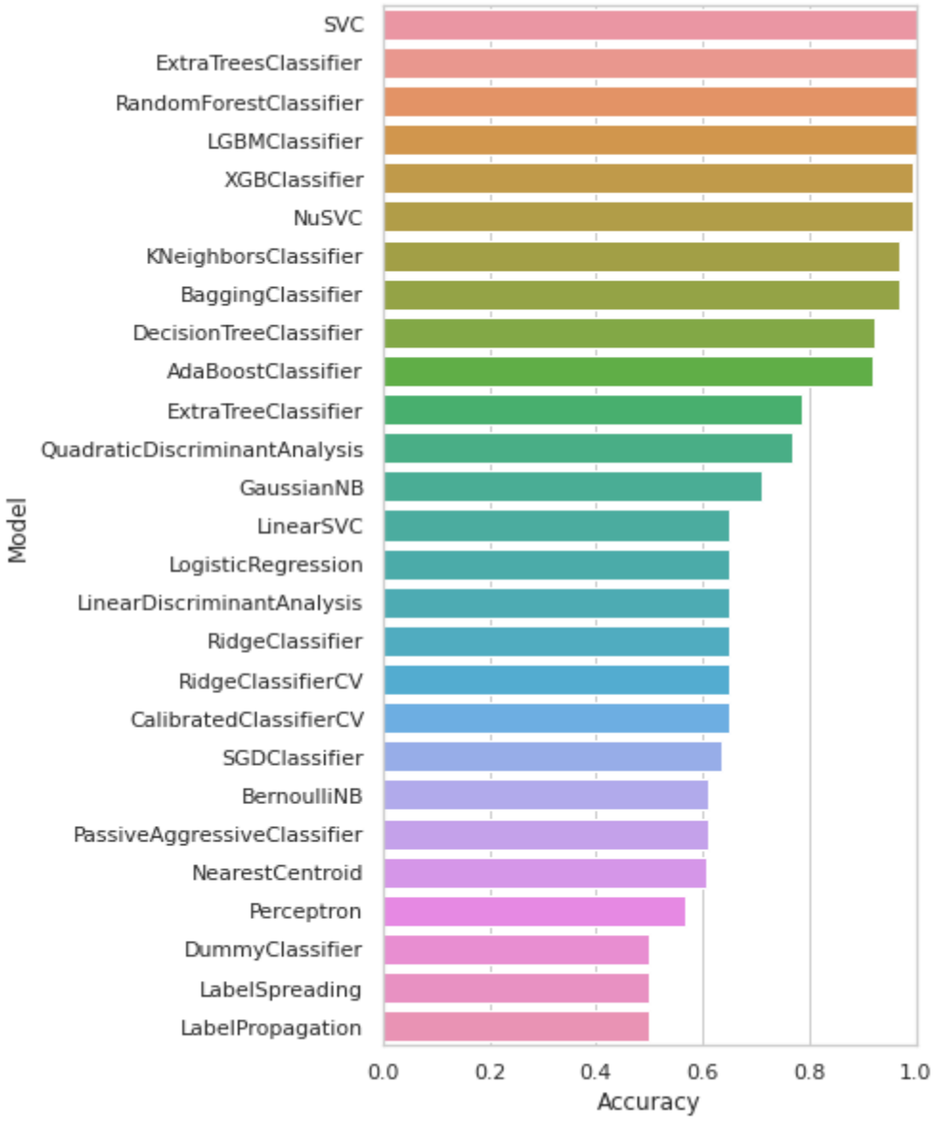


Figure 52- Bar plot for models

References

1- DRUG TARGET PREDICTIONS BASED ON HETEROGENEOUS GRAPH INFERENCE:
(https:/www.ncbi.nlm.nih.gov/pmc/articles/PMC3605000/#S2title)

2- Prediction of Drug–Target Interactions from Multi-Molecular Network Based on Deep Walk Embedding Model:

(https://www.frontiersin.org/articles/10.3389/fbioe.2020.00338/full)

3- A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information:

(https://www.nature.com/articles/s41467-017-00680-8#Sec24)

4- Drug–target interaction prediction by learning from local information and neighbors:

(https://academic.oup.com/bioinformatics/article/29/2/238/203064)

5- Machine learning basics with the k-nearest neighbors algorithm
(https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest neighbors-algorithm-6a6e71d01761)

6- Graph mining
https://www.researchgate.net/publication/228387780_Graph_mining_An_overvie w

7- Random forest algorithm: https://www.javatpoint.com/machine-learning-random-forest-algorithm

9- https://jalammar.github.io/illustrated-word2vec/ (skip gram)

10- https://arxiv.org/pdf/1301.3781.pdf (word2vec)

11- http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/ (softmax function)

13- https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/ (word2vec-skip gram-word embedding)

14- https://datascience.stackexchange.com/questions/51404/word2vec-how-to-choose-the-embedding-size-parameter (word size)

16- https://www.zstream.io/blog/what-is-the-agile-iterative-approach-and-where-is-it-used (iterative methodology)

17- https://www.innovativearchitects.com/KnowledgeCenter/basic-IT-systems/8-SDLC-models.aspx (iterative methodology)

18- https://www.researchgate.net/publication/51202763_PREDICT_A_method_for_inferring_novel_drug_indications_with_application_to_personalized_medicine

19- https://pubs.rsc.org/en/content/articlelanding/2012/mb/c2mb00002d#!divAbstract

20- https://www.researchgate.net/publication/51621103_Gaussian_interaction_profile_kernels_for_predicting_drug-target_interaction