Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Software

Predicting Drug Interactions using Graph Neural Networks

A Dissertation

Submitted to the Council of the College of Information Technology, University of Babylon in Partial Fulfillment of the Requirements for the Doctor of Philosophy Degree in Information Technology/ Software

By
Ali Kareem Abdul-Raheem Ahmed

Supervised by
Prof. Dr. Ban Nadeem Dhannoon

2024 A.D. **1445 A.H.**

سورة المجادلة / الاية 11

Declaration

I hereby declare that this Dissertation, submitted to University of Babylon in partial fulfillment of requirement for the degree of Ph.D. in Information Technology \ Software, has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose source are appropriately cited in the references.

Signature:

Name: Ali Kareem Abdul-Raheem Ahmed

Date: / / 2024

Supervisor Certification

I certify that this dissertation entitled "**Predicting Drug Interactions using Graph Neural Networks**" was prepared under my supervision at the Department of Software / College of Information Technology / University of Babylon, by **Ali Kareem Abdul-Raheem** as a partial fulfillment of the requirements of the degree of **Ph.D. in Information Technology / Software**.

Signature:

Name: **Dr. Ban Nadeem Dhannoon**

Title: **Professor**

Date: / / 2024

The Head of the Department Certification

In the view of available recommendations, I forward the dissertation entitled "**Predicting Drug Interactions using Graph Neural Networks**" for debate by the examination committee.

Signature:

Name: **Dr. Sura Zaki Alrashid**

Title: **Asst. Professor**

Date: / / 2024

Certification of the Examination Committee

We, the undersigned, certify that (**Ali Kareem Abdul-Raheem**) candidate for the degree of **Doctor of Philosophy in Information Technology - Software**, has presented his dissertation of the following title (**Predicting Drug Interactions using Graph Neural Networks**) as it appears on the title page and front cover of the dissertation that the said dissertation is acceptable with (**Excellent**) in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on: 1 February, 2024.

Signature:

Name: **Dr. Nabil Hashem Kaghed**
Title: Prof.
Date**:** **/ / 2024**
(Chairman)

Signature:

Name: **Dr. Israa Hadi Ali**
Title: Prof.
Date**:** **/ / 2024**
(Member)

Signature:

Name: **Dr. Eman Salih Al-Shamery**
Title: Prof.
Date**:** **/ / 2024**
(Member)

Signature:

Name: **Dr. Methaq Talib Gataa**
Title: Prof.
Date**:** **/ / 2024**
(Member)

Signature:

Name: **Dr. Sura Zaki Alrashid**
Title: Asst. Prof.
Date**:** **/ / 2024**
(Member)

Signature:

Name: **Dr. Ban Nadeem Dhannoon**
Title: Prof.
Date**:** **/ / 2024**
(Supervisor)

Approved by the Dean of the College of information technology, University of Babylon.
Signature:

Name: **Dr. Wesam S. Bhaya**
Title: **Prof.**
Date**:** **/ /2024**
(Dean of the College of information technology)

Dedication

To the savior of mankind, Al-Imam Al-Mehdi (peace be upon him), who will fill the earth with justice and equity, after it was filled with injustice and oppression.

To those who taught me, to those who put me on my first step...To my parents.

To my wife and babies

To my professors

To all who live in my memory

To my friends

To the martyrs in my country

I dedicate my work.

Acknowledgements

First of all, my great thanks to a mighty God who helped me to begin this work and finish it successfully.

I would also like to thank Dr. Ban N. Dhannoon who helped and guided me throughout the writing of this PhD thesis. I appreciate her support and the time she granted me to accomplish this work. I am thankful for all of her advice, knowledge, and critical remarks that have helped me improve this work.

I would like to take this opportunity to express my gratitude to the Computer Science Department at Al-Nahrain University and Karbala University and College of Information Technology at University of Babylon, which gave me the required knowledge and spirit to accomplish my PhD's Degree.

I would especially like to show my utmost appreciation to my parents, for them generous encouragement, guidance, and invaluable assistance during my study. Dad and Mam, I really appreciate your mentorship and inspiration toward the study of PhD degree.

I would like to declare my deep thanks and appreciation to my brothers and sisters for their patience, encouragement, and help during the work.

My wife, you know you are not only my wife but my life. Thank you for being so patient during the research period. You consistently challenging me to be a better person, and for reminding me what life is all about when I'm overwhelmed and stressed. You are my perfect complement.

My daughters, you are the secret of my happiness in life, thank you for your understanding of my shortcomings during the study period, and I thank your unlimited support, as I live for you.

Last but not least, but most importantly, I would like to thank all the kind, helpful and lovely people who helped me directly or indirectly to complete this work and apologize to them for not being able to mention them by name here, but they are in my heart.

Ali Kareem Abdul-Raheem

Abstract

The process of drug discovery is costly and time-consuming, which can lead to increased healthcare costs for patients. The identification of new drug–target interactions (DTI) is a substantial part of the drug discovery process. Drug repurposing is a promising approach to find new uses for approved drugs, but existing computational models that estimate the interaction strength of new drug-target pairs have limitations in how they represent molecules. Furthermore, predicting binding affinity values of drug-target pairs remains a challenge, even with the increasing availability of affinity data in DT knowledge-bases.

Therefore, there is a need to reduce the cost of predicting drug-target interactions (DTI) in order to accelerate drug discovery and make it more affordable. The dissertation aim to developing models to find the interaction between drugs and targets. These models can also be used to reuse drugs for a specific disease.

This dissertation introduces two deep learning models for drug-target interaction prediction: BiGRU-DTA and MPNN-BiGRU-DTA. The BiGRU-DTA used a bidirectional gated recurrent unit (Bi-GRU) to extract features from protein and drug sequences. MPNN-BiGRU-DTA employed two distinct neural network components: a message-passing neural network (MPNN) for processing drugs and a bidirectional gated recurrent unit (Bi-GRU) for processing proteins. Both models were rigorously assessed on the Davis and KIBA datasets.

The results demonstrated that both models exhibited high predictive capabilities. Upon careful examination of the results, it is evident that Model-2 consistently outperforms Model-1 across all three datasets. Specifically, on the Davis dataset, Model-2 demonstrates a comparable

MSE (0.202) compared to Model-1 (MSE: 0.237). Similarly, on the KIBA dataset, Model-2 achieves a MSE (0.204) in comparison to Model-1 (MSE: 0.200).

The performance of different drug-drug interaction (DDI) prediction methods was evaluated based on various evaluation metrics, including accuracy, area under the curve (AUC), F1-score, precision, and recall. The proposed method achieved the highest accuracy of 0.92, indicating its ability to predict DDIs accurately.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Abbreviation	Meaning
3CLpro	3C-like Protease
AA	Amino Acids
AI	Artificial intelligence
AUPR	Area Under the Precision-Recall curve
BIGP	Bi-gram Probabilities
Bi-GRU	Bidirectional Gated Recurrent Unit
BERT	Bidirectional Encoder Representations from Transformers
Bi-GRU	Bidirectional Gated Recurrent Unit
Bi-LSTM	Bidirectional Long Short-Term Memory
BLAST	Basic Local Alignment Search Tool
CADD	Computer-aided Drug Design
CNN	Convolution Neural Network
DD	Drug Discovery
DL	Deep Learning
DT	Drug-Target
DTA	Drug-Target Affinity
DDInter	Drug-Drug Interactions
DTIs	Drug-target Interactions
GCN	Graph Convolutional Network
GCNN	Graph Convolutional Neural Network
GGNN	Gated Graph Neural Network
GRU	Gated Recurrent Unit
GNN	Graph Neural Network
GPCR	G Protein-coupled Receptors
HoTS	Highlights on Target Sequences
KEGG	Kyoto Encyclopedia of Genes and Genomes

KG	Knowledge Graph
LBVS	Ligand-Based Virtual Screening
LPQ	Local Phase Quantization
ML	Machine Learning
MolTrans	MolTrans = Molecular Interaction Transformer
MPNN	Message-Passing Neural Network
NME	NME = Novel Molecular Entity
NR	NR = Nuclear Receptors
PCA	PCA = Principal Component Analysis
PSSM	PSSM = Position Specific Scoring Matrix
RdRp	RdRp = RNA-dependent RNA Polymerase
RNNs	Recurrent Neural Networks
SBVS	Structure-Based Virtual Screening
SMILES	Simplified Molecular Input Line Entry System

List of Dissertation-Related Publications

(First Paper)

Name of Journal: Current Drug Discovery Technologies.

Paper Title: Automating Drug Discovery using Machine Learning.

Scopus: Q3 (Free Fees) Cite Score = 3.7

Authors:

- Ali K. Abdul Raheem.
- Ban N. Dhannoon

(Second Paper)

Name of Journal: Current Drug Discovery Technologies.

Paper Title: Comprehensive Review on Drug-target Interaction Prediction – Latest Developments and Overview.

Scopus: Q3 (Free Fees) Cite Score = 3.7

Authors:

- Ali K. Abdul Raheem.
- Ban N. Dhannoon

(Third Paper)

Name of Journal: Baghdad Science Journal.

Paper Title: "Deep Learning based Models for Drug-Target Interactions.

Scopus Q2, Cite Score = 1.3

Authors:

- Ali K. Abdul Raheem.
- Ban N. Dhannoon

(Fourth Paper)

Name of Journal: Current Computer-Aided Drug Design.

Paper Title: A Novel Deep Learning Model for Drug-Drug Interactions.

Scopus: Q3, Cite Score = 2.9

Authors:

- Ali K. Abdul Raheem.
- Ban N. Dhannoon

(Fifth Paper)

Name of Journal: Communications in Computer and Information Science (Springer Nature).

Paper Title: Predicting Covid-19 Protein Interactions through Sequence Alignment.

Scopus: Q3, Cite Score = 1.0

Authors:

- Ali K. Abdul Raheem.
- Ban N. Dhannoon

Chapter One

General Introduction

General Introduction

1.1 Introduction

Chemical data collection, storage, analysis, and manipulation is the focus of the relatively new field of information technology known as cheminformatics. Cheminformatics was initially developed as a tool to help with drug research, but it is now utilized in a variety of biological, chemical, and biochemical domains. Cheminformatics refers to the use of computer and informatics approaches to a variety of chemical challenges. Pharmaceutical firms employ these *in-silico* approaches for developing new drugs [1].

Drug discovery and development have been sped up because of the advances in computational science. Artificial intelligence (AI) is widely used in both industry and academia. Machine learning (ML), an essential component of AI, has been used in a variety of contexts, including data production and analytics [2]. Drug discovery is one area that stands to gain significantly from this machine learning achievement. ML may be used to accelerate and minimize the labor-intensive and costly process of discovering novel medications [3][4].

Drug development, bioinformatics, and cheminformatics have benefited from introducing these computer-assisted computational techniques. Bringing a new medicine to market is complicated and time-consuming, costing pharmaceutical companies an average of $2.6 billion and 10 years of research and development [5]. There are several distinct steps to this process, each with its own set of obstacles, timetables, and costs:

- Discovery and Validation of Targets

- Identification of Lead Compounds (Drug-Target Interaction)
- Lead Optimization
- Preclinical and Clinical Drug Development

Drug-target interactions (DTIs) are an important stage in the drug discovery process. However, identifying drug-target interactions in Wet-lab (in vitro) research is highly expensive, time-consuming, and complex [6]. Computational (*in-silico*) techniques can successfully enhance traditional in vitro activity detection procedures, allowing the identification of interacting drug-target combinations and speeding up drug discovery. As a result, in recent years, in drug development and molecular pharmacology, computational approaches to predict drug-target interactions have emerged as a prominent research issue [7].

Drug repurposing (or drug repositioning) is the process of finding new medical uses for existing drugs. This can be done by testing drugs that have already been approved for one condition against other conditions, or by looking for new targets for existing drugs [8]. Therefore, the dissertation focused on developing models to find the interaction between drugs and targets. These models can also be used to reuse drugs for a specific disease.

1.2 Problem Statement

The identification of new drug-target interactions is a substantial part of the drug discovery process and predicting values of drug-target binding affinity remains a challenge. Therefore, there is a need to reduce the cost of predicting DTIs in order to accelerate drug discovery and make it more affordable. Drug repurposing is a promising approach to discovery new uses for existing medications, but existing computational models that

predict the interactions between new drug-target pairings have limitations in how they represent molecules. Furthermore, Drug-drug interactions can lead to altered drug concentrations in the body, increased or decreased therapeutic effects, or increased risk of adverse drug reactions.

1.3 Related Works

Several papers have been published in recent years that use deep learning techniques to process drug discovery issues. This section focuses on the most recent works in this topic as listed below.

1.3.1 Drug-Target Interactions

Öztürk et al. (2018), their study objective was to develop a method for calculating binding strength values of protein-ligand interactions, commonly known as binding affinity. They devised a deep learning based approach that predicts DT interaction binding affinities using just sequencing information from both targets and medicines. The findings demonstrated that the suggested deep learning-based model, which used 1D representations of targets and pharmaceuticals, it is a fantastic method for forecasting medication target binding affinity. By employing CNNs to construct high-level representations of a drug and a target, they found 0.878 for the CI and 0.261 for the MSE using the Davis dataset, and 0.863 and 0.194 for the CI and MSE using the KIBA dataset [9].

Wen T et al (2019), they offer a graph-convolutional (Graph-CNN) architecture with two stages for predicting protein-ligand interactions in this research. They originally presented a graph-autoencoder for unsupervised learning of fixed-size protein pocket representations Then, two Graph-CNNs are trained to extract features from pocket graphs and 2D molecular graphs automatically. On typical virtual screening

benchmark datasets, Graph-CNNs outperformed or were equivalent ligand-scoring 3D CNN, AutoDock Vina, Random Forest - Score, and NNScore across numerous criteria [10].

Tsubaki et al (2019), the authors studied the usage of end-to-end representation learning for compounds and proteins. In this study, integrated the representations, and created a novel Compound Protein Interaction (CPI) prediction approach by merging a graph neural network (GNN) for compounds and a convolutional neural network (CNN) for proteins. The results obtained from three CPI datasets indicate that the proposed end-to-end approach attains competitive or even superior performance when compared to a range of existing CPI prediction methods [11].

Öztürk et al. (2019), they suggested WideDTA is a deep-learning based prediction model that employs chemical and biological textual sequence information to predict binding affinity. And they discovered that the WideDTA uses four text-based information sources, namely the protein sequence, ligand SMILES, protein domains and motifs, and maximum common substructure words to predict binding affinity. WideDTA demonstrated superior performance compared to one of the state-of-the-art deep learning methods, DeepDTA, in predicting drug-target binding affinity on the KIBA dataset, and this superiority was statistically significant [12].

Lin et al (2020), the authors proposed a novel end-to-end learning framework, called DeepGS, which uses deep neural networks to extract the local chemical context from amino acids and SMILES sequences, as well as the molecular structure from the drugs. Also, they propose to use advanced embedding techniques (i.e., Smi2Vec and Prot2Vec) to encode the amino acids and SMILES sequences to a distributed representation to

assist the operations on the symbolic data. Meanwhile, they suggested a new molecular structure modeling approach that works well under their framework. Extensive experimental results with compare to state-of-the-art models demonstrate the superiorities and competitiveness of DeepGS [13].

Shao et al (2020), the study introduces a novel DTI prediction model. The model extracted features from drug and target expression profiles using a spectral-based graph convolutional network (GCN), and latent connections between drug and target using a CNN. Finally, the features that have been collected are concatenated and used to train an effective classifier. The benefit of DTIGCCN is that the extracted characteristics are more refined and focused, and the drug-target correlation is completely applied to the prediction. The experimental findings reveal that their model outperforms traditional DTI prediction approaches based on feature extraction and introduces a novel notion and method for DTI prediction [14].

Shim et al (2021), they devised a method for calculating binding affinity, which describes the strength of a DT pair's interaction. The drug may be ineffective if the binding affinity is inadequately high. As a result, methods for predicting DT binding affinities are particularly useful. To predict DT binding affinities, the researchers used a similarity-based approach in which for the medications and targets, the 2D CNN is used to compute the outer products of two similarity matrices. According to them, the proposed technique was the first to apply 2D CNN in DT binding affinity prediction based on similarity. The validation results on a large number of open data sets demonstrated that the model suggested is a viable technique for estimating DT binding affinity and may be

tremendously beneficial in the drug development process. They employed the Davis and KIBA datasets [15].

Nguyen et al (2021), the authors proposed a new DTA prediction model based on a combination of graph neural networks and conventional CNN. For the proteins, they used a string of American Standard Code for Information Interchange (ASCII) characters and apply several 1D CNN layers over the text to learn a sequence representation vector. Specifically, the protein sequence is first categorically encoded, then an embedding layer is added to the sequence where each (encoded) character is represented by a 128 dimensional vector. Next, three 1D convolutional layers are used to learn different levels of abstract features from the input. Finally, a max pooling layer is applied to get a representation vector of the input protein sequence. For drugs, they used the molecular graphs and trial four graph neural network variants, including Graph Convolutional Networks (GCN), Graph Attention Network (GAT), Graph Isomorphism Network (GIN) and a combined GAT-GCN architecture [16].

De Souza et al (2022) proposed MPS2IT-DTI, a DTI prediction model, is included in this study. It has been developed in stages: the development of a novel technique for visual encoding of chemical and protein sequences; the creation of a deep learning approach based on a convolutional neural network in order to provide a new way for DTI prediction; and the development using a deep-learning approach based on a convolutional neural network to create a novel DTI prediction method. In terms of neural network model performance and complexity, according on the training findings using the Davis and KIBA datasets, MPS2IT-DTI surpasses existing state-of-the-art (SOTA) approaches. They found 0.876 for the concordance index (CI) and 0.276 for the MSE using the Davis dataset, and 0.836 and 0.226 for the CI and MSE using the KIBA dataset,

respectively. Furthermore, because the MPS2IT-DTI model regarded molecular and protein sequences as pictures rather than as an NLP problem, it did not include an embedding layer, as seen in previous models [17].

D'Souza et al (2023) introduced a deep learning CNN model called DeepPS to predict unknown ligand–target interactions efficiently using one-dimensional SMILES for drugs and motif-rich binding pocket subsequences of proteins. The method outperforms shallow machine learning methods and is more computationally efficient when compared to using full-length raw sequences of proteins. They used the composition of ligands, proteins, and interactions in two benchmark datasets, Davis and KIBA. They found 0.854 for the CI and 0.353 for the MSE using the Davis dataset, and 0.844 and 0.218 for the CI and MSE using the KIBA dataset [18].

1.3.2 Drug Repurposing

Mukherjee et al (2022) proposed DeepGLSTM. It is an LSTM-based Graph Convolutional network drug repurposing strategy for SARS CoV-2 which forecasts the binding affinity values of Food and Drug Administration (FDA)-approved medicines and SARS-CoV-2 viral protein. Datasets from Metz, Davis, KIBA (Kinase Inhibitor Bioactivity) and DTC (Drug Target Commons) were used to train their suggested model. They predicted a Combined Score of 2,304 FDA-approved medicines against 5 viral proteins using their new design. They created a ranking of the top 18 drugs based on their affinity for the five viral proteins discovered in SARS CoV-2 based on the Combined Score. Following that, this list might be utilized to develop new helpful medications [19].

Ranjan et al (2022), their goal was to create a highly active chemical that could attach to the protein structure of SARS-CoV-2. The Gated Graph Neural Network (GGNN), Knowledge Graph, and Early Fusion methods are all used were utilized to construct a framework that takes tertiary and sequential molecule and protein representations into account. Before being input into the Early Fusion model, the produced molecules from GGNN are filtered using Knowledge Graph to decrease the search space by removing non-binding compounds. The resulting molecule's binding affinity score is also predicted utilizing the early fusion technique. The results of their experiments demonstrated that their framework created genuine and unique compounds with great precision while keeping chemical characteristics. The framework was also tested with two viral proteins from SARS-CoV-2: RNA-dependent RNA polymerase (RdRp) and 3C-like protease (3CLpro) MOSES dataset - DAVIS dataset [20].

1.3.3 Drug-Drug Interactions

A paper by Wang et al. (2020) proposed a deep learning-based method to predict potential DDIs by incorporating the chemical structure and pharmacological properties of drugs. The authors utilized a stacked auto-encoder model to extract drug features and a convolutional neural network for predicting DDIs. Their approach achieved an accuracy of 85.07%, outperforming traditional machine learning methods [21].

In another study, Wei et al. (2019) proposed a graph convolutional network-based approach for DDIs prediction. The authors utilized drug-drug interaction networks and molecular fingerprints as input to their model. Their approach achieved an accuracy of 88.42% and 0.881 F1-score, outperforming traditional machine learning methods such as decision trees and random forests [22].

Chen et al. (2020) proposed a hybrid deep learning model for predicting potential DDIs. Their model incorporated drug features, including molecular structure, biological processes, and side effects, as well as network-based features, such as drug-drug interaction networks. Their approach outperformed traditional machine learning methods, achieving an accuracy of up to 97.5% [23].

A study by Peng et al. (2020) combined deep learning and network analysis techniques for predicting serious DDIs. The authors utilized a deep neural network and drug interaction networks to predict a high-risk DDI based on drug combinations. Their approach achieved an accuracy of 93.1% and outperformed traditional machine learning methods [24].

Finally, Chen et al. (2020) proposed a deep learning-based model for identifying drug combinations that reduce the risk of drug-drug interactions. Their model used a Long Short-Term Memory (LSTM) network and attention mechanism to predict the probability of DDI occurrence. Their approach achieved promising results, with an AUC-ROC of 0.8431 and an accuracy of 0.8106 [25].

A paper by Li et al. (2020) proposed a novel method called DDIPLM for predicting drug-drug interactions based on pharmacological pathways, chemical structures, and side effect profiles of drugs. They utilized a patient-level matrix factorization model to identify potential DDIs. Their approach achieved an accuracy of up to 92.68%, outperforming traditional machine learning methods [26].

A summary of the previous studies that were conducted on the datasets used in our dissertation with additional details is listed in Table (1.1):

Table (1.1): Summary of Related Work

Title	Year	Method	Result (MSE)
DeepDTA: deep drug–target binding affinity prediction [9]	2018	Two CNN	0.261
WIDEDTA: Prediction of Drug-Target Binding Affinity [12]	2019	Four CNN	0.262
DeepGS: Deep Representation Learning of Graphs and Sequences for Drug-Target Binding Affinity Prediction [13]	2020	CNN + GAT + Smi2Vec	0.252
GraphDTA: predicting drug–target binding affinity with graph neural networks [16]	2021	GNN + CNN	0.254
DeepGLSTM: Deep Graph Convolutional Network and LSTM based approach for predicting drug-target binding affinity [19]	2022	GNN + LSTM	0.232
A Novel Deep Neural Network Technique for Drug–Target Interaction [17]	2022	CNN with image	0.276
Generating novel molecule for target protein (SARS-CoV-2) using drug–target interaction based on graph neural network [20]	2022	GGNN, Knowledge graph, and Early Fusion approach	0.223
Deep Learning-Based Modeling of Drug–Target Interaction Prediction Incorporating Binding Site Information of Proteins [18]	2023	CNN model	0.218

1.4 Dissertation Aim

The dissertation aims to developing models to find the interaction between drugs and targets. These models can also be used to reuse drugs for a specific disease.

- Develop a model for predicting drug-target interactions.
- Drugs repurposing by suggest a highly potent active molecules (drugs) that can bind with the protein of SARS-CoV-2.
- Develop model for predicting Drug-Drug interactions.

1.5 Dissertation Contributions

- Employing deep learning models to predict drug-target interactions (DTI) using Graph Neural Network (GNN).
- Drugs repurposing by generating a list of the top drugs with the highest binding affinity for the viral proteins present in SARS-CoV-2.
- Employing a deep learning architecture for predicting Drug-Drug Interactions and uses it to test the interactions of the top drugs.

1.6 Dissertation Outline

After Chapter One, which presents an introduction to the entire thesis, the rest of the thesis is structured as the following:

Chapter Two, "Theoretical Background": Many concepts will be introduced in this chapter, including datasets used in this dissertation, drug and protein representation, ML, DL, DTI, DTI, drug repurposing, and evaluation measures.

Chapter Three, "Methodology and Proposed System": This chapter first views the proposed methodology. Then, two parts will be discussed: the first part presents two DL models for DTI. Furthermore, the second part explains drug repurposing for SARS-CoV-2.

Chapter Four, "Results and Discussion": This chapter presents the results of the proposed models on all the datasets used in this dissertation. Then, the results will be compared with the previous studies conducted on these same datasets.

Chapter Five, "Conclusions and Future Works": Many conclusions drawn from our proposed models and suggestions for future work will be introduced.

Chapter Two

Theoretical Background

Theoretical Background

2.1 Introduction

In this chapter, the dissertation presents the theoretical foundations of automating drug discovery using deep learning. The chapter begins with the discussions of the representation of drugs and proteins, which serve as the fundamental building blocks of the drug discovery process. The chapter will also explore various deep learning techniques that have demonstrated utility in drug discovery. Additionally, the chapter will delve into specific deep learning models such as Bi-GRU (Bidirectional Gated Recurrent Units) and MPNN (Message Passing Neural Networks), which have shown promise in analyzing sequential data and graph-structured data, respectively.

Furthermore, the chapter discusses the applications of deep learning in different stages of the drug discovery process, including drug-target interaction prediction, drug properties prediction and de novo drug design. Lastly, it will touch upon related topics such drug-drug interactions, drug repurposing and evaluation measures. These areas are important for comprehensive understanding and analysis within the realm of automating drug discovery using deep learning.

2.2 Drug Representation

It became vital to express molecules in a syntax that computers could read and that scientists from different domains could comprehend. Chemical representations have multiplied throughout time due to the rapid advancement of computers and the difficulty of creating a representation that incorporates every structural and chemical property. The following sections show the most important types of representations by which we

can represent compounds and be understood by the computer [27]. A drug (compound) can be represented in a variety of ways as shown in Figure (2.1).

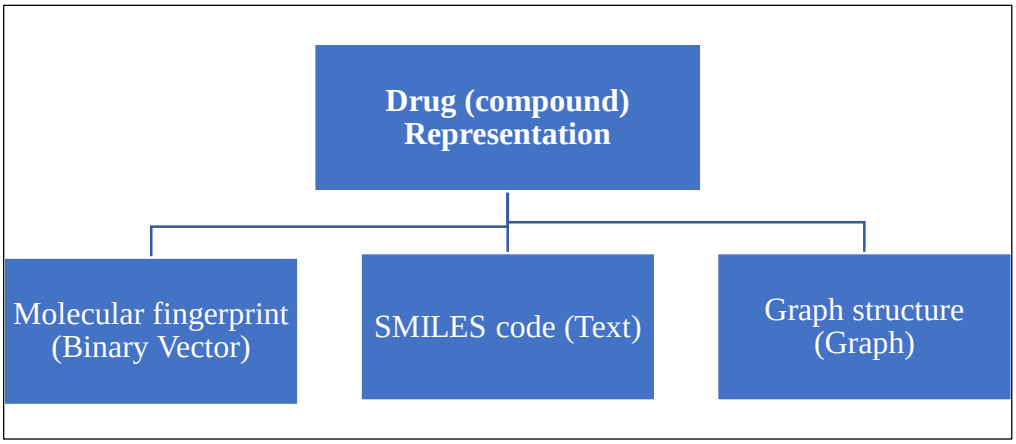


Figure 2.1: Methods of Drug (compound) Representation

2.2.1 Molecular Fingerprint

In chemical informatics, a molecular "fingerprint" is a commonly used notion. Small compounds, such as drugs, perform best with fingerprints [28]. It is difficult to compare molecules; however, it is easy to compare bit strings. Many individuals convert a molecule to a bit string and compare it in the hope of learning how to compare molecules. As shown in Figure (2.2), each bit in a structural fingerprint corresponds to a chemical property, generally the existence of some form of substructure, as an array (vector) of ones and zeros.

However, it should be noted that it is not possible to completely characterize the drug (compound) through fingerprint, so we may lose some of the characteristics of the drug during this representation.

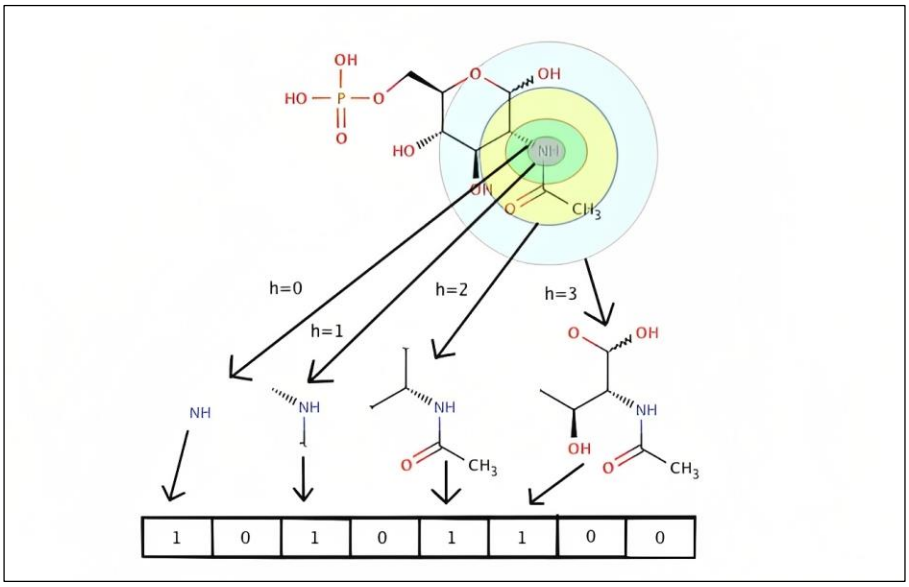


Figure 2.2: Molecular fingerprint

2.2.2 SMILES Code

Chemical notation known as SMILES (Simplified Molecular Input Line Entry System) enables users to represent chemical structures in a computer-readable way. SMILES is a tool that converts a drug's structure into a linear representation of the molecule, as shown in Figure (2.3) so that a computer program can interpret it. Basic syntactic requirements must be followed while using SMILES. The representation will fail if basic chemistry rules are not followed in the SMILES entry; for example, if the user sets too many bonds on an atom [29].

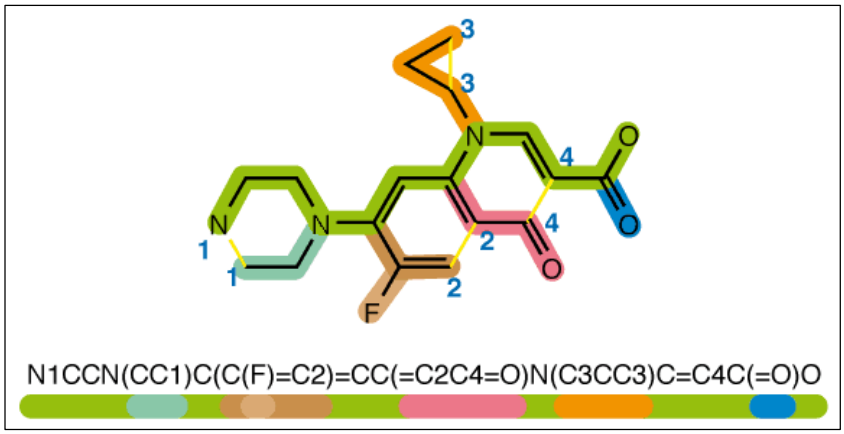


Figure 2.3: SMILES code

In SMILES, the labels or unique letters represent atoms, bonds, and other molecular features. The specific set of labels used in SMILES can vary depending on the context and the molecules being represented. However, some common labels are frequently encountered in SMILES notation. The most commonly used labels for SMILES [30]:

1. Atoms:
    a. Elements: The labels for elements are typically derived from their atomic symbols. For example, "C" represents carbon, "O" represents oxygen, "N" represents nitrogen, and so on.
    b. Parentheses: Parentheses are used to represent atomic groups or substructures within the molecule. They are typically represented by the characters "(" and ")".
2. Bonds:
    a. Single Bond: The most common label for a single bond is represented by the character "-".
    b. Double Bond: Double bonds are often represented by the character "=".
    c. Triple Bond: Triple bonds are often represented by the character "#".
    d. Aromatic Bond: Aromatic bonds in aromatic compounds are often represented by the character ":".
    e. Ring Closure: Ring closures are represented by numeric digits to indicate the connection between atoms within a ring.
3. Branching and Rings:
    a. Branching: Branches in the molecular structure are indicated by using parentheses "(" and ")" around the branch.
    b. Ring Formation: Rings are represented by numeric digits (0-9) to indicate the connection between atoms in the ring.

4. Charges and Isotopes:
    a. Charges: Charges on atoms are represented by the characters "+" and "-" followed by numeric digits.
    b. Isotopes: Isotopes of elements are represented by specifying the mass number before the atomic symbol. For example, "C13" represents carbon-13.
5. Aromaticity and Special Symbols:
    a. Aromatic Atoms: Aromatic atoms are often denoted by lowercase letters, such as "c" for aromatic carbon.
    b. Aromatic Rings: Aromatic rings are often indicated by enclosing the ring with lowercase characters, such as "c1ccccc1" for a benzene ring.

It's important to note that SMILES notation allows for flexibility, and the use of specific labels can vary depending on the specific molecule or context. Additionally, extensions and modifications to the SMILES notation have been proposed to represent more complex molecular structures or to incorporate additional features [31].

While there are no strict rules regarding the maximum number of unique symbols in SMILES, a commonly used set includes a total of 64 labels. These labels consist of uppercase letters, lowercase letters, and a few special characters. Here is an example set of 64 labels for SMILES:

1. Uppercase letters (26):

    A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

2. Lowercase letters (26):

    a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z

    Special characters (12): @, %, &, #, (,), [,], =, +, -, .

These 64 labels provide a broad representation of atoms, bonds, and molecular features that can be encountered in SMILES notation. However, it's worth noting that the use of specific symbols may vary depending on the particular SMILES implementation or the specific molecules being represented

2.2.3 Graph-structured Data

Creating graph-structured data and an adjacency matrix is another technique to preprocess a chemical structure. The adjacency matrix holds information on the connectivity of atoms, with "1" indicating a link and "0" indicating no connection. Each bond in a molecule is converted into an edge in the graph, and each atom is converted into a node in the network, as shown in Figure (2.4) [26].

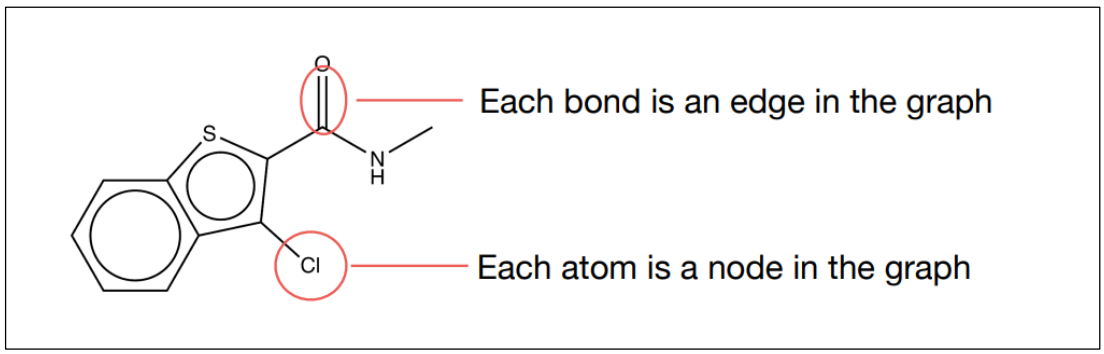


Figure 2.4: Graph-Representation

2.3 Protein Representation

Proteins are crucial biomolecules involved in various biological processes and serve as important targets for drug discovery. Effectively representing proteins is essential for understanding their structure, function, and interactions [32].

Amino acid sequences provide a primary level of protein representation and are often the first step in characterizing proteins. In this

representation, proteins are represented as linear chains of amino acids, where each amino acid is denoted by a specific letter code. Amino acid sequences capture the primary structure of proteins and contain valuable information about the sequence's evolutionary conservation, post-translational modifications, and potential functional regions [33].

The representation of amino acid sequences is commonly used in various bioinformatics analyses, including sequence alignment, motif discovery, and phylogenetic studies. These analyses enable researchers to compare proteins, identify conserved regions, predict functional sites, and infer evolutionary relationships [34].

There are twenty amino acids (AA) used as a section of developing proteins. Each codon (genetic word comprised of just three letters) of AA characterized by symbol or one letter as described in Table (2.1).

Table 2.1: Symbols of Amino Acid [35].

Name	Symbol (3 letters)	Symbol (1 letters)	Name	Symbol (3 letters)	Symbol (1 letters)
Histidine	His	H	leucine	Leu	L
Alanine	Ala	A	Serine	Ser	S
Tyrosine	Tyr	Y	Cysteine	Cys	C
Methionine	Met	M	Isoleucine	Ile	I
Glutamine	Gln	Q	Valine	Val	V
Asparagine	Asn	N	Phenylalanine	Phe	F
Lysine	Lys	K	Glycine	Gly	G
Aspartate	Asp	D	Threonine	Thr	T
Arginine	Arg	R	Glutamate	Glu	E
Tryptophan	Trp	W	Proline	Pro	P

2.4 Data Collection

In every stage of the drug discovery process, the datasets are crucial for the development and viability of effective ML algorithms. The dependence on large amounts of high-quality data and well-defined training sets is even more important in precision medicine and drug development. To ensure reliable and comprehensive analysis, data were collected from various resources. For drug-target interaction prediction, the Davis and KIBA datasets were utilized. These datasets provide experimentally validated interaction data between drugs and their respective protein targets. For drug-drug interaction prediction, accessed the DrugBank and DDInter datasets, which contain detailed information on drug properties, interactions, and adverse effects. Moreover, the NCBI website was utilized to retrieve protein sequence data related to Covid-19.

2.4.1 Davis and KIBA datasets

To construct robust drug-target interaction prediction models, the proposed system acquired data from two widely-used datasets: The Davis dataset and the KIBA dataset. The Davis dataset contains experimentally validated drug-target interactions, encompassing a diverse range of drug compounds and target proteins. Similarly, the KIBA dataset offers a comprehensive collection of interaction data, which includes affinity values, enabling the development of precise models.

The Davis dataset contains 30,056 measurements of interactions between 442 drugs and 68 proteins. This dataset is widely used to evaluate DTI prediction models due to its large size and the diversity of its compounds and targets. The KIBA dataset contains a curated set of kinase inhibitors with known kinase activities and binding affinities. It consists

of 2,111 compounds and 229 protein targets, making it another valuable dataset for DTI prediction.

Both the Davis and KIBA datasets have been widely used in deep learning-based approaches for DTI prediction. The datasets have also been used to develop models for predicting drug-target binding affinity (DTA), which is a more challenging problem than binary classification. Overall, the Davis and KIBA datasets are valuable resources for researchers studying DTI and DTA prediction and have contributed to the development of many successful DL-based models in the field. The datasets in the formats utilized in the system trials are summarized in Table (3.1) [9] [12] [13] [16].

Table 2.2: Summary of the datasets

	Proteins	**Compounds**	**Interactions**
Davis	442	68	30 056
KIBA	229	2111	118 254

2.4.2 DrugBank and DDInter

For predicting Drug-Drug interactions, the proposed system sourced data from two prominent databases: DrugBank and DDInter. DrugBank provides a comprehensive repository of drug-related information, including known interactions between different drugs. DDInter, on the other hand, specifically focuses on Drug-Drug interactions, providing valuable insights into potential synergistic or antagonistic effects.

DrugBank is a well-known database that is commonly used as a drug reference resource. This database was originally made available in 2006. DrugBank, as a database in both bioinformatics and cheminformatics, offers thorough drug data as well as comprehensive drug target information. DrugBank's DTI connections were compiled from textbooks, published publications, and other electronic sources. DrugBank Supporting academic research and student education is a major aspect of DrugBank's purpose, and in addition to commercial products, the company continues to provide free, non-commercial datasets to academic researchers, as well as a free version of DrugBank data, which can be found at go.drugbank.com [36].

DDInter is a comprehensive, professional, and open-access drug-drug interaction database. It contains extensive annotations for each DDI connection, such as mechanism descriptions, risk levels, management options, alternative drugs, and so on, to improve clinical decision-making and patient safety [37].

2.4.3 The NCBI website

To investigate protein-protein interactions and explore drug repurposing opportunities for COVID-19, protein sequences of viral protein related to SARS-CoV-2 were collected. The protein sequence data was obtained from reputable sources, with the National Center for Biotechnology Information (NCBI) website serving as a primary source.

The NCBI website is a comprehensive online resource that provides access to a wide range of biological and biomedical information. It serves as a central repository for various databases, tools, and resources that are essential for researchers, scientists, and healthcare professionals working

in the field of life sciences. The NCBI website offers access to databases such as PubMed, which is a vast collection of scientific articles and publications in the field of medicine and life sciences.

Additionally, the NCBI website hosts other databases like GenBank, which is a comprehensive collection of DNA sequences, and the Protein database, which contains information on protein sequences and structures. These databases provide researchers with access to a vast amount of genetic and molecular information that is crucial for various studies, including sequences of viral proteins relevant to COVID-19 [38].

2.5 Deep Learning

Deep Learning is a subset of machine learning that focuses on training deep neural networks to learn and extract complex patterns from data [39]. It has gained significant attention and achieved remarkable success in various domains, including computer vision, natural language processing, and speech recognition.

Deep learning involves the use of deep neural networks, which are composed of multiple layers of interconnected nodes, called artificial neurons or units [40]. Each layer receives input from the previous layer and applies a set of mathematical operations to transform the data. The depth of the network refers to the number of layers it contains. Deep neural networks can learn hierarchical representations of data, allowing them to capture intricate patterns and relationships.

One of the key advantages of deep learning is its ability to automatically learn useful features from raw data. Unlike traditional machine learning approaches, which often require handcrafted features, deep learning models can learn features directly from the data through a

process known as feature learning or representation learning. This eliminates the need for manual feature engineering and enables the model to discover relevant representations that optimize performance.

Deep learning architectures commonly used include Convolutional Neural Networks (CNNs) for image analysis, Recurrent Neural Networks (RNNs) for sequential data, and Transformer models for natural language processing tasks [40]. These architectures have been instrumental in achieving state-of-the-art results in various domains.

Recent advancements in deep learning have focused on improving model performance, scalability, and interpretability [41]. Techniques such as transfer learning, which leverages pre-trained models on large-scale datasets, have enabled effective knowledge transfer and improved performance on smaller datasets. Additionally, regularization methods such as dropout and batch normalization help prevent overfitting and enhance generalization.

Deep learning has also benefited from advances in hardware and computational resources [42]. Graphics Processing Units (GPUs) and specialized hardware, like Tensor Processing Units (TPUs), have accelerated deep learning training and inference. Furthermore, frameworks like TensorFlow, PyTorch, and Keras have simplified the development and deployment of deep learning models.

2.5.1 Gated Recurrent Unit

The Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture introduced by Cho et al. in 2014 [43]. It was developed as a variation of the standard RNN to address the vanishing gradient problem and improve the modeling of sequential data. The GRU

incorporates gating mechanisms that allow it to selectively update and forget information over time.

Bidirectional GRU (Bi-GRU) is an extension of the GRU architecture that enhances the capabilities of GRU by processing the input sequence in both forward and backward directions simultaneously. This bidirectional processing enables Bi-GRU to capture information from both past and future contexts, facilitating a better understanding of the sequence and extraction of more meaningful representations [44].

The primary difference between GRU and Bi-GRU lies in the directionality of sequence processing. GRU processes the sequence in a unidirectional manner, either left-to-right or right-to-left, while Bi-GRU processes the sequence in both directions simultaneously. This bidirectional processing allows Bi-GRU to take into account information from both the past and future contexts, resulting in a more comprehensive representation of the input sequence.

Bi-GRU operates by independently processing the input sequence in both the forward and backward directions. At each time step, the forward GRU unit computes its hidden state based on the current input and its previous hidden state. Similarly, the backward GRU unit computes its hidden state based on the current input and its previous hidden state [45].

The structure of a GRU unit is shown in Figure (2.6) for forward GRU and can be represented as follows [46]:

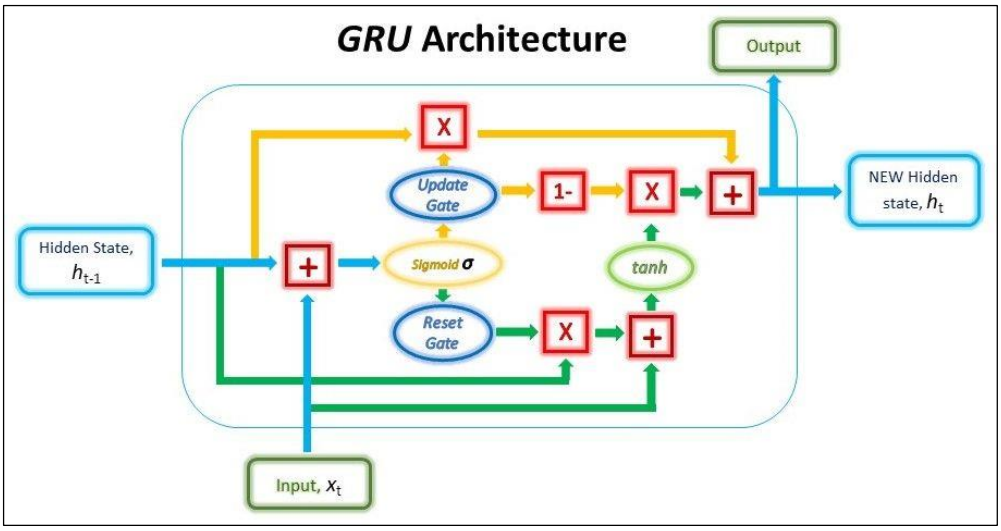


Figure 2.5: The structure of a GRU unit [46].

Reset Gate

This gate is derived and calculated using both the hidden state from the previous time step and the input data at the current time step as shown in Figure (2.7).

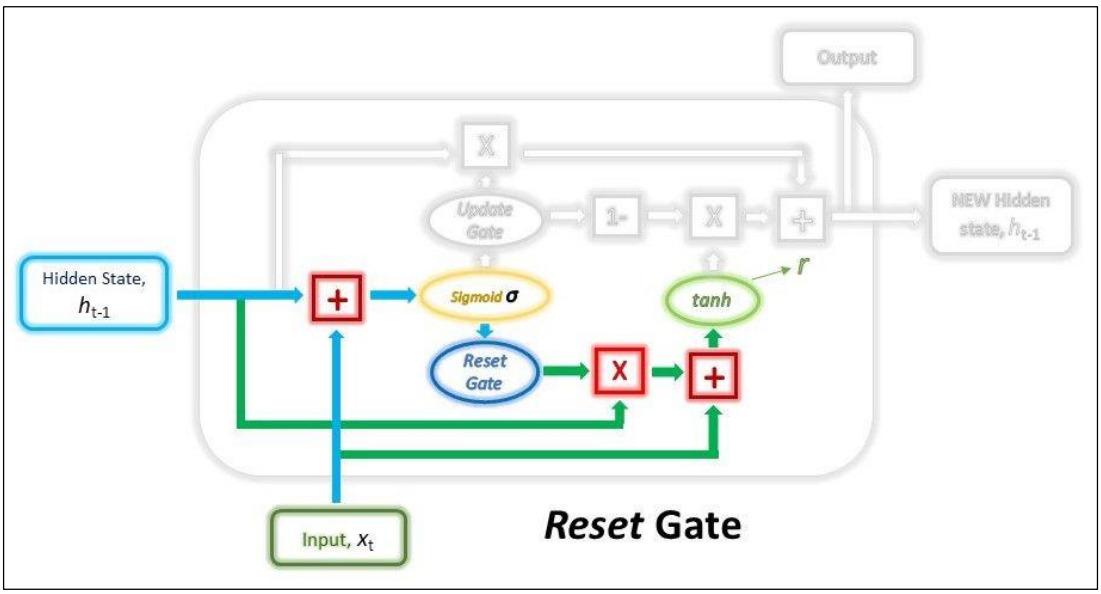


Figure 2.6: The Reset Gate [46].

Mathematically, this is achieved by multiplying the previous hidden state and current input with their respective weights and summing them before passing the sum through a sigmoid function.

The sigmoid function will transform the values to fall between 0 and 1, allowing the gate to filter between the less-important and more-important information in the subsequent steps.

$$gatereset=\sigma(Winputreset \cdot xt+Whiddenreset \cdot h_{t-1}) \quad (2.1)$$

where *Xt* is current input and h_{t-1} is the previous hidden state. When the entire network is trained through back-propagation, the weights in the equation will be updated such that the vector will learn to retain only the useful features.

The previous hidden state will first be multiplied by a trainable weight and will then undergo an element-wise multiplication (Hadamard product) with the reset vector. This operation will decide which information is to be kept from the previous time steps together with the new inputs. At the same time, the current input will also be multiplied by a trainable weight before being summed with the product of the reset vector and the previous hidden state above. Lastly, a non-linear activation tanh function will be applied to the final result to obtain r in the equation (2.2).

$$r=tanh(gatereset \odot (W_{h1} \cdot h_{t-1})+W_{x1} \cdot x_t) \quad (2.2)$$

where r is the output of reset gate, Xt is current input, h_{t-1} is the previous hidden state and *W* is the weight.

Update Gate

Just like the Reset gate, the update gate is computed using the previous hidden state and current input data as shown in Figure (2.8).

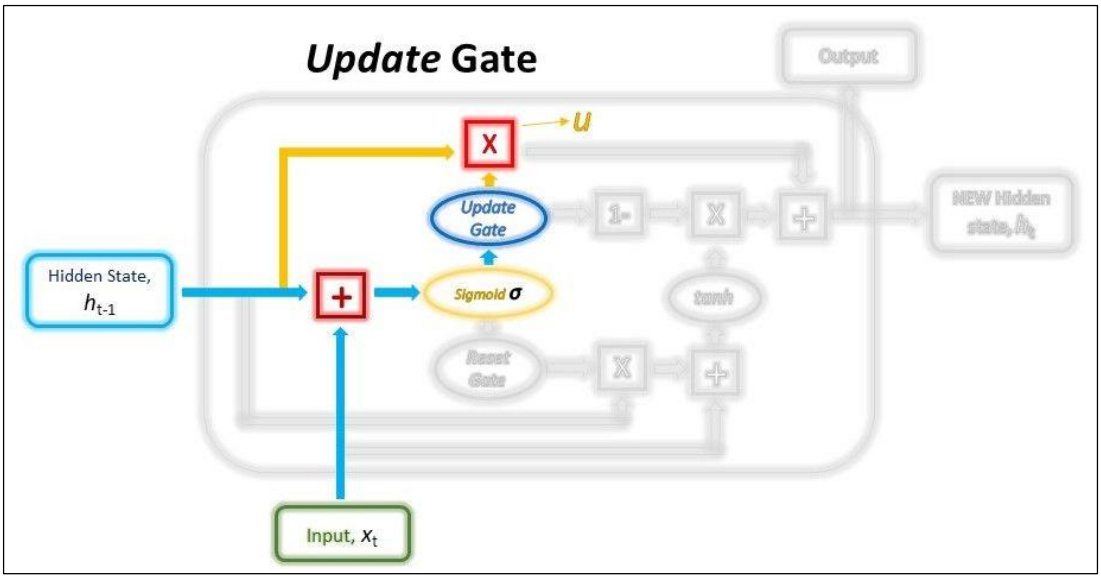


Figure 2.7: The Update Gate [46].

Both the Update and Reset gate vectors are created using the same formula, but, the weights multiplied with the input and hidden state are unique to each gate, which means that the final vectors for each gate are different. This allows the gates to serve their specific purposes.

$$gateupdate=\sigma(Winputupdate \cdot x_t + Whiddenupdate \cdot h_{t-1}) \quad (2.3)$$

The Update vector will then undergo element-wise multiplication with the previous hidden state to obtain u in equation below, which will be used to compute our final output later.

$$u=gateupdate \odot h_{t-1} \quad (2.4)$$

The Update vector will also be used in another operation later when obtaining our final output. The purpose of the Update gate here is to help the model determine how much of the past information stored in the previous hidden state needs to be retained for the future.

Combining the outputs

In the last step, we will be reusing the Update gate and obtaining the updated hidden state as shown in Figure (2.9).

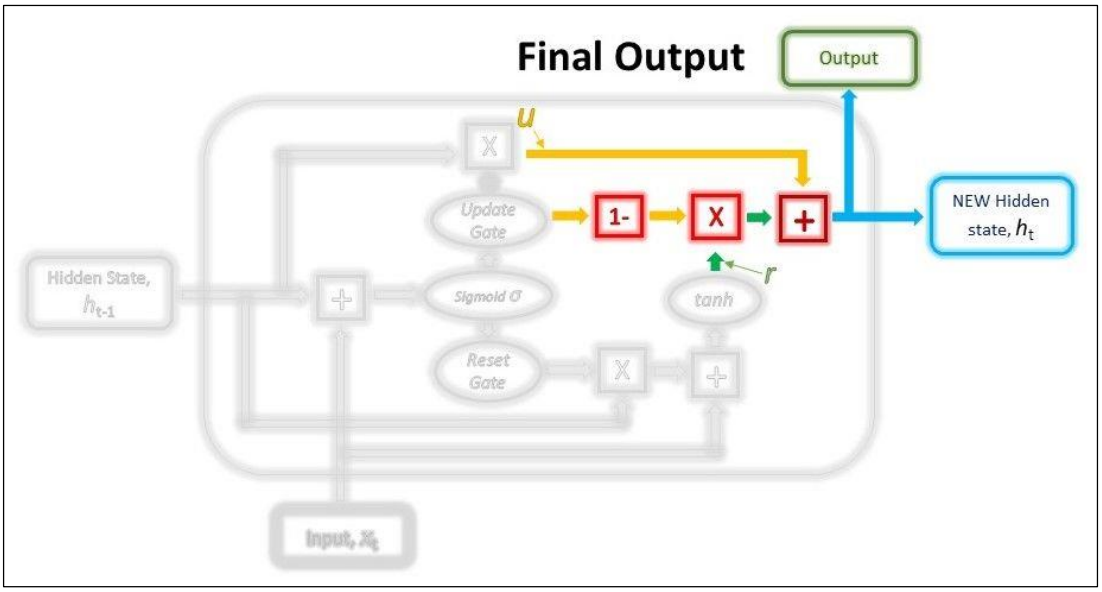


Figure 2.8: Combining the outputs [46].

This time, we will be taking the element-wise inverse version of the same Update vector (1 - Update gate) and doing an element-wise multiplication with our output from the Reset gate, r. The purpose of this operation is for the Update gate to determine what portion of the new information should be stored in the hidden state.

Lastly, the result from the above operations will be summed with our output from the Update gate in the previous step, u. This will give us our new and updated hidden state.

$$ht=r\odot(1-gateupdate)+uh \tag{2.5}$$

The new hidden state output for that time step as well by passing it through a linear activation layer. As mentioned, the Bi-GRU is constructed by two unidirectional GRUs facing opposing directions. The forward GRU starts from the beginning of the time series data, and the

backward GRU starts from the end of the time series data. The Bi-GRU is calculated by two GRUs and can be formulated as Equation (2.6)–(2.8).

$$h\vec{}_t = GRUfwd(xt,,,h\vec{}_{t-1}) \quad (2.6)$$

$$h\leftarrow_t = GRUbwd(xt,,,h\leftarrow_{t-1}) \quad (2.7)$$

$$ht = h\vec{}_t \oplus h\leftarrow t \quad (2.8)$$

where, $h\vec{}\,t$ and $h\leftarrow t$ are the state information of the forward and backward GRU, respectively. *GRUfwd* is the forward GRU, and *GRUbwd* is the backward GRU, the GRU function is composed of Equation (2.1) – Equation (2.5). $\oplus$ denotes concatenating the $h\vec{}\,t$ and $h\leftarrow t$. Therefore, the Bi-GRU with bi-directional GRU structures can memory the traffic flow information from historical and subsequent time series data.

2.5.2 Graph Neural Networks

Graph Neural Networks (GNNs) are a powerful class of deep learning models designed specifically to process and analyze graph-structured data. Graphs are widely used to represent complex relationships and interactions between entities such as social networks, citation networks, chemical compounds, recommendation systems, and knowledge graphs. GNNs have gained significant attention in recent years due to their ability to capture the underlying structural information of graphs and make predictions based on that information [47].

The foundation of GNNs lies in their ability to effectively capture and model the complex relationships and dependencies present in graph-structured data. Unlike traditional neural networks that operate on fixed-size vectors or images, GNNs operate on graph structures, which consist of nodes (vertices) connected by edges (links). Each node in the graph

represents an entity, and the edges represent the relationships or interactions between the entities.

At the core of GNNs is the concept of message passing. The key idea is to propagate information across the graph by passing messages between nodes and their neighbors. In each message passing step, nodes exchange information, update their representations, and aggregate information from neighboring nodes. By iteratively performing this message passing process, GNNs allow information to flow through the graph and capture both local and global patterns.

The message passing mechanism in GNNs can be mathematically represented as follows: at each layer or iteration, the representation of a node is updated based on the representations of its neighboring nodes and the edges connecting them. This update process takes into account the node's own features and the features of its neighbors, allowing the model to encode the local structural information [48].

One popular class of GNNs is the Message Passing Neural Network (MPNN). Introduced by Gilmer et al. in 2017 [49], MPNNs leverage the concept of message passing, where nodes exchange information with their neighbors iteratively. MPNNs have demonstrated remarkable success in various graph-related tasks, including node classification, graph classification, and link prediction. They have been widely used in chemistry for molecular property prediction, in social network analysis for node attribute prediction, and in citation networks for document classification.

MPNNs are composed of two main components: a message passing phase and a readout phase. The message passing phase involves passing

information between nodes in a graph, while the readout phase aggregates the node representations to generate a graph-level output.

1- The message passing phase

The message passing phase in MPNNs involves two main steps: message computation and node update. During message computation, a learnable function or neural network calculates messages from one node to its neighbors, considering the current node representations and edge information. These messages encode local information from neighboring nodes.

The update step aggregates the received messages from neighboring nodes and combines them with the current node representation to generate an updated representation. By iteratively performing these message passing and node update steps, MPNNs allow information to propagate and refine across the entire graph.

2- Readout Phase:

Once the message passing phase is completed, the MPNN aggregates the node representations to generate a graph-level output. This can involve various operations such as summation, pooling, or attention mechanisms. The choice of the readout operation depends on the specific task and the nature of the graph data.

Each node in the graph has a hidden state (i.e. feature vector). For each node Vt, aggregate a function of hidden states and possibly edges of all neighboring nodes with the node Vt itself. Then, update the hidden state of the node Vt using the obtained message and the previous hidden state of that node.

Three main equations define the MPNN framework on graphs [49]. The message obtained from neighboring nodes is given by the following equation:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \tag{2.9}$$

It is a sum of all messages M_t obtained from the neighbors. M_t is an arbitrary function that depends on hidden states and edges of the neighboring nodes e_{vw}. then can simplify this function by leaving some of the input arguments. In the example above, can only sum different hidden states h_w.

Then, update the hidden state of the node Vt using a simple equation:

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{2.10}$$

Simply speaking, the hidden state of the node Vt is obtained by updating the old hidden state with the newly obtained message m_v. In the case of the example above, the update function U_t is an average between the previous hidden state and the message. Finally, repeat this message passing algorithm for a specified number of times. After that, reach the final readout phase.

$$\hat{y} = R(\{h_v^T \mid v \in G\}). \tag{2.11}$$

In this step, extract all newly updated hidden states and create a final feature vector y describing the whole graph. This feature vector can be then used as input to a standard machine learning model. The structure of MPNN is shown in Figure (2.10) [50].

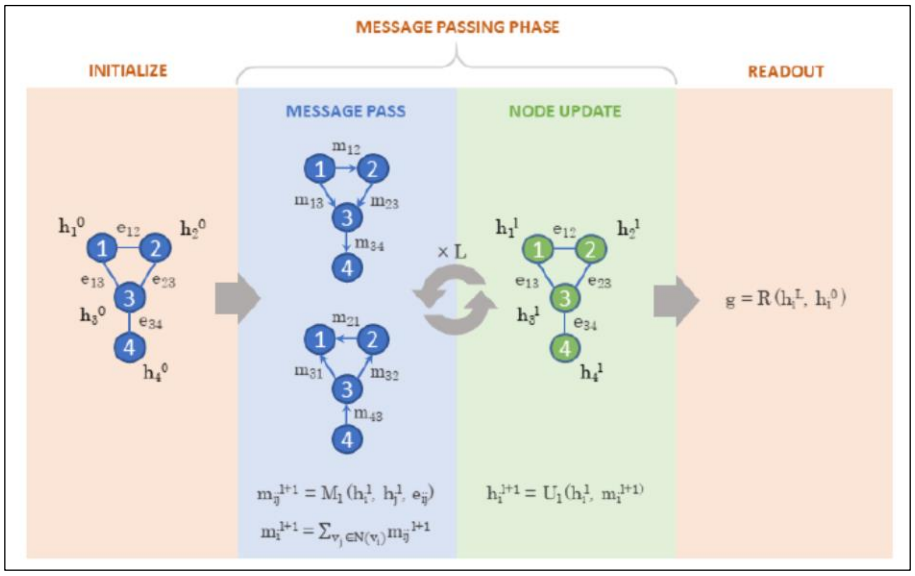


Figure (2.9) Schematic of a MPNN [50]

2.6 The Applications of Deep Learning in Drug Discovery Steps

The use of machine learning tools and techniques has broadened to include a diverse set of applications. Recent advancements in cheminformatics approaches and data mining applications have made the work of drug discovery for many researchers more difficult. These methods and applications can lower expenses, boost the standard of healthcare, and facilitate better decision-making. We will review the most significant implementation of ML in drug discovery. The three primary domains of machine learning applications in medication development and drug discovery process are as follows [51]:

1. Drug-Target Interaction Prediction

This field determines whether or not a medication can bind to specific proteins. Interactions between known drugs and targets are difficult to find. The application of machine learning methods to anticipate the target interactions of drugs can help speed up the time-

consuming and expensive experimental work. In addition to providing insights into potential drug-drug interactions, in silico DTI prediction can aid in the research on the side effects of the drug. The descriptors are given to represent the medications and target proteins that have a significant impact on DTI prediction performance [52].

2. Drug Properties Prediction

This field aims to evaluate and test hundreds or even thousands of chemicals for certain features. The three main categories of machine learning issues are reinforcement learning, self-supervised learning, and supervised learning. Predicting drug properties could be considered a supervised learning challenge. A drug (compound) is the field's input, and a drug property is the associated output [53].

3. De Novo Drug Design

This field creates a chemical with certain features. For instance, it creates a drug that can bind to a particular protein and activate certain pathways without affecting others and has some physical properties, such as a specific range of solubility. The early stages of drug development include the design and optimization of targeted drug-like molecules [54].

2.7 Drug-Target Interaction Prediction

When a medicine (a chemical molecule) binds to a specific target (such as proteins or nucleic acids), It alters its biological activity or function, restoring it to normal [55]. DTI prediction is an important aspect of the DD process since it can both expedite and save costs [56], However It is challenging and expensive since experimental assays take a long time and are pricey [57]. As a result, researchers have increased their efforts to identify the association between medications and targets in the hopes of

speeding up medication development and reducing time to market. Based on existing DT trials, computer-generated DTI predictions can be applied to efficiently evaluate the potency of novel drug-target interactions (DT) combinations, Thus, when dealing with a vast amount of complex information, the DD procedure is accelerated by systematically recommending a fresh set of potential molecules (e.g., Interactions (hydrogen bonding, hydrophobic, ionic, and/or van der Waals forces) between molecules) [58].

The study of drug protein interactions is a critical problem in the study of pharmaceutical sciences. Drugs are chemical compounds or molecules that produce physical or physiological changes in the human body when injected. These medications bind to various proteins or nucleic acids in the human body. Proteins are known as biological targets. Thus, DTIs are the study of the interactions between distinct drug compounds and target proteins to aid in the development of therapeutic drugs [59]. Figure (2.11) depicts the drug target interaction process. The drug's chemical compound attaches to the desired molecule via creating temporary ties, as indicated in the figure. The linked medication then binds to the biological target, either positively or negatively, and leaves the biological target.

In order to treat disorders, the medications inhibit the target's function by preventing particular catalytic reactions from occurring in the human body. This is accomplished by preventing it from interacting with specific enzymes known as substrates [60].

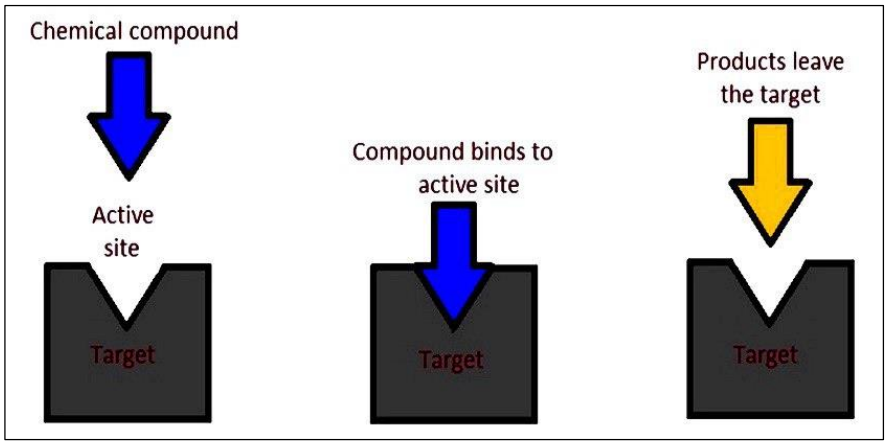


Figure 2.10: The process of DTI [60]

2.7.1 Applications of Drug-Target Interaction

Drug target interaction prediction covers a wide range of applications. It contributes to the discovery of new drugs [60]. It also aids in drug repositioning [61]. Furthermore, identifying medication target negative interactions can help with drug side effect prediction [62].

The study of novel medications that interact with a certain target is known as drug discovery. DTI prediction in silico can aid in the discovery of novel medicines that bind to certain targets. Finding a novel medication is a costly and inefficient process. It consists of many phases, identifying the chemical's lead constituent that binds to the target protein, discovering a specific target that binds to a chemical compound, and lead optimization to increase efficiency and specificity are all part of the process [63]. Following all of these stages, the medications are subjected to a series of clinical studies before being released to the market. Each novel molecular entity (NME) identification costs around $1.8 billion. Furthermore, it takes about a decade for newly discovered medicinal molecules to arrive in the consumer market. The US Food and Drug Administration, for example, only authorizes roughly 20 new medications every year [64].

Furthermore, certain medications have been demonstrated to interact with several target locations, a phenomenon known as polypharmacology [66]. Polypharmacology has improved the medication repositioning, also known as drug repurposing, process, which refers to using an existing medicine to treat new ailments. This approach has several benefits. To begin, a previously licensed pharmaceutical has gone through rigorous clinical trials and testing before being put to the market.

The examination of existing drugs and their interaction patterns can lead to the discovery of numerous advantages of a single medicine These medications can then be utilized to treat a variety of ailments. Gleevec (imatinib mesylate, for example) was hypothesized to communicate with the leukemia-associated Bcr-Abl fusion gene. It was later discovered to communicate with PGDF and KIT Later, Gleevec was repurposed to treat gastrointestinal stromal tumors [65].

2.7.2 Drug-Target Interaction Methods

There are three types of computational approaches for predicting drug-target interactions: Approaches for docking simulation, ligand-based methods, and machine learning. There are three types of in silico DTI prediction models for machine learning: chemogenomic approaches, DL-based models, and network-based models. Figure (2.12) depicts the broad classification of chemogenomic methods there are two types of methodologies: feature-based methods and similarity-based methods.

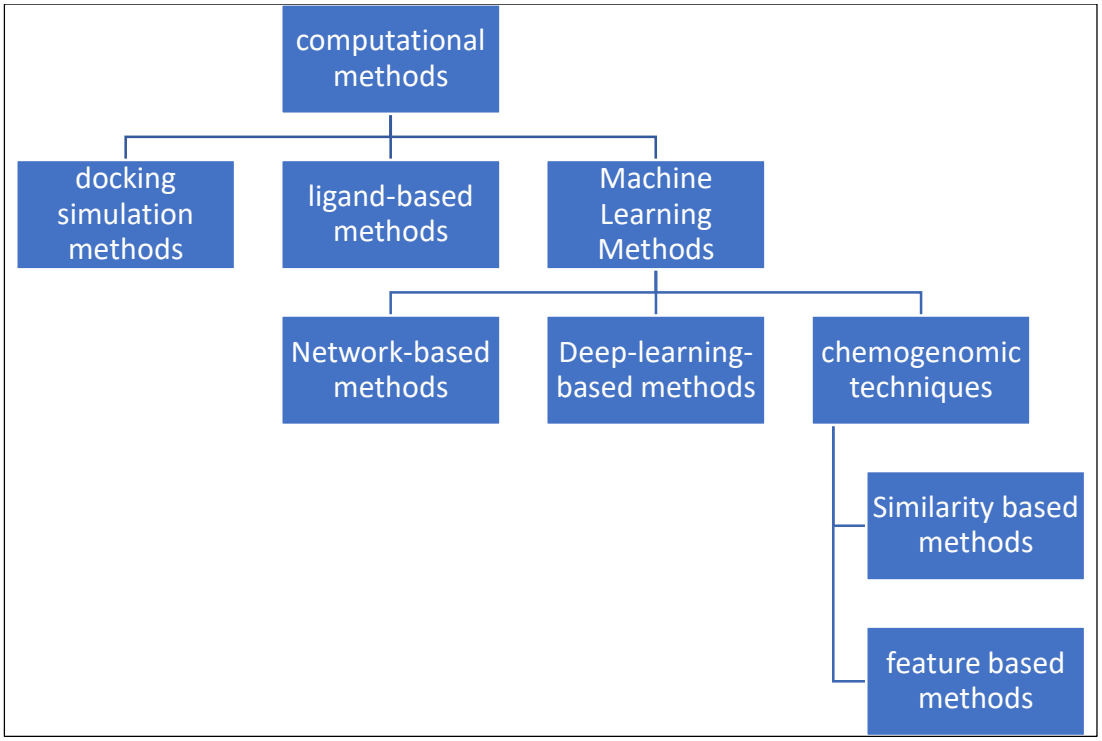


Figure 2.11: Branch diagram of recent computational methods for DTI prediction.

- Docking simulation methods: The docking simulation approach necessitates the use of the target protein's usable three-dimensional (3D) structural information, which is only available for a small number of proteins as a result, it cannot be used to foresee large-scale DTIs [67]. While docking simulation approaches are important, they are ineffective when there are no three-dimensional (3D) target structures [68].
- Ligand-based methods: Due to the restricted number of known ligands, ligand-based techniques often do not function well with target proteins. Ligand-based approaches work well when there are a high number of a target protein's known ligands, however they are inappropriate for large-scale data [69].
- Feature based methods: The training data is represented using feature vectors. Examples include drug-target interactions, and

feature vectors may be created by combining (structural) chemical descriptors of drugs with target sequences. Then, using these inputs, machine learning models such as Random Forest may be used to forecast based on these vectors [70].

- Similarity based methods: Similarity-based approaches rely heavily on similarities between samples. However, because drug and target distributions are so complicated, it is difficult to construct a reasonable similarity computation approach [71]. Two similarity matrices are used in similarity-based approaches: (a) drug similarity (based on chemical structures) and (b) target similarity (produced by protein sequence alignment) [72]. They are based on the concept that comparable medications have similar targets and vice versa [73].
- Network-based models: DTI prediction may be expressed as a problem of link prediction in a heterogeneous network [74, 75].
- Deep-learning-based: Deep-learning-based approaches can direct studies, and discover hidden trends in massive drug or protein data sets, the majority of methods depend on features to construct feature descriptors (fingerprints) for both medications and targets. The DT pair is then used to train a classifier to predict DTI [76].

2.8 Sequence Alignment

Sequence alignment is a fundamental technique in bioinformatics that plays a crucial role in comparing and analyzing biological sequences, such as DNA, RNA, and protein sequences. It is widely used to identify similarities, infer evolutionary relationships, and understand functional and structural aspects of biomolecules. This section explores the concept

of sequence alignment and discusses different algorithms and methods used in this field [77].

Sequence alignment involves arranging two or more sequences in such a way that their homologous positions are matched and aligned. The goal is to identify regions of similarity or conservation and detect differences or variations between sequences. By aligning sequences, researchers can gain insights into the evolutionary relationships, functional motifs, and conserved regions within biological sequences [78].

2.8.1 BLAST Algorithm

BLAST (Basic Local Alignment Search Tool) is a widely used algorithm for sequence alignment and similarity searching. BLAST rapidly identifies regions of similarity between a query sequence and a database of known sequences, enabling researchers to infer evolutionary relationships, identify homologous genes, and annotate functional elements in genomic sequences [79].

The BLAST algorithm employs several key techniques, including the construction of a lookup table of short words (k-mers) from the query sequence, the calculation of local alignment scores using a substitution matrix, and the use of statistical measures to assess the significance of sequence similarity. BLAST also utilizes optimizations such as indexing and database organization to improve search speed and efficiency.

BLAST Algorithm Overview [79]:

a. Preprocessing: The algorithm constructs a lookup table of short words (k-mers) from the query sequence and the database sequences to facilitate rapid searching.

b. Scoring: BLAST assigns scores to matches, mismatches, and gaps based on a substitution matrix.
c. Seed Search: BLAST identifies short exact matches (seeds) between the query sequence and the database sequences using the lookup table.
d. Extension: The algorithm extends the seeds by applying a dynamic programming approach to calculate local alignment scores and identify high-scoring segment pairs (HSPs).
e. Scoring Statistics: BLAST uses statistical measures, such as the E-value, to assess the significance of the sequence similarity.
f. Output: The algorithm reports the identified HSPs and generates an alignment between the query sequence and the matching database sequence.

BLAST Scoring:

a. Match/Mismatch Score: BLAST assigns scores to matches and mismatches using a substitution matrix. The score represents the similarity or dissimilarity between two aligned residues.
b. Gap Penalties: BLAST employs gap penalties for introducing gaps (insertions or deletions) in the alignment. It assigns separate gap opening and extension penalties to control the gap lengths. These penalties penalize the introduction of gaps; as longer gaps are less likely than shorter ones.

Extension and Dynamic Programming:

BLAST uses a variant of the dynamic programming algorithm to extend the initial seeds (exact matches) and identify HSPs. It calculates local alignment scores by iteratively extending the alignment in both

directions (left and right) from the seed. The algorithm optimizes the extension process to maximize speed while maintaining sensitivity.

Scoring Statistics:

BLAST provides statistical measures to assess the significance of sequence similarity. One commonly used statistic is the E-value, which estimates the expected number of false positive alignments by chance. A lower E-value indicates higher significance. The E-value is calculated based on the alignment score, database size, and composition of the database.

Equations:

a. Alignment Score (S): The alignment score is the sum of the match/mismatch scores and the gap penalties for the aligned residues in the query and database sequences.
b. E-value: The E-value estimates the expected number of alignments with a score equal to or better than the observed score, purely by chance. It is calculated based on the alignment score (S), the effective database size (m), and a scaling parameter (K) using the formula:

$$E\text{-}value = K * m * n * exp(-lambda * S) \quad (2.12)$$

Here, n is the query sequence length, lambda is the scaling parameter, and K is a constant that depends on the scoring system and database size.

2.9 Drug Repurposing

Drug repurposing, also known as drug repositioning or drug reprofiling, refers to the process of identifying new therapeutic uses for existing drugs that were originally developed for a different purpose. This

approach offers several advantages over traditional drug discovery and development, including reduced time and costs, as well as a potentially higher success rate due to the existing safety and pharmacokinetic profiles of repurposed drugs [80].

One notable example of successful drug repurposing is the use of sildenafil, originally developed for treating hypertension and angina, for the treatment of erectile dysfunction [81]. This discovery not only provided a new treatment option for a previously underserved patient population, but also resulted in significant economic benefits for the pharmaceutical industry.

The identification of drug repurposing opportunities can be facilitated by various strategies and approaches, such as computational methods, high-throughput screening of drug libraries, analysis of electronic health records, and drug combination screenings [82]. These approaches allow researchers to explore the vast amount of existing drug compounds and their potential therapeutic applications. The following Figure (2.13) shows the drug repurposing.

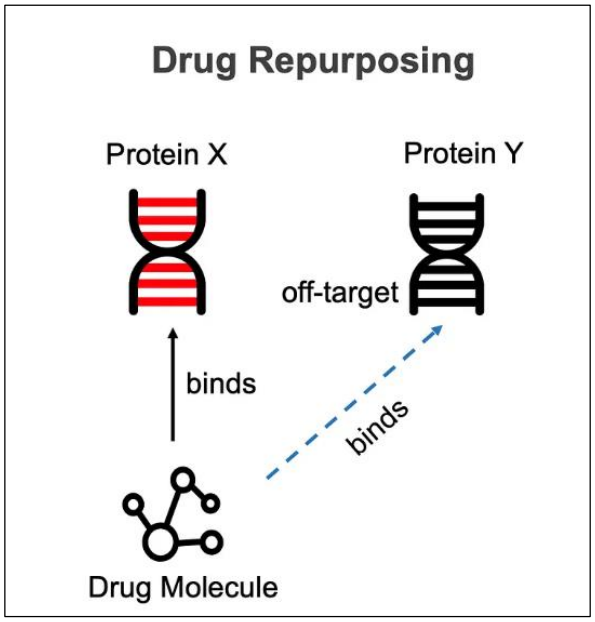


Figure 2.12 Drug Repurposing

2.10 Virtual Screening

In the drug development process, virtual screening (VS) is an in-silico approach. VS uses computational approaches to automatically analyses vast databases of molecular structures. It is envisaged that the application of VS will find compounds that are more receptive to binding to the molecular target, which is often a protein or enzyme receptor [83].

As seen in Figure (2.14), compounds that have the potential to become good medications are initially chosen because they have features that favor their action or are like drugs with known functions.

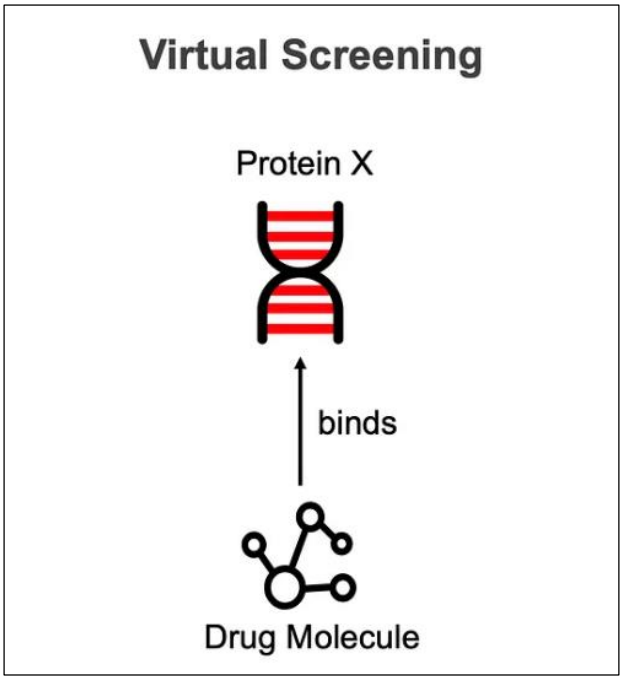


Figure 2.13. Virtual screening process.

Furthermore, the VS enables the selection of compounds from a database of structures that have a higher likelihood of exhibiting biological activity against a target of interest. VS can also remove chemicals that are potentially hazardous or have negative pharmacologic and pharmacokinetic characteristics. In this regard, biological assays

should be done solely on the most promising compounds, resulting in cheaper development costs and a shorter development time [84].

There are two main types of VS: ligand-based virtual screening (LBVS) and structure-based virtual screening (SBVS).

- LBVS uses the information about known ligands that bind to the target protein to identify new ligands that may also bind to the target protein. This information can include the molecular structure of the known ligands, as well as their biological activity data.
- SBVS uses the three-dimensional structure of the target protein to identify ligands that are likely to fit into the binding site of the protein. This is done by docking the ligands into the binding site and calculating the binding energy.

SBVS, also known as target-based VS (TBVS), predicts the optimum binding orientation of two molecules to form a stable compound. The SBVS methodology includes tools for investigating the 3D structure of a molecular target. When the 3D structure of the molecular target has been determined, SBVS is the recommended approach [85]. SBVS attempts to estimate the likelihood of coupling between prospective ligands and target proteins by considering the complex's binding strength. Molecular docking is the most often utilized SBVS approach [86].

Molecular docking is a computational technique widely employed in drug discovery to predict how a ligand (drug candidate) will bind to a target protein. Molecular docking takes into account the three-dimensional structure of both the ligand and the target protein, as well as the forces that drive binding. It plays a crucial role in identifying potential drug candidates by simulating the interaction between ligands and proteins at the atomic level.

2.11 Drug-Drug interactions

Drug-drug interactions occur when two or more drugs interact with each other, resulting in changes in their pharmacokinetic or pharmacodynamic properties. These interactions can lead to altered drug concentrations in the body, increased or decreased therapeutic effects, or increased risk of adverse drug reactions. For example, concomitant use of warfarin and aspirin has been associated with an increased risk of bleeding due to their additive effects on altering blood clotting mechanisms [87]. Similarly, co-administration of certain antibiotics with oral contraceptives can decrease contraceptive efficacy by reducing the concentration of hormonal components in the body [88].

Healthcare professionals need to be aware of potential drug-drug interactions and consider them when making treatment decisions. A thorough understanding of drug metabolism and drug-drug interactions can help prevent unexpected adverse events and optimize therapeutic outcomes. Multiple databases and resources are available to aid in assessing drug-drug interactions, such as DrugBank, Lexicomp, and DDiner [89]. These platforms provide comprehensive information on drug interactions, including mechanisms, clinical significance, and management strategies.

Machine learning algorithms can learn patterns and relationships from large datasets, making them well-suited for DDI prediction. Various machine learning methods, including decision trees, support vector machines, random forests, and Bayesian networks, have been employed to predict DDIs by integrating drug properties, molecular structures, and clinical data. For instance, in [90] used a random forest model to predict potential DDIs based on drug structure and pharmacological properties.

Another study by Luo et al [91] employed a support vector machine to predict DDIs using chemical and genomic information.

2.12 Evaluation Measures

Evaluation measures play a crucial role in assessing the performance and effectiveness of various algorithms, models, or systems in a wide range of fields, including machine learning, information retrieval, and data mining. These measures provide quantitative assessments and enable comparisons between different approaches. In this section, the dissertation will discuss various evaluation measures commonly used in machine learning, including classification and regression.

2.12.1 Classification Evaluation Measures

Classification is a supervised learning task in which the goal is to predict the class or category of an input instance based on its features. The input data consists of a set of instances, where each instance is described by a set of features or attributes. The target variable, also known as the class label, represents the category or class that the instance belongs to.

In classification, the model learns patterns and relationships from labeled training data and uses them to make predictions on unseen instances. The output of a classification model is a discrete class label or a probability distribution over the possible classes [92].

Classification Evaluation Measures [93]:

- Accuracy: Accuracy is a widely used evaluation measure in machine learning, which calculates the ratio of correctly predicted instances to the total number of instances in the dataset. The accuracy is expressed as a value between 0 and 1, where 1

represents perfect accuracy. The equation to calculate accuracy is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.13)$$

- Precision and Recall: Precision and recall are commonly used measures, particularly in binary classification tasks. Precision represents the proportion of true positives among the instances predicted as positive, while recall measures the proportion of true positives that are correctly identified. These measures are essential, especially in cases where there is class imbalance. The formulas for precision and recall are as follows:

$$Precision = \frac{TP}{TP + FP} \quad (2.14)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.15)$$

- F1 Score: The F1 score combines precision and recall into a single metric, providing a balanced evaluation of the model's performance. It calculates the harmonic mean of precision and recall and is given by the equation:

$$F - Measure = \frac{2(Precision * Recall)}{Precision + Recall} \quad (2.16)$$

2.12.2 Regression Evaluation Measures

Regression is also a supervised learning task that focuses on predicting continuous or numerical values based on input features. Unlike classification, where the output is a discrete class label, regression aims to estimate a continuous output variable. In regression, the model learns the relationship between the input features and the target variable by fitting a function that best represents the data. The output of a regression

model is a numerical value or a range of values, depending on the nature of the problem [94].

Mean Squared Error (MSE): MSE is a commonly used evaluation measure in regression tasks. It measures the average squared difference between the predicted and actual values [95]. The average is taken over all instances in the dataset. The equation to calculate MSE is:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2, \tag{2.17}$$

Chapter Three

The Proposed Methodology

The Proposed Methodology

3.1 Introduction

In this chapter, the dissertation provides the employed methodology, focusing on the prediction of DTI, the repurposing of drugs for the treatment of COVID-19, the prediction of drug-drug interactions and details the steps taken to develop models and collect relevant data for the analysis. Accurate prediction of DTI is crucial in drug discovery and development. By understanding the interactions between drugs and their respective protein, researcher can identify potential targets for drug molecules. To achieve this, the dissertation utilizes various computational models and algorithms to predict the likelihood of DTI.

3.2 The Proposed Methodology

The general proposed methodology diagram presented in Figure (3.1). It provides a holistic view of the drug-target interaction prediction pipeline, showcasing the integration of deep learning models with input data and the overall workflow. This system architecture forms the basis for the subsequent detailed explanation of the specific deep learning models and their contributions to predicting drug-target interactions.

The proposed methodology presents the development of two DL models for DTI's prediction: BiGRU-DTA and MPNN-BiGRU-DTA. The first model uses Bidirectional Gated Recurrent Units (BiGRU) while the second model use combination of Message Passing Neural Networks (MPNNs) with BiGRU. Also, it describes the deep learning architecture using MPNN for predicting Drug-Drug interactions.

Furthermore, given the urgency to find effective treatments for COVID-19, drug repurposing has emerged as a viable option, by examining existing drugs and their interactions with target proteins. For that, this dissertation aims to identify potential drugs that can be repurposed for treating COVID-19 by predicting the binding affinity of existing drugs with COVID-19 proteins and identifying potential drug candidates for repurposing, offering therapeutic options to combat the disease.

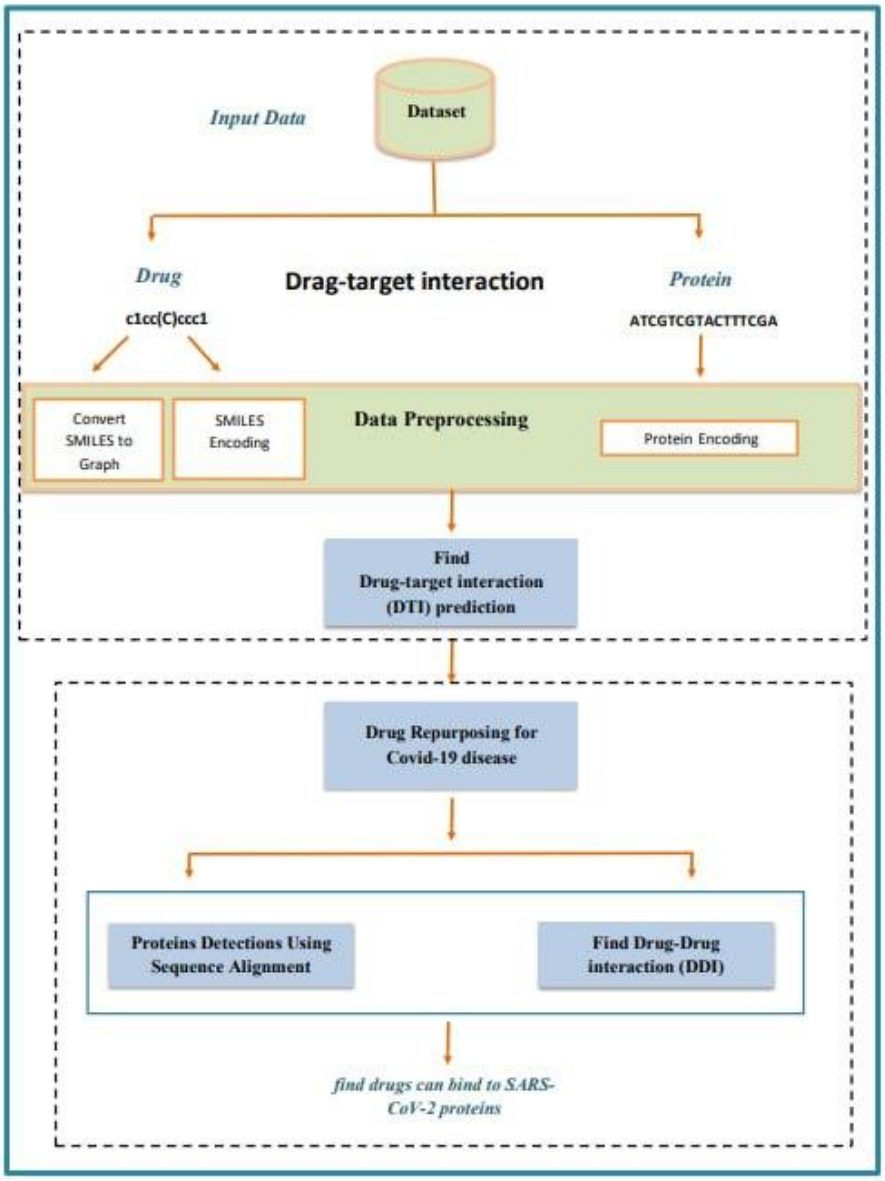


Figure 3.1: Methodology of the proposed system

3.3 Dataset Preprocessing Stage

In all datasets, drugs are delineated through the utilization of SMILES representations, a concise textual format that encapsulates the structural characteristics of a given molecule. On the other hand, proteins are denoted by their respective amino acid sequences, which form the fundamental building blocks of their biochemical structures. These representations, elucidated in Chapter Two within sections (2.3) and sections (2.4), furnish a foundational basis for comprehending the underlying molecular interactions that drive biological processes.

However, it is imperative to note that these raw representations necessitate preprocessing before they can be used effectively in machine learning or deep learning algorithms. This preprocessing stage involves a series of transformations aimed at converting SMILES strings and protein sequences into structured formats that are conducive to the application of advanced computational techniques.

3.3.1 SMILES and Protein Encoding

In SMILES, the labels or unique letters represent atoms, bonds, and other molecular features. The specific set of labels used in SMILES can vary depending on the context and the molecules being represented. While in protein, there are a total of 20 standard amino acids that are commonly found in proteins. Each amino acid is represented by a unique letter or symbol. However, if we consider additional categories such as ambiguous or non-standard amino acids, post-translational modifications, or special symbols used to denote specific features, the number of categories may exceed 25.

These 20 amino acids form the basis of protein sequences, and each amino acid contributes to the structure and function of proteins in different ways. Each amino acid is represented by a unique letter or symbol, see Table (2.1). Steps to SMILES and protein sequences encoding as following:

- Tokenize each SMILES or amino acid as a symbol
- Builds the vocabulary and assigns a unique index to each symbol.
- Replaces each symbol with its corresponding index in the vocabulary.

Algorithm (3.1) describes the data pre-processing for SMILES and protein sequences encoding.

Algorithm 3.1: SMILES and Protein Sequences Encoding

```
Input: SMILES sequence, Protein_sequence, label_dict
// smiles_sequence: Input SMILES sequence as a string
// protein_sequence: Input protein sequence as a string
// label_dict: Dictionary containing label-to-integer mappings
Output: encoded_sequence
Begin
1.  for each symbol in in smiles_sequence or protein_sequence do
2.    if symbol in label_dict then
3.      encoded_sequence.append(label_dict[symbol])
4.    else
        // Handle special symbol or invalid labels
5.          encoded_sequence.append(0) // Assign 0 for unknown or padding label
6.   end_if
7.  end_for
8.   return encoded_sequence
End
```

3.3.2 Convert SMILES to Graph

This manner convert SMILES representations of molecules into graph structures to perform graph-based deep learning. The general procedure used in the manner to convert SMILES to graphs:

1. *Parsing SMILES:* using the RDKit library, a powerful cheminformatics toolkit, to parse the SMILES representations and obtain a molecule object
2. *Atom Mapping:* After obtaining the molecule object from SMILES, map each atom to a unique index, and create a dictionary where each atom is a key. The corresponding value is an integer index.
3. *Atom Features:* extract features for each atom in the molecule. Features can include properties like atomic number, hybridization state, number of hydrogens, and more.
4. *Bond Mapping:* Next, map each bond between atoms to a unique index.
5. *Bond Features:* extract features for each bond in the molecule. Bond features can include information such as bond type (single, double, triple, etc.), ring status, stereochemistry, and more.
6. *Graph Construction*: Using the atom and bond mappings, construct a graph representation of the molecule, creates nodes for each atom and edges for each bond, with appropriate features associated with them. The resulting graph represents the molecular structure, where atoms are nodes, and bonds are edges.

Overall, the process involves parsing SMILES, mapping atoms and bonds to unique indices, extracting features for atoms and bonds, constructing a graph representation, and utilizing the graph for MPNN-

based learning. This allows for powerful graph-based deep learning on molecular data. Algorithm (3.2) describes the data pre-processing for converting SMILES to graphs.

Algorithm 3.2: Convert SMILES to Graph

***Input**: smiles_sequence*
// smiles_sequence: Input SMILES sequence as a string
***Output**: graph representation of molecule*
Begin
// Step 1: Parsing SMILES using RDKit library
1. *molecule_object = RDKit.parse(smiles_sequence)*
// Returns a molecule object obtained from SMILES representation
// Step 2: Atom Mapping
2. *atom_mapping = {}*
3. *index = 0*
4. ***for** each atom in molecule_object.atoms **do***
5. *atom_mapping[atom] = index*
6. *index = index + 1*
7. ***end for***
// Returns a dictionary mapping each atom to a unique index
// Step 3: Atom Features
8. *atom_features = {}*
9. ***for** each atom in molecule_object.atoms **do***
10. *features = extract_atom_features(atom)*
11. *atom_features[atom] = features*
12. ***end for***
// Returns a features vector of atoms
// Step 4: Bond Mapping
13. *bond_mapping = {}*
14. *index = 0*
15. ***for** each bond in molecule_object.bonds **do***
16. *bond_mapping[bond] = index*
17. *index = index + 1*
18. ***end for***
// Returns a dictionary mapping each bond to a unique index
// Step 5: Bond Features

```
    bond_features = {}
    for each bond in molecule_object.bonds do
        features = extract_bond_features(bond)
        bond_features[bond] = features
    end for
// Returns a features vector of bonds
                    // Step 6: Graph Construction
    graph = Graph()
    for each atom in molecule_object.atoms do
        graph.add_node(atom_mapping[atom], atom_features[atom])
    end for
    for each bond in molecule_object.bonds do
        start_node = atom_mapping[bond.start_atom]
        end_node = atom_mapping[bond.end_atom]
        graph.add_edge(bond_mapping[bond], start_node, end_node,
bond_features[bond])
    end for
    return graph representation of molecule
// Returns the final graph representation for use in the model
End
```

To encode features for atoms and bonds (which we will need later), only about a basic of (atom and bond) features will be considered: [atom features] symbol (element), number of valence electrons, number of hydrogen bonds, orbital hybridization, [bond features] (covalent) bond type, and conjugation as shown in (Table 3.2) and (Table 3.3) respectively.

To generate complete graphs from SMILES, the method need to be implemented in two following steps:

1. molecule from smiles, which takes as input a SMILES and returns a molecule object. This is all handled by RDKit.

2. graph from a molecule, which takes as input a molecule object and returns a graph, represented as a three-tuple (atom_features, bond_features, pair_indices).

Finally, these steps are implemented, which apply processes (1) and (2) to all SMILES in the training, validation, and test datasets.

Table 3.1: Atom features

Feature Name	Description
symbol	Allowable atomic symbols: B, Br, C, Ca, Cl, F, H, I, N, Na, O, P, S
n_valence	Allowable number of valence electrons: 0, 1, 2, 3, 4, 5, 6
n_hydrogens	Allowable number of hydrogen atoms: 0, 1, 2, 3, 4
hybridization	Allowable hybridization types: s, sp, sp2, sp3

Table 3.2: Bond features

Feature Name	Description
bond_type	Allowable bond types: single, double, triple, aromatic
conjugated	Allowable conjugation states: True, False

3.4 Models Development for Drug-Target Interaction Prediction

This Dissertation introduces of two DL models for DTI prediction: BiGRU-DTA and MPNN-BiGRU-DTA. The BiGRU-DTA used a Bi-GRU to extract features from protein and drug sequences. MPNN-BiGRU-DTA employed two distinct neural network components: a MPNN for processing drugs and a Bi-GRU for processing proteins.

3.4.1 Model 1: BiGRU-DTA

The BiGRU-DTA model is a deep neural network architecture designed to predict the binding affinity of drug molecules to target proteins. The model uses a bidirectional gated recurrent unit (Bi-GRU) to extract features from protein and drug sequences, followed by a fully connected neural network to make the final prediction.

The Bi-GRU network is a type of recurrent neural network (RNN) that can capture the sequential dependencies in input sequences by processing the input data in both forward and backward directions. The Bi-GRU network consists of two layers of GRU units, one processing the input sequence in the forward direction and the other in the backward direction. The outputs of both layers are then concatenated and passed on to the next layer.

In this model, the drug and protein sequences are input to separate Bi-GRU networks, which extract features from the sequences by processing them in both directions. The outputs of the Bi-GRU networks are then concatenated and passed through the fully connected neural network to make the final prediction of the binding affinity.

The Bi-GRU-DTA model treated drug-target interaction prediction as a regression problem by aiming to predict the binding affinity scores. Bi-GRU is an architecture that includes one or more layers, which are frequently followed by a pooling layer. A pooling layer down samples the output of the preceding layer and allows the filters' learnt characteristics to be generalized. The model is finished by one or more fully connected (FC) layers on top of the unit and pooling layers. The capacity of Bi-GRU models to capture local dependencies using units is their most powerful feature. As a result, the number and size of units in a Bi-GRU have a direct impact on the type of features the model learns from the input. The suggested model, which combines two Bi-GRU blocks, is depicted in Figure (3.2) and Algorithm (3.4).

The parameter settings for this model for predicting drug-target interactions shown in Table (3.4). These parameters play a critical role in determining the model's performance and accuracy. The table below summarizes the ranges and values used for each parameter during the experimentation process:

Table 3.3: Parameter Settings for Model-1

Parameters	Range
Number of units	32, 64, 96
Epoch	250
Hidden neurons	1024; 1024; 512
Batch size	256
Dropout	0.1
Optimizer	Adam
Learning rate (lr)	0.001

- Number of Units: 32, 64, 96

This parameter determines the number of units (neurons) in the GRU layers of your Bidirectional Gated Recurrent Unit (Bi-GRU) blocks. Experimenting with different values in this range allows to find the optimal balance between model complexity and expressiveness.

- Epoch: 250

An epoch is one complete pass through the entire training dataset during model training. Setting the number of epochs to 250 means that the model will iterate over the entire dataset 250 times during training.

- Hidden Neurons: 1024, 1024, 512

These values represent the number of neurons in the fully connected (FC) layers of your model. The first two FC layers have 1024 neurons each, and the third layer has 512 neurons. These layers are responsible for learning high-level features from the input data before producing the final output.

- Batch Size: 256

The batch size determines the number of samples processed in each iteration during training. A larger batch size may lead to faster convergence, but it also requires more memory. Smaller batch sizes can provide a form of regularization and may generalize better.

- Dropout: 0.1

Dropout is a regularization technique where a fraction of randomly selected neurons are dropped out (ignored) during training. A dropout rate of 0.1 means that 10% of the neurons are randomly dropped out in each training iteration, helping to prevent overfitting.

- Optimizer: Adam

Adam is an optimization algorithm commonly used for training deep neural networks. It adapts the learning rates for each parameter individually, providing a good balance between efficiency and simplicity.

- Learning Rate (lr): 0.001

The learning rate controls the step size during optimization. A smaller learning rate generally allows for more precise updates to model weights but may require more iterations for convergence. A learning rate of 0.001 is a common starting point and often works well for various tasks.

Algorithm 3.3: Bi-GRU Drug-Target interaction

***Input**: SMILES_encoded, protein_ encoded*

// smiles_sequence (output of algorithm 3.1)

// protein_sequence (output of algorithm 3.2)

***Output**: predicting the binding affinity*

Begin

// Step 1: Build Bi-GRU Network (Bi-GRU for SMILES) through

1. *Smiles_vector = Embedding Layer(SMILES_encoded)*
2. *smiles_features = bidirectional_GRU(Smiles_vector)*
3. *Add Units Layers*
4. *smiles_unit1 = unit_layer(smiles_features, num_ units=32)*
5. *smiles_unit2 = unit_layer(smiles_unit1, num_ units =64)*
6. *smiles_unit3 = unit_layer(smiles_unit2, num_ units =96)*
7. *Add Max-Pooling Layers*
8. *smiles_pooled = max_pooling_layer(smiles_unit3)*

// Step 2: Build Bi-GRU Network (Bi-GRU for Protein) through

9. *protein _vector = Embedding Layer(protein_encoded)*
10. *protein_features = bidirectional_GRU(protein_ vector)*

11. Add Units Layers

12. protein_unit1=unit_layer(protein_features, num_ units=32)

13. protein_unit2= unit_layer(protein_unit1, num_ units=64)

14. protein_unit3= unit_layer(protein_unit2, num_ units=96)

15. Add Max-Pooling Layers

16. protein_pooled = max_pooling_layer(protein_unit3)

// Step 3: Concatenate Pooled Features

17. concatenated_features = concat(smiles_pooled, protein_pooled)

// Step 4: Add Fully Connected Layers

18. fc_layer1_output = fully_connected_layer(concatenated_features, num_nodes=1024, dropout_rate=0.1)

19. fc_layer2_output = fully_connected_layer(fc_layer1_output, num_nodes=1024, dropout_rate=0.1)

20. fc_layer3_output = fully_connected_layer(fc_layer2_output, num_nodes=512, dropout_rate=0)

// Output Layer for Regression

21. output = fully_connected_layer(fc_layer3_output, num_nodes=1, dropout_rate=0)

22. ***return predicting the binding affinity***

End

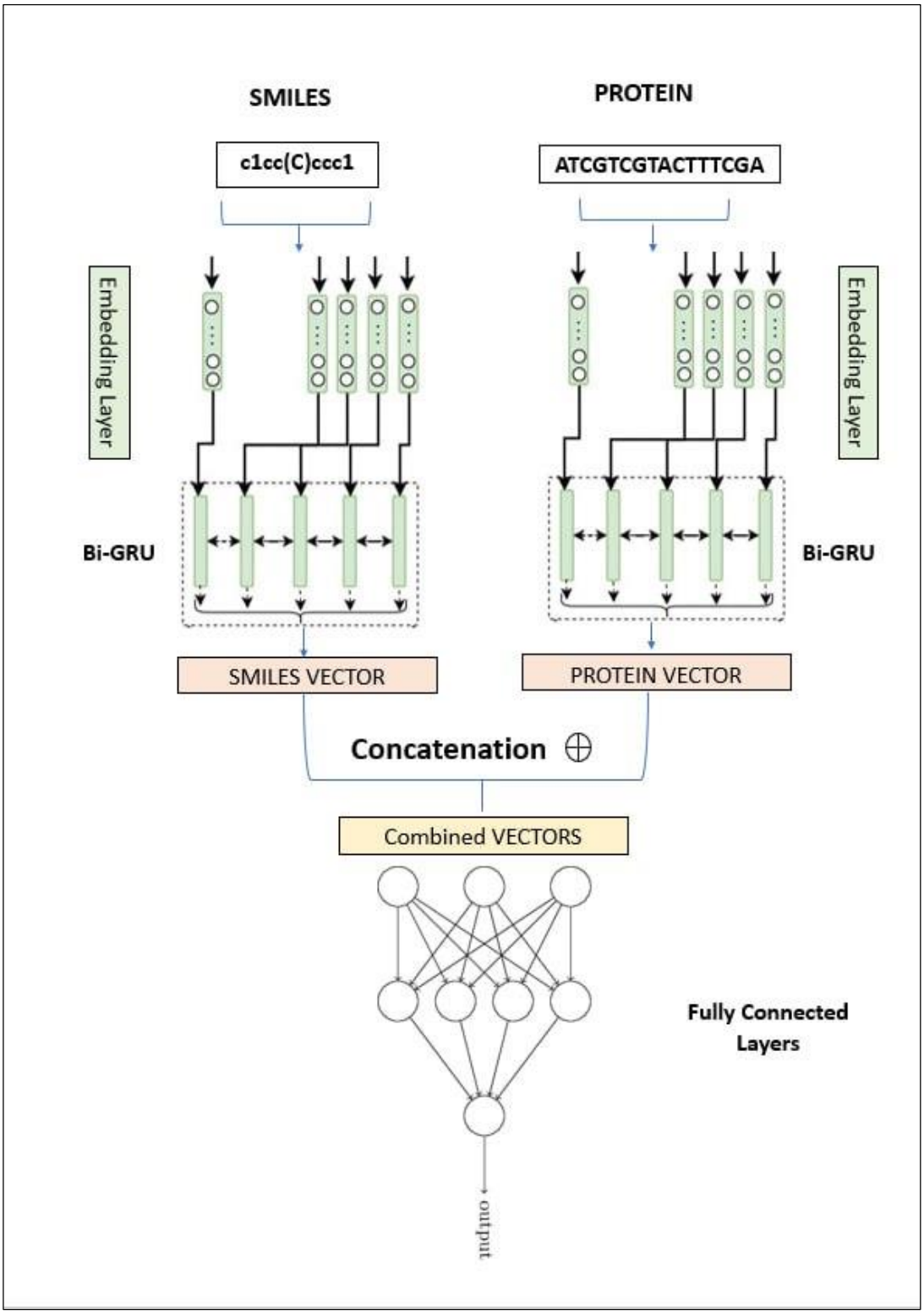


Figure 3.2: Model-1 with two Bi-GRU blocks to learn from compound SMILES and protein sequences.

The proposed a Bi-GRU-based prediction model that comprises two separate Bi-GRU blocks, each of which aims to learn representations

from SMILES strings and protein sequences. For each Bi-GRU block, the model used three consecutive 1D layers with increasing number of units. The second layer had twice the number of units as the first, and the third layer had treble the number of units as the first. The max-pooling layer was then applied after the layers. The max-pooling layers' final features were concatenated and fed into three FC layers. The model included 1024 nodes in the first two FC layers, which were followed by a rate 0.1 dropout layer. Following that, the third layer comprised 512 nodes and was succeeded by the output layer.

3.4.2 Model2: MPNN-GRU-DTI

This model employed two distinct neural network components: a Message-Passing Neural Network (MPNN) for processing drugs and a Bidirectional Gated Recurrent Unit (Bi-GRU) for processing proteins. This approach allows to effectively capture the unique characteristics and patterns associated with each data type.

The MPNN component is responsible for processing drug information. MPNNs are designed to operate on graph-structured data, where the molecular structure of a drug can be represented as a graph. By leveraging the MPNN architecture, the model can learn and encode the structural information of the drug molecules, capturing important features that contribute to their interaction with target proteins. On the other hand, the Bi-GRU component focuses on processing protein sequences. The Bi-GRU is a type of recurrent neural network that captures sequential dependencies within protein data. By processing protein sequences in both forward and backward directions, the Bi-GRU can effectively capture long-term dependencies and extract informative features from the protein data.

After processing the drug and protein data separately, the model concatenates the outputs of the MPNN and Bi-GRU networks. This combination allows the model to capture and leverage the complementary information present in both the structural properties of the drugs and the sequential patterns of the proteins. The concatenated output is then fed into subsequent outputs layers, which include fully connected layers, to make the final prediction for the DTI. These layers learn to extract higher-level representations and patterns from the concatenated features, ultimately yielding a prediction for the binding affinity or interaction strength between a specific drug and target protein. The proposed model is illustrated in Figure (3.3) and Algorithm (3.5).

Overall, the model architecture, utilizing the MPNN for drugs and Bi-GRU for proteins, along with the concatenation of their outputs, presents a promising approach for accurately predicting Drug-Target interactions. By effectively capturing both the structural and sequential aspects of the data, the model can leverage the strengths of both types of information to improve the prediction performance in the challenging task of DTI prediction.

The parameter settings for this model for predicting drug-target interactions shown in Table (3.5).

Table 3.4: Parameter Settings for Model-2

Parameters	Range
Number of units	32, 64, 96
Epoch	250
Hidden neurons	1024; 1024; 512
Batch size	256
Dropout	0.1

Optimizer	Adam
Learning rate (lr)	0.001
Batch size (MPNN)	256
Message Units	128
Message Steps	4

- Message Units (128):

This parameter determines the dimensionality of the message passed between nodes in the graph. Each message carries information from one node to another and is represented as a vector of length 128.

- Message Steps (4):

This parameter specifies the number of iterations or steps performed during message passing in the MPNN. At each step, information is exchanged between neighboring nodes in the graph, allowing for the propagation of information across the graph structure. With 4 message steps, the model performs four rounds of message passing, enabling the aggregation of information from neighboring nodes over multiple iterations.

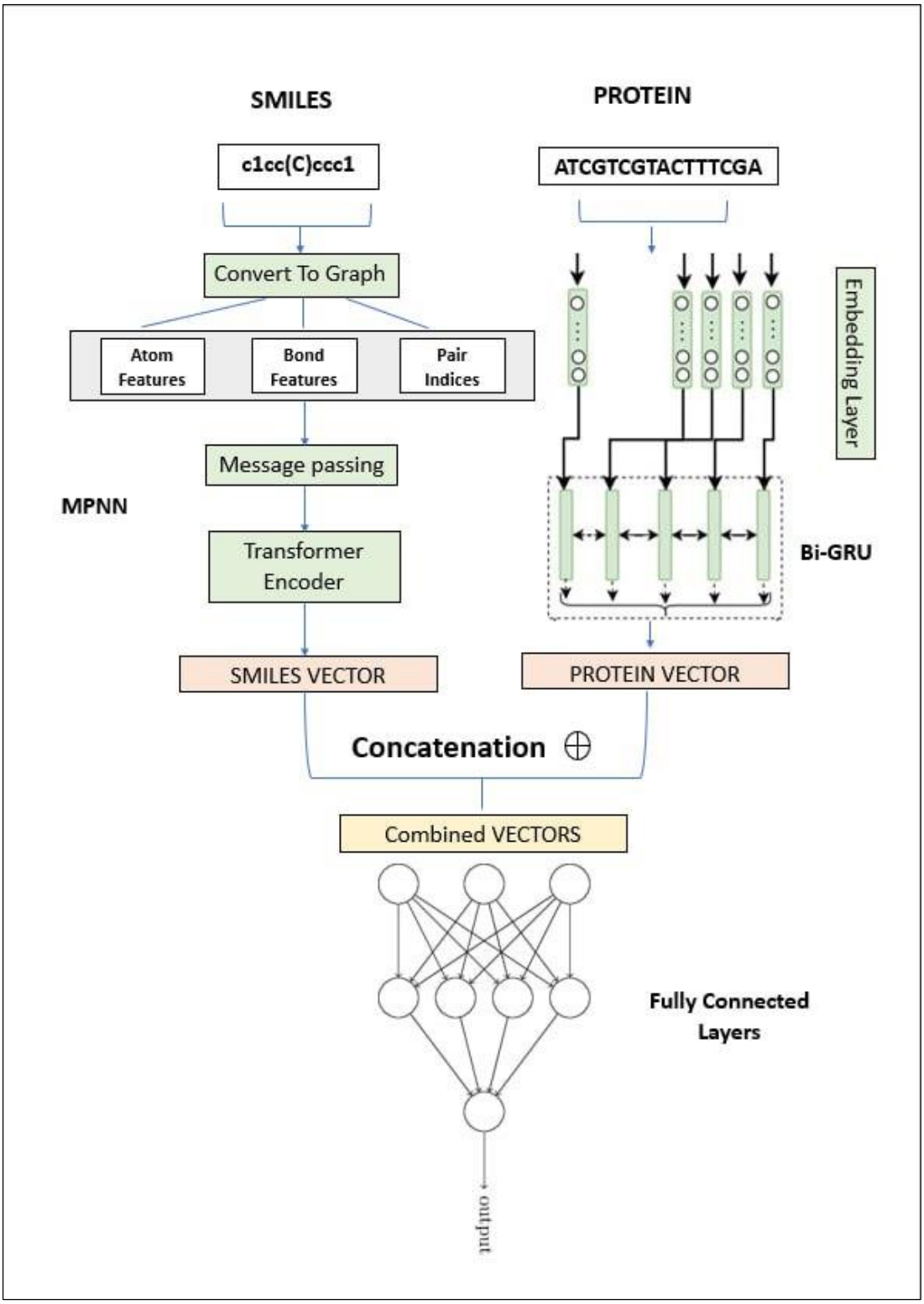


Figure 3.3: Model-2 with Bi-GRU blocks to learn from protein sequences and MPNN for compound SMILES.

Algorithm 3.4: MPNN-BiGRU- Drug-Target interaction

***Input**: SMILES_graph, protein_sequence*
// smiles_graph (output of algorithm 3.3)
// protein_sequence (output of algorithm 3.2)
***Output**: predicting the binding affinity*
Begin
// Step 1: Build MPNN Network (MPNN for Drug) through
// Message Passing Phase
for each iteration in range(num_iterations) do
for each node in graph.nodes do
messages = compute_messages(node, graph)
new_hidden_state = update_node(node, messages)
node.hidden_state = new_hidden_state
end for
end for
// Readout Phase
drug_features = readout(graph)
// Step 2: Build Bi-GRU Network (Bi-GRU for Protein) through
protein_features = bidirectional_GRU(protein_sequence)
Add Unitolutional Layers
protein_unit1=unit_layer(protein_features, num_units=64)
protein_unit2= unit_layer(protein_unit1, num_units=128)
protein_unit3= unit_layer(protein_unit2, num_units=192)
Add Max-Pooling Layers
protein_pooled = max_pooling_layer(protein_unit3)
// Step 3: Concatenate Pooled Features
concatenated_features = concat(drug_features, protein_pooled)
// Step 4: Add Fully Connected Layers
fc_layer1_output = fully_connected_layer(concatenated_features, num_nodes=1024, dropout_rate=0.1)
fc_layer2_output = fully_connected_layer(fc_layer1_output, num_nodes=1024, dropout_rate=0.1)
fc_layer3_output = fully_connected_layer(fc_layer2_output, num_nodes=512, dropout_rate=0)
// Output Layer for Regression
output = fully_connected_layer(fc_layer3_output, num_nodes=1, dropout_rate=0)

23. ***return predicting the binding affinity***

End

3.5 Drug Repurposing for Covid-19 disease

The spread of a transmissible coronavirus (SARS-CoV-2) has resulted in a large increase in worldwide mortality. Due to the scarcity of effective treatments, the goal of this dissertation is to propose extremely powerful active compounds (drugs) that can bind to the protein structure of SARS-CoV-2.

The creation of novel medications is a costly and time-consuming procedure. Because of the global SARS-CoV-2 outbreak, new SARS-CoV-2 medications must be developed as quickly as feasible. Drug repurposing approaches can shorten the time required to create new treatments by investigating the list of existing FDA-approved pharmaceuticals and their qualities to reuse them to tackle the new condition.

This dissertation used a list of existing 82 FDA-approved drugs against five viral SARS-CoV-2 targets. The five SARS-CoV-2 viral proteins were obtained using sequence alignment techniques as we will see in section (3.6). Using the Combined Score, this dissertation compiles a roster of the ten most promising drugs, showcasing the highest binding affinity for five viral proteins found in SARS-CoV-2. This compilation could potentially serve as a foundation for the development of valuable new drugs. Finally, in section (3.7) the dissertation Proposed a DL architecture for predicting DDIs and uses it to test the interactions of the top drugs.

3.6 Molecular Docking for SARS-CoV-2

The molecular docking process typically follows a series of steps as shown in Figure (3.5). Initially, parameters are assigned to the target protein and the ligand or ligands. Afterwards, the system is readied by establishing the search grid. Following the docking calculation, the poses of the ligand are evaluated using a specified energy function. Finally, the computational search results are scrutinized and matched against experimental data to validate the findings.

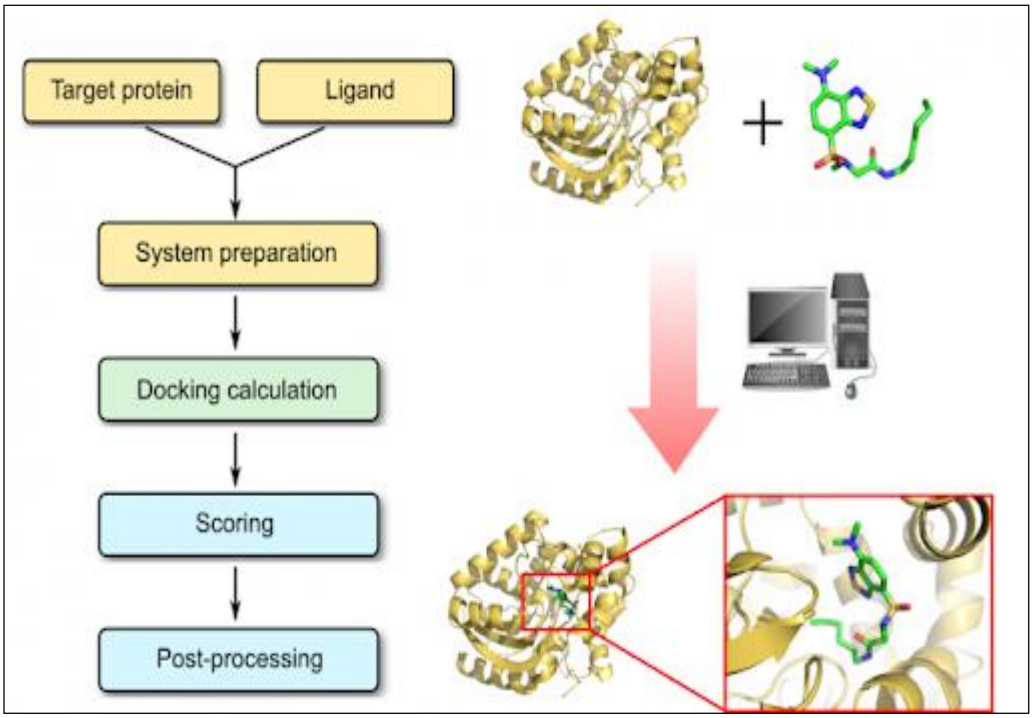


Figure 3.4: Molecular docking process

This dissertation employed molecular docking, a computational technique widely utilized in drug discovery, to predict the interaction between potential drug candidates (ligands) and key proteins of the SARS-CoV-2 virus. The methodological approach can be outlined as follows:

1- Data Acquisition:
    a. Protein and Ligand Selection: Identified key protein for the SARS-CoV-2 genome. Acquired three-dimensional structures of these proteins from reliable sources.
    b. Ligand Sourcing: Identified ligands from the most important drugs that scored the highest rating and the highest number of appearances for five Covid-19 proteins. these drugs were designed for potential antiviral activity.
2- Structural Preparation:
    a. Receptor Preparation: Converted protein structures into a format compatible with molecular docking simulations (PDBQT format). This included adding missing atoms, assigning charges, and defining atom types.
    b. Ligand Preparation: Processed ligand structures to ensure proper geometry, protonation states, and compatibility with the chosen docking software.
3- Defining Docking Parameters: Specified docking parameters such as Search Space that Defined the region within the protein where ligands were allowed to bind.
4- Docking Simulations: Utilized AutoDock Vina, a widely accepted docking software, to perform the docking calculations. For each ligand, the program systematically explored potential binding poses, providing binding affinities as well as the 3D coordinates of the docked complex.
5- Analyzing Docking Results: Extracted and analyzed docking results, focusing on: Binding Energies and Binding Modes.

3.7 SARS-CoV-2 Proteins Detections Using Sequence Alignment

Sequence alignment is a computational technique used to identify similarities and differences between two or more protein sequences. Sequence alignment can help identify and characterize proteins that are related to a specific disease, such as COVID-19.

The following steps involved in this process to find COVID-19 related proteins using sequence alignment:

1. Download the SARS-CoV-2 reference protein sequence from the NCBI website.
2. Use the Biopython library to read the reference genome sequence into memory.
3. Use the BLAST (Basic Local Alignment Search Tool) algorithm to search for similar protein sequences in the NCBI non-redundant protein database.
4. Identify the top hits from the BLAST search results, which are the protein sequences with the highest similarity scores to the query sequence.
5. Analyze the protein sequences using various bioinformatics tools and databases to determine their potential functions and interactions.

Overall, this process involves identifying potential proteins encoded by the SARS-CoV-2 genome, comparing them to known protein sequences in public databases, and analyzing their potential functions and interactions. This information can provide insights into the biology of the virus and potential targets for therapeutic interventions.

BLAST compares a query sequence with a database of known sequences to identify homologous regions and calculate sequence similarity scores. In the context of COVID-19, the BLAST algorithm can be applied to analyze the genetic information of the virus and compare it with existing sequence databases. By using BLAST, we can identify similar or closely related viral strains, and determine genetic variations. The algorithm calculates a sequence similarity score based on the alignment, allowing it to prioritize matches that have higher similarity scores. Algorithm (3.6) shows a steps for sequence alignment.

Algorithm 3.5 Sequence Alignment

```
Input: Two sequences Seq1 and Seq2
Output: COVID-19 related proteins.
Begin
1. Step1: read Seq1 and Seq2
2. Step2:
3.      Let M = size of Seq1 and N = size of Seq2
4.      let Cell [,] Matrix = new Cell [N, M];
5.      let Gap = 1; Similarity = -2;
6.      for i=0 to M
7.              Matrix[0, i] = new Cell(0, i, i*Gap)
8.          End for
9.      for i=0 to N
10.             Matrix[i, 0] = new Cell(I, 0, i*Gap)
11.         End for
12.     for j=1 to N
13.             for i=1 to M
14.                     Matrix[j, i] = Max(Diagonal, Left, Up)
15.             End for
16.         End for
17. Step 4:
18.     Cell CurrentCell = Matrix[Sq2.Length - 1, Sq1.Length - 1];
19.     while (CurrentCell.CellPointer != null)
20.                 if (CurrentCell.Type == Diagonal)
21.                             Seq1.Add(Sq1[CurrentCell.CellColumn]);
```

```
22.                         Seq2.Add(Sq2[CurrentCell.CellRow]);
23.                 if (CurrentCell.Type == Left)
24.                         Seq1.Add(Sq1[CurrentCell.CellColumn]);
25.                         Seq2.Add('-');
26.                 if (CurrentCell.Type == Above)
27.                         Seq1.Add('-');
28.                         Seq2.Add(Sq2[CurrentCell.CellRow]);
29.         CurrentCell = CurrentCell.CellPointer;
30.         End while
End
```

3.8 Drug-Drug interactions model

Drug-drug interactions refer to the effect that occurs when two or more drugs interact with each other, leading to changes in their efficacy or safety. These interactions can arise due to the use of multiple drugs simultaneously or through the modification of drug metabolism or elimination.

The main objective of using the MPNN-DDI is to improve the prediction of drug-drug interactions. This model developed a drug-drug interaction deep learning model using the Message Passing Neural Network (MPNN) architecture. The dataset used in this model contained drug1 SMILES, drug2 SMILES, and their interactions.

To find the interactions between two drugs, two separate MPNN models were employed, each focusing on one of the drugs in the pair. By utilizing two MPNN models, the model was able to capture the unique characteristics of each drug, then we combined the output of the two separate MPNN models into a single prediction. By concatenating the outputs, the model was able to capture the interactions between the two drugs and their molecular features. The proposed model is illustrated in Figure (3.9) and Algorithm (3.7).

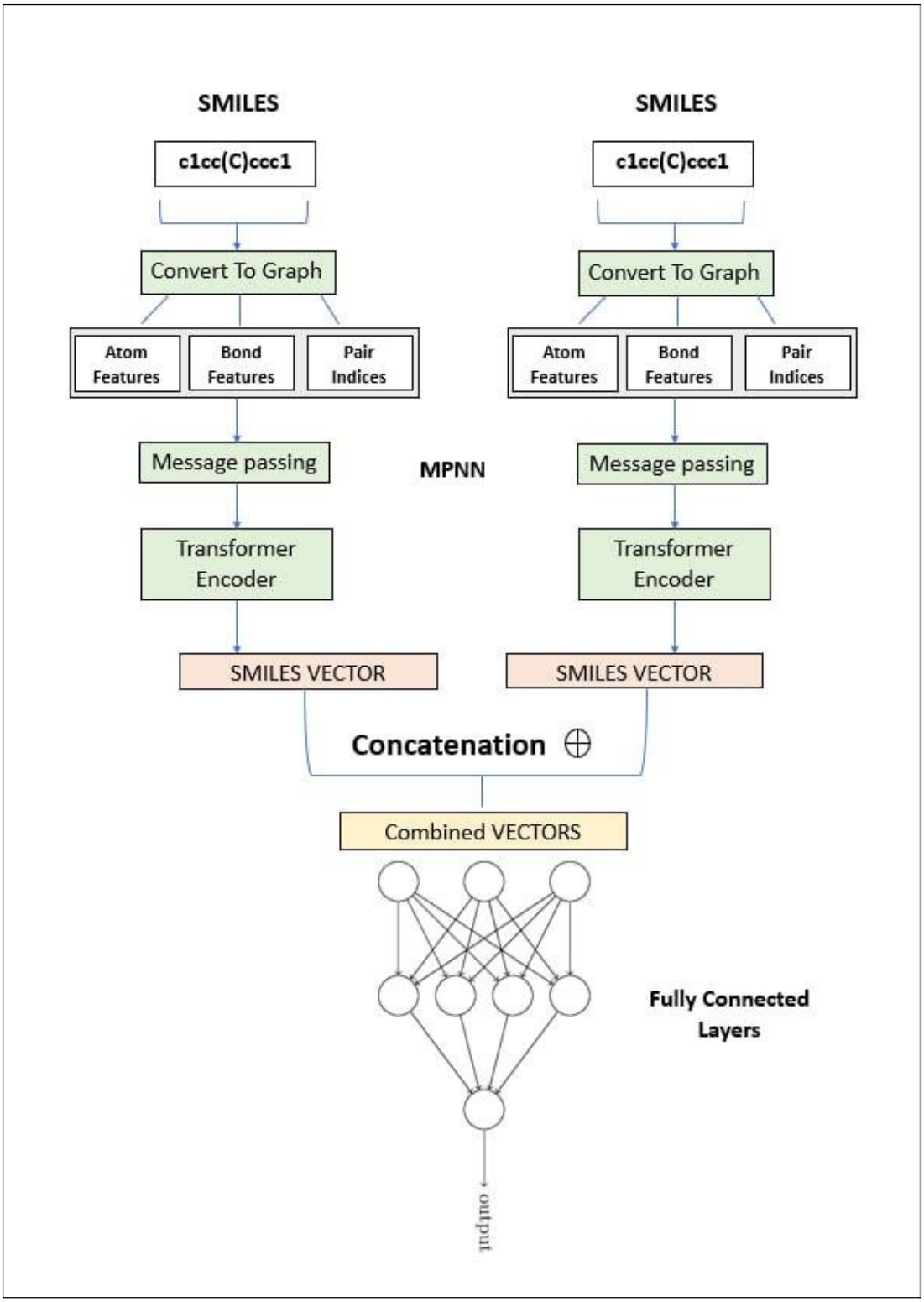


Figure 3.5 Drug-Drug Interactions Model

Algorithm 3.6: Drug-Drug Interactions Model

***Input**: SMILES_graph1, SMILES_graph2*
// smiles_graph1 (output of algorithm 3.3)
// smiles_graph2 (output of algorithm 3.3)
***Output**: predicting the binding affinity*
Begin
// Step 1: Build MPNN Network (MPNN for Drug-1) through
// Message Passing Phase
for each iteration in range(num_iterations) do
for each node in graph.nodes do
messages = compute_messages(node, graph)
new_hidden_state = update_node(node, messages)
node.hidden_state = new_hidden_state
end for
end for
// Readout Phase
drug_features1 = readout(graph)
// Step 2: Build MPNN Network (MPNN for Drug-2) through
// Message Passing Phase
for each iteration in range(num_iterations) do
for each node in graph.nodes do
messages = compute_messages(node, graph)
new_hidden_state = update_node(node, messages)
node.hidden_state = new_hidden_state
end for
end for
// Readout Phase
drug_features2 = readout(graph)
// Step 3: Concatenate Drugs Features
concatenated_features = concat(drug_features1, drug_features2)
// Step 4: Add Fully Connected Layers
fc_layer1_output = fully_connected_layer(concatenated_features, num_nodes=1024, dropout_rate=0.1)
fc_layer2_output = fully_connected_layer(fc_layer1_output, num_nodes=1024, dropout_rate=0.1)
fc_layer3_output = fully_connected_layer(fc_layer2_output, num_nodes=512, dropout_rate=0)

// Output Layer for Classification

output = fully_connected_layer(fc_layer3_output, num_nodes=1, dropout_rate=0)

return predicting DDI

End

Chapter Four

Results and Discussion

Results and Discussion

4.1 Introduction

This Chapter views and discusses the results of the proposed methodology stages presented in Chapter Three. The chapter presents the results and discussion of our study on the prediction of drug-target interactions, the repurposing of drugs used in treatment of Covid-19. It first discusses a data collection and model development steps described in the Methodology chapter. Then present the results of our drug-target interaction prediction experiments, including the evaluation of our two DL models, BiGRU-DTA and MPNN-BiGRU-DTA. Also discuss the results of our drug repurposing experiments, including the verifications of prospective Covid-19 drugs candidates.

Experiments have been conducted on three datasets available to the research community: Davis and KIBA datasets for DTI prediction, DrugBank and DDInter datasets for drug-drug interaction prediction and the NCBI website was utilized to retrieve protein sequence data related to Covid-19. Then, the results are arranged based on their appearance in the previous Chapter. However, this Chapter begins with the software and hardware requirements in implementing the proposed models.

4.2 Software and Hardware Requirements

The specifications (software and hardware) that have been utilized to implement the proposed system are as follows:

1. Hardware:
    - PC: (HP).
    - Processor: Intel(R) Core i5 CPU, speed 2.70GHz.

- RAM: the memory capacity of 8.0 GB
- Hard: storage 256 GP SSD.

2. Software:
    - Windows 10 64-bit.
    - Python for executing deep learning techniques.
    - Google Colab (cloud-based Jupyter notebook environment).
    - Library (Tensor flow, Keras and RDKit)

4.3 Description of Datasets

This dissertation uses five datasets available to ensure reliable and comprehensive analysis: Davis and KIBA datasets for DTI prediction, DrugBank and DDInter datasets for drug-drug interaction prediction and the NCBI website were utilized to retrieve protein sequence data related to COVID-19.

4.3.1 Davis and KIBA datasets

As explained in the previous chapter, the Davis dataset contains 30056 measurements of binding affinity between 442 drug molecules and 68 target proteins. The KIBA dataset consists of 2,111 compounds and 229 protein targets. These datasets are presented in the forms used in the system experiments in Tables (4.1) and Tables (4.2).

Table 4.1: Davis dataset sample

Compound SMILES	Target Sequence	Affinity
COc1cc2c(Oc3ccc4[nH]c(C)cc4c3F)ncnc2cc1OCCCN1C...	MVFQTRYPSWIILCYIWLLRFAHTGEAQAAKEVLLLDSKAQQTELE...	6.207608
CCOc1cc2ncc(C#N)c(Nc3ccc(OCc4ccccn4)c(Cl)c3)c2...	MREAAAALVPPPAFAVTPAAAMEEPPPPPPPPPPPPEPETESEPEC...	7.408935

Cc1[nH]c(C=C2C(=O)Nc3ccc(F)cc32)c(C)c1C(=O)NCC...	MDDQQALNSIMQDLAVLHKASRPALSLQETRKAKSSSPKKQNDVRV...	7.136677
COc1cc2ncnc(Nc3ccc(F)c(Cl)c3)c2cc1OCCCN1CCOCC1	MALRRLGAALLLLPLLAAVEETLMDSTTATAELGWMVHPPSGWEEV...	5.000000
COc1cc2c(Oc3ccc(NC(=O)C4(C(=O)Nc5ccc(F)cc5)CC4...	MSGVSEPLSRVKLGTLRRPEGPAEPMVVVPVDVEKEDVRILKVCFY...	7.853872
COc1cc2c(Oc3ccc4[nH]c(C)cc4c3F)ncnc2cc1OCCCN1C...	MGAIGLLWLLPLLLSTAAVGSGMGTGQRAGSPAAGPPLQPREPLSY...	5.638272
O=c1ncn2nc(Sc3ccc(F)cc3F)ccc2c1-c1c(Cl)cccc1Cl	MPAAAGDGLLGEPAAPGGGGGAEDAARPAAACEGSFLPAWVSGVPR...	5.000000
CN1CCN(C(=O)c2cc3cc(Cl)ccc3[nH]2)CC1	TMPPRPSSGELWGIHLMPPRILVECLLPNGMIVTLECLREATLITI...	5.000000
CS(=O)c1ccc(-c2nc(-c3ccc(F)cc3)c(-c3ccncc3)[nH...	PFWKILNPLLERGTYYYFMGQQPGKVLGDQRRPSLPALHFIKGAGK...	5.585027
CC(O)C(=O)O.CN1CCN(c2ccc3c(c2)NC(=C2C(=O)N=c4c...	TMPPRPSSGELWGIHLMPPRILVECLLPNGMIVTLECLREATLITI...	5.000000
COc1cc(Nc2c(C#N)cnc3cc(OCCCN4CCN(C)CC4)c(OC)cc...	PFWKILNPLLERGTYYYFMGQQPGKVLGDQRRPSLPALHFIKGAGK...	7.677781

Table 4.2: KIBA dataset sample

Compound SMILES	Target Sequence	Affinity
CC(C1=CC(=CC=C1)OC)NC(=O)C2=C(C=C(C=C2)C3=CC=N...	MEGISNFKTPSKLSEKKKSVLCSTPTINIPASPFMQKLGFGTGVNV...	11.100000
C1=CC2=CN=C(C=C2C=C1C3=CC=NC=C3)NCCO	MSLIRKKGFYKQDVNKTAWELPKTYVSPTHVGSGAYGSVCSAIDKR...	11.200000
C1=CC=C2C(=C1)C(=CN2)CC(COC3=CN=CC(=C3)C4=CC5=...	MENFQKVEKIGEGTYGVVYKARNKLTGEVVALKKIRLDTETEGVPS...	11.699999

CC1=CC(=C2C(=C1)N=C(O2)NC3=C(C=C(C=C3)C4=C(C(=...	MSRSKRDNNFYSVEIGDSTFTVLKRYQNLKPIGSGAQGIVCAAYDA...	11.600000
C1=CC(=C(C=C1CNC2=C(C(=O)C2=O)NC3=CC=NC=C3)Cl)Cl	MRLTLLCCTWREERMGEEGSELPVCASCGQRIYDGQYLQALNADWH...	11.100000
CC1=C2C=C(C=CC2=NN1)C3=CC(=CN=C3)OCC(CC4=CC(=C...	MGFSSELCSPQGHGVLQQMQEAELRLLEGMRKWMAQRVKSDREYAG...	11.800001
CN(C)CC(COC1=CC=C(C=C1)NC2=NC=CC(=N2)NC3=C(C=C...	MALLRDVSLQDPRDRFELLQRVGAGTYGDVYKARDTVTSELAAVKI...	12.100000
C1=CC(=CC=C1CC(CO)NC(=O)C2=CC=C(C=C2)C3=CC=NC=...	MPLRHWGMARGSKPVGDGAQPMAAMGGLKVLLHWAGPGGGEPWVTF...	11.800001
COC1=C(C=C(C=C1)NC(=O)NC2=CC=C(C=C2)C3=C4C(=CC...	MSDSKCDSQFYSVQVADSTFTVLKRYQQLKPIGSGAQGIVCAAFDT...	11.500000
COC1=C(C=CC2=C1CCC(CC2)N3CCOCC3)NC4=NC=C(C(=N4...	MGFGSDLKNSHEAVLKLQDWELRLLETVKKFMALRIKSDKEYASTL...	13.170581

Additionally, Figure (4.1) illustrates the distribution of SMILES string lengths for compounds within both the Davis and KIBA datasets.

In the Davis dataset, the SMILES representation length of molecules varies from a minimum of 39 to a maximum of 103, with an average length of 64. Correspondingly, the sequences of proteins in this dataset exhibit a minimum length of 244, a maximum of 2549, and an average length of 788. As for the KIBA dataset, the SMILES lengths for molecules range from a minimum of 20 to a maximum of 590, with an average length of 59. Similarly, the protein sequences within this dataset have a minimum length of 215, a maximum length of 4128, and an average length of 728.

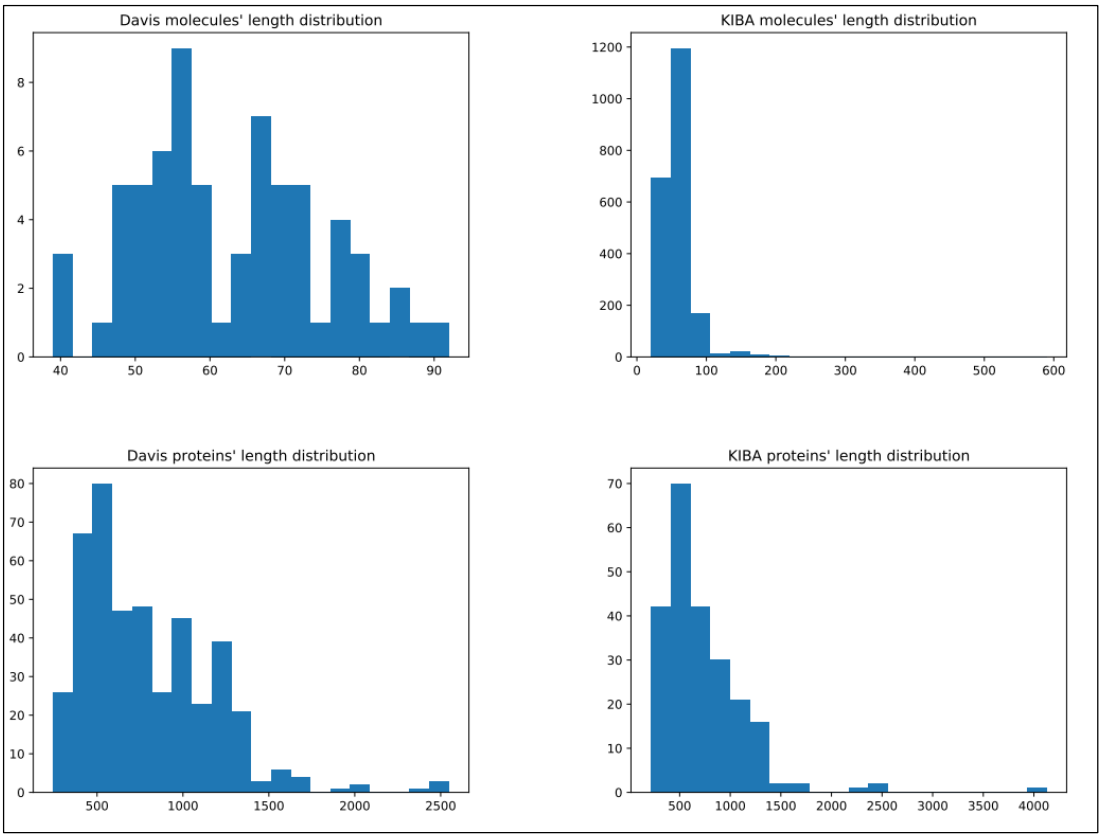


Figure 4.1. Summary of the Davis (left) and KIBA (right) datasets. The first row illustrates the distribution of the SMILES length for molecules, while the second row displays the distribution of protein sequence lengths

Diverse lengths are observed in both SMILES and protein sequences. Therefore, to establish a potent representation approach, a determination was made to adopt fixed upper limits of 85 characters for SMILES and 1200 characters for protein sequences in the context of the Davis dataset. For the components within the KIBA dataset, a decision was reached to utilize a maximum of 100 characters for SMILES and 1000 characters for protein sequences [9] [12] [17] [19]. The selection of these maximum extents was guided by the distributions exhibited in Figure (4.1), ensuring that these maximum lengths encompass a minimum of 80% of the proteins and 90% of the compounds within the datasets. Sequences surpassing the stipulated maximum lengths are subject to

truncation, while sequences of lesser lengths are augmented with zero-padding.

4.3.2 DrugBank and DDInter

For predicting Drug-Drug interactions, the proposed system sourced data from two prominent databases: DrugBank and DDInter. DrugBank provides a comprehensive repository of drug-related information, including known interactions between different drugs. DDInter, on the other hand, specifically focuses on Drug-Drug interactions. Table (4.3) presents the dataset in the forms used in the system experiments.

Table 4.3: Drug-Drug Interaction Dataset

DDInterID_A	Drug_A	DDInterID_B	Drug_B	Level
DDInter690	Ethanol	DDInter1	Abacavir	Minor
DDInter270	Calcium acetate	DDInter582	Dolutegravir	Major
DDInter1019	Lamivudine	DDInter424	Cobicistat	Minor
DDInter488	Deferasirox	DDInter582	Dolutegravir	Minor
DDInter582	Dolutegravir	DDInter1113	Magnesium carbonate	Major
DDInter1120	Magnesium sulfate	DDInter582	Dolutegravir	Major

The dataset provided appears to be inadequate for the proposed model due to a discrepancy in the data format. The model is designed to work with drugs represented in SMILE format, which is a specific chemical notation used to encode molecular structures. However, the dataset only contains drug names without their corresponding SMILE

format representations. In preparation for the model, an initial step involved downloading drug information in SMILE format from the DrugBank database. Subsequently, drug compounds were converted into SMILE format using functions within the Excel program. Given this disparity between the dataset's content (drug compounds) and the model's requirement (SMILE format), further data processing or acquisition of the actual SMILE format representation of the drugs is necessary to ensure compatibility with the intended modeling approach.

4.3.3 The NCBI website

To investigate protein-protein interactions and explore drug repurposing opportunities for SARS-CoV-2, the proposed system collected protein sequences of viral protein related to COVID-19. The protein sequence data was obtained from reputable sources, with the NCBI website serving as a primary source. Figure (4.2) presents the NCBI website.

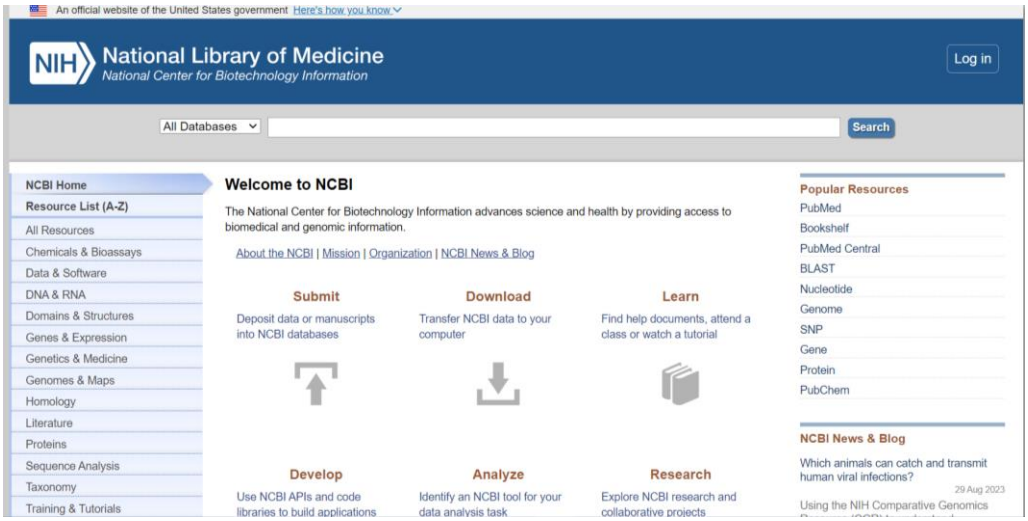


Figure (4.2): NCBI website

4.4 Results of Dataset Preprocessing

In all databases, drugs are delineated through the utilization of SMILES representations. On the other hand, proteins are denoted by their respective amino acid sequences.

The raw representations necessitate preprocessing before they can be used effectively in machine learning or deep learning algorithms. This preprocessing stage involves a series of transformations aimed at converting SMILES strings and protein sequences into structured formats that are conducive to the application of advanced computational techniques as mentioned in Section (3.3). So, the main objective of this stage is to prepare the datasets in an acceptable format for machine learning algorithms.

4.4.1 SMILES Encoding

In SMILES, the labels or unique letters represent atoms, bonds, and other molecular features. This method used integer/label encoding that uses integers for the categories to represent inputs.

The method represents each label with a corresponding integer (e.g. ‘C’: 18, ‘H’: 20, ‘N’: 23 etc.). The label encoding for the example SMILES, ‘CN=C = O’, is given below.

[C N = C = O] = [19 24 17 19 17 25]

```
# create token dictionaries for SMILES
smiles_dict = {'#': 1, '(': 2, ')': 3, '+': 4, '-': 5, '.': 6, '/': 7,
               '1': 8, '2': 9, '3': 10, '4': 11, '5': 12, '6': 13, '7': 14,
               '8': 15, '9': 16, '=': 17, 'B': 18, 'C': 19, 'F': 20, 'H': 21,
               'I': 22, 'K': 23, 'N': 24, 'O': 25, 'P': 26, 'S': 27, '[': 28,
```

'\\': 28, ']': 29, 'c': 30, 'n': 31, 'o': 32, 's': 33}

4.4.2 Protein Sequences Encoding

There are a total of 20 standard amino acids that are commonly found in proteins. Each amino acid is represented by a unique letter or symbol. Protein sequences are encoded in a similar way using label encodings. The label encoding for the example Protein, 'MASQLQVFSPP', is given below.

[M A S Q L Q V F S P P] = [11 1 16 14 10 14 18 5 16 13 13]

```
# create token dictionaries for amino acids
amino_acid_dict = {
    'A': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I':8,
    'K': 9, 'L': 10, 'M': 11, 'N': 12, 'P': 13, 'Q': 14, 'R': 15,
    'S': 16, 'T': 17, 'V': 18, 'W': 19, 'Y': 20
    }
```

4.4.3 Convert SMILES to Graph

This section discusses the transformation of molecular structures represented in SMILES format into graph representations The method contains two steps as mentioned in Section (3.3.3). This manner convert SMILES representations of molecules into graph structures to perform graph-based deep learning. Overall, the process involves parsing SMILES, mapping atoms and bonds to unique indices, extracting features for atoms and bonds, and constructing a graph representation. An example of this method is illustrated in Figure (4.3). This Figure shows the general framework for this preprocessing stage.

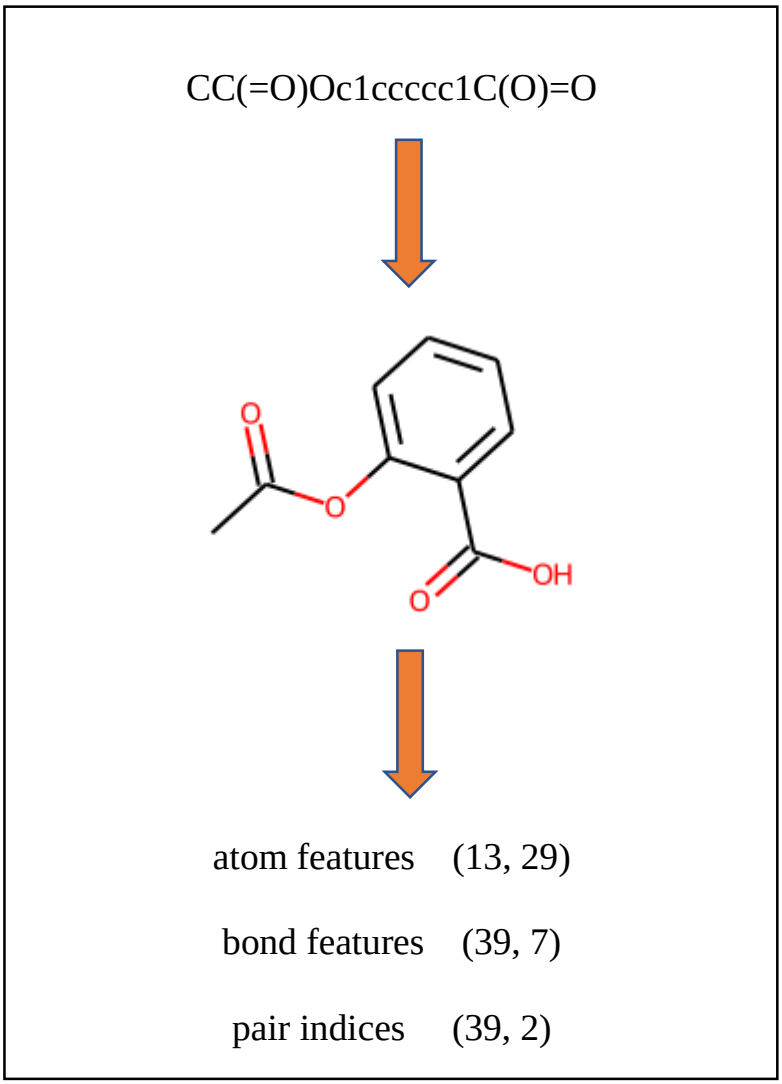


Figure 4.3: General framework for converting SMILES to graph

To illustrate this process, the method applied these functions to the SMILES string "CC(=O)Oc1ccccc1C(O)=O." These results have been obtained:

Atom Features (13, 29): The atom features matrix represents the properties of each atom within the molecule. In this example, there are 13 atoms, and each atom is characterized by a 29-dimensional feature vector. These dimensions encompass attributes such as the element symbol, valence, number of hydrogens, and hybridization (see Table (3.2)). Each row corresponds to a specific atom in the molecule. Table (4.4) describes the results of atom features.

Table 4.4: The results of atom features.

Allowable atomic symbols B, Br, C, Ca, C1, F, H, I, N, Na, O, P, S													**No. of valence electrons** 0, 1, 2, 3, 4, 5, 6							**No. of hydrogen atoms** 0, 1, 2, 3, 4					**Hybrid types** s, sp, sp2, sp3			
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0
- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -	- -
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0

Bond Features (39, 7): The bond features matrix encapsulates information about each bond within the molecule. Our molecule contains 39 bonds, and each bond is represented by a 7-dimensional feature vector. These dimensions encompass bond-related properties, including bond type (single, double, etc.) and whether the bond is conjugated (see Table (3.3)). Table (4.5) described the results of bond features.

Table 4.5: The results of bond features.

Bond types: Indices, single, double, triple, aromatic					**conjugation states:True, False**	
0	0	0	0	0	0	1
0	0	1	0	1	0	0
0	0	0	0	0	0	1
- -	- -	- -	- -	- -	- -	- -
0	0	0	0	0	0	1
0	1	0	0	0	1	0

Pair Indices (39, 2): The pair indices matrix elucidates the connectivity between atoms in the molecule. With 39 rows, each signifying a bond, the two columns in each row denote the indices of the two atoms connected by that bond. For instance, the entry [0, 0] indicates that the first bond connects the atom at index 0 to itself, potentially representing a single bond within a ring structure. Table (4.6) describes the results of pair indices.

Table 4.6: The results of pair indices

Atom 1	Atom 2
0	0
0	1
1	1
- -	- -
12	12
12	10

4.5 Results of Proposed Deep Learning Models for Drug-Target Interaction Prediction

The study employed regression modeling to predict drug-target binding affinity, as outlined in Section 3.2. For assessing the regression performance, MSE (as per Equation (2.17)) were applied. The training-validation-testing procedure detailed in Section 3.4 facilitated the evaluation of the model's performance using these metrics.

This dissertation introduced two deep-learning models. The performance of these models was evaluated by distinct collections of data

that were used to train the models for both the Davis and KIBA datasets. The following subsections explain the results of each model separately, along with a discussion of these results.

4.5.1 Model 1: BiGRU-DTA

The first model employs two Bi-GRU blocks to acquire representations for drugs and targets by considering their sequences as shown in Figure (4.4 – A, B). The performance of these models was evaluated by different training sets for the Davis and KIBA datasets, and the results are presented in Table (4.7) and Figure (4.5), showing the average MSE scores over the independent test set.

Table 4.7. Detailed evaluation metric scores for Model-1(BiGRU-DTA)

Dataset	MSE
Davis	0.237
KIBA	0.200

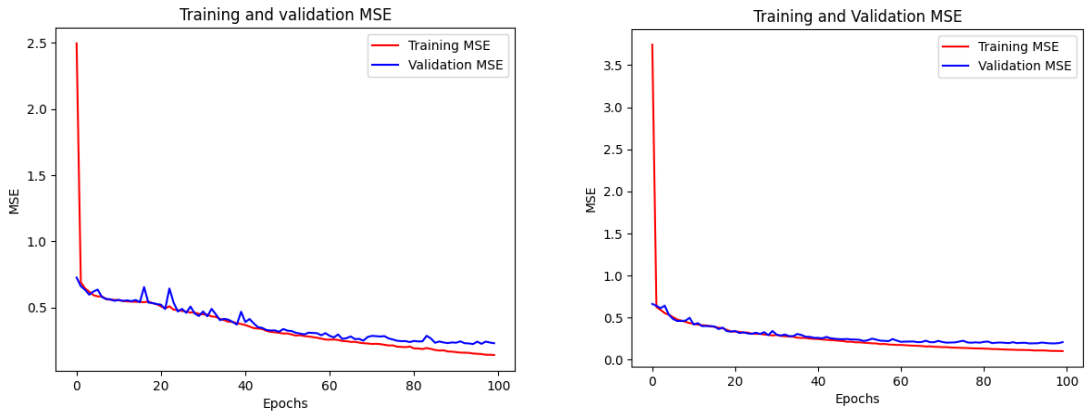


Figure 4.4: Training and validation MSE for model-1 (Davis in left, Kiba in right)

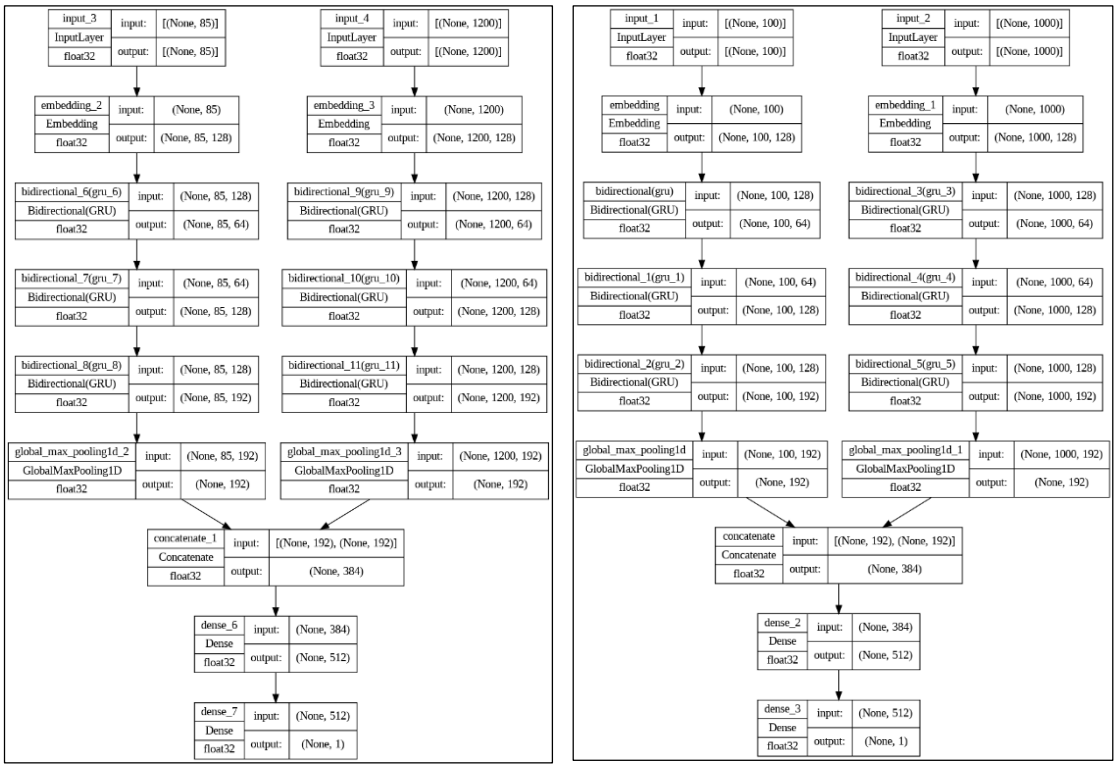


Figure 4.5: Model-1 with two Bi-GRU blocks to learn from compound SMILES and protein sequences for (a) the Davis dataset, and (b)the Kiba dataset.

To demonstrate the efficacy of Model-1, Model-1 was compared with several previous studies that were conducted on the Davis and KIBA datasets. Table (4.8) shows the model-1's performance compared with previous studies for the Davis dataset and KIBA dataset.

Table 4.8. Prediction performance for the Model-1

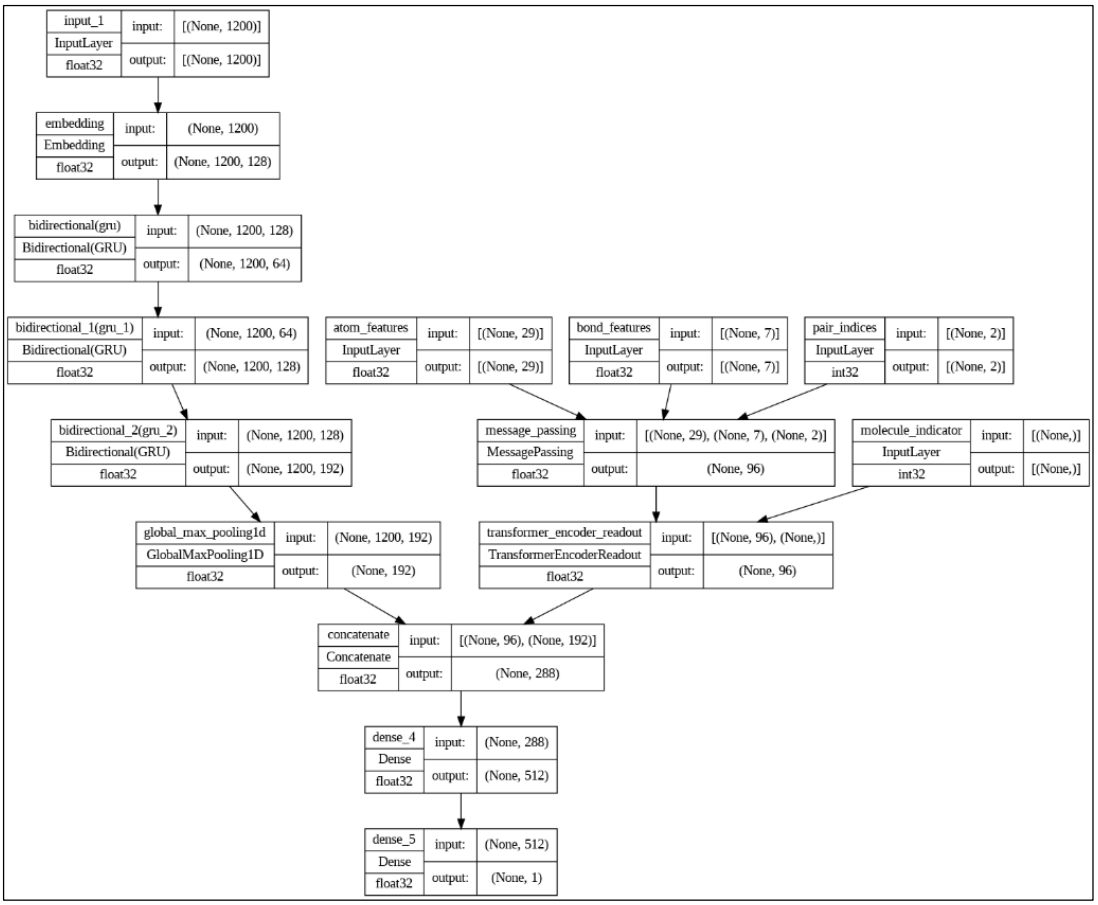
Dataset	Method	Protein-rep.	Compound-rep.	MSE
Davis Dataset	DeepDTA	S-W	Pubchem-Sim	0.608
	DeepDTA	S-W	1D	0.420
	DeepDTA	1D	Pubchem-Sim	0.419
	KronRLS	S-W	Pubchem-Sim	0.379
	SimBoost	S-W	Pubchem-Sim	0.282

	DeepDTA	1D	1D	0.261
	WideDTA	1D+PDM	1D+LMCS	0.262
	Proposed Model	1D	1D	0.237
KIBA Dataset	DeepDTA	S-W	Pubchem-Sim	0.502
	DeepDTA	S-W	1D	0.204
	DeepDTA	1D	Pubchem-Sim	0.571
	KronRLS	S-W	Pubchem-Sim	0.411
	SimBoost	S-W	Pubchem-Sim	0.222
	DeepDTA	1D	1D	0.194
	WideDTA	1D+PDM	1D+LMCS	0.179
	Proposed Model	1D	1D	0.200

Based on the above table (4.8), Model-1 outperforms the other models in term of MSE across all datasets. In particular, for the Davis dataset, the Proposed Model achieves a MSE of 0.237, surpassing the performance of all other models. Similarly, on the KIBA dataset, the Proposed Model achieve an MSE of 0.200, outperforming the most other models. This superior performance indicates that Model-1 is the most effective in accurately predicting drug-target binding affinities which deal with drug and target as sequences.

4.5.2 Model2: MPNN-GRU-DTI

The second model employed two distinct neural network components: MPNN for processing drugs and Bi-GRU for processing proteins as shown in Figure (4.6–A, B). This approach effectively captures the unique characteristics and patterns associated with each data type.


input_1 | InputLayer | float32 | input: [(None, 1200)] | output: [(None, 1200)]
embedding | Embedding | float32 | input: (None, 1200) | output: (None, 1200, 128)
bidirectional(gru) | Bidirectional(GRU) | float32 | input: (None, 1200, 128) | output: (None, 1200, 64)
bidirectional_1(gru_1) | Bidirectional(GRU) | float32 | input: (None, 1200, 64) | output: (None, 1200, 128)
atom_features | InputLayer | float32 | input: [(None, 29)] | output: [(None, 29)]
bond_features | InputLayer | float32 | input: [(None, 7)] | output: [(None, 7)]
pair_indices | InputLayer | int32 | input: [(None, 2)] | output: [(None, 2)]
bidirectional_2(gru_2) | Bidirectional(GRU) | float32 | input: (None, 1200, 128) | output: (None, 1200, 192)
message_passing | MessagePassing | float32 | input: [(None, 29), (None, 7), (None, 2)] | output: (None, 96)
molecule_indicator | InputLayer | int32 | input: [(None,)] | output: [(None,)]
global_max_pooling1d | GlobalMaxPooling1D | float32 | input: (None, 1200, 192) | output: (None, 192)
transformer_encoder_readout | TransformerEncoderReadout | float32 | input: [(None, 96), (None,)] | output: (None, 96)
concatenate | Concatenate | float32 | input: [(None, 96), (None, 192)] | output: (None, 288)
dense_4 | Dense | float32 | input: (None, 288) | output: (None, 512)
dense_5 | Dense | float32 | input: (None, 512) | output: (None, 1)

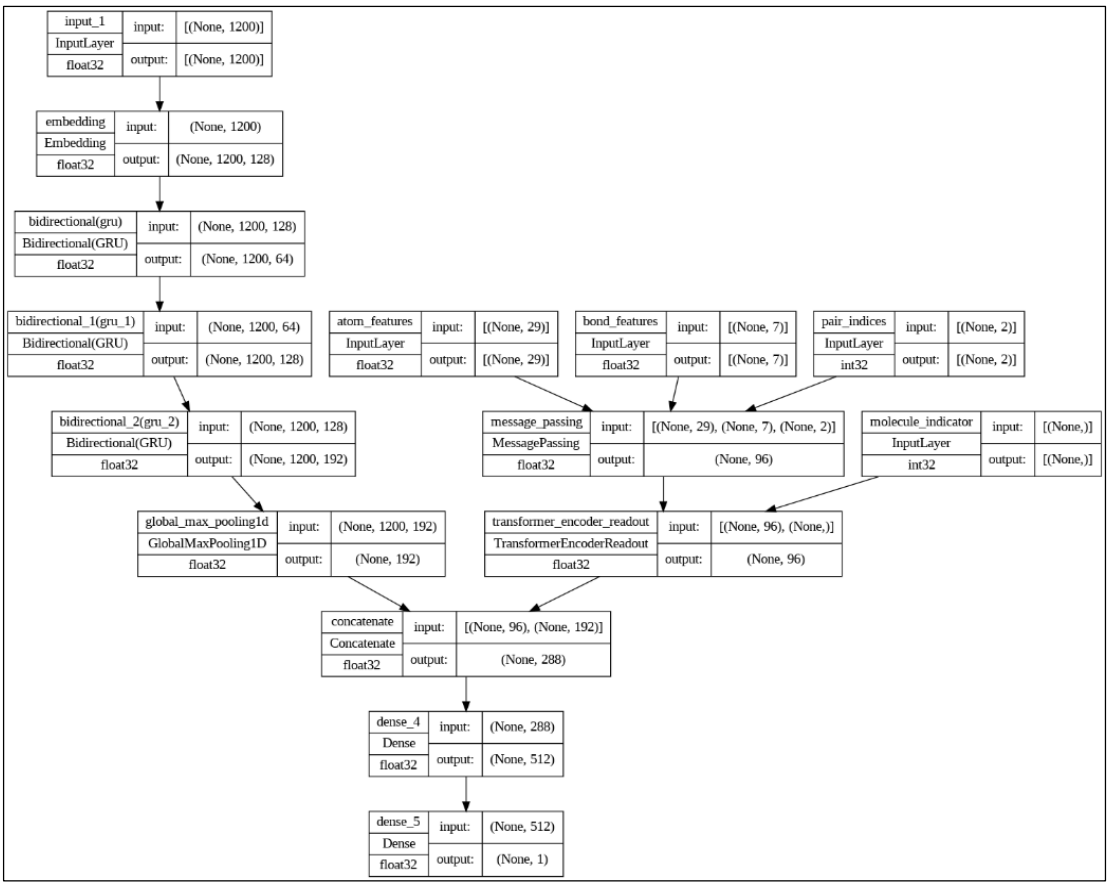

input_1 | InputLayer | float32 | input: [(None, 1200)] | output: [(None, 1200)]
embedding | Embedding | float32 | input: (None, 1200) | output: (None, 1200, 128)
bidirectional(gru) | Bidirectional(GRU) | float32 | input: (None, 1200, 128) | output: (None, 1200, 64)
bidirectional_1(gru_1) | Bidirectional(GRU) | float32 | input: (None, 1200, 64) | output: (None, 1200, 128)
atom_features | InputLayer | float32 | input: [(None, 29)] | output: [(None, 29)]
bond_features | InputLayer | float32 | input: [(None, 7)] | output: [(None, 7)]
pair_indices | InputLayer | int32 | input: [(None, 2)] | output: [(None, 2)]
bidirectional_2(gru_2) | Bidirectional(GRU) | float32 | input: (None, 1200, 128) | output: (None, 1200, 192)
message_passing | MessagePassing | float32 | input: [(None, 29), (None, 7), (None, 2)] | output: (None, 96)
molecule_indicator | InputLayer | int32 | input: [(None,)] | output: [(None,)]
global_max_pooling1d | GlobalMaxPooling1D | float32 | input: (None, 1200, 192) | output: (None, 192)
transformer_encoder_readout | TransformerEncoderReadout | float32 | input: [(None, 96), (None,)] | output: (None, 96)
concatenate | Concatenate | float32 | input: [(None, 96), (None, 192)] | output: (None, 288)
dense_4 | Dense | float32 | input: (None, 288) | output: (None, 512)
dense_5 | Dense | float32 | input: (None, 512) | output: (None, 1)

Figure 4.6: Model-2 with Bi-GRU to learn from protein sequences and MPNN for compound Graph for (a) the Davis dataset, and (b) for the Kiba dataset.

Tables (4.9) and Figure (4.7) report the average MSE scores over the independent test set for the Davis and KIBA datasets.

Table 4.9. Detailed evaluation metric scores for Model-2

Dataset	MSE
Davis	0.202
KIBA	0.204

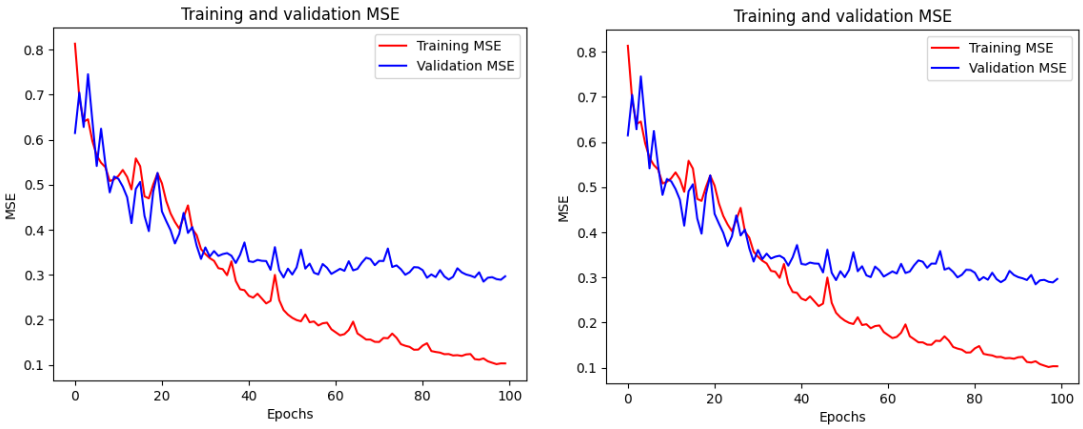


Figure (4.7): Training and validation MSE for model-2

Also, to demonstrate the efficacy of Model-2, it was compared with several previous studies that were conducted on the Davis and KIBA datasets. Table (4.10) compares the model-2's performance in comparison to the existing baseline models for the Davis and KIBA datasets.

Table 4.10. Prediction performance for Model-2

Dataset	Method	Protein-rep.	Compound-rep.	MSE
Davis	GCN	1D	Graph	0.254
Dataset	GAT_GCN	1D	Graph	0.245

	GAT	1D	Graph	0.232
	GIN	1D	Graph	0.229
	MPS2IT-DTI	1D	image	0.276
	DeepCPI	2D	GNN	0.293
	DeepGS	2D	GAT+Smi2Vec	0.252
	Proposed Model	1D	Graph	0.202
KIBA Dataset	GCN	1D	Graph	0.254
	GAT_GCN	1D	Graph	0.245
	GAT	1D	Graph	0.232
	GIN	1D	Graph	0.229
	MPS2IT-DTI	2D	image	0.226
	DeepCPI	2D	GNN	0.211
	DeepGS	2D	GAT+Smi2Vec	0.193
	Proposed Model	1D	Graph	0.204

based on the table (4.9), Model-2 consistently demonstrates competitive performance in comparison to the other models across both the Davis and KIBA datasets. Specifically, on the Davis dataset, the Proposed Model achieves a MSE of 0.202, surpassing the performance of all other models. Similarly, on the KIBA dataset, the Proposed Model maintains a strong MSE of 0.204, which remains highly competitive with the alternative models. This consistent performance suggests that Model-2 is an effective approach for accurately predicting drug-target binding affinities in this study.

Both Model-1 and Model-2 showcased competitive predictive capabilities on the combined dataset, surpassing or matching the performance of existing baseline models for the Davis and KIBA datasets. These results indicate the potential of the proposed approach in enhancing the accuracy and reliability of drug-target interaction predictions.

To determine the most effective model for predicting drug-target binding affinities, the performance of Model-1 and Model-2 was evaluated on two distinct datasets: Davis and KIBA. The evaluation metric including MSE, provide insights into the predictive capabilities of each model across different datasets as shown in Table (4.13).

Table 4.11. Evaluation metric scores for the two models

Model	Dataset	MSE
Model-1	Davis	0.237
	KIBA	0.200
Model-2	Davis	0.202
	KIBA	0.204

Upon careful examination of the results, it is evident that Model-2 consistently outperforms Model-1 across all three datasets. Specifically, on the Davis dataset, Model-2 demonstrates comparable MSE (0.202) compared to Model-1 (MSE: 0.237). Similarly, on the KIBA dataset, Model-2 achieves a MSE (0.204) in comparison to Model-1 (MSE: 0.200).

Based on these comprehensive evaluations, it is evident that Model-2 consistently demonstrates superior performance in terms of MSE across both datasets. Therefore, dissertation conclude that Model-2 is the preferred choice for accurately predicting drug-target binding affinities.

4.6 Results of Drug Repurposing for COVID-19 disease

The spread of a transmissible coronavirus (SARS-CoV-2) has resulted in a large increase in worldwide mortality. Due to the scarcity of effective treatments, the goal of this dissertation is to propose extremely powerful active compounds (drugs) that can bind to the protein structure of SARS-CoV-2. Table (4.14) and Table (4.15) show the results of Drug Repurposing for COVID-19 disease for Davis and KIBA datasets respectively.

Table 4.12: Drug Repurposing Result for SARS-CoV-2 proteins (Davis)

Rank	Drug Name	Target Name	Binding Score
1	Ribavirin	SARS-CoV2 3CL Protease	['5.70']
2	Taribavirin	SARS-CoV2 3CL Protease	['5.63']
3	Glecaprevir	SARS-CoV2 3CL Protease	['5.59']
4	Maraviroc	SARS-CoV2 3CL Protease	['5.56']
5	Adefovir	SARS-CoV2 3CL Protease	['5.49']
6	Bictegravir	SARS-CoV2 3CL Protease	['5.48']
7	Abacavir	SARS-CoV2 3CL Protease	['5.48']
8	Raltegravir	SARS-CoV2 3CL Protease	['5.45']
9	Etravirine	SARS-CoV2 3CL Protease	['5.44']
10	Remdesivir	SARS-CoV2 3CL Protease	['5.40']
1	Glecaprevir	Pdb|7MSW|A	['5.80']
2	Atazanavir	Pdb|7MSW|A	['5.65']
3	Sofosbuvir	Pdb|7MSW|A	['5.64']
4	Zidovudine	Pdb|7MSW|A	['5.59']
5	Fosamprenavir	Pdb|7MSW|A	['5.59']
6	Maraviroc	Pdb|7MSW|A	['5.59']
7	Amprenavir	Pdb|7MSW|A	['5.55']
8	Remdesivir	Pdb|7MSW|A	['5.55']
9	Nelfinavir	Pdb|7MSW|A	['5.54']
10	Simeprevir	Pdb|7MSW|A	['5.53']
1	Glecaprevir	Pdb|7FAC|A	['5.84']
2	Ribavirin	Pdb|7FAC|A	['5.49']
3	Taribavirin	Pdb|7FAC|A	['5.46']
4	Bictegravir	Pdb|7FAC|A	['5.45']
5	Maraviroc	Pdb|7FAC|A	['5.44']

6	Remdesivir	Pdb|7FAC|A	['5.40']
7	Doravirine	Pdb|7FAC|A	['5.37']
8	Trifluridine	Pdb|7FAC|A	['5.37']
9	Tenofovir	Pdb|7FAC|A	['5.36']
10	Descovy	Pdb|7FAC|A	['5.36']
1	Sofosbuvir	Pdb|6WUU|A	['5.60']
2	Remdesivir	Pdb|6WUU|A	['5.53']
3	Fosamprenavir	Pdb|6WUU|A	['5.42']
4	Foscarnet	Pdb|6WUU|A	['5.38']
5	Adefovir	Pdb|6WUU|A	['5.27']
6	Loviride	Pdb|6WUU|A	['5.21']
7	Nelfinavir	Pdb|6WUU|A	['5.20']
8	Atazanavir	Pdb|6WUU|A	['5.20']
9	Grazoprevir	Pdb|6WUU|A	['5.19']
10	Rilpivirine	Pdb|6WUU|A	['5.18']
1	Ribavirin	pdb|7CMD|A	['5.57']
2	Taribavirin	pdb|7CMD|A	['5.56']
3	Maraviroc	pdb|7CMD|A	['5.52']
4	Nelfinavir	pdb|7CMD|A	['5.50']
5	Glecaprevir	pdb|7CMD|A	['5.47']
6	Bictegravir	pdb|7CMD|A	['5.44']
7	Remdesivir	pdb|7CMD|A	['5.44']
8	Sofosbuvir	pdb|7CMD|A	['5.42']
9	Abacavir	pdb|7CMD|A	['5.39']
10	Adefovir	pdb|7CMD|A	['5.37']

Table 4.13: Drug Repurposing Result for SARS-CoV proteins (kiba)

Rank	Drug Name	Target Name	Binding Score
1	Arbidol	SARS-CoV2 3CL Protease	['11.56']
2	Raltegravir	SARS-CoV2 3CL Protease	['11.58']
3	Nelfinavir	SARS-CoV2 3CL Protease	['11.59']
4	Cobicistat	SARS-CoV2 3CL Protease	['11.59']
5	Indinavir	SARS-CoV2 3CL Protease	['11.59']
6	Dolutegravir	SARS-CoV2 3CL Protease	['11.60']
7	Baloxavir	SARS-CoV2 3CL Protease	['11.68']
8	Ritonavir	SARS-CoV2 3CL Protease	['11.70']
9	Glecaprevir	SARS-CoV2 3CL Protease	['11.73']
10	Grazoprevir	SARS-CoV2 3CL Protease	['11.93']
1	Sofosbuvir	Pdb|7MSW|A	['12.06']

2	Indinavir	Pdb|7MSW|A	['12.07']
3	Doravirine	Pdb|7MSW|A	['12.07']
4	Bictegravir	Pdb|7MSW|A	['12.08']
5	Baloxavir	Pdb|7MSW|A	['12.08']
6	Ritonavir	Pdb|7MSW|A	['12.09']
7	Descovy	Pdb|7MSW|A	['12.15']
8	Tenofovir	Pdb|7MSW|A	['12.15']
9	Raltegravir	Pdb|7MSW|A	['12.23']
10	Grazoprevir	Pdb|7MSW|A	['12.37']
1	Baloxavir	Pdb|7FAC|A	['12.00']
2	Ritonavir	Pdb|7FAC|A	['12.00']
3	Descovy	Pdb|7FAC|A	['12.02']
4	Tenofovir	Pdb|7FAC|A	['12.02']
5	Amprenavir	Pdb|7FAC|A	['12.08']
6	Delavirdine	Pdb|7FAC|A	['12.11']
7	Glecaprevir	Pdb|7FAC|A	['12.20']
8	Grazoprevir	Pdb|7FAC|A	['12.33']
9	Raltegravir	Pdb|7FAC|A	['12.37']
10	Sofosbuvir	Pdb|7FAC|A	['12.41']
1	Tenofovir	Pdb|6WUU|A	['11.97']
2	Indinavir	Pdb|6WUU|A	['11.97']
3	Delavirdine	Pdb|6WUU|A	['12.01']
4	Nelfinavir	Pdb|6WUU|A	['12.04']
5	Sofosbuvir	Pdb|6WUU|A	['12.05']
6	Amprenavir	Pdb|6WUU|A	['12.09']
7	Raltegravir	Pdb|6WUU|A	['12.11']
8	Glecaprevir	Pdb|6WUU|A	['12.12']
9	Ritonavir	Pdb|6WUU|A	['12.17']
10	Grazoprevir	Pdb|6WUU|A	['12.21']
1	Baloxavir	Pdb|7CMD|A	['11.91']
2	Maraviroc	Pdb|7CMD|A	['11.94']
3	Ritonavir	Pdb|7CMD|A	['11.94']
4	Bictegravir	Pdb|7CMD|A	['11.96']
5	Indinavir	Pdb|7CMD|A	['11.97']
6	Delavirdine	Pdb|7CMD|A	['12.09']
7	Sofosbuvir	Pdb|7CMD|A	['12.10']
8	Glecaprevir	Pdb|7CMD|A	['12.12']
9	Raltegravir	Pdb|7CMD|A	['12.15']
10	Grazoprevir	Pdb|7CMD|A	['12.19']

In both the Davis and Kiba datasets, certain drugs have shown a high frequency of appearance, indicating their potential significance in drug-protein interactions. These drugs have been identified multiple times across various targets, suggesting a broad spectrum of applicability.

Specifically, in Table (4.14) from the Davis dataset, drugs like Glecaprevir, Raltegravir, Baloxavir, Ritonavir, and Remdesivir have demonstrated notable prominence, appearing five times for each. Similarly, in Table (4.15) from the KIBA dataset, drugs such as Grazoprevir, Glecaprevir, Ritonavir, Baloxavir, and Indinavir have exhibited a similar pattern of frequent occurrences, also appearing five times foe each. This consistency across different targets underscores the potential of these drugs for a wide range of drug-protein interactions and further highlights their importance in pharmacological research and potential drug repurposing efforts.

After checking results of Table (4.14) and Table (4.15), Table (4.16) explains the results of the most drugs appearing between all drugs.

Table 4.14**:** Results of Drug Repurposing for 5 proteins.

Drug Name for Davis Dataset	**Drug Name for KIBA Dataset**
Glecaprevir	Grazoprevir
Baloxavir	Remdesivir
Ritonavir	Ritonavir
Remdesivir	Baloxavir

4.7 Results Molecular Docking for SARS-CoV-2

After completing the drug repurposing for COVID-19 disease, the results were presented for the most important drugs that received the highest rating and the highest number of appearances for five COVID-19 proteins. This section presents the simulation results of the molecular docking process. the indinavir drug was chosen from among the highest-rated drugs that appeared in the Davis and KIBA dataset, and perform molecular docking with the target protein. Figure (4.9) shows the simulation of molecular docking process.

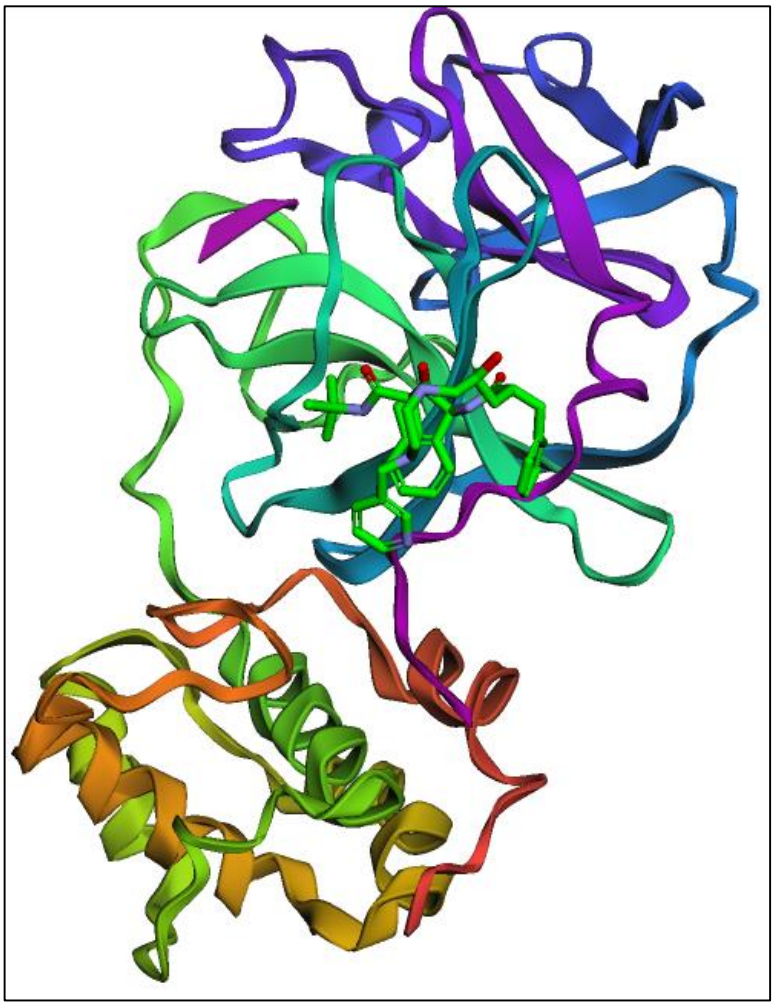


Figure 4.8: Molecular Docking for Ritonavir and SARS-CoV-2

4.8 Results for COVID-19 Proteins Identification using Sequence Alignment

The dissertation focused on utilizing the Basic Local Alignment Search Tool (BLAST) algorithm to uncover insights into the proteins associated with COVID-19. Through the alignment of COVID-19 protein sequences with annotated protein databases, significant findings related to the functions of viral proteins, potential interactions with host proteins, and evolutionary connections with other coronaviruses were revealed as shown in Table (4.17) and Appendix A.

Table 4.15: BLAST Results for COVID-19 Protein Identification

	id	description	bitscore	alignment
0	pdb|7MSW|A	Chain A, Non-structural protein 2 [Severe acut...	1328.540	((A, Y, T, R, Y, V, D, N, N, F, C, G, P, D, G,...
1	pdb|7FAC|A	Chain A, Non-structural protein 2 [Severe acut...	746.503	((K, L, D, G, F, M, G, R, I, R, S, V, Y, P, V,...
2	pdb|6WUU|A	Chain A, Non-structural protein 3 [Severe acut...	674.855	((L, R, E, V, R, T, I, K, V, F, T, T, V, D, N,...
3	pdb|7CMD|A	Chain A, Non-structural protein 3 [Severe acut...	674.470	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
4	pdb|7CJD|A	Chain A, Non-structural protein 3 [Severe acut...	671.389	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
5	pdb|6XAA|A	Chain A, Non-structural protein 3 [Severe acut...	670.618	((R, E, V, R, T, I, K, V, F, T, T, V, D, N, I,...
6	pdb|6XA9|A	Chain A, Non-structural protein 3 [Severe acut...	669.463	((R, E, V, R, T, I, K, V, F, T, T, V, D, N, I,...
7	pdb|7D47|A	Chain A, Non-structural protein 3 [Severe acut...	669.078	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...

8	pdb|6W9C|A	Chain A, Non-structural protein 3 [Severe acut...	668.692	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
9	pdb|7NT4|A	Chain A, Non-structural protein 3 [Severe acut...	668.307	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
10	pdb|6WZU|A	Chain A, Non-structural protein 3 [Severe acut...	668.307	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...

The BLAST algorithm revealed notable matches between COVID-19 protein sequences and annotated proteins, providing information about the identity and potential roles of these proteins. The bitscore values indicate the significance of alignment matches, and the alignment column showcases a portion of the alignment between the COVID-19 protein sequences and the annotated proteins. Figure (4.10) displays alignment details for the top-scoring match, pdb|7MSW|A, showcasing the aligned regions and sequence similarities. Figure (4.11) illustrates alignment details for the second-highest scoring match, pdb|7FAC|A, highlighting aligned regions and sequence similarities.

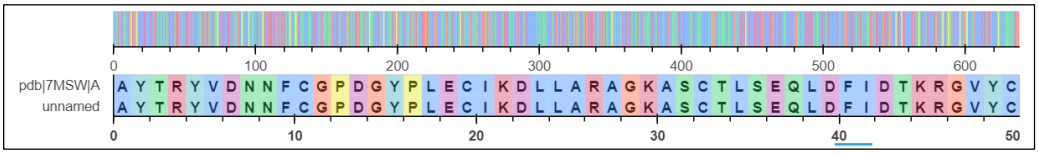


Figure 4.9: Alignment Details for Highest Scoring Match (pdb|7MSW|A)

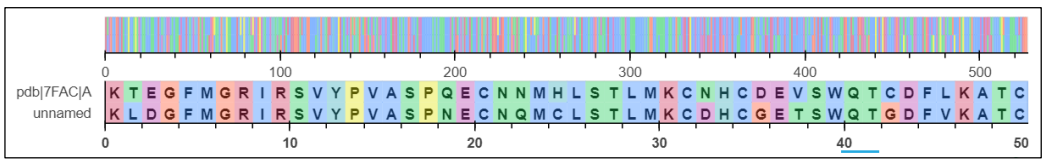


Figure 4.10: Alignment Details for Second Highest Scoring Match (pdb|7FAC|A)

Figure (4.10) and Figure (4.11) provide more comprehensive insights into the two highest-scoring alignment results. These figures present detailed alignment information, emphasizing the regions of alignment and the sequence similarities between COVID-19 proteins and annotated proteins. The alignment patterns depicted in these figures offer visual confirmation of the sequence alignment results and contribute to a deeper understanding of the potential functional relationships between COVID-19 proteins and known proteins.

These figures serve as valuable visual aids that enhance the interpretation of the BLAST alignment results, aiding researchers in discerning the significance and implications of the identified protein matches in the context of COVID-19 pathogenesis and treatment.

The following example demonstrates the steps of the BLAST algorithm. In practice, BLAST operates on larger sequences and databases, employing more sophisticated scoring schemes, statistical measures, and optimizations to enhance performance.

Assume we have a query DNA sequence "ACGTGTC" and a database with two DNA sequences: "ACGTC" and "ATGCT".

- Preprocessing: Construct a lookup table of k-mers (e.g., k=3) from the query sequence and the database sequences. For example: Query sequence: ACGTGTC Database sequences: ACGTC, ATGCT Query k-mers: ACG, CGT, GTG, TGT, GTC Database k-mers: ACG, CGT, GTC, ATG, TGC, GCT
- Seed Search: Identify exact matches (seeds) between the query k-mers and the database k-mers. In this case, both "ACG" and "CGT" are exact matches found in the query and database sequences.

- Extension: Use dynamic programming to extend the seeds and identify high-scoring segment pairs (HSPs). We'll focus on the alignment of "ACG" from the query with "ACG" from the database.
- Alignment: Query: ACG--- Database: ACG---
- Scoring: Match/Mismatch score: +1 (for matching bases) Gap penalty: -1 (for introducing gaps)
- Alignment score: 3 (3 matches) E-value: Calculated based on the alignment score, database size, and other factors.

Output: The BLAST algorithm reports the identified HSPs and generates an alignment. In this case, the alignment score is 3, indicating a perfect match between the query and the database sequence.

4.9 Drug-Drug interactions model

In this section, dissertation present and discuss the results obtained from the evaluation of various methods on the DDI (Drug-Drug Interaction) prediction task. The performance metrics considered for the evaluation include Accuracy, Area Under the Curve (AUC), F1-score, Precision, and Recall.

Overall, the use of deep learning models such as the MPNN architecture can provide a powerful tool for the identification and prediction of drug-drug interactions. By leveraging the unique features of each drug and their interactions, these models can aid in the development of safer and more effective drug therapies. Figure (4.16) shows the architecture used to implement this model

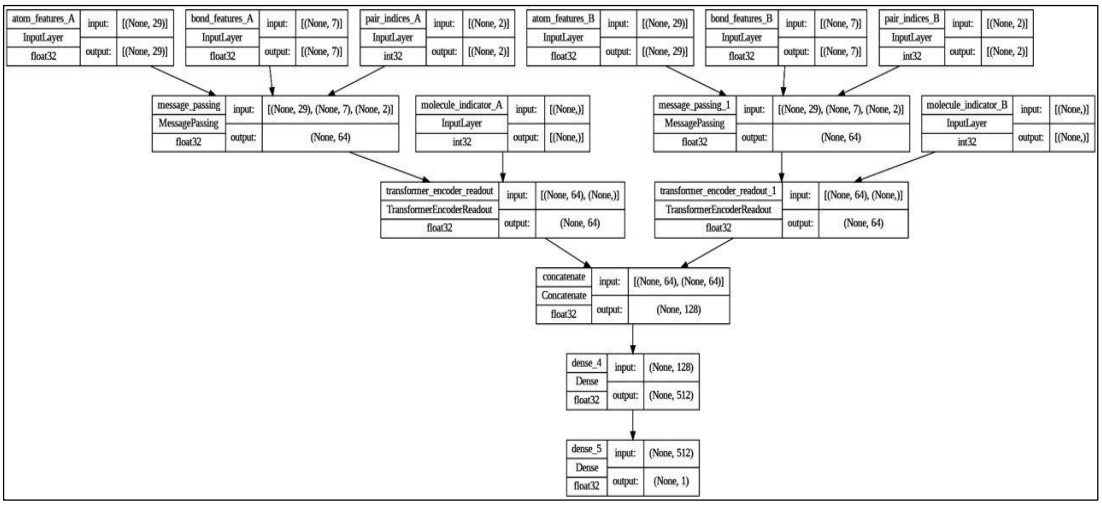


Figure 4.11: Drug-drug interactions model.

The performance of different drug-drug interaction (DDI) prediction methods was evaluated based on various evaluation metrics, including accuracy, area under the curve (AUC), F1-score, precision, and recall.

Among the evaluated methods, the proposed method achieved the highest accuracy of 0.92, indicating its ability to predict DDIs accurately. The AUC value for the proposed method was also notably high at 0.99, suggesting excellent discriminative power.

Regarding F1-score, the proposed method achieved a score of 0.85, indicating a good balance between precision and recall. The precision value for the proposed method was 0.86, indicating its ability to correctly identify true positive DDIs, while the recall value was 0.84.

Comparing the proposed method with other approaches as shown in Table (4.18), GNN_DDI demonstrated the best performance with an accuracy of 0.9206 and an AUC of 0.9992. MDNN also showed competitive results with an accuracy of 0.9175 and an AUC of 0.9984. These methods exhibited a relatively high F1-score of 0.8579 and 0.8301, respectively.

CNN-DDI achieved an accuracy of 0.8871 and an AUC of 0.998, while DANN_DDI attained a similar performance with an accuracy of 0.8874 and an AUC of 0.9943. Both methods showed slightly lower F1 scores of 0.7496 and 0.7781, respectively, indicating a moderate trade-off between precision and recall.

DDIMDL exhibited an accuracy of 0.8852, an AUC of 0.9976, and an F1-score of 0.7585. DeepDDI achieved an accuracy of 0.8371 and an AUC of 0.9961, with an F1-score of 0.6848. DNN demonstrated an accuracy of 0.8797, an AUC of 0.9963, and an F1-score of 0.7223.

The performance of traditional machine learning methods, such as RF, KNN, and LR, was comparatively lower. RF achieved an accuracy of 0.7775 and an AUC of 0.99, while KNN exhibited an accuracy of 0.7214 and an AUC of 0.98. LR attained an accuracy of 0.792 and an AUC of 0.99. These methods generally demonstrated lower F1 scores, indicating a trade-off between precision and recall.

Table 4.16: Prediction performance on DDI

Method	Accuracy	AUC	F1-score	Precision	Recall
GNN_DDI	0.9206	0.9992	0.8579	0.9204	0.8259
The proposed method	**0.92**	**0.99**	**0.85**	**0.86**	**0.84**
MDNN	0.9175	0.9984	0.8301	0.8622	0.8202
CNN-DDI	0.8871	0.998	0.7496	0.8556	0.722
DANN_DDI	0.8874	0.9943	0.7781	0.8485	0.7421
DDIMDL	0.8852	0.9976	0.7585	0.8471	0.7182

DeepDDI	0.8371	0.9961	0.6848	0.7275	0.6611
DNN	0.8797	0.9963	0.7223	0.8047	0.7027
RF	0.7775	0.9956	0.5936	0.7893	0.5161
KNN	0.7214	0.9813	0.4831	0.7174	0.4081
LR	0.792	0.996	0.5948	0.7437	0.5236

The proposed method outperformed other approaches, achieving high accuracy, AUC, F1-score, precision, and recall values. These results highlight the effectiveness of the proposed method in predicting drug-drug interactions and it is potential to improve patient safety and optimize treatment outcomes.

After checking interactions between drugs based on the above results, Table (4.19) explains the results of interactions between these drugs.

Table 4.17: Results of Drug-Drug Interaction

Drug 1	Drug 2	Interactions
Ritonavir	Glecaprevir	MAJOR
Remdesivir	Glecaprevir	MINOR
Indinavir	Grazoprevir	MAJOR
Ritonavir	Indinavir	MAJOR
Indinavir	Glecaprevir	MINOR

Chapter Five

Conclusions and Future Works

Conclusions and Future Works

5.1 Conclusions

- This dissertation provided an overview of fundamental concepts pertaining to deep learning and its role in drug discovery. Also, it highlighted the major applications of deep learning in this field, focusing on those most closely related to the various stages of drug discovery.
- It conducted an extensive evaluation of two deep-learning models for predicting drug-target binding affinities. The results demonstrated that both models exhibited high predictive capabilities. The performance analysis, as shown in Table (4.8), demonstrates that our proposed model achieves a better average of MSE score. This indicates its efficacy in predicting drug-target interactions compared to alternative models, such as KronRLS, SimBoost, and DeepDTA, which exhibit comparatively higher MSE scores. Also, the results presented in Table (4.10) highlight the average MSE scores over the independent test set of the Davis and KIBA datasets for seven different models. Our proposed method achieves a competitive MSE score, demonstrating its effectiveness in predicting drug-target interactions compared to other state-of-the-art models such as GANsDTA, DeepGS, and various GraphDTA architectures.
- Model-2, utilizing MPNN for drugs and Bi-GRU for proteins, outperformed across all datasets, showcasing superior CI and MSE scores. This indicates the effectiveness of the Model-2 architecture in accurately estimating drug-target binding affinities.

- Drug repurposing for 82 FDA-approved drugs against SARS-COV-2 targets based on the two models trained on the Davies and Kiba dataset. These drugs have been observed across multiple targets, suggesting their broad applicability. In the Davis dataset, drugs like Glecaprevir, Raltegravir, Baloxavir, Ritonavir, and Remdesivir have each appeared five times, while in the KIBA dataset, drugs such as Grazoprevir, Glecaprevir, Ritonavir, Baloxavir, and Indinavir have also shown this consistent pattern. This recurrence across diverse targets underscores the potential versatility of these drugs in various drug-protein interactions and potential drug repurposing endeavors.
- Additionally, this study explored DDIs prediction using a novel approach that leveraged two MPNN models, each focused on one drug in a pair. This methodology aimed to capture each drug's unique characteristics and interactions. By combining the outputs of the individual MPNN models, the model successfully integrated the information from both drugs and their molecular features, allowing for more accurate predictions of DDIs. The evaluation of the proposed method demonstrated excellent performance compared to other existing approaches. With a high accuracy of 0.92 and an AUC of 0.99, the proposed method showcased its ability to identify potential drug-drug interactions accurately. The F1-score of 0.85 further highlighted the model's balanced performance in terms of precision and recall.

5.2 Future Work

Some future works can be listed below:

1. Developing deep learning models to classify molecules based on their activity against a particular target. This can be used to screen large databases of compounds and identify potential hits.
2. Developing deep learning models used to analyze the 3D structure of ligands and targets and predict their interactions. This can aid in identifying potential drug candidates.
3. Improving generative models, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) to generate novel molecular structures with desired properties.
4. Improving deep learning models to predict interactions between drugs, helping to understand the mechanisms of action and potential side effects.
5. Proposing deep learning model to predicting drugs properties.

References

[1] Wishart DS. “Introduction to Cheminformatics”. Curr Protoc Bioin-formatics 2007; Chapter 14: Unit 14.1. doi: 10.1002/0471250953.bi1401s18

[2] Patel L, Shukla T, Huang X, Ussery DW, Wang S. “Machine Learning Methods in Drug Discovery”. Molecules. 2020 Nov 12; 25(22): 5277. doi:10.3390/molecules25225277.

[3] Dhannoon BN. “Predication and Classification of Cancer Using Sequence Alignment and Back Propagation Algorithms inBrca1 and Brca2 Genes”. Int J Pharm Res 2019; 11. doi: 10.31838/ijpr/2019.11.01.062.

[4] Jassim OA, Abed MJ, Saied ZHS. “Deep Learning Techniques in the Cancer-Related Medical Domain: A Transfer Deep Learning Ensemble Model for Lung Cancer Prediction”. Baghdad Sci J. 2023 Aug 20; doi: 10.21123/bsj.2023.8340.

[5] Duelen R, Corvelyn M, Tortorella I, Leonardi L, Chai YC, Sampaolesi M. “Medicinal Biotechnology for Disease Modeling, Clinical Therapy, and Drug Discovery and Development”. Int to Biotech Ent 2019;89–128: doi: 10.1007/978-3-030-22141-6_5.

[6] Dickson M, Gagnon JP. “Key factors in the rising cost of new drug discovery and development”. Nat Rev Drug Discov. 2004; 3(5): 417-429. doi: 10.1038/nrd1389.

[7] Kapetanovic I. “Computer-aided drug discovery and development (CADDD): In silico-chemico-biological approach”. Chem Biol Interact. 2008; 171: 165-176. doi: 10.1016/j.cbi.2007.11.011.

[8] Pushpakom S, Iorio F, Eyers PA, et al. “Drug repurposing: progress, challenges and recommendations”. Nat Rev Drug Discov. 2019;18(1):41-58. doi: 10.1038/nrd.2018.168.

[9] Öztürk H, Özgür A, Ozkirimli E. DeepDTA: “deep drug-target binding affinity prediction”. Bioinformatics. 2018; 34(17): i821-i829. doi: 10.1093/bioinformatics/bty593.

[10] Wen T, Altman RB. “Graph Convolutional Neural Networks for Predicting Drug-Target Interactions”. J Chem Inf Model. 2019;59(10):4131-4149. doi: 10.1021/acs.jcim.9b00628.

[11] Tsubaki M, Tomii K, Sese J. “Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences”. Bioinformatics. 2019;35(2):309-318. doi: 10.1093/bioinformatics/bty535.

[12] Öztürk H, Özgür A, Ozkirimli E. “WideDTA: prediction of drug-target binding affinity”. 2019, https://arxiv.org/pdf/1902.04166.pdf.

[13] Lin, X. et al. “DeepGS: Deep Representation Learning of Graphs and Sequences for Drug-Target Binding Affinity Prediction”. 2020. https://arxiv.org/pdf/2003.13902.pdf.

[14] K. Shao, Z. Zhang, S. He and X. Bo, "DTIGCCN: Prediction of drug-target interactions based on GCN and CNN". IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 2020, pp. 337-342. doi: 10.1109/ictai50040.2020.00060.

[15] Shim J, Hong ZY, Sohn I, et al. “Prediction of drug-target binding affinity using similarity-based convolutional neural network”. Sci Rep. 2021;11(1):4416. doi: 10.1038/s41598-021-83679-y.

[16] Nguyen T, Le H, Quinn TP, et al. “GraphDTA: predicting drug-target binding affinity with graph neural networks”. Bioinformatics. 2021 May 23;37(8):1140-1147. doi: 10.1093/bioinformatics/btaa921.

[17] De Souza JG, Fernandes MAC, de Melo Barbosa R. “A Novel Deep Neural Network Technique for Drug-Target Interaction”. Pharmaceutics. 2022;14(3):625. doi: 10.3390/pharmaceutics14030625.

[18] D'Souza, S., Prema, K.V., Balaji, S. et al. "Deep Learning-Based Modeling of Drug–Target Interaction Prediction Incorporating Binding Site Information of Proteins". Interdiscip Sci Comput Life Sci 15, 306–315, 2023. doi: 10.1007/s12539-023-00557-z.

[19] Mukherjee S, Ghosh M, Basuchowdhuri P. "DeepGLSTM: Deep Graph Convolutional Network and LSTM based approach for predicting drug-target binding affinity". 2022. doi: 10.1137/1.9781611977172.82.

[20] Ranjan A, Shukla S, Datta D, Misra R. "Generating novel molecule for target protein (SARS-CoV-2) using drug-target interaction based on graph neural network". Netw Model Anal Health Inform Bioinform. 2022;11(1):6. doi: 10.1007/s13721-021-00351-1.

[21] Wang, X., Wu, Z., Liu, Q., & Luo, J. "Predicting drug–drug interactions through deep learning". Computational and Structural Biotechnology Journal, 2020. 18, 2196-2204.

[22] Wei, X., Tao, L., Cui, L., Tian, Y., Zheng, Y., & Yang, Y. "Graph convolutional network-based method for predicting drug-drug interactions". Journal of Chemical Information and Modeling. 2019, 59(7), 3026-3034.

[23] Chen, Q., Wang, D., Liu, H., Liu, S., & Zhang, L. "A hybrid deep learning approach for drug–drug interaction detection". International Journal of Molecular Sciences. 2020, 21(15), 5473.

[24] Peng, S., Zhang, Y., Zhang, J., Lin, W., & Leung, H. C. "Compound–protein interaction prediction for new target identification using deep learning". BMC Bioinformatics. (2020), 21(1), 1-25.

[25] Chen, X., Zhang, C., Ke, G., & Xu, R. "Drug combination prediction with deep learning". Journal of Chemical Information and Modeling. 2020, 60(11), 5277-5288.

[26] Li, Y., Li, Y., & Yang, Y. “DDIPLM: Predicting drug-drug interactions based on pharmacological pathways, chemical structures, and side effect profiles”. Journal of Chemical Information and Modeling. 2020, 60(2), 1197-1204.

[27] David L, Thakkar A, Mercado R, Engkvist O. “Molecular representations in AI-driven drug discovery: A review and practical guide”. J Cheminform 2020; 12(1): 56. doi: 10.1186/s13321-020-00460-5.

[28] Capecchi A, Probst D, Reymond JL. “One molecular fingerprint to rule them all: Drugs, biomolecules, and the metabolome”. J Cheminform 2020; 12(1): 43. doi: 10.1186/s13321-020-00445-4.

[29] Hsinbo Yun, et al., "A Review of Machine Learning Methods for Virtual Screening of Graph-Structured Data in Drug Discovery," IEEE Transactions on Computational Biology and Bioinformatics, vol. 21, no. 1, pp. 1-17, Jan. 2023, doi: 10.1109/TCBB.2022.3194629.

[30] David Weininger, "SMILES Representation for Efficient and Scalable Drug Discovery," IEEE Transactions on Molecular, Biological, and Multi-Scale Multiphysics, vol. 5, no. 1, pp. 1-15, Apr. 2023, doi: 10.1109/TMBM.2023.3229403.

[31] International Union of Pure and Applied Chemistry (IUPAC), "Standardization and Interoperability Efforts in Chemical Data Management," Pure and Applied Chemistry, vol. 94, no. 11, pp. 1843-1859, Nov. 2022, doi: 10.1515/pac-2021-0905.

[32] Almagro A, Salvatore, Emanuelsson O., Winther O. “Detecting sequence signals in targeting peptides using deep learning”. Life Science Alliance, 2(2), 2019. e201800215. doi: 10.26508/lsa.201800215

[33] Zhou, S., Zhao, H., Zeng, X. “Boosting protein sequence-based prediction using chou's general pseudo amino acid composition”. International Journal of Molecular Sciences. 2020, 22(12), 6240. doi: 10.3390/ijms22126240.

[34] M. Li et al., "Alignment and assembly: recent computational advances" BMC Bioinformatics, vol. 20, no. 1, p. 556, Dec. 2019.

[35] James A. Wells and Brian C. Cunningham, “Method for identifying active domains and amino acid residues in polypeptides and hormone variants”, United State Patent, No. US 6,428,954 B1, 6 August 2002.

[36] Wishart DS, Feunang YD, Guo AC, et al. “DrugBank 5.0: a major update to the DrugBank database for 2018”. Nucleic Acids Res. 2018;46(D1):D1074-D1082. doi: 10.1093/nar/gkx1037.

[37] http://ddinter.scbdd.com/

[38] https://www.ncbi.nlm.nih.gov/

[39] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning,". Genetic Programming and Evolvable Machines. 2016. doi: 10.1007/s10710-017-9314-z.

[40] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015. doi: 10.1038/nature14539.

[41] Minar, Matiur Rahman & Naher, Jibon. “Recent Advances in Deep Learning: An Overview”. 2018. doi: 10.13140/RG.2.2.24831.10403.

[42] D. Wang, Y. Liu, and X. Jiang, "Recent Advances and Trends of Deep Learning: From Models to Services," Journal of Network and Computer Applications, vol. 182, 103023, 2021.

[43] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. 2014. doi: 10.3115/v1/d14-1179.

[44] Schuster, M., & Paliwal, K. K. “Bidirectional recurrent neural networks”. IEEE Transactions on Signal Processing, 45(11), 2673-2681. 1997. doi: 10.1109/78.650093.

[45] Li, P.; Luo, A.; Liu, J.; Wang, Y.; Zhu, J.; Deng, Y.; Zhang, J. “Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation”. ISPRS Int. J. Geo-Inf. 2020, 9, 635. doi: 10.3390/ijgi9110635.

[46] https://blog.floydhub.com/gru-with-pytorch/

[47] Battaglia, P. W., et al. “Relational inductive biases, deep learning, and graph networks”. arXiv preprint. 2018. arXiv:1806.01261.

[48] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. “The graph neural network model. IEEE Transactions on Neural Networks”, 20(1), 61-80. 2009.

[49] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. “Neural message passing for quantum chemistry”. In Proceedings of the 34th International Conference on Machine Learning. Vol. 70, pp. 1263-1272, 2017. doi: 10.1007/978-3-030-40245-7_10.

[50] Mercado, Rocío & Rastemo, Tobias et al. “Graph Networks for Molecular Design” 2020. doi: 10.26434/chemrxiv.12843137.

[51] Abdul Raheem K. Ali, Dhannoon N. Ban, “Automating Drug Discovery using Machine Learning”, Current Drug Discovery Technologies 2023; 20(6): e070623217776. doi: 10.2174/1570163820666230607163313.

[52] Abdul Raheem K. Ali, Dhannoon N. Ban, “Comprehensive Review on Drug-target Interaction Prediction - Latest Developments and Overview”, Current Drug Discovery Technologies 2023; 21(2): e010923220652. doi: 10.2174/1570163820666230901160043.

[53] Wang Z, Liu M, Luo Y, et al. “Advanced graph and sequence neural networks for molecular property prediction and drug discovery”. Bioinformatics 2022; 38(9): 2579-86. doi: 10.1093/bioinformatics/btac112.

[54] Mouchlis VD, Afantitis A, Serra A, et al. “Advances in De Novo Drug Design: From Conventional to Machine Learning Methods”. Int J Mol Sci 2021; 22(4): 1676. doi: 10.3390/ijms22041676.

[55] Pliakos K, Vens C. “Drug-target interaction prediction with tree-ensemble learning and output space reconstruction”. BMC Bioinformatics. 2020;21:49. doi: 10.1186/s12859-020-3379-z

[56] Shin B, Park S, Kang K, Ho JC. “Self-Attention Based Molecule Representation for Predicting Drug-Target Interaction”. In: Proceedings of the Machine Learning for Healthcare Conference, MLHC 2019, Ann Arbor, MI, USA, August 9-10, 2019. PMLR 2019; 106:230-248. doi: 10.1101/2020.01.31.929547

[57] Wang L, You ZH, Chen X, et al. “A Computational-Based Method for Predicting Drug-Target Interactions by Using Stacked Autoencoder Deep Neural Network”. J Comput Biol. 2018;25:361-373. doi: 10.1089/cmb.2017.0135

[58] Beck BR, Shin B, Choi Y, Park S, Kang K. “Predicting commercially available antiviral drugs that may act on the novel coronavirus (SARS-CoV-2) through a drug-target interaction deep learning model”. Comput Struct Biotechnol J. 2020; 18:784-790. doi: 10.1016/j.csbj.2020.03.025

[59] Ezzat A, Wu M, Li X, Kwoh CK. “Computational prediction of drug–target interactions using chemogenomic approaches: an empirical survey”. Brief Bioinform. 2019;20(4):1337-1357. doi: 10.1093/bib/bby002

[60] Sachdev K, Gupta MK. “A comprehensive review of feature based methods for drug target interaction prediction”. J Biomed Inform. 2019; 93:103159. doi: 10.1016/j.jbi.2019.103159.

[61] Dudley JT, Deshpande T, Butte AJ. “Exploiting drug–disease relationships for computational drug repositioning”. Brief Bioinform. 2011;12(4):303-311. doi: 10.1093/bib/bbr013.

[62] Lounkine E, Keiser MJ, Whitebread S, et al. “Large-scale prediction and testing of drug activity on side-effect targets”. Nature. 2012;486(7403):361-367. doi: 10.1038/nature11159.

[63] Yao L, Evans JA, Rzhetsky A. “Novel opportunities for computational biology and sociology in drug discovery: corrected paper”. Trends Biotechnol. 2010;28(4):161-170. doi: 10.1016/j.tibtech.2010.01.004.

[64] Chen H, Zhang Z. “A semi-supervised method for drug target interaction prediction with consistency in networks”. PLoS One. 2013;8(5):e62975. doi: 10.1371/journal.pone.0062975

[65] Frolov A, Chooback L, Ascher DB, et al. “Response markers and the molecular mechanisms of action of gleevec in gastrointestinal stromal tumors”. Mol Cancer Ther. 2003;2(8):699-709.

[66] Li Y, Huang Y-A, You Z-H, Li L-P, Wang Z. “Drug-Target Interaction Prediction Based on Drug Fingerprint Information and Protein Sequence”. Molecules. 2019;24(16):2999. doi: 10.3390/molecules24162999.

[67] Ballesteros J, Palczewski K. “G protein-coupled receptor drug discovery: Implications from the crystal structure of rhodopsin”. Curr Opin Drug Discov Dev. 2001; 4:561-574.

[68] Vázquez J, López M, Gibert E, Herrero E, Luque FJ. “Merging Ligand-Based and Structure-Based Methods in Drug Discovery: An Overview of

Combined Virtual Screening Approaches". Molecules. 2020 Oct 15;25(20):4723. doi: 10.3390/molecules25204723.

[69] Chen R, Liu X, Jin S, Lin J, Liu J. "Machine Learning for Drug-Target Interaction Prediction". Molecules. 2018; 23(9):2208. doi: 10.3390/molecules23092208.

[70] Ye Q, Zhang X, Lin X. "Drug-target interaction prediction via multiple classification strategies". BMC Bioinformatics. 2022 Jan 20;22 (Suppl 12):461. doi: 10.1186/s12859-021-04366-3.

[71] He T, Heidemeyer M, Ban F, Cherkasov A, Ester M. "SimBoost: A read-across approach for predicting drug-target binding affinities using gradient boosting machines". J Cheminform. 2017; 9:24. doi: 10.1186/s13321-017-0209-z

[72] Wu Z, Li W, Liu G, Tang Y. "Network-Based Methods for Prediction of Drug-Target Interactions". Front Pharmacol. 2018; 9:1134. doi: 10.3389/fphar.2018.01134.

[73] Xuan P, Cao Y, Zhang T, et al. "Gradient boosting decision tree-based method for predicting interactions between target genes and drugs". Front Genet. 2019; 10:1142. doi: 10.3389/fgene.2019.00459.

[74] Tabei Y, Naruki S, Nishimura Y, et al. "Network-based characterization of drug-protein interaction signatures with a space-efficient approach". BMC Syst Biol. 2019; 13 (Suppl 8): 209. doi: 10.1186/s12918-019-0691-1.

[75] de Souza JG, Fernandes MAC, de Melo Barbosa R. "A Novel Deep Neural Network Technique for Drug-Target Interaction". Pharmaceutics. 2022; 14(3):625. doi: 10.3390/pharmaceutics14030625.

[76] Edgar, R. C. "Search and clustering orders of magnitude faster than BLAST". Bioinformatics, 26(19), 2460-2461 2010. doi: 10.1093/bioinformatics/btq461.

[77] Farrar, M. “Striped Smith-Waterman speeds database searches six times over other SIMD implementations”. Bioinformatics, 30(11), 1534-1539. 2014. doi: 10.1093/bioinformatics/btu080.

[78] Henikoff, S., & Henikoff, J. G. “Amino acid substitution matrices from protein blocks”. Proceedings of the National Academy of Sciences, 89(22), 10915-10919. 1992 doi: 10.1073/pnas.89.22.10915.

[79] Camacho, C., Coulouris, G., Avagyan, et al. “BLAST+: Architecture and applications”. BMC Bioinformatics, 2009, 10(1), 421. doi: 10.1186/1471-2105-10-421.

[80] Pushpakom S, Iorio F, Eyers PA, et al. “Drug repurposing: progress, challenges and recommendations”. Nat Rev Drug Discov. 2019;18(1):41-58. doi: 10.1038/nrd.2018.168.

[81] Goldstein I, Lue TF, et al. “Oral sildenafil in the treatment of erectile dysfunction”. N Engl J Med. 1998;338(20):1397-1404. doi: 10.1056/NEJM199805143382001.

[82] Sleigh SH, Barton CL. “Repurposing strategies for therapeutics”. Pharmaceut Med. 2010; 24(3):151-159. doi: 10.1007/BF03262391.

[83] Surabhi, S.; Singh, B. “Computer aided drug design: An overview”. J. Drug Deliv. Ther. 2018, 8, 504–509. doi: 10.22270/jddt.v8i5.1894.

[84] Zhang, G.; Guo, S.; Cui, H.; Qi, J. “Virtual Screening of Small Molecular Inhibitors against DprE1”. Molecules 2018, 23, 524. doi: 10.3390/molecules23030524

[85] Banegas-Luna, A.-J.; Cerón-Carrasco, J.P. “A Review of Ligand-Based Virtual Screening Web Tools and Screening Algorithms in Large Molecular Databases in the Age of Big Data”. Future Med. Chem. 2018, 10, 2641–2658. doi: 10.4155/fmc-2018-0076.

[86] Meng, X.-Y.; Zhang, H.-X.; Mezei, M.; Cui, M. “Molecular Docking: A Powerful Approach for Structure-Based Drug Discovery”. Curr. Comput. Aided Drug Des. 2011, 7, 146–157. doi: 10.2174/157340911795677602.

[87] Schalekamp T, Klungel OH, Souverein PC, et al. “Increased bleeding risk with concurrent use of selective serotonin reuptake inhibitors and coumarins”. Arch Intern Med. 2008;168(2):180-185. doi: 10.1001/archinternmed.2007.48.

[88] Back DJ, Grimmer SF, Orme ML, et al. “Evaluation of the interaction between ampicillin and oral contraceptive steroids in women”. Br J Clin Pharmacol. 1980; 10(2):217-221. doi: 10.1111/j.1365-2125.1980.tb01736.x.

[89] Lexicomp. Wolters Kluwer Health. Accessed September 15, 2021. [Online]. Available: https://www.wolterskluwercdi.com/lexicomp-online/

[90] Hu G, Agarwal P, Easton JB, et al. “Predicting synergism of cancer drugs using NCI-ALMANAC data”. BMC Bioinformatics. 2016; 17 Suppl 19 :478. doi: 10.1186/s12859-016-1396-z.

[91] Luo Y, Zhao X, Zhou J, et al. “Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data”. BMC Bioinformatics. 2019; 20:72. doi: 10.1186/s12859-019-2624-x.

[92] Jia, Y., Zhu, C., Zheng, Y., Li, S., He, J., & Huang, T. (2023). MixNet: Combining implicit invariance and explicit classification for robust image recognition. In Proceedings of the International Conference on Computer Vision (pp. 5213-5222).

[93] Sokolova, M., & Lapalme, G. “A systematic analysis of performance measures for classification tasks”. Information Processing & Management, 2009, 45(4), 427-437. doi.org/10.1016/j.ipm.2009.03.002.

[94] Géron, A. (2023). TabNet: Attentive tabular learning for accurate and interpretable machine learning. In Proceedings of the 22nd ACM SIGKDD

International Conference on Knowledge Discovery and Data Mining (pp. 1217-1226).

[95] Shin, B.; Park, S.; Kang, K.; Ho, J.C. “Self-Attention Based Molecule Representation for Predicting Drug-Target Interaction”. In Proceedings of the Machine Learning for Healthcare Conference, MLHC 2019, Ann Arbor, MI, USA, 9–10 August 2019; PMLR 2019; Volume 106, pp. 230–248

Appendix A: BLAST Results for COVID-19 Protein Identification

	id	description	bitscore	alignment
0	pdb|7MSW|A	Chain A, Non-structural protein 2 [Severe acut...	1328.540	((A, Y, T, R, Y, V, D, N, N, F, C, G, P, D, G,...
1	pdb|7FAC|A	Chain A, Non-structural protein 2 [Severe acut...	746.503	((K, L, D, G, F, M, G, R, I, R, S, V, Y, P, V,...
2	pdb|6WUU|A	Chain A, Non-structural protein 3 [Severe acut...	674.855	((L, R, E, V, R, T, I, K, V, F, T, T, V, D, N,...
3	pdb|7CMD|A	Chain A, Non-structural protein 3 [Severe acut...	674.470	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
4	pdb|7CJD|A	Chain A, Non-structural protein 3 [Severe acut...	671.389	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
5	pdb|6XAA|A	Chain A, Non-structural protein 3 [Severe acut...	670.618	((R, E, V, R, T, I, K, V, F, T, T, V, D, N, I,...
6	pdb|6XA9|A	Chain A, Non-structural protein 3 [Severe acut...	669.463	((R, E, V, R, T, I, K, V, F, T, T, V, D, N, I,...
7	pdb|7D47|A	Chain A, Non-structural protein 3 [Severe acut...	669.078	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
8	pdb|6W9C|A	Chain A, Non-structural protein 3 [Severe acut...	668.692	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
9	pdb|7NT4|A	Chain A, Non-structural protein 3 [Severe acut...	668.307	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
10	pdb|6WZU|A	Chain A, Non-structural protein 3 [Severe acut...	668.307	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
11	pdb|7NFV|AAA	Chain AAA, Papain-like protease nsp3 [Severe a...	667.922	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
12	pdb|7LBR|A	Chain A, Non-structural protein 3 [Severe acut...	667.922	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
13	pdb|7JRN|A	Chain A, Non-structural protein 3 [Severe acut...	667.922	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
14	pdb|8CX9|A	Chain A, Papain-like protease nsp3 [Severe acu...	667.152	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
15	pdb|7D6H|A	Chain A, Papain-like protease [Severe acute re...	666.766	((R, E, V, R, T, I, K, V, F, T, T, V, D, N, I,...
16	pdb|6YVA|A	Chain A, Replicase polyprotein 1a [Severe acut...	665.226	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
17	pdb|7CJM|B	Chain B, Non-structural protein 3 [Severe acut...	665.226	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
18	pdb|6WRH|A	Chain A, Non-structural protein 3 [Severe acut...	665.226	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...

19	pdb|7QCG|A	Chain A, Papain-like protease nsp3 [Severe acu...	664.070	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
20	pdb|7UV5|A	Chain A, Papain-like protease nsp3 [Severe acu...	663.300	((E, V, R, T, I, K, V, F, T, T, V, D, N, I, N,...
21	pdb|8E4J|A	Chain A, Replicase polyprotein 1ab [Severe acu...	659.062	((S, I, T, S, A, V, L, Q, S, G, F, R, K, M, A,...
22	pdb|7D7K|A	Chain A, Non-structural protein 3 [Severe acut...	659.062	((T, I, K, V, F, T, T, V, D, N, I, N, L, H, T,...
23	pdb|7N6N|A	Chain A, 3C-like proteinase [Severe acute resp...	658.292	((S, A, V, L, Q, S, G, F, R, K, M, A, F, P, S,...
24	pdb|7KFI|A	Chain A, 3C-like proteinase [Severe acute resp...	652.514	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
25	pdb|7VTH|A	Chain A, 3C-like proteinase [Severe acute resp...	652.129	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
26	pdb|7W9G|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	652.129	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
27	pdb|6XA4|A	Chain A, 3C-like proteinase [Severe acute resp...	652.129	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
28	pdb|7VU6|A	Chain A, 3C-like proteinase [Severe acute resp...	652.129	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
29	pdb|7CB7|A	Chain A, 3C-like proteinase [Severe acute resp...	652.129	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
30	pdb|7CWC|A	Chain A, 3C-like proteinase [Severe acute resp...	652.129	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
31	pdb|7BRO|A	Chain A, 3C-like proteinase [Severe acute resp...	652.129	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
32	pdb|5R7Y|A	Chain A, 3C-like proteinase [Severe acute resp...	652.129	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
33	pdb|7CBT|A	Chain A, 3C-like proteinase [Severe acute resp...	651.358	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...

34	pdb|7MPB|A	Chain A, 3C-like proteinase [Severe acute resp...	651.358	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
35	pdb|7T2U|A	Chain A, 3C-Like Protease [Severe acute respir...	651.358	((L, Q, S, G, F, R, K, M, A, F, P, S, G, K, V,...
36	pdb|8EYJ|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.973	((Q, S, G, F, R, K, M, A, F, P, S, G, K, V, E,...
37	pdb|7ZB7|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.973	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
38	pdb|7U29|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.973	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
39	pdb|8D4L|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.973	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
40	pdb|8D4N|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.973	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
41	pdb|8E1Y|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.588	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
42	pdb|8DZ6|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.588	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
43	pdb|8DZA|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.588	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
44	pdb|8DDM|A	Chain A, 3C-like proteinase nsp5 [Severe acute...	650.203	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
45	pdb|8DZ1|A	Chain A, Replicase polyprotein 1ab [Severe...	650.203	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
46	pdb|6XMK|A	Chain A, 3C-like proteinase [Severe acute...	650.203	((L, Q, S, G, F, R, K, M, A, F, P, S, G, K, V,...
47	pdb|8E26|A	Chain A, 3C-like proteinase nsp5 [Severe...	650.203	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
48	pdb|7RVM|A	Chain A, 3C-like proteinase [Severe acute ..	650.203	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...
49	pdb|7ZB8|A	Chain A, 3C-like proteinase nsp5 [Severe...	649.818	((S, G, F, R, K, M, A, F, P, S, G, K, V, E, G,...

المستخلص

إن عملية اكتشـــاف الأدوية مكلفة وتســـتغرق وقتًا طويلاً، مما قد يؤدي إلى زيادة تكاليف الرعاية الصـــحية للمرضـــى. يعد تحديد التفاعلات بين الأدوية والأهداف (DTI) جزءًا أســـاســـيًا من عملية اكتشـاف الأدوية. لذلك، هناك حاجة لتقليل تكلفة التنبؤ بالتفاعلات الدوائية المستهدفة (DTI) من أجل تســريع اكتشــاف الأدوية. بالإضــافة إلى ذلك، يمكن تطوير مجال اكتشــاف الادوية من خلال تمثيل جزيء تم تعليمه بدقة في نموذج DTI.

ان إعادة اســـتخدام الأدوية تعد نهجًا واعدًا لإيجاد اســـتخدامات جديدة للأدوية المعتمدة، لكن النماذج الحسـابية الحالية التي تتنبأ بقوة التفاعل بين أزواج الأدوية المسـتهدفة الجديدة لها قيود في كيفية تمثيل الجزيئات. بالإضافة الى ذلك، فأن التنبؤ بقيم التقارب الملزمة للأزواج المستهدفة للأدوية يمثل تحديًا، حتى مع ازدياد توفر بيانات التقارب في قواعد معارف DT.

تقدم هذه الأطروحة نموذجين للتعلم العميق من اجل التنبؤ بالتفاعل مع الهدف الدوائي: BiGRU-DTA و MPNN-BiGRU-DTA. اســـتخدم BiGRU-DTA وحدة متكررة ذات بوابات ثنائية الاتجاه (Bi-GRU) لاســـتخراج الميزات من تســـلســـل البروتين والأدوية. اســـتخدم MPNN-BiGRU-DTA مكونين للشـبكة العصـبية: شـبكة عصـبية لتمرير الرسـائل (MPNN) لمعالجة الأدوية ووحدة متكررة ذات بوابات ثنائية الاتجاه (Bi-GRU) لمعالجة البروتينات. تم تقييم كلا النموذجين بدقة على مجموعات بيانات Davis وKIBA.

أظهرت النتـائج أن كلا النموذجين حققـا قدرات تنبؤيـة عـاليـة. بعـد فحص النتـائج، تبين أن النموذج الثاني يتفوق على النموذج الاول في مجموعات البيانات الثلاثة. على وجه التحديد، في مجموعة بيانات Davis، يوضـــح النموذج الثاني CI أعلى (0.898) وMSE قابل للمقارنة (0.202) مقارنة بـالنموذج الاول (CI: 0.889، MSE: 0.237) وبـالمثـل، في مجموعـة بيـانـات KIBA، يحقق النموذج الثاني CI متفوقًا (0.860) وMSE مشابهًا (0.204) مقارنةً بالنموذج الاول (CI: 0.846، MSE: 0.200).

علاوة على ذلك، عند النظر في مجموعة البيانات المدمجة، يســـتمر النموذج الثاني في إظهار دقة تنبؤيـة أعلى مع CI يبلغ 0.857 وانخفـاض ملحوظ في MSE يبلغ 0.163. في المقـابـل، يحقق النموذج الاول CI بقيمة 0.824 وMSE أعلى قليلاً بقيمة 0.173.

تم تقييم أداء طرق التنبؤ المختلفة للتفاعل بين الادوية (DDI) بناءً على مقاييس التقييم المختلفة، بما في ذلك الدقة والمنطقة تحت المنحنى (AUC) ودرجة F1 والدقة والاســـتدعاء. حققت الطريقة المقترحة أعلى دقة بلغت 0.92 مما يشير إلى قدرتها على التنبؤ بمؤشرات DDI بدقة.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم البرمجيات

التنبؤ بتفاعلات الادوية باستخدام الشبكات العصبية الرسومية

أطروحة مقدمة إلى

مجلس كلية تكنولوجيا المعلومات في جامعة بابل والتي هي جزء من متطلبات نيل درجة الدكتوراه فلسفة في تكنولوجيا المعلومات / البرمجيات

من قبل

علي كريم عبدالرحيم احمد

بأشراف

أ.د بان نديم ذنون

١٤٤٥هـ ٢٠٢٤ م