Dissertation on

"Importance of Drug Features in Drug-Drug Interaction: A Comparative Study"

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE20CS461A – Capstone Project Phase - 2

Submitted by:

Chetana N Patil	**PES2UG20CS504**
Naveen B N	**PES2UG20CS217**
Mekala Sanjana	**PES2UG20CS194**
Kumari Shivangi	**PES2UG20CS173**

Under the guidance of

Dr.Prajwala.T .R
Associate Professor
PES University

June - Nov 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'Importance of Drug Features in Drug-Drug Interaction: A Comparative Study'

is a bonafide work carried out by

Chetana N Patil	**PES2UG20CS504**
Naveen B N	**PES2UG20CS217**
Mekala Sanjana	**PES2UG20CS194**
Kumari Shivangi	**PES2UG20CS173**

In partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE20CS461A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period June 2023 – Nov. 2023. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature **Dr.Prajwala.T.R** Associate Professor	Signature Dr. Sandesh B J Chairperson	Signature Dr. B K Keshavan Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **"Importance of Drug features in Drug-Drug Interaction: A Comparative Study"** has been carried out by us under the guidance of Dr.Prajwala T R , Associate Professor and submitted in partial fulfillment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester June – Nov. 2023. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES2UG20CS173	**Kumari Shivangi**
PES2UG20CS194	**Mekala Sanjana**
PES2UG20CS217	**Naveen B N**
PES2UG20CS504	**Chetana N Patil**

ACKNOWLEDGEMENT

I would like to express my gratitude to Dr.Prajwala T R, Associate Professor, Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE20CS461A - Capstone Project Phase – 2.

I am grateful to the Capstone Project Coordinator, Dr.Sarasvathi V, Professor and Dr. Sudeepa Roy Dey, Associate Professor, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Sandesh B J, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof.Jawahar Doreswamy, Pro Chancellor – PES University, Dr.Suryaprasad J, Vice-Chancellor, PES University and Prof.Nagarjuna Sadineni, Pro-Vice Chancellor - PES University, for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

Drug target interaction is used to represent how a drug molecule interacts with a specific biological target, such as a protein or another drug, in order to develop a therapeutic effect. Drug repurposing is the process of identifying new therapeutic uses for existing drugs that were originally developed for a different purpose. It works as an attractive strategy in drug discovery, because of the several advantageous offers it has over traditional drug development, including reduced development time and costs, as well as increased safety and efficacy compared to newly developed drugs. The drug-target interaction is an important event in Drug repurposing as it defines whether the interaction is positive or negative. If an effect is positive then, mostly drugs can be repurposed. If it's negative then, repurposing will be mostly a failure.

In this project, Deep Learning techniques will be used for Drug Repurposing as it can train a model on a large dataset of known drug-target interactions. The model then uses the knowledge of all these interactions of drugs to predict potential drug-target interactions for existing drugs that have not been previously tested. These predictions can then be validated experimentally, and the most promising drug-target interactions can be further evaluated for their potential as new therapies. Additionally, deep learning can be used to predict the potential efficacy and toxicity of drugs, which can help to guide the selection of promising drug candidates for further development.

Thus, Using Deep Learning techniques as an approach for Drug Repurposing can accelerate the drug discovery process and help to bring new therapies to patients more quickly and efficiently.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview:

Drug-Drug interaction is a process in which a drug molecule interacts with another drug molecule. People all across the world consume multiple drugs all day long. Different combinations of drugs are administered-prevalent to underlying diseases or treatments. Combination of few drugs when administered in small doses daily may accumulate side-effects. We therefore need to learn the essence of such interaction and prevent future consequences.

As the world is progressing, there has been a huge upsurge in the number of different diseases and healthcare issues like vitamin deficiency, flues, and other life threatening diseases. Recently, we have faced a huge pandemic caused due to COVID-19 which has shown the need for improvement in the drug discovery process which can be possible using ML models.

Also, according to the stats generated in recent years, at least more than 60% of old age populations and some amount of the working population have health issues such that they have to consume multiple drugs simultaneously. The intake of different drugs all together can lead to side effects which could prove to be life threatening.

1.2 Scope of the Project

Our project has future in the Pharmaceutical industries as it can

- Reduce the time required for new drug discovery.
- Gives an essence for Drug Repurposing.

CHAPTER 2

PROBLEM DEFINITION

2.1 Problem Statement

The project depicts the importance of drug features in the interaction of two drugs when ingested. Drugs interact with each other and can produce synergistic effects (enhance the effect of medication), antagonistic effects (reduce the effect of medication) or neutral.

Each Drug has its own characteristics i.e., features. Some features might relate to its chemical structure whilst others depict its chemical properties. In any Drug - Drug Interaction, properties of metabolism, play an important role in deciding Pharmacokinetics ie, one drug might affect absorption or excretion of another drug ultimately leading to loss of efficacy of ingested drugs or causing adverse side effects. Likewise Drugs with similar structure might take similar metabolic pathways ultimately leading to interactions. Drugs with equivalent binding affinity might interfere with the receptor, increasing or decreasing their concentration in the body .Therefore, it becomes a major source of task to identify underlying causes and provide insights on such problems.

We tend to analyze the importance of these features in the Drug - Drug Interaction. For attaining such we aim to build multiple ML models and make a comparative study for Chemical properties.

CHAPTER 3

LITERATURE SURVEY

3.1 Inferring drug-target interactions based on random walk and convolutional neural network [1]

DTI-Pred is a framework implemented using random walk and CNNs to predict DTIs, The model works by calculating the drug similarity using the data on the drug-drug side effect association. Then, the existing information about various drug-protein interactions and similarities as well as other interactions and similarities are used to create a heterogeneous network. Since, DTI-Pred has CNNs so it provides good feature extraction for the representation of drug and protein. The advantages of the model are discovery of Novel drug–protein interactions. It gives best prediction performance due to the large amount of knowledge on the multiple drug-target interactions. First k candidates are correctly classified as the recall rate is high. But, due to imbalance in class (i.e; the positive example and the negative example) of the training data, the ROC cannot be used.Utilizing data on drug-drug side effect associations, the model calculates drug similarity and constructs a heterogeneous network by integrating various drug-protein interactions and their similarities. CNNs enhance feature extraction, providing a robust representation of drugs and proteins. DTI-Pred excels in discovering novel drug-protein interactions, leveraging extensive knowledge on multiple drug-target interactions for high predictive performance.

It exhibits strong recall rates, effectively classifying top-k candidates.However, challenges arise due to class imbalance in training data, limiting the use of the ROC curve. Despite this, DTI-Pred's predictive accuracy and emphasis on uncovering new interactions underscore its potential for drug discovery and target identification. The integration of random walk and CNNs showcases a thoughtful combination of network-based approaches and deep learning, contributing to its success in capturing complex interactions within heterogeneous biological data.

3.2 DDI-Pred: Graph Convolutional Network-based Drug-Drug Interactions Prediction using Drug Chemical Structure Embedding. [2]

SMILES representation is converted to SELFIES. (Syntactic, semantically correct and valid molecules).SELFIES are converted to Vectors by embedding Techniques. Link in graph says that interaction is present. Take the feature vectors and Existing DDI prediction to give new DDI styles. 4 layers GCN with 16d hidden node feature, Activation Function -RELU, Adam Optimizer and dropout at 10%.Link generator takes 2 nodes and checks whether a link is possible between nodes(drug)and outputs the same.

The advantages are: The values for AUPRC and the AUROC were closer to one which means prediction is good. SELFIES outperform SMILES representation with Accuracy, AUROC and Loss values.t-SNE visualization of the vectors gave an outright synergistic combination of DDI's and model beat other models in DDI predictions; an example would be mol2vec.

The drawbacks are: There is a requirement that more number of hidden layers would have been a bonus for further increase in accuracy. It requires registered DDI's for prediction and if input of DDI network is very sparse, it provides poor results.

In the DDI-Pred model for predicting Drug-Drug Interactions (DDIs), it employs a Graph Convolutional Network (GCN) based on drug chemical structure embedding. SMILES representation is converted to SELFIES, ensuring syntactically and semantically valid molecules. SELFIES are then transformed into vectors using embedding techniques.

The GCN comprises four layers with a 16-dimensional hidden node feature, utilizing ReLU activation, Adam optimizer, and 10% dropout. The model excels with high AUPRC and AUROC values, outperforming SMILES representations. However, drawbacks include the need for more hidden layers for increased accuracy and dependency on registered DDIs, leading to poorer results in sparse DDI networks.

3.3 Graph Convolutional Auto-encoder and Generative Adversarial Network-Based Method for Predicting Drug-Target Interactions [3]

A heterogeneous drug-target network is used to concatenate different drug-target connections, drug-target or drug-target similarities and interactions, and drug-target interactions. A graph convolutional auto-encoder was used to learn network embedding of drug and target nodes in a low-dimensional feature space, and the auto-encoder deeply combined different kinds of connections in the network. A GAN was used to regularize the feature vectors of the nodes into a Gaussian distribution. Severe imbalance between known and unknown DTI. Therefore, we developed an ensemble learning model-based classifier, Light GBM, to estimate drug-target interaction propensities. This classifier made full use of all unknown DTIs and balanced the negative effect of class imbalance. Experimental results have shown that GAN DTI outperforms several state-of-the-art methods for DTI prediction.

The advantages are: By building several decision trees, light GBM may effectively alleviate the effect of class imbalance. It trains distinct trees using a different dataset of negative samples, ensuring that the negative samples are fully utilized.

The limitations are: Complex architecture. Difficulty in handling noise.In the approach to predicting Drug-Target Interactions (DTIs), a heterogeneous drug-target network is employed to amalgamate diverse drug-target connections and similarities. A graph convolutional auto-encoder facilitates learning network embeddings in a low-dimensional feature space, while a Generative Adversarial Network (GAN) regularizes feature vectors into a Gaussian distribution. To address class imbalance, an ensemble learning model-based classifier, Light GBM, is utilized. The GAN DTI method outperforms state-of-the-art DTI prediction methods. Advantages include effective alleviation of class imbalance through decision tree-based Light GBM, but drawbacks include a complex architecture and challenges in handling noise.

3.4 HS-DTI: Drug-target Interaction Prediction Based on Hierarchical Networks and Multi-order Sequence Effect [4]

The drug encoder obtains functional group information from hierarchical learning on the molecule graph constructed from the SMILES sequence. The protein encoder has a convolution block and a multi-order sequence learning block in which the 1st and 2nd order sequence information is assembled to extend the protein encoder and extract different levels of features from the convolution process. Drug functional group information and multi-order protein sequence information are obtained efficiently. Moreover, we found that the HS-DTI model containing these two modules provides the best performance. Both the drug property and the protein property are concatenated and fed to a multilayer perceptron to predict the interaction.

The advantages are: Reduced experimental costs and time. Ability to handle heterogeneous data. HS-DTI seeks to extract numerous properties of drugs and proteins through functional group structures and incorporating multi-line sequence information.

The limitations are: The structure and function of certain functional groups in pharmaceutical preparations and the multiple sequence effect of proteins were not taken into account in this model. Data quality issues: The accuracy of the HS-DTI model is strongly dependent on the quality of the input data. Inaccuracies or biases in the data can lead to incorrect predictions. Limited data availability: The HS-DTI model requires large amounts of data for training, which may be limited in some cases, leading to over-fitting or poor performance.

The HS-DTI model for Drug-Target Interaction Prediction employs a hierarchical approach to gather functional group information from drug molecules and utilizes a protein encoder with a convolution block and a multi-order sequence learning block. By efficiently combining drug functional group details and multi-order protein sequence information, the model achieves optimal performance. It reduces experimental costs and time, handles heterogeneous data, and extracts diverse drug and protein properties. However, limitations include neglecting certain functional group characteristics and protein sequence effects, dependency on data quality, and potential issues with limited data availability.

3.5 A Graph Convolution-Transformer Neural Network for Drug-Target Interaction Prediction [5]

Three different Datasets from Drug Bank. Drugs-SMILES were converted to graphs using an adjacency list (containing atoms and bond) and also maintained feature representation of 9 attributes. (Mol2Vec).Proteins-FASTA sequence was converted into a vector using Word2Vec.Transformer is a Deep learning model encompassing the nature of "self-attention", meaning predicting linkages between nodes (query, key and value vectors).Output of Mol2Vec is passed onto the GCN layer and extracts information, whilst protein sequences undergo 1d convolution and attention mechanism .The outputs namely molecules and protein are transformed/converted to similar dimension with ending a fully connected layer. Lastly, inter-sequence self-attention is provided by the decoder (3 layers) and predicts binary class (present/not).

The advantages include -Mapping attention weights to atoms displaying the model decision mechanism. The model also showed by paying attention to the atoms, which enormously contributed to the interaction's chemical bond formation. Self-attention mechanism along with the GCN transformer was a novel approach.

The disadvantages that follow using the model would be that it did not outperform the Deep DTA model in terms of 2 datasets. Partially extract the molecular information in an interaction. The performance decay on the Kinome imbalanced dataset shows that the dataset can dominantly influence performance. The model focuses on H-bond donors (oxygen, nitrogen).

The Graph Convolution-Transformer Neural Network for Drug-Target Interaction Prediction utilizes three datasets from DrugBank, converting drug SMILES and protein FASTA sequences into graphs and vectors, respectively. The model employs a Transformer with self-attention and a GCN layer, showcasing a unique approach. Advantages include mapping attention weights to atoms, revealing the decision mechanism, while limitations involve performance disparities with specific datasets and focusing on H-bond donors.

3.6 Using Supervised Machine Learning Algorithms For Drug Target Interaction Prediction [6]

This study obtains a low-dimensional vector for graph nodes using the node2vec method. Node2vec framework is used to get low dimensionality for nodes. Different types of supervised learning algorithms have been used to estimate drug-target associations from low-level representations. Multiple machine learning methods have been used to predict drug-target interactions. To evaluate the proposed models, the calculated AUROC and AUPRC values and obtained results indicate that nonlinear SVM and logistic regression performed better than other models with AUROC and AUPRC values of 0.8317 and 0.8260, respectively. A bipartite network is used.

The advantages are: Down sampling was used to try to fix the imbalance in the dataset. The Node2vec framework is used to create small dimensions for a node.

The cons of the model are: It may lead to overfitting. Logistic regression is not suitable for modeling complex relationships between variables, such as interactions or nonlinear effects. Logistic regression is limited to linear relationships.

In the study focusing on Drug Target Interaction Prediction, the node2vec method is employed to obtain low-dimensional vectors for graph nodes, utilizing the framework to achieve dimensionality reduction. Supervised machine learning algorithms, including nonlinear SVM and logistic regression, are applied to estimate drug-target associations from these low-level representations. The models are evaluated using AUROC and AUPRC values, with nonlinear SVM and logistic regression demonstrating superior performance (AUROC of 0.8317 and AUPRC of 0.8260). To address dataset imbalance, down sampling is implemented. However, potential drawbacks include the risk of overfitting, especially when working with smaller datasets. Additionally, while logistic regression excels in simplicity, its limitation to linear relationships might constrain its effectiveness in capturing complex interactions or nonlinear effects, which could impact the model's predictive capabilities in scenarios with intricate drug-target interactions.

3.7 Deep learning in drug design: protein-ligand binding affinity prediction [7]

Deep Atom is a framework to predict the protein-ligand binding affinity. A 3D grid box is used on the protein ligand 3D structure at their binding site so as to get information about it. PC Max algorithm is used to collect information from the binding site surroundings. Light weighted CNN is used as feature extractors to get the useful information regarding the interaction according to the binding affinity labels. The positive aspects that we inferred from the study include, Due to light weighted CNN, the feature extraction was more accurate than other models and in future gives the model ability to be more accurate as the amount of training data increases. Reduced cost and computation even though 3D models are used. No restrictions on the color channels for protein and ligand representations unlike in computer vision (where we have to show RGB).Through training and Overfitting is avoided using a shared weights FC (Fully Connected) layer. Improved performance and reduced variance in prediction. But, the performance is not that good as a limited amount of training data was used. Due to 3D CNNs computation cost and trainable parameters increases. Increased resolution gives more clarity on the interaction but causes a huge computational cost. Therefore, the optimal resolution is used instead of the best solution.

The Deep Atom framework for predicting protein-ligand binding affinity introduces the use of a 3D grid box around the binding site to extract information from the protein-ligand 3D structure. Leveraging the PC Max algorithm for collecting contextual information from the binding site surroundings, a lightweight CNN serves as a feature extractor to enhance interaction information based on binding affinity labels. Positive aspects include the model's accurate feature extraction, especially notable with the potential for increased accuracy as training data expands. The framework also boasts reduced computational costs despite utilizing 3D models, avoiding color channel restrictions present in computer vision. Shared weights in the fully connected layer prevent overfitting, leading to improved performance and reduced prediction variance. However, challenges arise from the limited training data impacting performance, and the use of 3D CNNs increases computational costs, emphasizing the need for optimal resolutions to balance clarity and efficiency in predicting protein-ligand binding affinities.

3.8 Drug target interaction prediction: end-to-end deep learning approach [8]

A model using CNNs to get the 1D representations from protein sequences and compound SMILES strings. The output received can be used in FCNNs for binary classification as CNNs are good at feature detection which gives useful information about patterns or other dependencies. The deep learning model is better than any traditional descriptor for classification of positive as well as negative interactions between the drugs. Thus, good and efficient identification of meaningful regions for the study of drug-target interaction as CNNs feature extraction is good. But, not all deep learning models are that efficient in comparison to traditional models. (like CNN with random forest walk).Also, additional information may prove to be useful to correctly identify positive interactions. As a CNN, auto encoder and FCNN combined model has more information. Hence, more sensitivity than the used model.

The Drug-Target Interaction Prediction model adopts an end-to-end deep learning approach, utilizing Convolutional Neural Networks (CNNs) to extract 1D representations from protein sequences and compound SMILES strings. These representations are then employed in Fully Connected Neural Networks (FCNNs) for binary classification, leveraging CNNs' proficiency in feature detection. The deep learning model demonstrates superiority over traditional descriptors in classifying both positive and negative drug interactions. CNNs excel in extracting meaningful features for studying drug-target interactions, enhancing the identification of relevant regions. However, the study acknowledges that not all deep learning models surpass traditional ones; for instance, a combination of CNNs with random forest walk may offer an alternative efficient approach. Additionally, the model's performance could benefit from incorporating supplementary information to enhance the identification of positive interactions. Combining CNNs, autoencoders, and FCNNs in a unified model provides increased information and sensitivity, potentially improving the overall predictive capability.

The Drug-Target Interaction Prediction model employs CNNs for 1D representations, outperforming traditional descriptors. Additional information and model combinations enhance sensitivity.

3.9 Drug-Target Interaction Prediction Based on Multisource Information Weighted Fusion [9]

Most present research presume that there are no known interactions between medicines and targets. Un-labelled samples are defined in this paper as those for which the link between medicines and targets has not been determined. Three approaches are used to screen un-labelled samples: drug similarity, random walk with restart, and WNN-GIP. Finally, the training set's interaction matrix is changed based on the fusion results, and the BLM-NII model is used to forecast interactions. Enzymes (Es), ion channels (ICs), G-protein-coupled receptors (GPCRs), and nuclear receptors (NRs) are all represented in the databases. To demonstrate the validity of the multisource information fusion-revised dataset in the proposed technique is by comparing accuracy, sensitivity, specificity, and precision to the BLM-NII method.

The advantages are: Experiments show that the suggested method has significantly higher specificity, sensitivity, precision, and accuracy than the BLM-NII method. DTI prediction concerns, samples with uncertain labels are frequently viewed as negative samples, influencing the results and imposing certain constraints. The weighted fusion method improves the efficacy and dependability of screening outcomes.

The disadvantages are: It performs better in datasets with more samples, but its generalization ability deteriorates in datasets with fewer examples. The prediction accuracy needs to be increased, especially for datasets with fewer positive results.

The Drug-Target Interaction Prediction method integrates multisource information through three screening approaches—drug similarity, random walk with restart, and WNN-GIP—to identify unlabelled samples. The BLM-NII model is then utilized for interaction prediction, demonstrating superior specificity, sensitivity, precision, and accuracy compared to conventional methods. While the approach performs well in datasets with ample samples, generalization weakens in datasets with fewer examples, urging a focus on improving accuracy, particularly for datasets with sparse positive results.

3.10 Drug-drug interaction extraction from biomedical texts based on multi-attention mechanism [10]

In this model again we look at attention mechanisms to calculate import scores on the neural network. Layer called BI-LSTM performs conversion of each word into multidimensional vectors and then to learn meaningful and useful sentences or creating their representation is done through Bi directional LSTM layers. As we know in the attention layer we calculate the contribution of each particular word against all others. The use of [CLS] tags provokes reduction of training time and performance improvement.

If we look at key takeaways from the article we get to see that the model allows us to calculate the attention weights in the neural network, which can be utilized to measure the contributions of different words while the model makes decisions. Key features include classified tag words, named as [CLS] used to learn the global information for DDI classification, experimental results have shown that our model has a competitive advantage in extracting the relation between the two candidate drugs in one instance.

The Drug-Drug Interaction Extraction model employs a multi-attention mechanism, utilizing a BI-LSTM layer to convert words into multidimensional vectors and create meaningful sentence representations. The attention layer calculates the importance of each word in relation to others. Notably, the use of [CLS] tags enhances training efficiency and performance. The model's key feature lies in its ability to calculate attention weights, aiding in measuring word contributions during decision-making. The [CLS] tags, designed for learning global information, contribute to the competitive advantage of the model in extracting relationships between candidate drugs. Experimental results affirm its efficacy in drug-drug interaction classification, demonstrating a notable competitive edge.The Drug-Drug Interaction Extraction model employs a multi-attention mechanism with BI-LSTM layers, calculating word contributions using [CLS] tags. This approach enhances training efficiency, and experimental results show the model's competitive advantage in accurately extracting drug-drug interaction relationships.

3.11 Drug-target interaction prediction using Multi Graph Regularized Nuclear Norm Minimization [11]

Multi-Graph Regularized Nuclear Norm Minimization, a new technology, predicts drug-target interaction networks utilizing three inputs: known drug-target interaction networks, similarities across drugs, and those across targets. To capture the proximities, numerous drug-drug and target-target similarities are used as multiple graph Laplacian (across drugs/targets) regularization terms. The four newly incorporated similarities computed from the interaction matrix are Cosine similarity, Pearson's linear correlation, Hamming similarity, and Jaccard similarity. They used Singular value shrinkage and Multi Graph regularized Nuclear Norm Minimization, both of which are efficient solvers for Nuclear Norm minimization. To compare our method to five other state-of-the-art methods (WGRMF, CMF, RLS_WNN, NRLMF, TMF), they employed three separate cross-validation settings: CVS1 (random drug-target combinations excluded), CVS2 (entire drug profile excluded), and CVS3 (entire target profile omitted).

The advantages are: The multi graph regularized version of Nuclear norm minimization tries to prevent over-fitting and considerably improve generalization capabilities. The suggested method focuses on constructing a low-rank interaction matrix based on the proximities of medicines and targets encoded by graphs.

The limitations are: The issue with traditional nuclear norm minimization (NNM) is that it does not support linked information such as Similarity matrices for Drugs and Targets.

The Drug-Target Interaction Prediction model employs Multi Graph Regularized Nuclear Norm Minimization, utilizing known drug-target interactions, drug similarities, and target similarities. Multiple graph Laplacian regularization terms capture proximities, incorporating novel similarities like Cosine, Pearson's correlation, Hamming, and Jaccard. Efficient solvers like Singular value shrinkage and Multi Graph Regularized Nuclear Norm Minimization prevent overfitting, enhancing generalization. The model constructs a low-rank interaction matrix based on graph-encoded drug and target proximities.

3.12 Predicting Drug-Drug Interactions Using Deep Neural Networks. [12]

This paper includes a model which would predict about 80 different Drug-Drug Interaction patterns using deep neural networks as its model. Information pertaining to the structure of drugs was taken from the Drug bank. Drug description (descriptors) of drugs is taken through Chemical databases.3,126,838 drug-drug pair interaction was refined and added 2216 additional features to existing ones and predicted interaction (4432). 4 hidden layers, Activation function -RELU, Loss-Categorical Cross Entropy. Mini batch size of 256 to improve performance. Dropouts of 20% (randomly) to improve accuracy.

If we were to look at the positive aspects of the paper we find that the performance obtained from this model is relatively good with values of Area under the Curve (AUC) -93.2% and for test case (set) was 94%.The estimation was applied on IBD drugs which showed positive synergistic effect. Showed a significant 5% rise in accuracy compared to SVM implementation. Increased antipsychotic activities.

Paper also has few key points of failure which include that the ROC curve of certain DDI types did not show optimized results due to fewer samples(smaller dataset -5,134 drugs and more hidden layers would have been a bonus for further increase in accuracy.

The paper introduces a deep neural network model predicting 80 Drug-Drug Interaction patterns using DrugBank's structural information and chemical descriptors. After refining 3,126,838 drug-drug pair interactions, 2216 features were added, resulting in 4432 predicted interactions. The model, with 4 hidden layers, RELU activation, and categorical cross-entropy loss, exhibited strong performance (AUC 93.2%, test set 94%). Notably, it demonstrated positive synergistic effects in IBD drugs, outperforming SVM with a 5% accuracy increase. However, limitations include suboptimal ROC curves for certain DDI types due to a smaller dataset and potential for further accuracy improvement with more hidden layers.

CHAPTER 4

DATA

4.1 OVERVIEW:

Drug Bank is a comprehensive, publicly available online collection of medication and drug target information. It is a valuable resource for researchers, healthcare professionals, and students in the pharmaceutical and biotech industries. The Drug Bank dataset is huge and consists of information about numerous drug molecules.

Data regarding interaction of Drugs and SMILES were obtained from the link given below,

(https://github.com/haichengyi/DDIPred)[10].

Properties of Drugs were obtained from Kaggle for 8289 Drugs with their associated Drug Bank id given below

(https://www.kaggle.com/code/abdelrahmankhalil/drug-bank-smiles)

	A	B	C	D	E	F	G
1	drugbank_id	cas	logP ALOGPS	logP ChemAxon	solubility ALOGPS	pKa (strongest acidic)	pKa (strongest basic)
2	DB00006	128270-60-0	-0.76	-14	4.64e-02 g/l	2.79	11.88
3	DB00007	53714-56-0	1.04	-2.4	3.38e-02 g/l	9.49	11.92
4	DB00014	65807-02-5	0.3	-5.2	2.83e-02 g/l	9.27	10.82
5	DB00035	16679-58-6	-1	-6.1	1.10e-01 g/l	9.5	11.77
6	DB00050	120287-85-6	1.33	-1.7	6.94e-03 g/l	9.5	11.79
7	DB00067	11000-17-2	-1.4	-7.2	1.24e-01 g/l	7.65	11.5
8	DB00080	103060-53-3	-0.47	-9.4	1.73e-02 g/l	2.98	9.59
9	DB00093	56-59-7	-1.1	-5.8	4.53e-02 g/l	11.39	10.18
10	DB00104	83150-76-9	0.42	-1.4	1.22e-02 g/l	11.4	10.17
11	DB00106	183552-38-7	2.84	-0.46	3.71e-03 g/l	9.47	10.66
12	DB00114	54-47-7	-0.55	-2.1	5.70e+00 g/l	1.68	4.11
13	DB00115	68-19-9	2.66	-3.2	2.02e-02 g/l	1.82	8.68
14	DB00116	135-16-0	-0.96	-4.2	2.69e-01 g/l	3.51	3.58
15	DB00117	71-00-1	-2.7	-3.6	7.13e+01 g/l	1.85	9.44
16	DB00118	29908-03-0	0.05	-5.3	1.62e+00 g/l	1.7	9.41
17	DB00119	127-17-3	-0.38	0.066	1.34e+02 g/l	2.93	-9.6
18	DB00120	63-91-2	-1.4	-1.2	4.14e+00 g/l	2.47	9.45
19	DB00121	58-85-5	0.17	0.32	1.22e+00 g/l	4.4	-1.9

Fig. 4.1: The content of the dataset.

The figure 4.1 represents the properties of drugs which includes pKa acidic and basic values, logP ALOGPS, logP ChemAxon, solubility ALOGPS.

CHAPTER 5

SYSTEM REQUIREMENTS SPECIFICATIONS

5.1 Product Perspective

The following section deeply describes the essential product representatives that are essential to our model.

5.1.1 Product Features

1. Predictive Modeling: The product can look through large amounts of data and levels of information and predict and model the required drug candidates effectively.

2. High Throughput Screening: The obtained interaction prediction is subjected to testing and validation for further use.

3. Personalized Medicine: One can predict based on one's health and genetic data and can be monitored.

4. Good Visualization: Interaction can be visualized using 3d modeling techniques.

5. Downloadable: The predicted interaction can be stored on local storage.

6. Patient-specific recommendations: The ability to generate patient-specific recommendations for managing drug-drug interactions based on the patient's medical history and current medication regimen.

7. Integration with electronic health records: The ability to integrate with electronic health records (EHRs) to access patient data and provide real-time alerts and recommendation.

8. Customizable alerts: A feature that allows users to set alerts for specific drugs or drug combinations that may be of particular concern.

5.1.2 User Classes and Characteristics

1. Industrial User: These are pharmaceutical companies that are used to synthesize new drugs or are involved in a quality driven process.

2. Pharmacists and Research Professionals: Drug Discovery involves a lot of steps from drug design to toxicity prediction. They can use this as a source of information and process them effectively.

3. Diagnostic User: He is responsible for diagnosing a disease or medical condition using drugs.

4. Regulatory agencies: These are governmental organizations responsible for validation and approval of drugs.

5. Prophylactic User: They are to prevent onset of pandemic/epidemic diseases.

5.1.3 Operating Environment

Operating Systems: Windows, Linux, Mac-OS.

Graphics: Nvidia. High Graphics processor /GPU preferred.

Operating Processors: 4.

RAM: 8GB

5.1.4 General Constraints, Assumptions and Dependencies

1. GPU: Calculations and large datasets are used so the processing power should be good.

2. Dataset: Lack of biological and structural data.

3. AI Model: It is an AI model which cannot give perfect accuracy. If an AI model leads to incorrect prediction of target and it may harm patients.

4. Complex to interpret: Being computer science Students, we lack the knowledge of molecules and Pharmacology.

We assume that, Drug reaches the depicted target and interacts with that. Drug combines with the target to provide desired effects. Drug -target interaction speeds up the drug discovery process.

5.1.5 .Risks

1. GPU's are required for faster processing.
2. Resources and Scientific Research pertaining the same are at the bottom.
3. Lack of visualization and protein structure information.
4. Chemical knowledge and libraries are less.

5.2 Functional Requirements

1. The system would be highly interactive and visualizable.
2. It shall take varied levels of drugs, an input (with a wide range of different properties),protein sequences and predict the possibility of interaction.
3. The interaction would be depicted in high dimensional resolution.
4. The interaction model would also depict labels on 3d models of different amino acids and gene sequences.
5. Chemical descriptions would also be provided.
6. The interaction depiction can be easily downloaded.
7. Drug screening would depict further use of drugs for either drug repurposing or new drug synthesis.

5.3. External Interface Requirements

5.3.1 User Interfaces:

Command-Line Interfaces (CLIs): CLIs are text-based interfaces that enable users to interact with the system using command-line input. They are often used by advanced users and can provide more flexibility and control than GUIs, but can be less user-friendly for novice users.

5.3.2 Hardware Requirements

1. GPU for computations.- Nvidia
2. Good Graphics for displaying interaction effectively.
3. Capable RAM size.-8GB
4. Operating Processors: 4 (like 2.20 GHz Intel i7-8750H)

5.3.3 Software Requirements:

5.3.3.1 Operating System

Windows, Linux, Mac-OS

5.3.3.2 Tools and libraries

1) Tkinter: A python library used to create graphical user interface.

2) Numpy: NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

3) Pandas: Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

4) Keras: Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation.

5) Tensorflow: The TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining.

6) Some libraries are imported and used for implementation of machine learning algorithms.

5.4 Non-Functional Requirements

5.4.1 Performance Requirement:

1. Interaction should be deprecated as fast as possible. The latency and delay in the display must be as less as possible.

2. The system should be designed to perform efficiently and effectively, with acceptable response times for all interactions. It should be able to handle large datasets and complex computations without sacrificing performance.

5.4.2 Security Requirements:

1. Data security: The tool must comply with data security and privacy regulations, including secure storage and transmission of patient data.

2. Adverse event reporting: The tool should have a mechanism for healthcare professionals to report adverse events related to drug-drug interactions, which could help improve the tool's accuracy and safety over time.

3. Systematic review: The tool must undergo systematic reviews and evaluations to ensure its safety and effectiveness, which could help identify and address potential safety issues.

5.4.3 Safety requirements:

1. Real-time updates: The tool must be regularly updated with the latest drug information, interactions, and adverse event data to ensure the accuracy of its recommendation.

2. Integration with EHRs: The tool must integrate with EHRs securely to avoid errors and to provide accurate patient-specific recommendations.

5.5 Other Requirements

1. Scalability: It should be able to handle a growing number of interactions and users without experiencing significant degradation in performance.

2. Maintainability: The system should be designed to be easily maintained, upgraded, and enhanced over time. It should be well-documented and designed with modularity and extensibility in mind to facilitate future development and improvements.

3. Usability: The system that is built should have essential key requirements that are simple to use, can be learnt easily, and navigated easily . It must be designed or decorated with the needs and skill level of the user in mind and provide a user-friendly interface that enables efficient drug target interaction.

4. Reliability: The system should be dependable and operate without errors or interruptions. It should be designed to prevent data loss, ensure data integrity, and maintain consistent performance under varying conditions.

5. Interoperability: It will be operable on multiple OS's as researchers use Windows and Linux more So, main focus will be there.

6. Performance: Speed at which the interaction depiction displayed to the user is of utmost importance. Multiple users accessing the site at a time should not impact the performance.

7. Security: Researchers performing critical research can have utmost privacy of their work and will not be made public until they owe to.

8. Reusability: Certain parts of code as interaction can be made up for reuse. We also permit the system to be used up in De novo Drug Design applications if created in future to use up our methodology for Interaction prediction.

9. Application Compatibility: We highly motivate ourselves to use Browser-stack or Cross Browser Testing tools to make our application compatible across different platforms.

CHAPTER 6

SYSTEM DESIGN

6.1 Introduction

Drug target interaction is the process by which a drug molecule interacts with a specific biological target, such as a protein or receptor, in order to elicit a therapeutic effect. Drug repurposing, also known as drug repositioning, is the process of identifying new therapeutic uses for existing drugs that were originally developed for a different purpose. Drug repurposing has become an attractive strategy in drug discovery, as it can offer several advantages over traditional drug development, including reduced development time and costs, as well as increased safety and efficacy compared to newly developed drugs.

In our project, we are going to use a deep learning approach, a type of machine learning that uses neural networks to analyse large-scale data sets. The following documentation will discuss design considerations that include existing designs, design details, and architecture style used and talk about reliability and performance of the system. The general high level design diagram, architecture style diagram, logical data flow diagram, proposed approach diagram, as well as a UI diagram will be discussed.

6.2 Current System

Predicting whether two drugs or drug and protein will interact with each other.We used Random forest algorithm for prediction as other algorithms were having less accuracy.It will consider two drugbank_id's as input and returns whether the interaction is there or not.If interaction found returns 1, else 0 based on drug features, along with the most feature importance.

6.2.1 Decision Tree

A supervised learning method used for classification and regression. A Decision Tree is a tree in which each internal node represents a feature test, each branch a possible test outcome, and

each terminal node a class label. The optimum splitting criteria for each node must be determined at each node, potentially making decision trees computationally expensive. Decision Trees are valuable for visualizing the decision-making process and identifying key factors influencing student performance. Pruning is a method of reducing the size of a decision tree. By restricting the size of the tree. it decreases the possibility of overfitting.

6.2.2 Random Forest

A random forest is a grouping of multiple individual decision trees that work together to create an ensemble model. Each decision tree in the random forest makes a class prediction, and the class with the most votes become the model's prediction.

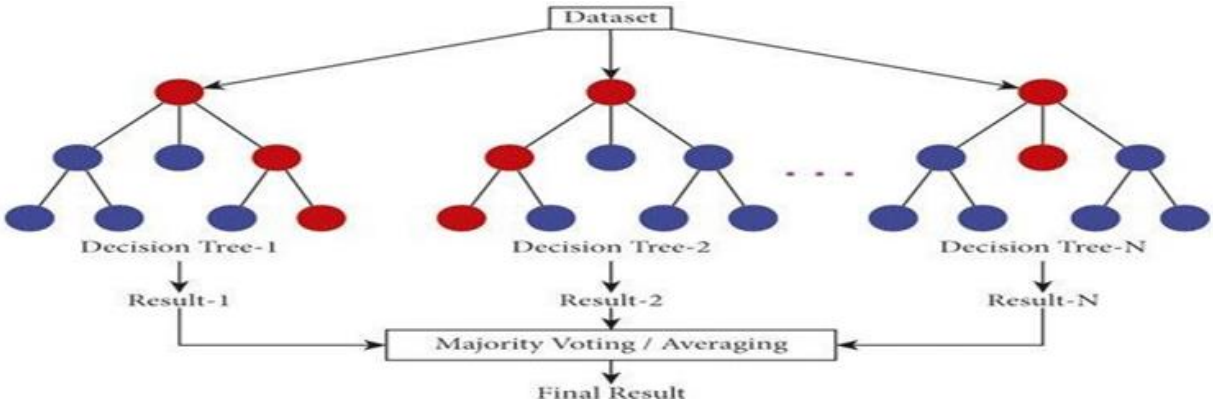


Fig 6.1: Graphical representation of Random forest

Random Forest introduces randomness both in the selection of features for each tree and the data points used to build them. By combining diverse trees, the algorithm provides robust and interpretable predictions, making it well-suited for visualizing the impact of various factors on student performance.

6.3 General Low level Design Diagram

The high level design aims to shed some light on the broad approach taken during the design of our system. Our design consists of a simple but efficient three module system. The basic components consists of Data preprocessing module, Descriptor (Characteristic property of the drug compounds) Derivation module and finally the Machine learning model.

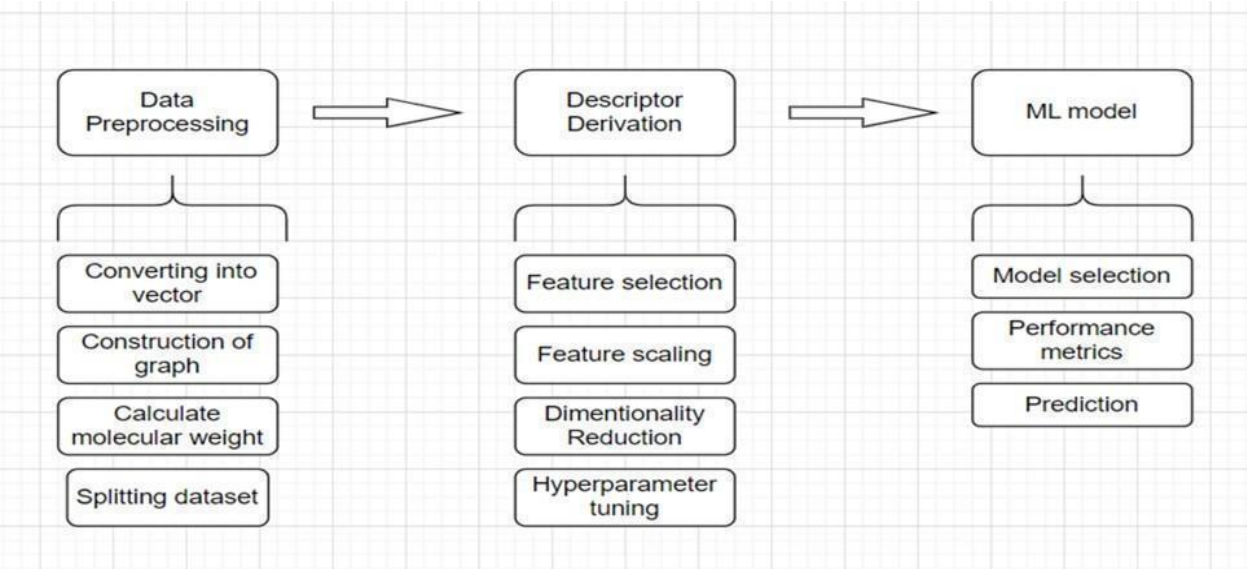


Fig 6.2: An overview of the low level design of the system.

6.3.1 Exploratory Data Analysis(Data Visualization)

It refers to the visual representation of data. It is an effective way of conveying the data when it is numerous. Various plots and graphs such as scatter plots, bar graphs, pie chart, feature importance charts, hyperplane animation, ROC curve and confusion matrix allow us to interpret the variability of the data and uncover patterns.

A Receiver Operating Characteristic (ROC) curve: is a graphical representation of a binary classification model's performance across different threshold settings. It plots the true positive rate against the false positive rate, providing insights into the trade-off between sensitivity and specificity. The area under the ROC curve (AUC) is a common metric to assess the overall performance of a classification model, with a higher AUC indicating better discrimination capability.

Confusion matrix: is a table used in machine learning and statistics to assess the performance of a classification model. It summarizes the results of classification by showing the counts of true positive, true negative, false positive, and false negative predictions.

Hyperplanes: are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features.

6.3.2 Data Pre-Processing

Data pre-processing refers to various activities which prepare the raw dataset and make it suitable for training in a machine learning model. It is an essential step in machine learning. This step ensures that the input data for algorithm visualization is clean and ready for analysis.

6.3.3 Feature Engineering

Feature engineering includes feature selection, feature extraction and column transformations. Relevant features might include previous exam scores, study hours, participation in extracurricular activities, etc.

Feature selection for a ML model is imperative to the final prediction as in this step the highest contributing features are selected based on a metric known as feature importance. Features with low feature importance will inevitably be dropped.

The column transformations can include encoding categorical data into numerical data that the machine learning model will be able to interpret with techniques such as one-hot encoding, binary encoding, target encoding, frequency encoding, label-encoding etc.

6.4 ML Model Development

6.4.1 Train and Test Split

One way to evaluate the performance of machine learning algorithms is to split training and testing.

It can be applied to tasks involving classification, regression, and any supervised learning technique.

This step requires splitting the data set into 80% training subset and 20% testing subset. The training set is the initial subset used to fit the model. The model was not trained on the test subset. Instead, it is fed into an input array of datasets and its predictions are constructed and compared to the expected values. This contrast highlights the model's performance when given previously unseen data.

Train Dataset: To fit machine learning model.
Test Dataset: To evaluate the fit machine learning model.

The premise of the split is to evaluate the performance of machine learning model using unused data to train the model.

6.4.2 Evaluate Performance

ML model performance is evaluated using metrics such as accuracy, precision, recall, F1 score and area under ROC curve. Accuracy measures the overall correctness of the model's predictions. Precision assesses the accuracy of positive predictions, while recall evaluates the model's ability to capture all relevant instances. F1-score is the harmonic mean of precision and recall. The ROC curve provides a graphical representation of the trade-off between true positive rate and false positive rate, and the area under the curve (AUC) quantifies the model's ability to distinguish between classes. These metrics collectively offer a comprehensive evaluation of an ML model's performance in predicting student outcomes, providing insights into its strengths and areas for improvement.

6.5 ML GUI Deployment

Tkinter is a GUI application framework library in Python. It will use the ML models in backend to predict interaction and gives the most important feature which is responsible for drug interaction.

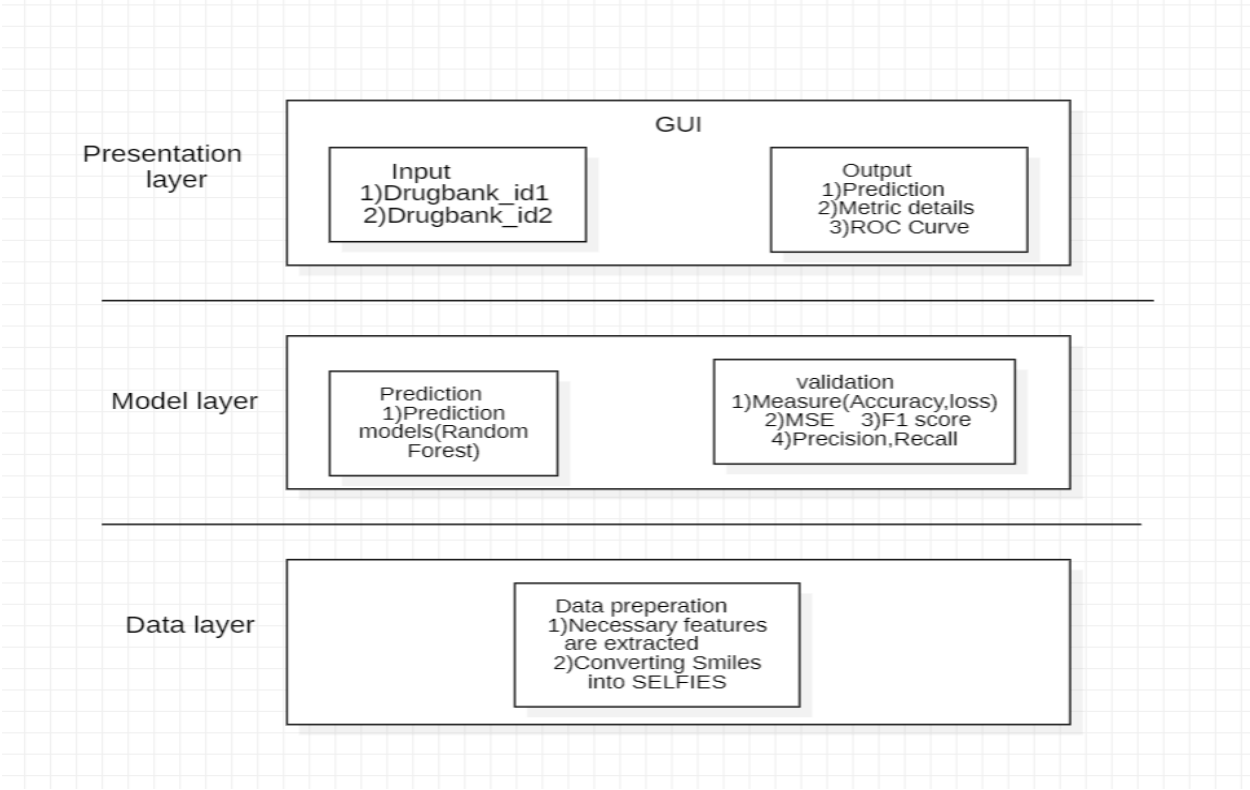


Fig 6.3: An overview of the design of the system.

6.6 Architecture Choice

Among all the available classification methods, random forests provide the highest accuracy. The random forest technique can also handle big data with numerous variables running into thousands. It can automatically balance data sets when a class is more infrequent than other classes in the data. Random forest is a commonly-used machine learning algorithm which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

6.7 Constraints, Assumptions And Dependencies

Like any other machine learning method, the use of Random forest for drug target interaction has certain constraints, assumptions, and dependencies. Here are a few:

Constraints:

Data availability: The effectiveness of GCNs in drug target interaction prediction is dependent on the availability and quality of data on drug-target interactions, which can be limited in some cases.
Model complexity: GCNs can be computationally expensive, which can be a constraint when dealing with large datasets or limited computing resources.
Interpretability: GCNs are sometimes viewed as a "black box" method, making it difficult to understand how the model arrived at its predictions.

Assumptions:

•There should be some actual values in the feature variables of the dataset, which will give the classifier a better chance to predict accurate results, rather than provide an estimation.
•The predictions from each tree must have very low correlations.

Dependencies:

Data quality: The accuracy of Random forest predictions depends on the quality of the data used to train the model. Noisy or incomplete data can lead to inaccurate predictions.
Hyper-parameter tuning: The performance of Random forest can be sensitive to the choice of hyper-parameters, such as learning rate and number of layers. Appropriate hyper-parameter tuning is crucial to achieve optimal performance.
Model architecture: The effectiveness of Random forest for drug target interaction prediction is also dependent on the specific architecture chosen for the model, as different architectures may have varying levels of performance on different datasets.

CHAPTER 7

PROPOSED METHODOLOGY

We considered various algorithm to predict interaction between two drugs, Random forest and XgBoost gave high accuracy compared to other models.

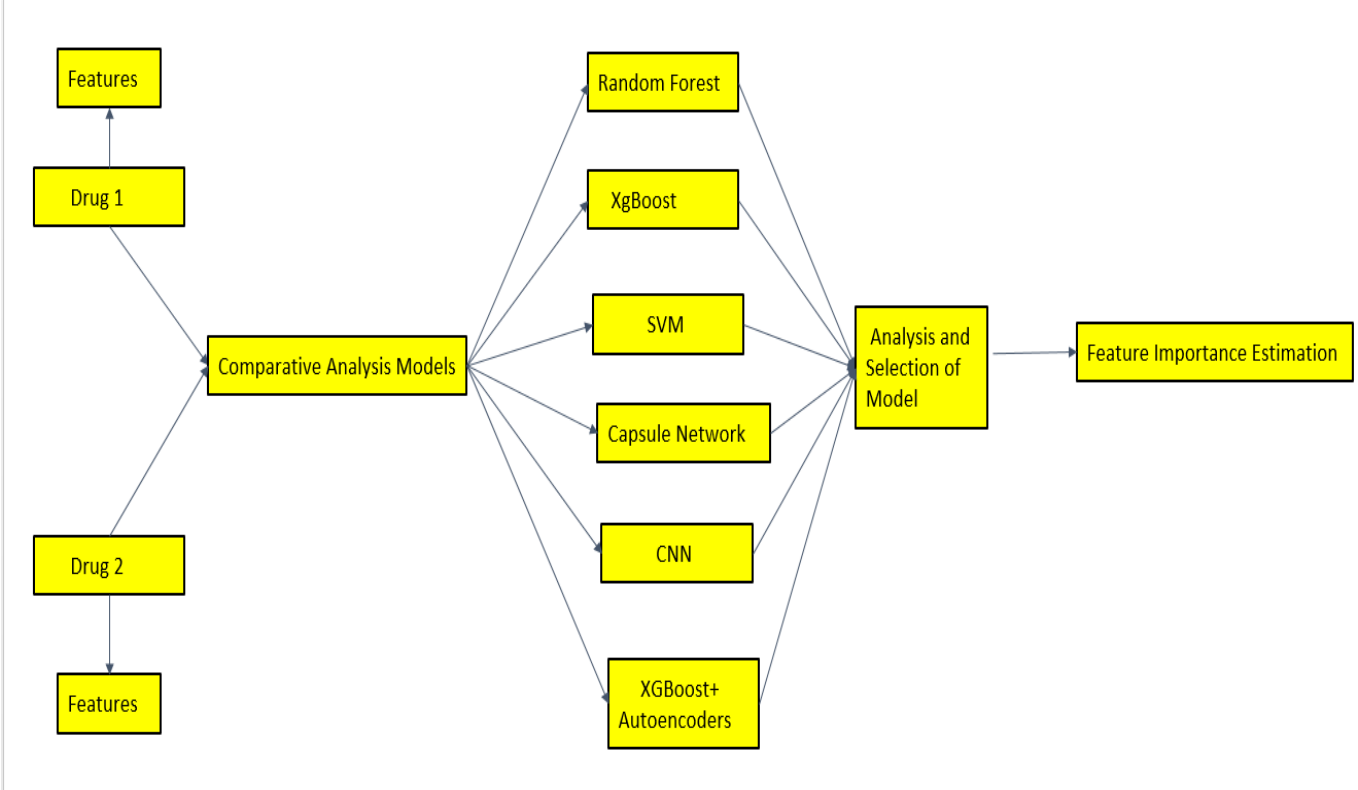


Fig 7.1: Figure represents an overview of the high level design of the system.

Finally we used Random forest algorithm to predict interaction between two drugs. Random forest uses group of decision trees to make decision. Random forest is a commonly-used machine learning algorithm which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

Each drug is characterized by five features mainly dealing with solubility and permeability namely logP ChemAxon, logP ALOGPS, solubility, pKa acidic value and pKa basic value.We

take supervised learning approach to solve the problem.The data for source and target interaction was studied with features.Then the dataset was analyzed using multiple models.

A. Dataset Preprocessing

Individual preprocessing of the dataset is carried out.

Step 1: Removal of Unwanted Features:

We proceed with use of ‘drugbankid’ as our reference id for the identification of drug and ‘cas’ drug identifier column is unnecessary and hence is removed from Drug Features Dataset. ‘SMILES’ column present in Drug Names dataset provides no use to our estimation and is eliminated.

Step 2: Joining of datasets:

First part of this step includes inner join on two datasets i.e, Drug Features Dataset and Drug Names Dataset on the column ’drugbank id’. New dataset formed is named as the Drug Feat_Names dataset. In the next part we join the ‘source’ column from the Drug-Drug Interaction dataset with the ‘drugbank id’ column of Drug Feat Names dataset. Results yielding from this join are saved as Source Dataset. We rename columns from ‘drugbank id’, ‘logALOGPS’, ’logPChemAxom’, ’solubility’,’ LOGPS’, ’pKa(strongest acidic)’ and ‘pKa(strongest basic)’ to ‘source drugbank id’, ‘source logP’, ’source logP CA’, ’source solubility’, ’source acidic pKa’ and ‘source basic pKa’. Similarly we tend to join the ’target’ column from the Drug-Drug Interaction dataset with the ‘drugbank id’ column of Drug Feat Names dataset.Results yielding from this join are saved as Target Dataset. We rename columns from ‘drugbank id’, ‘logALOGPS’, ’logPChemAxom’, ’solubility’, ‘LOGPS’, ’pKa(strongest acidic)’ and ‘pKa(strongest basic)’ to ‘target drugbank id’, ‘target logP’, ’target logP CA’, ’target solubility’, ’target acidic pKa’ and ‘target basic pKa’. As a part of the last step we merge Source Dataset and Target Dataset based on Drug Drug Interaction dataset and save as Merged Dataset.This dataset comprises of columns ‘source drugbank id’, ‘source logP’, ’source logP CA’, ’source solubility’, ’source acidic pKa’, ‘source basic pKa’, ‘target drugbank id’, ‘target logP’, ’target logP CA’, ’target solubility’, ’target acidic pKa’ and ‘target basic pKa’.

Step 3: Removal of Null values

Any row entries having NA values are removed in this step. After this step our dataset is ready to be utilized by the models.

B. Application of Models:

RandomForestClassifier: RandomForestClassification technique creates multiple de cision tree estimators by the utilization of different dataset subsets. We split the dataset for training and testing by 80:20 ratio. And then train the model using 100 tree estimators to output the predictions.

XGBoost: XGBoost is the most powerful form of gradient boosting techniques. The model comprises decision tree estimators ana lyzed sequentially rather than parallel as in case of RandomForestClassification. Here again we make use of 100 decision tree estimators for classification and 80 data for training.

XGBoost + Autoencoders: This model takes the predictions from XGBoost and com bines with the encoded features obtained from the autoen coders. A simple classifier layer is trained above the encoded features . Here again data is split in the ratio 80:20.

SVM: SVM draws a hyperplane separating classes .Here the hyperplane studies the drug features and solely separates into two parts i.e, the drug classes that interact and the ones which do not.

CNN: Input layer receives features and passes down to two layers of convolution and a flatten layer to extract essential features. Activation unit used in the Convolution layer is RELU.A dense layer added on top of it for classification purposes. Activation unit used in the dense layer is Sigmoid. Then we compile the model with binary cross entropy loss, Optimiser as Adam and activation unit as RELU and is made to run for 20 epochs to obtain desirable results.

Capsule Networks: Capsule Networks are one of the widely used Deep learning technique. This model comprises three layers. Input layer reads the features and is passed to the first layer of Capsule Network i.e. Primary Capsule layer which extracts essential features in the form of

vectors. Followed by that extracted features are sent to Digit Capsule Layer where Squash function is used to extract the spatial relationship amongst the vector data. The relationship learnt in this layer is utilized by Class Capsule layer to predict the results function squash(vectors) squared norm = sum(square(vectors)) scale = squared norm / (1 + squared norm) / sqrt(squared norm + epsilon) squashed vectors = scale * vectors.

C. Selection of model:

Metrics of performance are calculated on each of the models we discussed above for selection of good result yielding model. Metrics studied include Accuracy score, Precision Score, Recall Score, F1 score , Mean Squared Error (MSE) and ROC- AUC values.

D. Feature Importance Estimation:

We make use of the trained RandomForest model above and add upon an interface for predicting and testing the results. Tkinter library of python was used to build GUI. The GUI checks firstly whether an interaction exists between the drugs and if at all the interaction is found out then it plots the feature importance of the two interacting drugs. The GUI also outputs the highest feature importance.

CHAPTER 8

IMPLEMENTATION AND PSEUDOCODE

8.1 Random Forest

Random forest is a commonly-used machine learning algorithm which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Train a Random Forest classifier
clf = RandomForestClassifier()
clf.fit(X_train, y_train)

# Predict on the test set
y_pred = clf.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy}')
```

```
curacy: 0.9536755110917791
```

```
from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score

# Predict on the test set
y_pred = clf.predict(X_test)

# Calculate F1 score, precision, and recall
f1 = f1_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

# Print the results
print(f'Accuracy: {accuracy}')
print(f'F1 Score: {f1}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
```

Fig 8.1 : Code for Implementation of Random forest

The above figure 8.1 represents the use of in-built libraries i.e, RandomForestClassifier, then trains the model using training dataset.For prediction uses the test data. Calculates accuracy of the model and other metrics also.

8.2 SVM(Support Vector Machine)

A supervised learning algorithm that works by finding the optimal hyperplane that best separates data points of different classes in a high-dimensional space. SVM is particularly effective in scenarios with complex decision boundaries and high-dimensional feature spaces. The algorithm aims to maximize the margin, the distance between the hyperplane and the nearest data points of each class. This results in robust generalization to new, unseen data. SVM is versatile, accommodating linear and non-linear classification tasks through the use of different kernel functions.

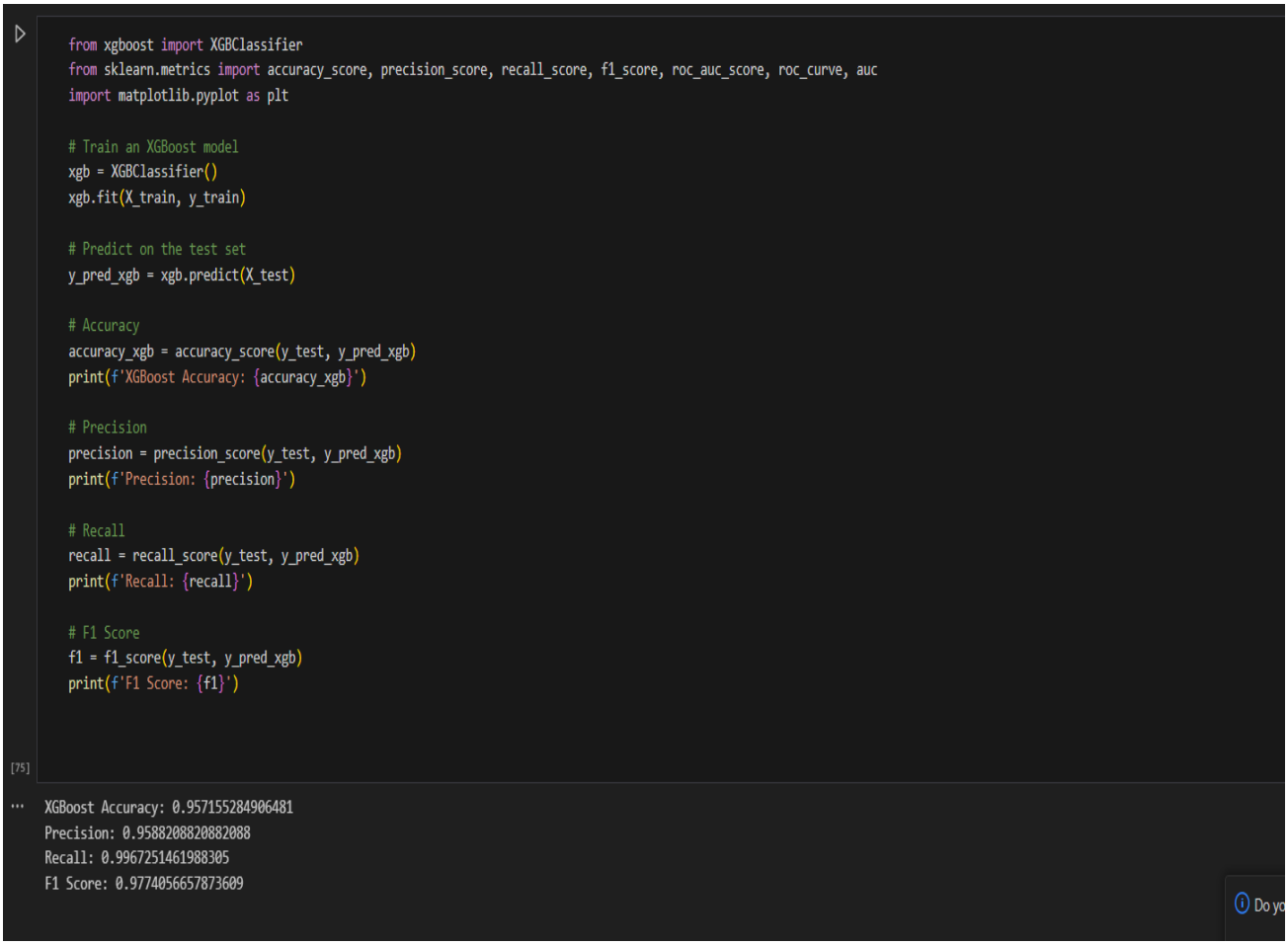
```
from sklearn.svm import SVC

# Train an SVM model
svm = SVC()
svm.fit(X_train, y_train)

# Predict on the test set
y_pred_svm = svm.predict(X_test)

# Evaluate the model
accuracy_svm = accuracy_score(y_test, y_pred_svm)
print(f'SVM Accuracy: {accuracy_svm}')



#####
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

# Train an SVM model
svm = SVC()
svm.fit(X_train, y_train)

# Predict on the training and testing sets
y_pred_svm_train = svm.predict(X_train)
y_pred_svm_test = svm.predict(X_test)

# Calculate training and testing accuracy
accuracy_svm_train = accuracy_score(y_train, y_pred_svm_train)
accuracy_svm_test = accuracy_score(y_test, y_pred_svm_test)

print(f'SVM Training Accuracy: {accuracy_svm_train}')
print(f'SVM Testing Accuracy: {accuracy_svm_test}')
```

Fig 8.2: Code for implementation of SVM

The above figure 8.2 represents the implementation of SVM, we should use the inbuilt library i.e, SVC. Then train models using training dataset. Predict output using testing data and then evaluate the model using different metrics.

8.3 XgBoost

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. Flexibility: XGBoost supports a variety of data types and objectives, including regression, classification, and ranking problems. Regularization: XGBoost incorporates regularization techniques to avoid overfitting and improve generalization performance.



```
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, roc_curve, auc
import matplotlib.pyplot as plt

# Train an XGBoost model
xgb = XGBClassifier()
xgb.fit(X_train, y_train)

# Predict on the test set
y_pred_xgb = xgb.predict(X_test)

# Accuracy
accuracy_xgb = accuracy_score(y_test, y_pred_xgb)
print(f'XGBoost Accuracy: {accuracy_xgb}')

# Precision
precision = precision_score(y_test, y_pred_xgb)
print(f'Precision: {precision}')

# Recall
recall = recall_score(y_test, y_pred_xgb)
print(f'Recall: {recall}')

# F1 Score
f1 = f1_score(y_test, y_pred_xgb)
print(f'F1 Score: {f1}')
```

[75]

```
XGBoost Accuracy: 0.957155284906481
Precision: 0.9588208820882088
Recall: 0.9967251461988305
F1 Score: 0.9774056657873609
```

Fig 8.3: Code for implementation of XgBoost

The above figure 8.3 represents the implementation of XgBoost,we should use the in-built library i.e, XgBoostClassifier. Then train models using training dataset. Predict output using testing data and then evaluate the model using different metrics i.e, Accuracy, Precision, Recall, F1 Score.

8.4 XGBoost + AutoEncoders

An autoencoder could misclassify input errors that are different from those in the training set or changes in underlying relationships that a human would notice. Another drawback is you may eliminate the vital information in the input data. The XGBoost model outperforms the deep learning models on most datasets.

```
class GradientBoostingClassifier:
    def __init__(self, n_estimators=100, learning_rate=0.1):
        self.n_estimators = n_estimators
        self.learning_rate = learning_rate
        self.models = []

    def fit(self, X, y):
        y_pred = np.zeros(y.shape)
        for _ in range(self.n_estimators):
            residuals = y - y_pred
            tree = DecisionTreeRegressor(max_depth=3)
            tree.fit(X, residuals)
            update = self.learning_rate * tree.predict(X)
            y_pred += update
            self.models.append(tree)

    def predict(self, X):
        y_pred = np.zeros(X.shape[0])
        for tree in self.models:
            y_pred += self.learning_rate * tree.predict(X)
        return y_pred

# Train an XGBoost-like model
xgb = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1)
xgb.fit(X_train, y_train)

# Predict on the test set
y_pred_xgb = xgb.predict(X_test)

# Combine predictions with encoded features
encoded_dim = 32  # Adjust the dimension based on your problem
encoder = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(X_train.shape[1],)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(encoded_dim, activation='relu'),
])
decoder = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(encoded_dim,)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(X_train.shape[1], activation='sigmoid'),
])
autoencoder = tf.keras.Sequential([encoder, decoder])
autoencoder.compile(optimizer='adam', loss='mean_squared_error')

# Fit the autoencoder on the training data
autoencoder.fit(X_train, X_train, epochs=20, batch_size=64, validation_split=0.1, verbose=2)

# Encode the features
encoded_features = encoder.predict(X_train)
encoded_features_test = encoder.predict(X_test)

# Train a simple classifier on top of the encoded features
classifier = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(encoded_dim,)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Fit the classifier on the encoded features
classifier.fit(encoded_features, y_train, epochs=20, batch_size=64, validation_split=0.1, verbose=2)

# Predict using the classifier
y_pred_classifier = classifier.predict(encoded_features_test)
y_pred_classifier = (y_pred_classifier > 0.6).astype(int)

# Calculate metrics
accuracy_combined = accuracy_score(y_test, y_pred_classifier)
precision = precision_score(y_test, y_pred_classifier)
```

Fig 8.4: Code for implementation of XgBoost + Autoencoder

The above figure 8.4 represents the implementation of XgBoost+Autoencoder. We should use the in-built library i.e,GradientBoostingClassifier and use encoder and decoder. Then train models using training dataset. Predict output using testing data and then evaluate the model using different metrics i.e, Accuracy, Precision, Recall, F1 Score.

8.5 Capsule Network

In a capsule network, each capsule is made up of a group of neurons with each neuron's output representing a different property of the same feature. This provides the advantage of recognizing the whole entity by first recognizing its parts. The input to a capsule is the output (or features) from a CNN. Capsules leverage vectors for more detailed representation, rather than scalars.

```
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow import keras
from tensorflow.keras.layers import Input, Conv1D, Flatten, Dense, Reshape, Lambda
from tensorflow.keras import backend as K


# Split your data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize your data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform (X_test)

# Define the Capsule Network architecture
input_dim = X_train.shape[1]

# Primary Capsules
x = Input(shape=(input_dim,))
x_reshaped = Reshape(target_shape=(input_dim, 1))(x)
primary_capsules = Conv1D(filters=32, kernel_size=3)(x_reshaped)

# Digit Capsules
def squash(vectors, axis=-2):
    s_squared_norm = K.sum(K.square(vectors), axis, keepdims=True)
    scale = s_squared_norm / (1 + s_squared_norm) / K.sqrt(s_squared_norm + K.epsilon())
    return scale * vectors

digit_capsules = Lambda(squash)(primary_capsules)

# Class Capsules
num_classes = 1  # Binary classification
class_capsules = Conv1D(filters=num_classes, kernel_size=input_dim - 2)(digit_capsules)
class_capsules_reshaped = Reshape(target_shape=(num_classes,))(class_capsules)

# Define the Capsule Network model
capsule_model = keras.models.Model(x, class_capsules_reshaped)

# Compile the model
capsule_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the Capsule Network
capsule_model.fit(X_train, y_train, epochs=10, batch_size=64)

# Evaluate the model on the test data
accuracy = capsule_model.evaluate(X_test, y_test)
print(f'Accuracy: {accuracy[1]}')
```

Fig 8.5: Code for implementation of Capsule Network

The above figure 8.5 represents the implementation of the capsule network in python. It uses CNN and squash definition to scale the vectors. It compiles the model and evaluates the model. Calculates the model accuracy.

8.6 Convolutional Neural Network

CNN classifier for image classification is a CNN-based model specifically designed to classify images into different predefined classes. It learns to extract relevant features from input images and map them to the corresponding classes, enabling accurate image classification.

```
# Split your data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize your data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Define the Convolutional Neural Network architecture
input_dim = X_train.shape[1]

model = keras.Sequential()
model.add(Input(shape=(input_dim, 1)))
model.add(Conv1D(filters=32, kernel_size=3, activation='relu'))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))  # Output layer for binary classification

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Define a callback for ROC AUC calculation
class ROCCallback(keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        self.aucs = []

    def on_epoch_end(self, epoch, logs={}):
        if self.validation_data is None:
            return

        X_val, y_val = self.validation_data[0], self.validation_data[1]
        y_pred = self.model.predict(X_val)
        auc = roc_auc_score(y_val, y_pred)
        self.aucs.append(auc)
        print(f'Epoch {epoch + 1} - AUC: {auc:.4f}')

roc_callback = ROCCallback()

# Train the CNN model
history = model.fit(X_train, y_train, epochs=10, batch_size=64, validation_data=(X_test, y_test), callbacks=[roc_callback])

# Evaluate the model on the test data
y_pred = model.predict(X_test)
y_pred_binary = (y_pred > 0.5).astype(int)

accuracy = accuracy_score(y_test, y_pred_binary)
precision = precision_score(y_test, y_pred_binary)
recall = recall_score(y_test, y_pred_binary)
f1 = f1_score(y_test, y_pred_binary)

print(f'Testing Accuracy: {accuracy:.4f}')
print(f'Precision: {precision:.4f}')
print(f'Recall: {recall:.4f}')
print(f'F1 Score: {f1:.4f}')


och 1/10
3/288 [==============================] - 2s 4ms/step - loss: 0.2545 - accuracy: 0.9265 - val_loss: 0.2295 - val_accuracy: 0.9302
och 2/10
```

Fig 8.6: Code for implementation of Convolution Neural Network

The above figure 8.6 represents the python code for implementation of CNN. It uses convolutional layer and pooling layer, fully connected layer, dense layer. It uses adam as optimizer, RELU as activation layer.

CHAPTER 9
RESULTS AND DISCUSSION

9.1 Random forest

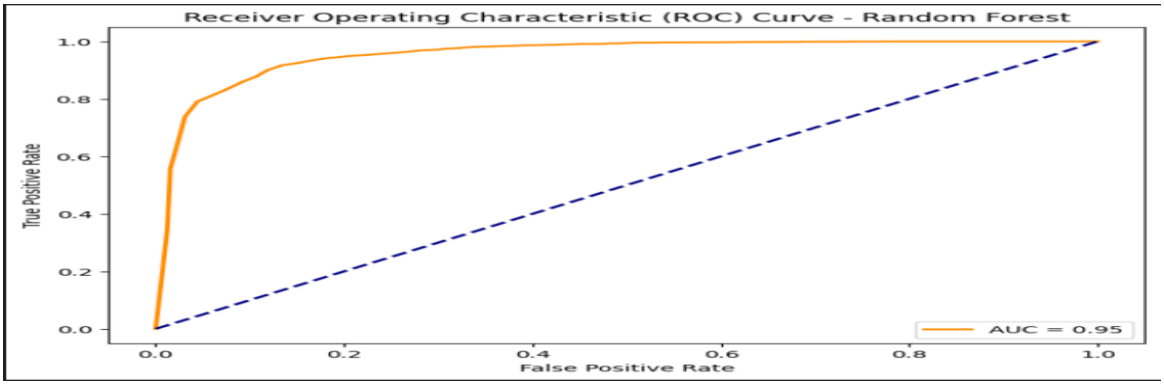


Fig. 9.1: ROC Curve for Random Forest

As shown in Fig.9.1, AUC of 0.95 indicates a very good model performance. i.e; it has high ability to distinguish between positive and negative instances.

9.2 SVM

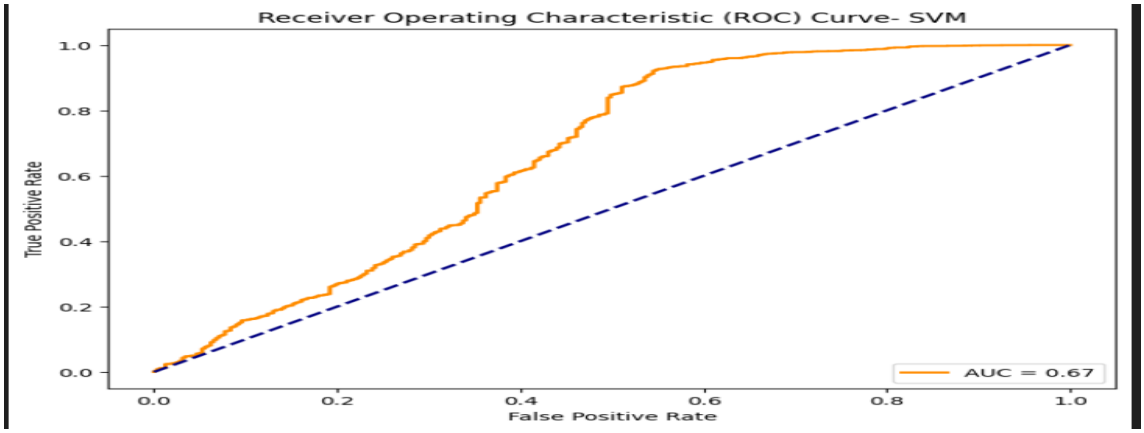


Fig. 9.2: ROC Curve for SVM

As shown in Fig.9.2, AUC of 0.67 indicates moderate performance. i.e, there is a good balance between true positive rate and false positive rate.

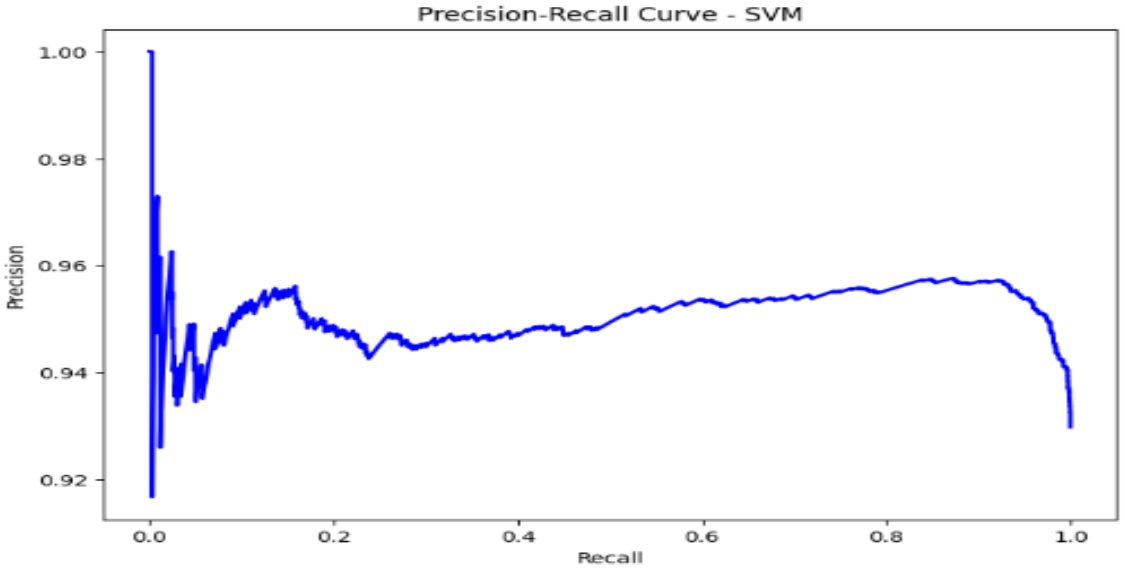


Fig. 9.3: Precision-Recall Curve for SVM

As shown in Fig.9.3, With a precision and recall of 93.3% and 99.9% respectively indicates that the instances predicted by the model as positive are indeed positives.

9.3 XGBoost

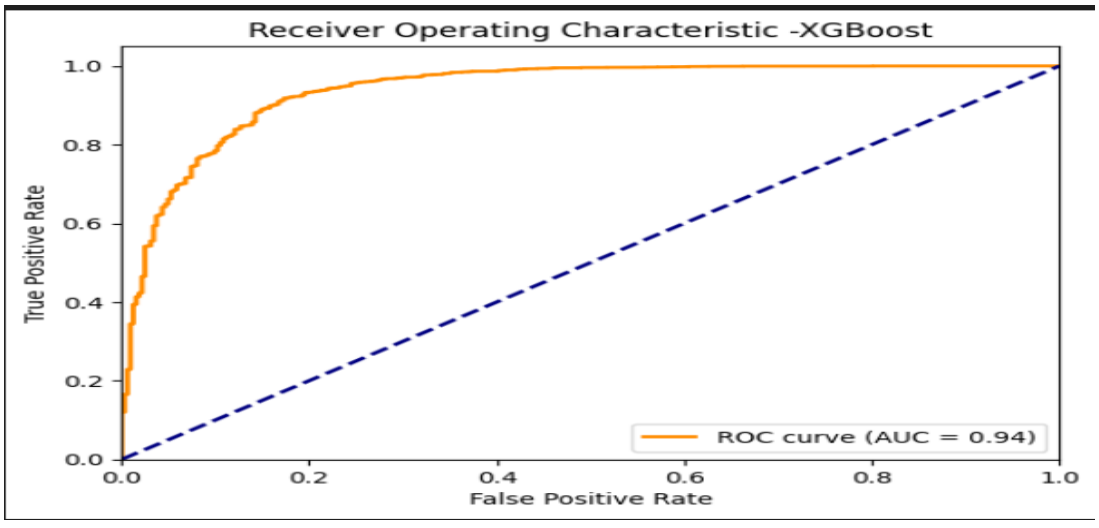


Fig. 9.4: ROC Curve for XGBoost

As shown in Fig.9.4, With an AUC of 0.94 this model is also a very good model like Random Forest Classifier. i.e, it has a high ability to distinguish between positive and negative instances.

9.4 XGBoost + Autoencoders

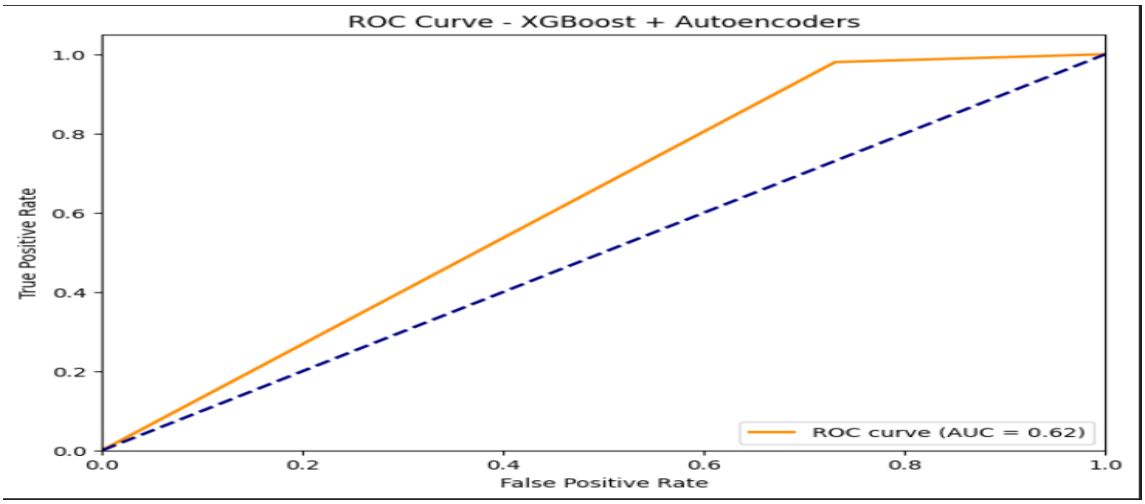


Fig. 9.5: ROC Curve for XGBoost + Autoencoders

As shown in Fig.9.5, AUC of 0.62 indicates lower performance compared to all the other models.

9.5 Capsule Networks

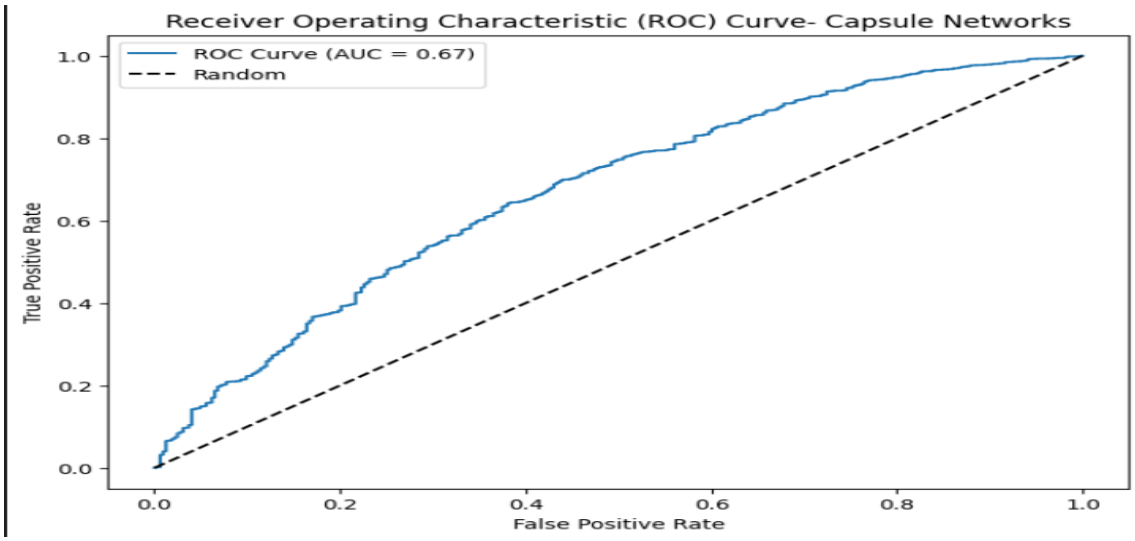


Fig. 9.6: ROC Curve for Capsule Networks

As shown in Fig.9.6, AUC of 0.67 is similar to SVM suggesting comparable performance.

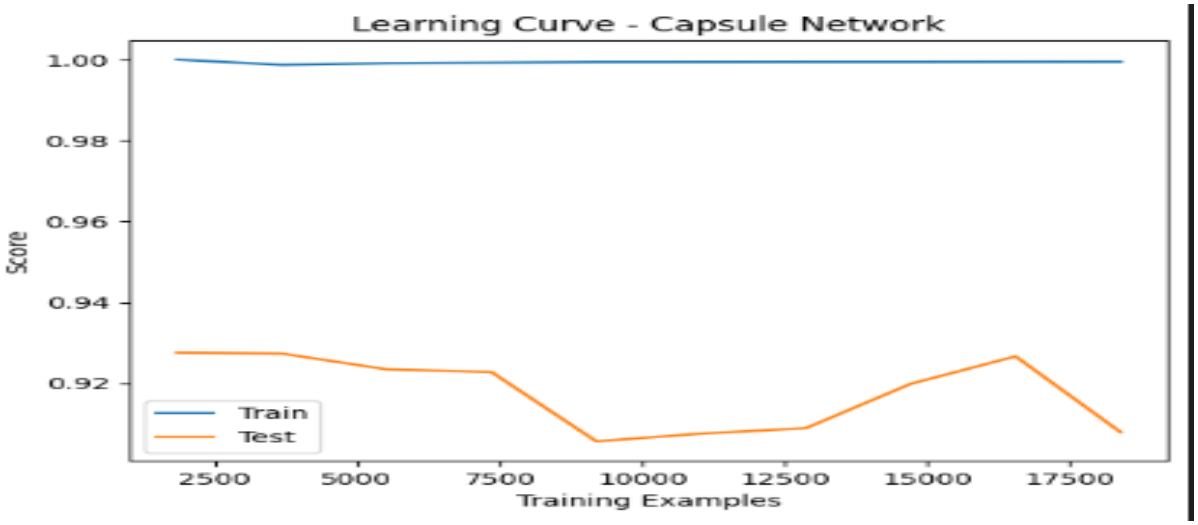


Fig. 9.7: Learning Curve for Capsule Networks

As shown in Fig.9.7, The results of the learning curve for capsule network in Fig 9.7 shows that the model is a strong performer and could be useful in applications where capturing positive instances is crucial.

9.6 CNN

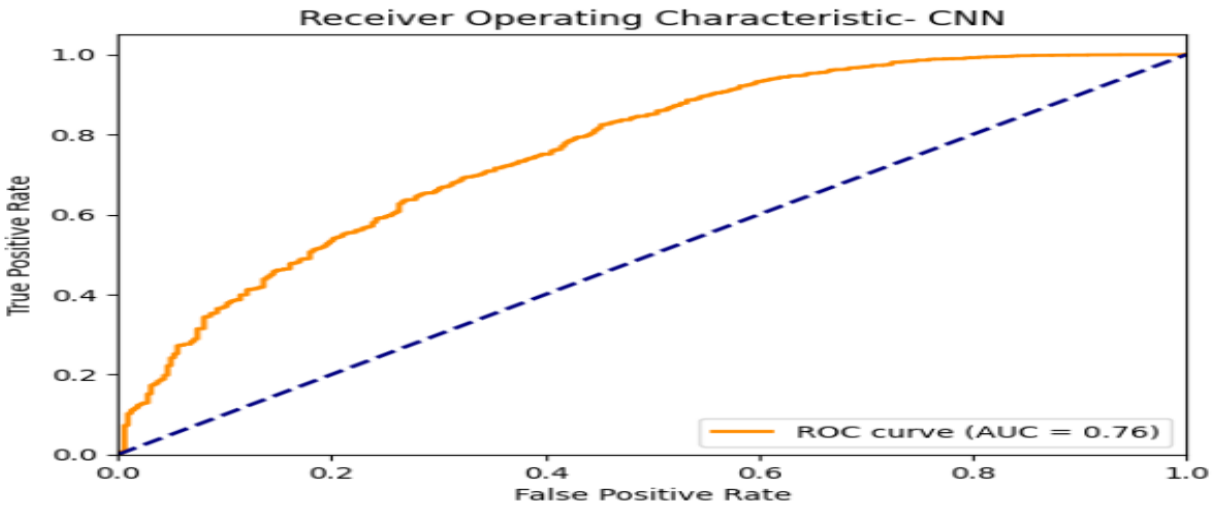


Fig. 9.8: ROC Curve for CNN

As shown in Fig.9.8, AUC of 0.76 suggests that the model performed reasonably well i.e; there is a good balance between true positive rate and false positive rate.

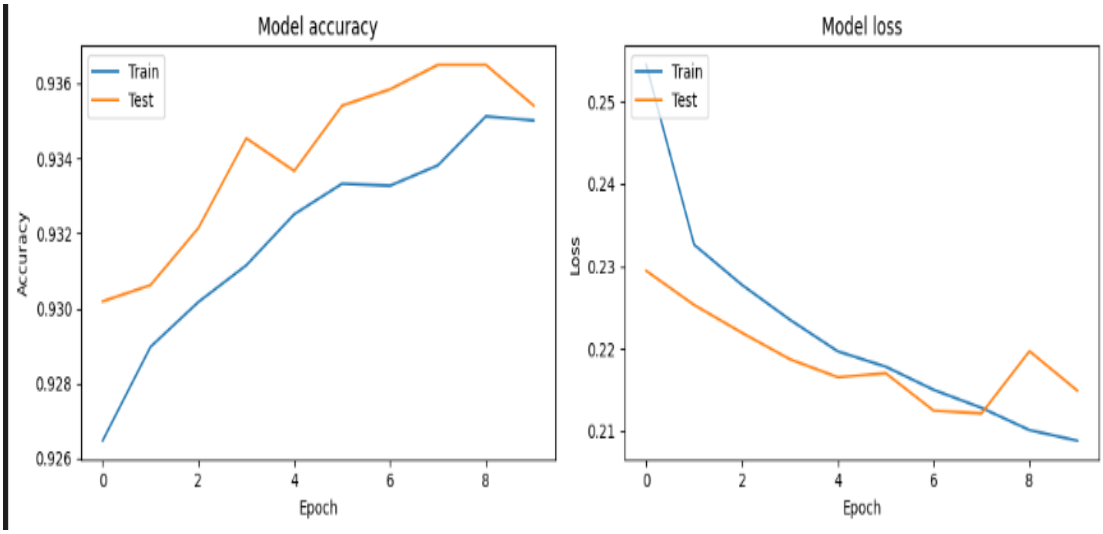


Fig. 9.9: Model accuracy and loss Curve for CNN

As shown in Fig.9.9, With an accuracy of 93.61% the correctly predicts the class labels for a substantial portion of the dataset.

9.7 Table for Comparison

Algorithm	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Random Forest	95.60	95.63	99.83	0.97	0.96
XGBoost	95.71	95.88	99.67	0.97	0.94
SVM	93.32	93.31	99.97	0.96	0.67
Capsule Networks	92.90	92.97	99.92	0.96	0.67
CNN	93.61	94.03	99.44	0.96	0.77
XGBoost+Autoencoder	93.30	94.42	98.61	0.96	0.61

Fig. 9.10: Metrics analyzed for different algorithm

Now, let us make conclusions on the results that we have obtained.

Accuracy which provides understanding on overall correct prediction, is fundamental for any comparison. Here again we see that XGBoost and Random Forest Classifier performed slightly better than other models with accuracy values 95.7 and 95.4 respectively as shown in Fig. 9.10. However the

models Capsule Networks, CNN, SVM and AutoEncoders+XGBoost were good enough proving their values 93.9, 93.6, 93.3 and 095.7 respectively as shown in Fig. 9.10. Performance of all models were nearly equivalent.

Accuracy alone cannot justify the comparison. Hence other comparison metrics like Precision, F1 score and Recall also contribute to effective estimation.

Now let us analyze the **Precision score** among models, which measures the accuracy of the positive predictions made by the model. It is the ratio of true positive predictions to the total number of positive predictions made by the model. Capsule network predicted accurately with the score of 99.8 followed by XGboost and Random Forest Classifier with scores of 95.8 and 95.5 respectively. XGboost+Autoencoders, CNN and SVM also predicted well with precision values 94.7, 93.8 and 93.3 respectively as shown in Fig. 9.10.

Recall also goes hand in hand with precision metrics which measure out of relevant features how many did we get right? Here we see that SVM, Random Forest Classifier, Capsule Networks and CNN played its game well with scores of 99.9, 99.8, 99.8 and 99.7 respectively as shown in Fig. 9.10. XGBoost and XGBoost +Autoencoders were in no lag with values 95.8 and 94.7 respectively.

It also becomes essential to calculate **F1 score** when we have both Recall and Precision score .The results here also show that performance of models were nearly equivalent. XGBoost. Random Forest Classifier, CNN, Capsule Network and SVM had their scores 97.7,97.6,96.7,96.3 and 96.3 respectively as shown in Fig. 9.10. XGBoost + Autoencoders had a score of 95.8.

Receiver Operating Characteristic Curve (ROC) which provides insights on true positive rate and false positive rates was plotted for all models and was found that Random Forest Classifier and XGBoost performed better with AUC 0.95 and 0.94 respectively. The ROC for models CNN, SVM, Capsule Network and AutoEncoders + XGboost performed above satisfactory with AUC values 0.76 ,0.67 ,0.67 and 0.62 respectively as shown in Fig. 9.10.

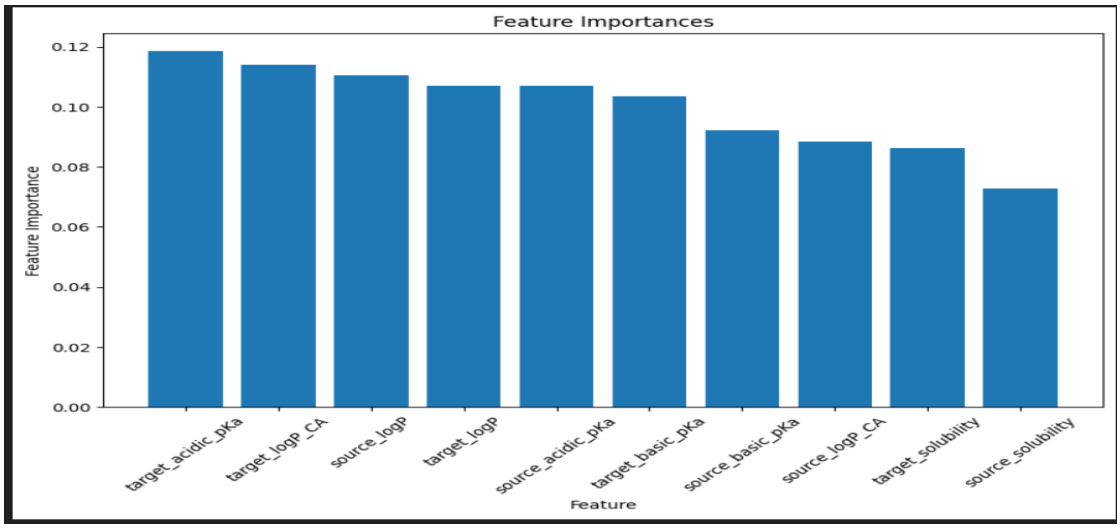


Fig. 9.11: importance of features

At last to predict the importance of features, we used Random Forest Classification as the model proved better prediction and the calculated result is displayed in the above Fig 9.11.

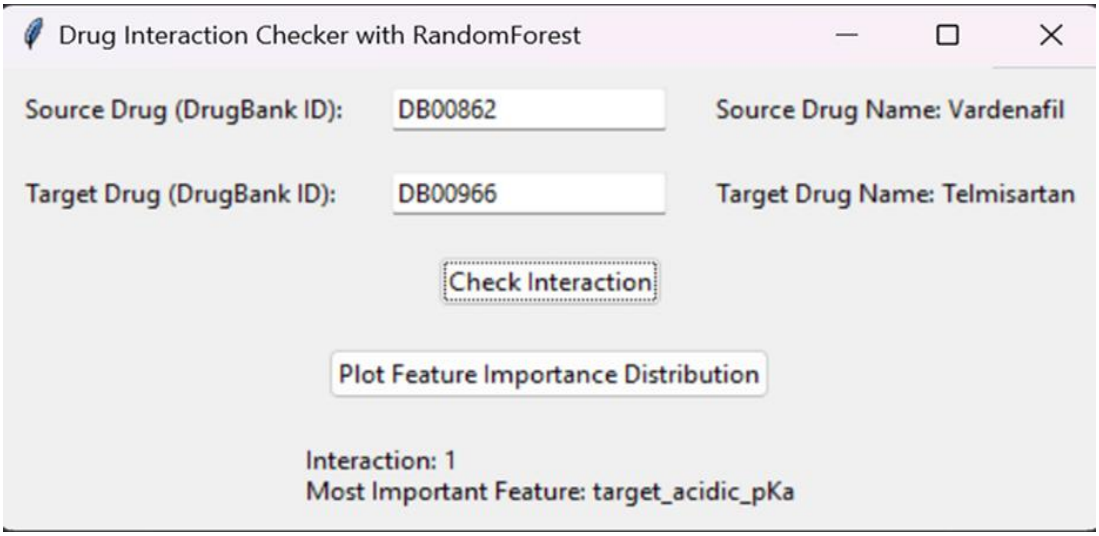


Fig 9.12: GUI of our system when interaction is present.

The above Fig 9.12 represents it will take input as two drugbank_id's of drugs. When you click on check interaction, it will print 1 as interaction is present between them. And returns the most important feature which causes interaction between those two drugs.

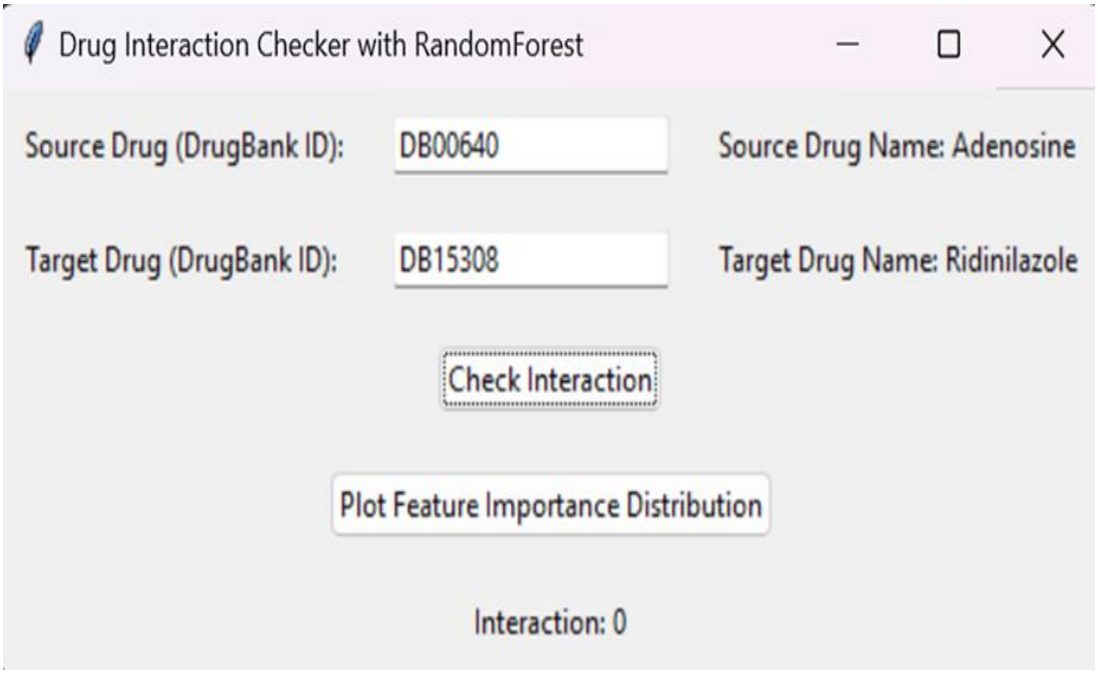


Fig 9.13: GUI of our system when there is no interaction.

The above Fig 9.13 represents it will take input as two drugbank_id's of drugs. When you click on check interaction, it will print 0 as interaction is not present between them. And it does not return the most important feature as there is no interaction between them.

CHAPTER 10
CONCLUSION AND FUTURE WORK

The primary objective of this project revolves around the prediction of drug-drug interactions (DDIs) by training and evaluating multiple models using five major features of drugs. The central focus is to discern the importance of these features when two drugs interact. However, it is acknowledged that the study currently has limitations, notably its confinement to the analysis of two drugs and reliance on only five major features.The study's methodology involves training various models, each potentially employing different algorithms or architectures. The performance of these models is assessed using standard metrics such as accuracy, precision, recall, and possibly the area under the receiver operating characteristic curve. This rigorous evaluation process is designed to identify the most effective model in predicting drug interactions based on the selected features.

Despite the project's merit, it is recognized that the complexity of drug interactions extends beyond the initial five features considered. The acknowledgment of this limitation is critical, emphasizing the intricate nature of biological systems and the multitude of factors influencing drug interactions.Looking ahead, the study outlines ambitious future directions. These include plans to broaden the scope by incorporating a more extensive array of features into the analysis. The intention is to delve deeper into the biochemical, molecular, and pharmacological attributes of drugs to capture a more holistic understanding of drug interactions. Furthermore, the future projects aspire to move beyond the restriction of studying interactions between only two drugs. Real-time scenarios often involve multiple drug interactions, necessitating models that can handle such intricacies.

In essence, this project serves as a foundational step in understanding drug-drug interactions. While constrained by the limitations of its current scope, the study lays the groundwork for future endeavors. The expansion of features and the aspiration to predict interactions involving more than two drugs reflect a commitment to advancing the sophistication and practical applicability of models in the complex landscape of pharmacology and healthcare.

REFERENCES

[1] Xiaoqiang Xu; Ping Xuan; Tiangang Zhang; Bingxu Chen; Nan Sheng,"Inferring Drug-Target Interactions based on Random Walk and Convolutional Neural Network" in IEEE/ACM Transactions on Computational Biology and Bioinformatics, Volume: 19, Issue: 4, 01 July-Aug. 2022.

[2] Shaghayegh Sadeghi , Alioune Ngom, "DDI PRED:Graph Convolutional Network based Drug-Drug Interactions Prediction using Drug Chemical Structure Embedding" in IEEE, 2022.

[3] Nilay Fatma , Yıldız Alper Özcan, "Graph Convolutional Autoencoder and Generative Adversarial Network-Based Method for Predicting Drug-Target Interactions " published in 2022.

[4] Xu Gong, Maotao Liu, Haichao Sun,Min Li, Qun Liu, "HS-DTI: Drug-Target Interaction Prediction based on Hierarchical Networks and Multi-Order Sequence Effect" published in 2022.

[5] Tianjung , Wang , Xin , Liu, "A Graph Convolution-Transformer Neural Network for Drug-Target Interaction Prediction", ICBBT 2022, May 27–29, 2022.

[6] Chang Sun, Ping Xuan , Tiangang Zhang and Yilin Ye, "Using Supervised Machine Learning Algorithms for Drug-Target Interaction Prediction" in IEEE/ACM Transactions On Computational Biology And Bioinformatics, Vol. 19, No. 1, January/February 2022.

[7] Mohammad A. Rezaei , Yanjun Li , Dapeng Wu, Xiaolin Li and Chenglong Li,"Deep Learning in Drug Design : Protein-Ligand Binding Affinity Prediction" in IEEE/ACM Transactions on Computational Biology and Bioinformatics, Volume: 19, Issue: 1, 01 Jan.-Feb. 2022.

[8] Nelson R. C. Monteiro , Bernardete Ribeiro , and Joel P. Arrais,"Drug Target Interaction Prediction :end-to-end Deep Learning Approach" in IEEE/ACM Transactions on Computational Biology and Bioinformatics, Volume: 18, Issue: 6 , 01 Nov.-Dec. 2021.

[9] Liu S., An J., Zhao J., Zhao S., Lv, H. & Wang S, "Drug-Target Interaction Prediction based on Multisource Information Weighted Fusion" from Drug-Target Interaction Prediction Based on Multisource Information Weighted Fusion.Contrast media & molecular imaging, 6044256, 2021.

[10] Chengkun Wu , Wei Wang , Xi Yang, Canqun Yang, "Drug-Drug Interaction extraction from Biomedical Texts based on Multi-Attention Mechanism",ICBRA, 2021.

[11] Mongia A, Majumdar A,"Drug-Target Interaction Prediction using Multi Graph Regularized Nuclear Norm Minimization", from PLoS ONE 15(1): e0226484, 2020.

[12] Xinyu Hao , Jiaying You , PingZhao Hu, "Predicting Drug-Drug Interactions using Deep Neural Networks", ICMLC '19, February 22–24, 2019.

APPENDIX A

ACRONYMS AND ABBREVIATIONS

1. **SMILES -** Simplified Molecular Input Line Entry System
2. **SELFIES -** Self-Referencing Embedded Strings
3. **DDI's -** Drug Drug Interactions
4. **HS-DTI** - Heterogeneous drug target interaction
5. **GCN -** Graph Convolution Network
6. **AUROC -** Area Under the Receiver Operating Characteristics
7. **AUPRC -** Area under Precision-Recall curve
8. **SVM -** Support vector machine
9. **CNN -** Convolutional Neural Network
10. **ML/DL -** Machine learning / Deep learning
11. **MGRNNM -** Multi-Graph Regularized Nuclear Norm Minimization
12. **WNN-GIP -** Weighted Nearest Neighbours – Gaussian interaction profile

Importance of Drug Features in Drug-Drug Interaction : A Comparative Study

Chetana N Patil
Department of CSE
PES University
Bengaluru, India
chetananpatil2002@gmail.com

Naveen B N
Department of CSE
PES University
Bengaluru, India
naveenbiligeri@gmail.com

Mekala Sanjana
Department of CSE
PES University
Bengaluru, India
sanjana.mekala135@gmail.com

Kumari Shivangi
Department of CSE
PES University
Bengaluru, India
shivangik325@gmail.com

Prajwala T R
Associate Professor Department of CSE
PES University
Bengaluru, India
prajwalatr@pes.edu

***Abstract*— A Drug-Drug Interaction is an interplay of drugs where one or more drugs interfere with the activity of other drugs. Synergy or Interaction between drugs may cause side effects that are unforeseen.They may also intervene and oppose each other's action and nullify the medication.Many of such interactions are often negligible but some might be harmful if not discovered at the earliest.Features or characteristics of drugs do play a vital role in interaction.It therefore becomes an essential study to untie the patterns of interaction.**

Our study aims to predict important features that are involved in the interaction of two drugs.Firstly we train multiple learning models on the dataset. These models are RandomForestClassifier,XGBoost,SVM,Autoencoders+XGBoost,CNN and Capsule Networks. Then compare the results of trained models and select the most appropriate performance yielding model. We then ask the model to predict the interaction between two drugs and also plot the feature importance distribution of two interacting drugs. In the study it was found that RandomForestClassifier did slightly better than XGBoost and outshined against all other models.Hence RandomForestClassifier was used as The BackBone model to predict feature importances amongst two interacting drugs.

***Keywords*—Drug-Drug Interaction, Random Forest Classifier,XGBoost,CNN,SVM,Capsule Networks, Autoencoders**

I. INTRODUCTION

In recent years AI driven drug discovery is advancing as scientific innovations in this field can reduce the number of years taken to develop and design the drugs. People all across the world, especially the elderly people ingest multiple drugs at a time for underlying medical causes due to the pervasiveness of polypharmacy[2].And these drugs when ingested simultaneously together might lead to an interaction either with another drugs or with a protein or any target molecules and might lead to Adverse Side Reactions(ADR) ADR are one of the main reasons for financial loss in the pharmaceutical company as ADR's might cause mortality and morbidity and ultimately leading to the withdrawal of the drugs sold across the markets[4].Therefore has paved a way for the discovery of these interactions .

As we all know that there could be multiple DrugDrug Interaction can take place.In our study we restricted ourselves to find the interaction taking place between two drugs .Each drug is characterized by its features namely permeability,solubility ,structural pattern etc.These all features tend to examine four major parameters of ingested drugs:absorption,distribution,metabolism and Excretion in short ADME's . It therefore becomes an essence to identify to study the interaction patterns that take place between two drugs by using these properties.

If we take any two drugs that are interacting , not all attributes of source and target drug molecule influence its importance in the interaction.Hence,it is vital for us to study these interaction patterns to predict the feature importances in the Drug-Drug Interaction.

We aim to build multiple models and then select the best model based on performance and then use that model to predict the importance of Drug features.The study not only help drug designers to develop new drugs but also help them rethink on the solution for repurposing some combination of drugs if they are found to be synergistic.This would then ultimately lead efficient and proficient Drug discovery.

II. LITERATURE REVIEW

3WDDI is a novel method for predicting Drug Drug interactions using a 3 way decision process as well as the application of the knowledge graph. Here CNN was used as a delay decision model and used as a decision function.They generated drug substructure similarity features and drug embeddings from the Drug Knowledge Graph (DRKG) using ComplEx for the Three-Way Drug-Drug Interaction (3WDDI)

prediction. Subsequently, the drug substructure similarity features were employed to delineate the boundary region and derive classification results for the remaining areas. The knowledge graph embeddings of a drug pair were introduced as novel supplementary features by 3WDDI for improved delay decision-making. The proposed method was implemented, and a comparative experiment was conducted on a widely adopted dataset. The experimental outcomes demonstrate the superior performance of 3WDDI compared to baseline Drug-Drug Interaction (DDI) prediction models.[1]

CNN-DDI, a refined architecture of CNN, was used to predict Drug interactions based on similarity.In this method features were extracted from various descriptors of drugs namely,drug categories,enzymes and pathways.Then they converted it into feature vectors and then spatial information amongst the vectors were obtained using Jaccard Similarity.Then based on the similarity metrics obtained ,they developed a new Convolutional Neural Network Model.But the accuracy of the model was 88.[3]

DDIPred was a model developed to find the importance of chemical structure of drugs in the interaction.The study involved use of SMILES(Simplified Molecular Input Line Entry System).Susceptibility of syntactic and semantic error is plausible in SMILES and hence was converted to SELFIES(SELFreferencIng Embedded Strings) to study the interaction importance.Graph Convolutional Network was used as a backbone to study the interaction.The method had good AUROC of 0.99.However the model studied only one feature that is Drug's chemical structure.[4]

Multiple machine learning methods have been used to predict drug-target interactions. To evaluate the proposed models, the calculated AUROC and AUPRC values and obtained results indicate that nonlinear SVM and logistic regression performed better than other models with AUROC and AUPRC values of 0.8317 and 0.8260, respectively.The cons of the model are: It may lead to overfitting. Logistic regression is not suitable for modeling complex relationships between variables, such as interactions or nonlinear effects. Logistic regression is limited to linear relationships. [6].

DANN DDI , a model developed using Deep Attention Neural Networks.Here in the model they attentively added the drug features to predict unforeseen drug drug interactions.Accuracy score of 88.74 was obtained in the overall model.Proper balanced data and proper noise removal in the data would lead to improved accuracy scores.[12]

When Deep Neural Network was used as a model to predict the interaction of drugs, accuracy of 93.2 on test dataset and 94.9 on validation dataset were obtained.This method involved small dataset of 5,134 drug molecules with their drug descriptors namely constitutional properties,topological properties ,geometric properties.The model used four hidden layers and activation unit as RELU.Further more on the study they added the testing to predict interaction amongst existingIBD drugs and found an increase in antipsychotic effects. of drugs.[14]

Below are some of the conclusion that were made from the entire literature Review :

- Majorly the ML and DL model used were SVM , Random Forest And Decision Tree, CNN, Autoencoders.
- They have compared the performances of these algorithms using F-1 score , accuracy,precision and AUC.
- We used ML algorithms to predict the drug interaction and the most important feature responsible for the interaction.

III. DATASET

Number of dataset that facilitated our study was three.

1. Drug-Drug Interaction Dataset
2. Drug features Dataset
3. Drug Names Dataset

A. Drug-Drug Interaction Dataset This dataset represents the presence or absence of interaction between two drugs.The columns present here include 'source', 'target' and 'interaction'. The 'interaction' column consists of two values 0 and 1.Value ' 0' indicates no interaction exists between the two drugs and Value '1' indicates the presence of interaction between 'source' and the 'target' columns as shown in Fig 1.

	source	target	interaction
0	DB00862	DB00966	1
1	DB01235	DB01275	1
2	DB01609	DB06212	1
3	DB01232	DB09291	1
4	DB00104	DB00908	1
...	...	...	...
22967	DB00632	DB15299	0
22968	DB00633	DB15300	0
22969	DB00634	DB15302	0
22970	DB00635	DB15307	0
22971	DB00640	DB15308	0

Fig. 1. Drug-Drug Interaction Dataset

B. Drug features Dataset The refined dataset comprises 22972 rows.This dataset represents features of individual drugs.The dataset spans over 8288 rows and 7 columns. The columns include 'drugbank id' and 'cas' drug identification Id followed by these columns are the features representing permeability and solubility attributes of drugs namely 'log ALOGPS','logP

hemAxom, 'solubility', 'LOGPS', 'pKa(strongest acidic)' and 'pKa(strongest basic)'features as seen in Fig 2 .

	drugbank_id	cas	logP ALOGPS	logP ChemAxon	solubility ALOGPS	pKa (strongest acidic)	pKa (strongest basic)
0	DB00006	128270-60-0	-0.76	-14.00	4.64e-02 g/l	2.79	11.88
1	DB00007	53714-56-0	1.04	-2.40	3.38e-02 g/l	9.49	11.92
2	DB00014	65807-02-5	0.30	-5.20	2.83e-02 g/l	9.27	10.82
3	DB00035	16679-58-6	-1.00	-6.10	1.10e-01 g/l	9.50	11.77
4	DB00050	120287-85-6	1.33	-1.70	6.94e-03 g/l	9.50	11.79

Fig. 2. Drug Features Dataset

C. Drug Names Dataset This data comprises three columns namely, 'drugbank id', 'name' and 'SMILES' columns.'name' column represents the name of the drug corresponding to 'drugbank id' as seen in Fig 3.

	drugbank_id	name	smiles
0	DB00006	Bivalirudin	CC[C@H](C)[C@H](NC(=O)[C@H](CCC(O)=O)NC(=O)[C@...
1	DB00007	Leuprolide	CCNC(=O)[C@@H]1CCCN1C(=O)[C@H](CCCNC(N)=N)NC(=...
2	DB00014	Goserelin	CC(C)C[C@H](NC(=O)[C@@H](COC(C)(C)C)NC(=O)[C@H...
3	DB00035	Desmopressin	NC(=O)CC[C@@H]1NC(=O)[C@H](CC2=CC=CC=C2)NC(=O)...
4	DB00050	Cetrorelix	CC(C)C[C@H](NC(=O)[C@@H](CCCNC(N)=O)NC(=O)[C@H...

Fig. 3. Drug Names Dataset

IV. METHODOLOGY

Proposed overflow involves four major steps,

1.Data Preprocessing

2.Application of Models

3.Selection of Model

4.Feature Importance Prediction

A. Dataset Preprocessing Individual preprocessing of the dataset is carried out.

Step 1 : Removal of Unwanted Features:

We proceed with use of 'drugbankid' as our reference id for the identification of drug and 'cas' drug identifier column is unnecessary and hence is removed from Drug Features Dataset.'SMILES' column present in Drug Names dataset provides no use to our estimation and is eliminated.

Step 2 : Joining of datasets:

First part of this step includes inner join on two datasets i.e, Drug Features Dataset and Drug Names Dataset on the column 'drugbank id'. New dataset formed is named as the Drug Feat Names dataset. In the next part we join the 'source' column from the Drug-Drug Interaction dataset with the 'drugbank id' column of Drug Feat Names dataset. Results yielding from this join are saved as Source Dataset. We rename columns from 'drugbankid', 'logALOGPS', 'logPChemAxom', 'solubility LOGPS', 'pKa(strongest acidic)' and 'pKa(strongest basic)' to 'source drugbank id', 'source logP', 'source logP CA', 'source solubility', 'source acidic pKa' and 'source basic pKa'. Similarly we tend to join the 'target' column from the Drug-Drug Interaction dataset with the 'drugbank id' column of Drug Feat Names dataset.Results yielding from this join are saved as Target Dataset. We rename columns from 'drugbankid', 'logALOGPS', 'logPChemAxom', 'solubility LOGPS', 'pKa(strongest acidic)' and 'pKa(strongest basic)' to 'target drugbank id', 'target logP', 'target logPCA', 'target solubility', 'target acidic pKa' and 'target basic pKa'. As a part of the last step we merge Source Dataset and Target Dataset based on Drug Drug Interaction dataset and save as Merged Dataset.This dataset comprises of columns 'source drugbank id', 'source logP', 'source logP CA', 'source solubility', 'source acidic pKa', 'source basic pKa', 'target drugbank id', 'target logP', 'target logP CA', 'target solubility', 'target acidic pKa' and 'target basic pKa'.

Step 3 : Removal of Null values

Any row entries having NA values are removed in this step.After this step our dataset is ready to be utilized by the models.

B. Models

1) RandomForestClassifier: RandomForestClassification technique creates multiple decision tree estimators by the utilization of different dataset subsets.We split the dataset for training and testing by 80:20 ratio.And then train the model using 100 tree estimators to output the predictions.

2) XGBoost: XGBoost is the most powerful form of gradient boosting techniques.The model comprises decision tree estimators analyzed sequentially rather than parallel as in case of RandomForestClassification. Here again we make use of 100 decision tree estimators for classification and 80 data for training.

3) XGBoost + Autoencoders: This model takes the predictions from XGBoost and combines with the encoded features obtained from the autoencoders.A simple classifier layer is trained above the encoded features . Here again data is split in the ratio 80:20.

4) SVM : SVM draws a hyperplane separating classes .Here the hyperplane studies the drug features and solely separates into two parts i.e,the drug classes that interact and the ones which do not.

5) CNN: Input layer receives features and passes down to two layers of convolution and a flatten layer to extract essential features. Activation unit used in the Convolution layer is RELU.A dense layer added on top of it for classification purposes. Activation unit used in the dense layer is

Sigmoid.Then we compile the model with binary cross entropy loss, Optimiser as Adam and activation unit as RELU and it is made to run for 20 epochs to obtain desirable results.

6) Capsule Networks: Capsule Networks are one of the widely used Deep learning technique.This model comprises three layers. Input layer reads the features and is passed to the first layer of Capsule Network i.e. Primary Capsule layer which extracts essential features in the form of vectors. Followed by that extracted features are sent to Digit Capsule Layer where Squash function is used to extract the spatial relationship amongst the vector data.The relationship learnt in this layer is utilized by Class Capsule layer to predict the results.

Function squash(vectors):
squared norm = sum(square(vectors)) scale = squared norm / (1 + squared norm) / sqrt(squared norm + epsilon) squashed vectors = scale * vectors

Fig. 4. Flow of activities

C. Selection of model:
Metrics of performance are calculated on each of the models we discussed above for selection of good result yielding model.Metrics studied include Accuracy score, Precision Score , Recall Score , F1 score , Mean Squared Error (MSE) and ROC- AUC values.
The performance estimation of any classification model can be judged by Receiver Operating Curve(ROC). It gives Area Under the Curve (AUC) value which helps in the estimation of performance. From Fig1 to Fig5 it is evident that RandomForestClassifier has yielded better results with AUC 0.96 which is slightly ahead of XGBoost's AUC- 0.94.

Algorithm	Accuracy	Precision	Recall	F1 Score	MSE	ROC-AUC
Random Forest	95.60	95.63	99.83	0.97	0.03	0.96
XG-Boost	95.71	95.88	99.67	0.97	0.04	0.94
SVM	93.32	93.31	99.97	0.96	0.06	0.67
Capsule Networks	92.90	92.97	99.92	0.96	0.10	0.67
CNN	93.61	94.03	99.44	0.96	0.05	0.77
XG-Boost + Auto-encoder	93.30	94.42	98.61	0.96	0.06	0.61

Fig. 5. Metrics of Performance

Now let us look at other plots for comparison i.e,.,Precision Recall Curve which attributes to estimate the trade offs that exist between True Positive Rate and actual predicted positive value.

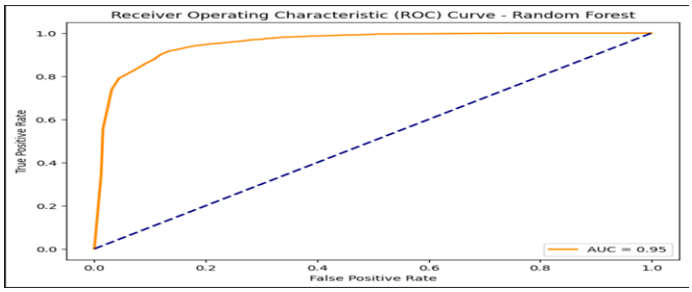


Fig. 6. ROC-RandomForestClassifier

The Fig6 represents the ROC Curve of Random forest. AUC of 0.95 indicates a very good model performance. i.e; it has a high ability to distinguish between positive and negative instances.

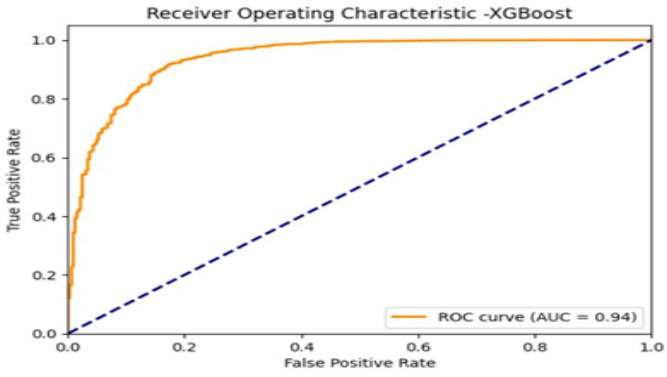


Fig. 7. ROC - XGBoost

The Fig7 represents the ROC Curve of XGBoost.With an AUC of 0.94 this model is also a very good model like Random Forest Classifier. i.e, it has a high ability to distinguish between positive and negative instances.

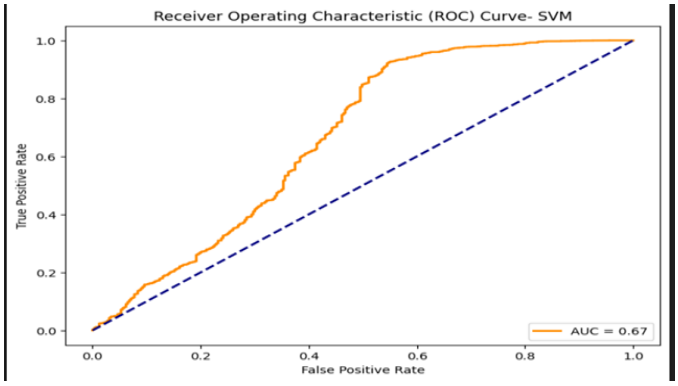


Fig. 8. ROC - SVM

The Fig8 represents the ROC Curve of SVM. AUC of 0.67 indicates moderate performance. i.e, there is a good balance between true positive rate and false positive rate.

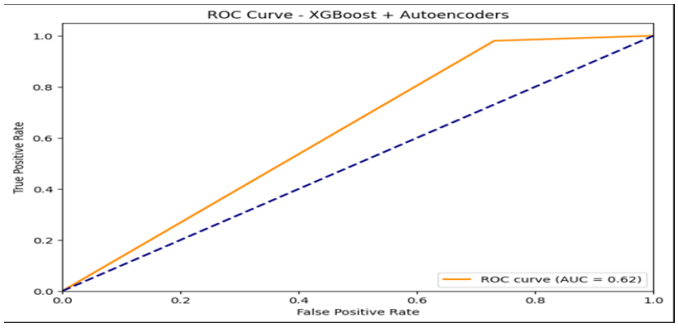


Fig. 9. ROC - XGBoost+Autoencoder

The Fig9 represents the ROC Curve of XGBoost+Autoencoders. AUC of 0.62 indicates lower performance compared to all the other models.

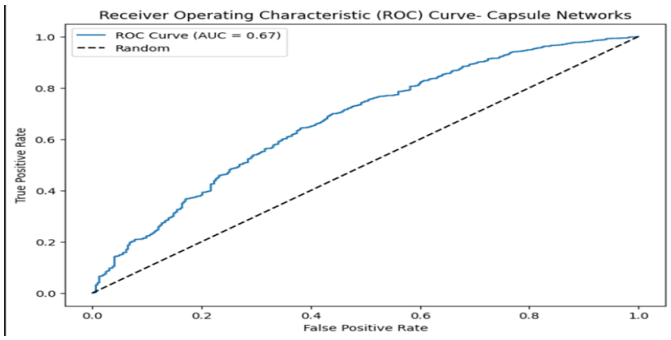


Fig. 10. ROC - Capsule Network

The Fig10 represents the ROC Curve of Capsule Network. AUC of 0.67 is similar to SVM suggesting comparable performance.

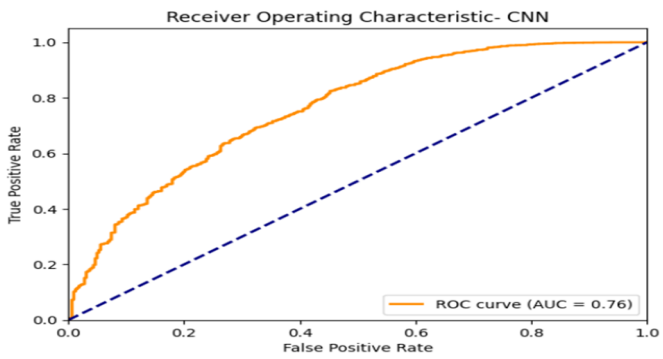


Fig. 11. ROC - CNN

The Fig11 represents the ROC Curve of CNN.AUC of 0.76 suggests that the model performed reasonably well i.e; there is a good balance between true positive rate and false positive rate.

By looking at the plots Fig 6 to Fig 11,again it is evident that RandomForest Classifier outshines other models. If we debate on the Mean Squared Error(MSE) values it it evident that RandomForestClassifier has the lowest value of 0.03 followed by XGBoost with 0.04 ,CNN with 0.05 , SVM and XGBoost+Autoencoders with 0.06 and lastly Capsule Network with 1.0. Hence we conclude that RandomForestClassifier is the model to be chosen for the Feature Importance analysis.

D. Feature Importance Estimation:

We make use of the trained RandomForest model above and add upon an interface for predicting and testing the results.Tkinter library of python was used to build GUI. The GUI checks firstly whether an interaction exists between the drugs and if at all the interaction is found out then it plots the feature importance of the two interacting drugs. The GUI also outputs the highest feature importance.

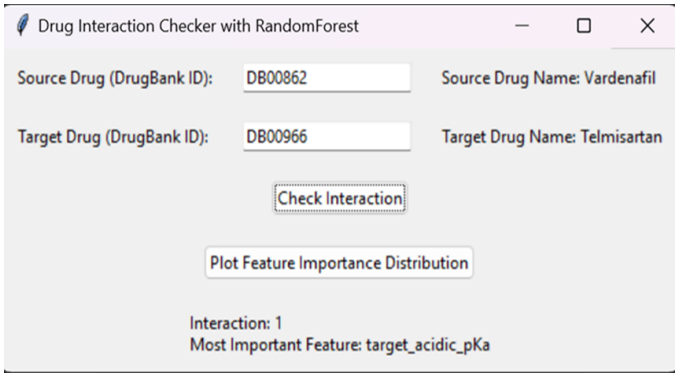


Fig. 12. GUI for interaction between drugs

Fig 12 depicts the interface developed using tkinter.It takes two inputs Source DrugBank ID and Target DrugBank ID.We

input the drugbank id's in the field. RandomForest Model trained on the dataset identifies the presence of interaction and outputs 1 if interaction exists between the drugs when clicked on 'Check Interaction' button.It also outputs their respective names and even the highest prevalent feature Importance.

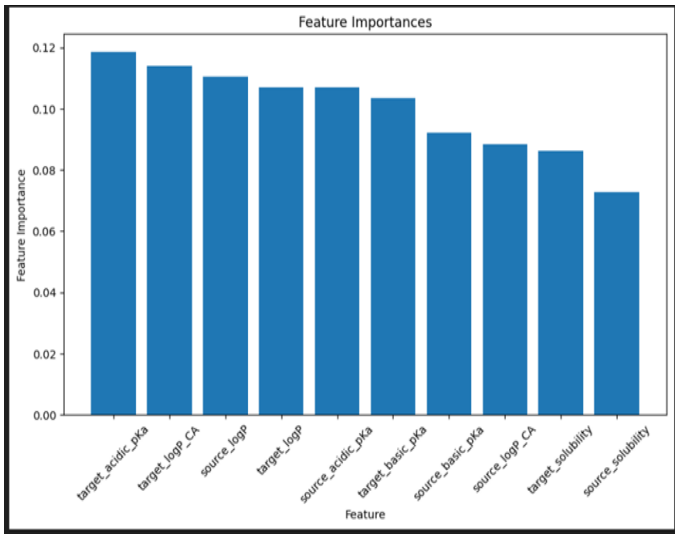


Fig. 13. Feature Importance Distribution

Next when clicked on 'Plot Feature Importance Distribution' the above graph Fig 13 is depicted on the screen.

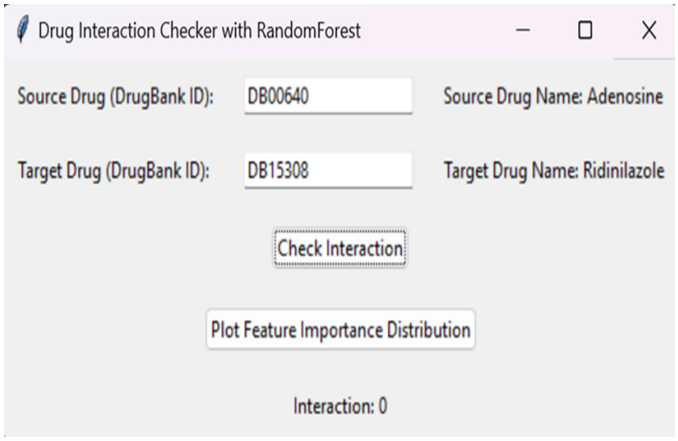


Fig. 14. GUI for no interaction between drugs

Fig 14 depicts two drugs posing no interaction when clicked on 'Check Interaction' button.If we Click on the button 'Plot feature Importance Distribution' button error message is displayed on the terminal 'Feature Importance graph exists only if there is an interaction.

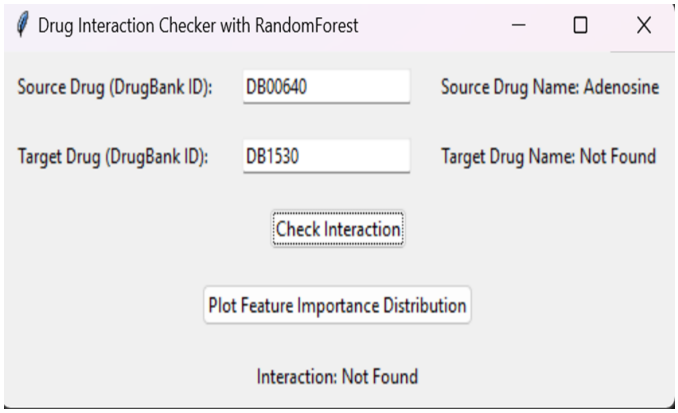


Fig. 15. GUI for interaction not found between drug

Fig 15 shows incorrect Target DrugBank Id which when clicked on 'Check Interaction' button displays that Interaction is not found.When clicked on 'Plot Feature Importance Distribution' error message will be displayed on the terminal showing 'No data found'.

V. RESULTS

The project yielded promising results. Accuracy metrics for the models showed strong performance, with Random Forest achieving an accuracy of 95.60 percentage, Support Vector Machine(SVM) achieving an accuracy of 93.3 percentage, XGBoost achieving an accuracy of 95.4 percentage.The system uses Random Forest algorithms to analyze interaction between drugs based on drug features and gives output of most responsible feature for interaction.

VI. CONCLUSION AND FUTURE WORK

The goal of the project was to train multiple models on drug-drug interactions. Evaluate and select the best model by comparing models performance metrics to predict the importance of features of two interacting drugs. We all know that not only these five major features of drugs are contributing to Drug-Drug Interaction there are numerous other attributes of drugs that influence the interaction.Our study was limited to two drugs and five major features but in real time multiple Drug-Drug Interaction could take place.In our future projects we aim to increase the number of features involved in the interaction as well as introduce the scope of learning multiple drug-drug interactions.

REFERENCES

[1] Xinkun Hao, Qingfeng Chen, Haiming Pan,Jie Qiu, Yuxiao Zhang,Qian Yu,Zongzhao Han,Xiaojing Du,"Enhancing drug–drug interaction prediction by three-way decision and knowledge graph embedding',Springler Nature,2022

[2] E. Kim and H. Nam, "Decide-ddi: interpretable prediction of drug drug interactions using drug-induced gene expressions," Journal of cheminformatics, vol. 14, no. 1, pp. 1–12, 2022

[3] ChengCheng Zang, Yao Lu ,Tianyi Zang ,"CNN-DDI: a learning-based method for predicting drug–drug interactions using convolution neural networks",BMC Bioinformatics,23,Article number :88(2022)

[4] Shaghayegh Sadeghi , Alioune Ngom, "DDI PRED:Graph Convolutional Network based Drug-Drug Interactions Prediction using Drug Chemical Structure Embedding" in IEEE, 2022.

[5] Nilay Fatma , Yıldız Alper Ozcan, "Graph Convolutional Autoencoder ¨ and Generative Adversarial Network-Based Method for Predicting DrugTarget Interactions " published in 2022.

[6] Xu Gong, Maotao Liu, Haichao Sun,Min Li, Qun Liu, "HS-DTI: DrugTarget Interaction Prediction based on Hierarchical Networks and MultiOrder Sequence Effect" published in 2022.

[7] Tianjung , Wang , Xin , Liu, "A Graph Convolution-Transformer Neural Network for Drug-Target Interaction Prediction", ICBBT 2022, May 27–29, 2022.

[8] Chang Sun, Ping Xuan , Tiangang Zhang and Yilin Ye, "Using Supervised Machine Learning Algorithms for Drug-Target Interaction Prediction" in IEEE/ACM Transactions On Computational Biology And Bioinformatics, Vol. 19, No. 1, January/February 2022.

[9] Mohammad A. Rezaei , Yanjun Li , Dapeng Wu, Xiaolin Li and Chenglong Li,"Deep Learning in Drug Design : Protein-Ligand Binding Affinity Prediction" in IEEE/ACM Transactions on Computational Biology and Bioinformatics, Volume: 19, Issue: 1, 01 Jan.-Feb. 2022.

[10] Nelson R. C. Monteiro , Bernardete Ribeiro , and Joel P. Arrais,"Drug Target Interaction Prediction : end-to-end Deep Learning Approach" in IEEE/ACM Transactions on Computational Biology and Bioinformatics, Volume: 18, Issue: 6 , 01 Nov.-Dec. 2021.

[11] Liu S., An J., Zhao J., Zhao S., Lv, H. Wang S, "Drug-Target Interaction Prediction based on Multisource Information Weighted Fusion" from Drug-Target Interaction Prediction Based on Multisource Information Weighted Fusion.Contrast media molecular imaging, 6044256, 2021.

[12] Shichao Liu, Yang Zhang, Yuxin Cui, Yang Qiu, Yifan Deng, Wen Zhang, Zhongfei Zhang, "Enhancing Drug-Drug Interaction Prediction Using Deep Attention Neural Networks ", IEEE TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS,2021.

[13] Mongia A, Majumdar A,"Drug-Target Interaction Prediction using Multi Graph Regularized Nuclear Norm Minimization", from PLoS ONE 15(1): e0226484, 2020.

[14] Xinyu Hao , Jiaying You , PingZhao Hu, "Predicting Drug-Drug Interactions using Deep Neural Networks", ICMLC '19, February 22–24, 2019..