UNIVERSIDADE D
COIMBRA

Nelson Rodrigo Carvalho Monteiro

End-to-End Deep Learning Approach for Drug-target Interaction Prediction

Thesis submitted to the University of Coimbra in compliance with the requirements for the degree of Master in Biomedical Engineering

July, 2019

UNIVERSIDADE D COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Nelson Rodrigo Carvalho Monteiro

End-to-End Deep Learning Approach for Drug-Target Interaction Prediction

Thesis submitted to the
University of Coimbra for the degree of
Master in Biomedical Engineering

Supervisors:
Prof. Dr. Joel P. Arrais
Prof. Dr. Bernardete Ribeiro

Coimbra, 2019

This work was developed in collaboration with:

Center for Informatics and Systems of the University of Coimbra

BSIM Therapeutics

Acknowledgments

Quero começar por agradecer ao Professor Joel P. Arrais por ter sempre acreditado nas minhas capacidades, por toda a responsabilidade depositada e pela liberdade de perseguir, investigar e explorar as minhas ideias. Os desafios colocados, as perguntas levantadas e todos os debates foram imprescindíveis para todo o trabalho desenvolvido e para o meu enriquecimento pessoal e intelectual. Fazer ciência é contribuir para ela, e não posso não reconhecer e não deixar de estar agradecido que me tenha permitido fazer parte dela.

Um agradecimento à Professora Bernardete Ribeiro por todos os comentários construtivos, pelas revisões detalhadas e essenciais ao projeto desenvolvido e pela disponibilidade e ajuda demonstrada.

Agradeço ao Carlos Simões da BSIM Therapeutics por toda a partilha de informação e conhecimento, por todas as questões e visões pertinentes e pela permanente disponibilidade prestada. Todas as conversas e debates foram fundamentais para o progresso do projeto.

Exprimo a minha maior e profunda gratidão à minha família por todo o sacrifício, esforço e por sempre, mesmo sempre, terem feito tudo por mim e para poder hoje estar aqui. Obrigado por acreditarem sempre em mim, por todo o apoio, por toda a força, por nunca me terem deixado desistir e por terem sempre acompanhado o meu percurso como se fosse o vosso. Sem vocês nada seria possível, nada teria feito sentido e espero poder sempre retribuir tudo aquilo que abdicaram.

Aos meus amigos e à minha família de Coimbra um grande e profundo obrigado por serem vocês e por estarem sempre lá. Guardo eternamente todos os momentos, todas as histórias, todas as memórias, todas as conversas, todas as partilhas, todas as noitadas e todos os trabalhos.

Finalmente, um grande obrigado à Junior Enterprise for Science and Technology pela incrível experiência, pelo reconhecimento e pela oportunidade. Obrigado por

todas as discussões, toda a partilha de conhecimento, todos os momentos aleatórios, mas essencialmente por ter feito parte de um grupo tão incrível e único que é a JEST.

Financing

This research has been funded by the Portuguese Research Agency FCT, through D4 - Deep Drug Discovery and Deployment (CENTRO-01-0145-FEDER-029266).

"É o tempo da travessia: e, se não ousarmos fazê-la, teremos ficado, para sempre, à margem de nós mesmos."

Fernando Pessoa

"Without data you are just another person with an opinion."

W. Edwards Deming

"La science n'a pas de patrie, parce que le savoir est le patrimoine de l'humanité, le flambeau qui éclaire le monde."

Louis Pasteur

"Das Leben ist wie ein Fahrrad. Man muß sich vorwärts bewegen, um das Gleichgewicht nicht zu verlieren."

Albert Einstein

Resumo

A descoberta de potenciais Interações Fármaco-Alvo é uma etapa determinante no processo de descoberta e reposicionamento de fármacos, uma vez que a eficácia do tratamento antibiótico disponível está a diminuir, provocado pelo aumento da sua utilização indevida. Apesar dos esforços colocados nos métodos tradicionais *in vivo* ou *in vitro*, o investimento financeiro farmacêutico foi reduzido ao longo dos anos. Desta forma, estabelecer métodos computacionais eficazes, é decisivo para encontrar novos propósitos clínicos para os fármacos disponíveis (*leads*) num tempo considerável.

Abordagens bem sucedidas, incluindo aprendizagem de máquina e profunda, foram apresentadas para resolver e identificar corretamente novos *leads* e DTIs, contudo, raramente são utilizados, em conjunto, dados estruturais e sequências de proteínas. Neste trabalho, propomos um modelo de arquitetura de aprendizagem profunda, que explora a habilidade particular das Redes Neuronais Convolucionais para automaticamente presumir e identificar regiões sequenciais e estruturais, e extrair representações 1D das sequências de proteínas (sequências de aminoácidos) e das SMILES *strings* dos compostos. Estas representações podem ser interpretadas como caraterísticas que expressam dependências locais ou padrões e, que por sua vez, podem ser usadas numa Rede Neural Completamente Conectada, funcionando como um classificador binário.

Os resultados alcançados demonstram que usar CNNs para obter representações dos dados, em vez dos descritores tradicionais, levam a um aumento do desempenho. O método proposto de aprendizagem profunda de ponta a ponta superou os métodos tradicionais de aprendizagem de máquina na classificação correta de interações positivas e negativas, alcançando elevados valores de sensibilidade (0.861) e especificidade (0.961).

Palavras-Chave: Reposicionamento de Fármacos, Interação Fármaco-Alvo, Aprendizagem Profunda, Rede Neuronal Convolucional, Rede Neuronal Completamente

Conectada.

Abstract

The discovery of potential Drug-Target Interactions is a determining step in the drug discovery and repositioning process, as the effectiveness of the currently available antibiotic treatment, arisen from the increased misuse, is declining. Although putting efforts on the traditional *in vivo* or *in vitro* methods, pharmaceutical financial investment has been reduced over the years. Thus, establishing effective computational methods is decisive to find new clinical purposes for the available drugs (leads) in a reasonable amount of time.

Successful approaches, including machine and deep learning, have been presented to solve and correctly identify new leads and DTIs, but seldom protein sequences and structured data are used together. In this work, we propose a deep learning architecture model, which exploits the particular ability of Convolutional Neural Networks to automatically surmise and identify important sequential and structural regions and extract 1D representations from protein sequences (amino acid sequences) and compounds SMILES strings. These representations can be interpreted as features that express local dependencies or patterns that can be used in a Fully Connected Neural Network, acting as a binary classifier.

The achieved results demonstrate that using CNNs to obtain representations of the data, instead of the traditional descriptors, lead to improved performance. The proposed end-to-end deep learning method outperformed traditional machine learning approaches in the correct classification of both positive and negative interactions, reaching high scores of sensitivity (0.861) and specificity (0.961).

Keywords: Drug Repositioning, Drug-Target Interaction, Deep Learning, Convolutional Neural Network, Fully Connected Neural Network.

Contents

List of Tables **xix**

List of Figures **xxi**

Abbreviations **xxiii**

1 Introduction **1**

1.1 Context 1

1.2 Motivation 4

1.3 Objectives 5

1.4 Workflow 5

1.5 Research Contributions 7

1.6 Document Structure 7

2 State of the Art **9**

2.1 Ligand Based 9

2.2 Docking Simulation 11

2.3 Chemogenomic 13

2.3.1 Machine Learning 14

2.3.2 Deep Learning 16

3 Data Preparation **21**

3.1 Processing 21

3.2 Representation 22
3.2.1 Protein Sequence Encoding 23
3.2.2 SMILES String Encoding 24

4 Model **25**
4.1 One-Hot Encoding Layer 25
4.2 Convolutional Neural Network 26
4.3 Fully Connected Neural Network 29
4.4 Model Overview 31
4.5 Hyperparameter Optimization Approach 33

5 Experimental Setup **35**
5.1 Datasets 35
5.1.1 Protein Sequences & SMILES Strings Dataset 35
5.1.2 Coelho et al. (2016) Descriptors Dataset 38
5.1.3 Specific Descriptors Dataset 38
5.2 Main Model 39
5.3 Random Forest 42
5.4 Fully Connected Neural Network 45
5.5 Support Vector Machine 45
5.6 CNN, Autoencoder and FCNN Combined Model 47
5.7 Evaluation Metrics 50

6 Results and Discussion **53**

7 Conclusion **59**

Bibliography **63**

List of Tables

3.1 Yu et al. (2010) [54] protein substitution. 23

3.2 SMILES char-integer dictionary. 24

5.1 Unique drugs, targets and DTIs. 36

5.2 Training and testing datasets after elimination. 37

5.3 Unique targets, drugs and number of targets for the training and testing datasets. 38

5.4 Parameter settings for the proposed model.* Initial number of epochs to allow convergence of the model, however early stopping and model checkpoint were used. 42

5.5 Parameter settings for the RF model. 44

5.6 Parameter settings for the FCNN model using descriptors as input. *Initial number of epochs to allow convergence of the model, however early stopping and model checkpoint were used. 45

5.7 Parameters Setting for the SVM Model. 46

5.8 Parameter settings for the autoencoder model. * Initial number of epochs, however early stopping and model checkpoint were applied. . 49

5.9 Parameter settings for the FCNN related to the combined model. * Initial number of epochs, however early stopping and model checkpoint were applied. 50

6.1 Prediction results of testing set for the deep learning approaches. . . . 53

6.2 Prediction results of testing set for the machine learning approaches. . 53

List of Figures

1.1 Drug discovery versus drug repositioning aligned with computational methods: The use of sequential and structural data combined. 6

2.1 Main computational approaches for DTI prediction. 9

2.2 Docking simulation process. Image from "CABS-dock web server for the flexible docking of peptides to proteins without prior knowledge of the binding site" [26]. 11

2.3 Yamanishi et al. (2008) bi-partite graph learning method. Image from "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces" [35]. 14

2.4 Comparison of a biological neuron with an artificial neuron. Image adapted from "`www.alive.com/health/can-rewire-brains`". 17

2.5 Deep feedforward neural network architecture. Image from "Boosting compound-protein interaction prediction by deep learning" [49]. . . . 18

2.6 Deep belief neural network architecture. Image from "Deep Learning Based Drug Target Interaction Prediction" [51]. 19

3.1 Processing methodology applied for the protein sequences and SMILES strings based on a length threshold derived from the lengths distributions and a chosen percentage of information. 22

3.2 Fraction and frequency based encoding. 22

3.3 Integer based encoding. 22

3.4 Word/Character embedding. 23

4.1 One-Hot encoding applied to myocyte-specific enhancer factor 2B. . . 25

4.2 Convolutional neural network architecture. 26

4.3 Convolution operation. 27

4.4 Global max pooling. 28

4.5 Two parallel CNNs, followed by a global max pooling, are applied to protein sequences and SMILES strings, resulting in deep representations that are concatenated into feature vectors. 29

4.6 Fully Connected Neural Network Architecture. 29

4.7 Dropout technique applied to a portion of a FCNN. 30

4.8 Feature vectors, obtained from the two parallel CNNs model, are used as the input of a FCNN, which is followed by a binary output layer. . 31

4.9 Drug-Target Interaction model architecture. 32

4.10 Hyperparameter optimization model based on grid search. 33

5.1 Distribution of the proteins and SMILES lengths of training and testing datasets. (a) Training protein sequences. (b) Testing protein sequences. (c) Training SMILES. (d) Testing SMILES. 37

5.2 Random Forest. 43

5.3 K-fold cross-validation. 44

5.4 FCNN model using descriptors as input. 45

5.5 Support Vector Machine. Image from "Support vector machines for drug discovery" [83]. 46

5.6 CNN, Autoencoder and FCNN combined model. 48

5.7 Autoencoder architecture. 49

5.8 Confusion matrix. 51

6.1 Confusion matrix of testing set classification for the proposed model. 54

Abbreviations

Adam Adaptive Moment Estimation

ADMET Absorption, Distribution, Metabolism, Excretion and Toxicity

ANN Artificial Neural Network

CNN Convolution Neural Network

CTD Composition, Transition, Distribution

DBN Deep Belief Networks

DNA Deoxyribonucleic Acid

DTI Drug-Target Interaction

FCNN Fully Connected Neural Network

FDA Food and Drug Administration

GPCR G-Protein-Coupled Receptors

Kd Constant of Dissociation

MACCS Molecular Access System

MSE Mean Squared Error

PDB Protein Data Bank

QSAR Quantitative Structure-Activity Relationship

RBF Radial Basis Function

ReLU Rectified Linear Unit

RF Random Forest

SMILES Simplified Molecular Input Line Entry System

SGD Stochastic Gradient Descent

SVM Support Vector Machine

1

Introduction

1.1 Context

The Role of Proteins

Human and other organisms, e.g., bacterias, proteome is constituted by a considerable amount of different proteins [1]. Proteins are identified as key working molecules that reside in every organism cells and are responsible for unique functions. Each protein consists of one or more polypetide chains, which are described as linear chains of amino acids connected by peptide bonds in a specific order, that fold into complex 3D structures [2]. The protein's biological function is determined by the unique amino acid sequence, as each amino acid has a particular chemical behavior and each polypeptide a specific amino acid order, and also by the interactions that occur within the chains, which are responsible for the folding and therefore the structure.

The primary structure of every protein, defined as the amino acid sequence, is determined by the DNA, specifically the gene, that encodes that protein. Hence, any mutation in the DNA sequence might affect the protein's structure and function. Additionally, proteins are also affected by chemical, biological and environmental factors, which might lead to the loss of shape or functionality [3].

Several unique functions are carried out by proteins, including structure, signaling, maintenance, substance transport, protection, storage and biochemical reactions. Thus, they are essential to maintain the biological, chemical and physiological balance within every organism.

Most proteins fulfill their role by interacting with other proteins or molecules, where their activity depends on the type of binding that occurs. Thus, a protein's function rate can easily be altered based on the interaction with potential "invasive" ligands that dominate over the natural ligands at the binding regions, leading to the rise or

decline of its natural function.

The Importance of Drugs

Drugs play an important role in the modern medicine, as many diseases, illnesses and infections are prevented, controlled and treated using these compounds. Although natural products (raw drugs) have been used since ancient times [4], most of the actual drugs are derived from combinatorial chemistry, chemical synthesis or related to substances produced by microorganisms.

These chemical compounds have been accountable for the majority of the humanity survival, as they are responsible for enhancing certain substances, e.g., proteins, and their activity (acting as agonists) or, on the contrary, having antagonistic effects, affecting even a bacteria's physiology and biochemistry, resulting in their cell death and cessation of growth [5].

Antibiotics, which target bacterias, have revolutionized human development and survival, enabling the human race to survive and prevent several bacterial infections. Additionally, they are also responsible for major advances in medicine, surgery, transplants and chemotherapy [6]. These type of drugs, starting with the Penicillin, discovered by Sir Alexander Fleming in 1928 [7] and used to treat bacterial infections during World War II, were the pinnacle for the modern medicine development.

The structure of a drug is usually divided into two main parts, specifically the essential part, which is involved in the drug-target interaction and therefore responsible for the biological/pharmacological response of the drug, and non-essential, allowing the transport of the drug and the interaction with secondary receptors, without changing drastically the drug's biological activity [8].

Drugs predominantly target proteins, including enzymes, receptors or transporters. The accurate identification of targets is essential for the pharmacological action of a drug.

The drug discovery process is usually associated with six different steps [9], including target discovery, lead discovery, lead optimization, pre-clinical, clinical trials and regulatory approval. Target and lead discovery are related to identifying which ligand binds to a certain drug or which drug binds to a certain ligand, respectively. On the other hand, lead optimization is associated with using lead compounds, known as potential drugs for new clinical purposes, and perform chemical modifications to improve potency, selectivity or pharmacokinetic features. Pre-Clinical, known as ADMET, is defined as a number of conditions that a certain drug must met to

be viable for human consumption. Clinical trials is one of the final stages and are divided into several steps, to evaluate all the effects of the drug and validate the effectiveness and the viability for consumption. These clinical trials usually start with animals and end in human trials. Finally, the drug needs to be registered and approved by a drug administration department, e.g., FDA in the USA.

Drug-Protein Interactions

The pharmacological action of a drug is the result of the intrinsic properties as well as the interaction with the complementary chemical groups of a specific cellular component, designed as receptor, which initiates several biological and physiological modifications, altering the function rate of that receptor.

The interaction between a protein and a drug, which can be reversible or irreversible, is the consequence of several bond types, including ionic, hydrogen, hydrophobic, Van der Waals and covalent [10]. The binding is usually divided into primary binding, responsible for a firm binding (ionic interaction), and secondary binding, which is a supplementary binding to hold the drug. Even though the complementary of the chemical groups is essential to form bonds, there are several factors that affect the interaction, including physical, chemical and physiological. Additionally, not all regions of the protein are responsible to form bonds, only specific regions, denominated of active or binding sites, interact with the drug [11].

Drugs are potential modulators of the functions performed by several proteins, therefore their ability to bind, defined as affinity, is essential for the interaction. However, the capacity to execute their pharmacological activity, identified as intrinsic activity, is determined by the recognition of certain functional groups in the 3D space, including their electron densities, by the receptor. These two factors, affinity and intrinsic activity, determine the role enforced by the drug, which can be an agonist, antagonist or partial agonist.

In order to reach the site of action, where the pharmacological task is achieved, drug molecules need to cross natural barriers and circulate in the blood stream [12]. Thus, there are several secondary interactions that are responsible and necessary for the transportation, storage, metabolism and excretion of the drug molecules. These interactions are essential to regulate the effects of the drug molecule in the body.

1.2 Motivation

Multi-drug resistant bacterias are a rising health concern to the overall population and pharmaceutical industry as more and more drugs are becoming ineffective and unresponsive to the symptoms and diseases associated with these kind of infections, leading to a situation where some infections have no cure [13, 14]. Modern medicine is aligned with antibiotic treatment, however the discovery of new, potential and effective drugs is declining, as there is an irrational and injudicious misuse of the current available medicine, causing a resistance effect to these kinds of agents [15], as well as a free path for the bacterias to evolve and start resisting these compounds.

The pharmaceutical financial investment has been reduced over the years, making it difficult for researchers to keep up with the current population and pharmaceutical needs [16]. The amount of new drugs discovered every year is declining, conversely to number of new variants of the already existing infections and diseases. Traditional *de novo* drug discovery is very time consuming, as it may take 10 to 17 years from concept to marketed drug [17], expensive, in the realm of thousands of millions, and it is associated with a low probability of success, as there is a considerable number of conditions to be met in order to be viable for human consumption. Aligning drug repositioning, that is, finding new clinical purposes for existing drugs, with computational methods [18] is crucial and decisive to find potential drug-target interactions and new leads (hit compounds) in a reasonable amount of time. Besides, drug repositioning allows to ignore some of the steps of the traditional *de novo* drug discovery, as most of the drug candidates have already been through the validation phases.

The process of developing new drugs has the common basis of using a compound to produce some kind of pharmacological response on a certain target, however, that drug must interact with several secondary targets before reaching its site of action. Taking into consideration the intrinsic characteristics of drug-target interactions, the fact that a certain protein target binds several drugs and a certain drug binds several protein targets and also the amount of unintended side effects that a drug may produce [19], it is possible to affirm that there are several new possibilities for a given drug. Thus, finding new purposes, new targets and all the side effects is important in the discovery of new leads.

Despite all the efforts and the existing successful approaches to find new DTIs and leads, protein sequences and structural data are rarely used together. Although there are several factors that affect the drug-target interaction, the protein sequence,

specifically the binding regions of the protein, and the chemical structure of the compound, which is divided essentially into two parts, one related to main interaction and the other to the secondary interactions, are determinant for the interaction. Thus, combining computational methods with sequential and structural information may lead to new decisive findings.

1.3 Objectives

The main goal of this master thesis is to develop an end-to-end deep learning approach capable of predicting DTIs using 1D raw data, protein amino acid sequences and SMILES strings, which represent the drug's chemical structure. Conventional physicochemical and/or structural descriptors are general descriptors of the whole sequence or chemical structure and therefore non relevant, in most cases, to a possible real interaction. Besides, the amount of available known 3D structures is limited or highly complex, which makes them impractical to use. Therefore, there are six objectives to fulfill:

1. Explore a pipeline to process and represent the 1D sequential and structural data.
2. Exploit the particular ability of CNNs to uncover deep patterns (representations or local dependencies) from raw data.
3. Build a deep learning architecture model based on two CNNs and a FCNN.
4. Evaluate the proposed model and compare it with machine and deep learning approaches.
5. Compare the differences of using deep representation over traditional and conventional descriptors of the proteins and compounds.
6. Evaluate the influence of specific descriptors in the prediction of DTIs.

1.4 Workflow

The rising antibiotic resistance and reduced financial investment in the traditional *in vitro* and *in vivo* drug discovery methodologies led to the pursuing of drug repositioning approaches. This works focus on aligning drug repositioning with computational methods to increase the reward-time trade-off and diminish the dependence of using traditional laboratory experimental methods to identify potential drug-target interactions and discover new leads. Besides, drug repositioning allows to skip some steps

of the traditional drug discovery process as most of the drugs used are already validated, reducing exponentially the time needed. Plus, a deep learning architecture is applied to automatically identify meaningful hidden and complex patterns and relationships on 1D sequential and structural data to the prediction of DTIs. On that account, it allows to learn and identify potential leads and DTIs using intrinsic information of the proteins and drugs, requiring less time and money, as it does not need to verify each interaction experimentally, and also surpassing the traditional approaches on its capacity to spot potential complex relationships.

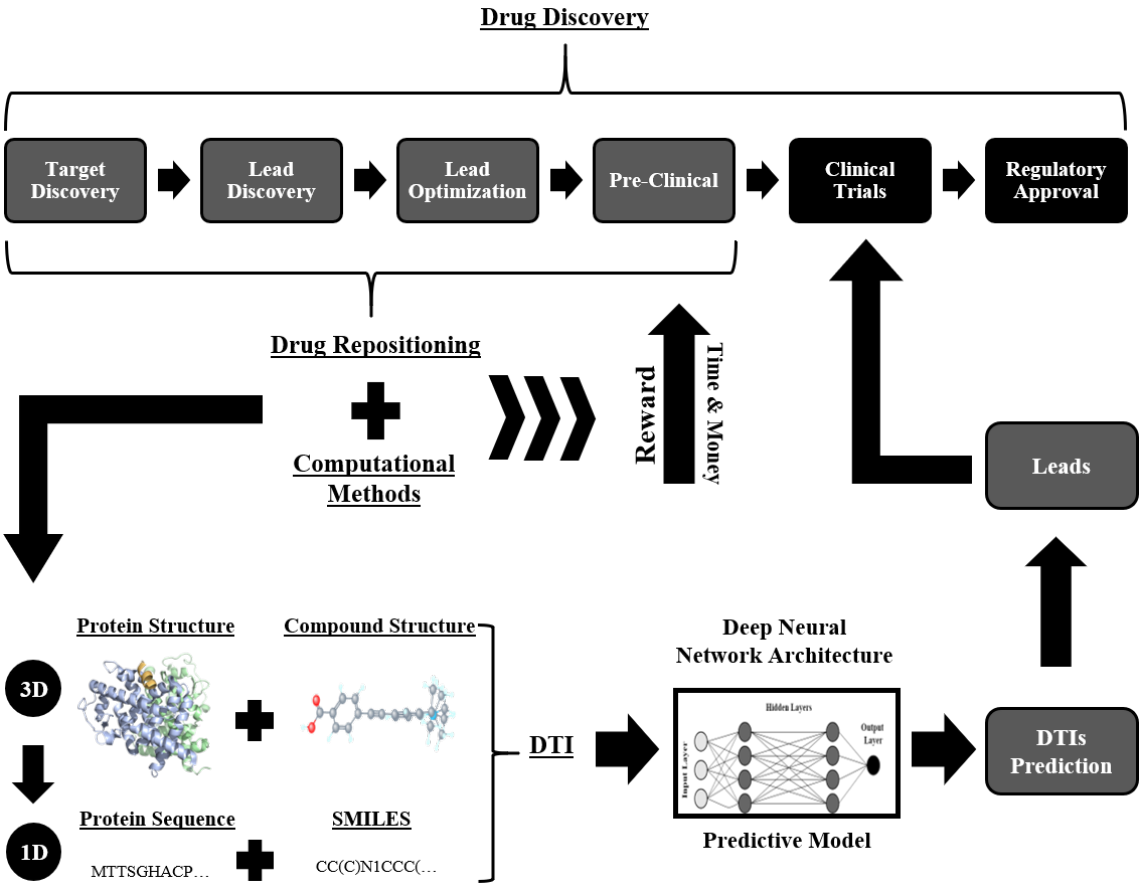

Figure 1.1: Drug discovery versus drug repositioning aligned with computational methods: The use of sequential and structural data combined.

1.5 Research Contributions

The work developed during this thesis resulted in the following contributions:

Papers

1. Nelson R. C. Monteiro, Bernardete Ribeiro, Joel P. Arrais. "Drug-Target Interaction Prediction: End-to-End Deep Learning Approach". IEEE/ACM Transactions on Computational Biology and Bioinformatics, IF: 2.428 (Submitted on 15th April 2019 as a full paper and under revision)
2. Nelson R. C. Monteiro, Bernardete Ribeiro, Joel P. Arrais. "Deep Neural Network Architecture for Drug-Target Interaction Prediction". ICANN2019, 28th International Conference on Artificial Neural Networks, CORE rank: B. (Submitted on 8th April 2019 and accepted as a short paper)

Posters

1. Nelson R. C. Monteiro, Bernardete Ribeiro, Joel P. Arrais. "Drug-Target Interaction Prediction: End-to-End Deep Learning Approach". EJIBCE2018, Structural Computational Biology Meeting of Junior Researchers (Poster Presentation on December 2018).
2. Nelson R. C. Monteiro, Bernardete Ribeiro, Joel P. Arrais. "Drug-Target Interaction Prediction: End-to-End Deep Learning Approach". BOD2019, Bioinformatics Open Days (Poster Presentation on February 2019).
3. Nelson R. C. Monteiro, Bernardete Ribeiro, Joel P. Arrais. "End-to-End Deep Learning Approach for Drug-Target Interaction Prediction". Ciência2019, Science and Technology Summit (Poster Presentation on July 2019).

1.6 Document Structure

The rest of this document is organized into 6 different chapters. The Chapter 2, State of the Art, provides a showdown of the principal computational approaches used in the drug-target interaction area, presenting several research works related to each one of them, as well as an explanation of the reasons behind them. The Chapter 3, Data Preparation, describes the used pipeline to process and encode the data. The Chapter 4, Model, presents the deep neural network architectures used to build the proposed model and the hyperparameter optimization approach applied.

The Chapter 5, Experimental Setup, describes the different stages of the experimental setup, including the construction of the datasets and the different models used to compare the performance of the proposed setup. The Chapter 6, Results and Discussion, shows the results obtained and the discussion of the whole process, including the advantages and disadvantages. Finally, the Chapter 7, Conclusion, concludes the master thesis and presents future approaches and possibilities for the proposed work.

2

State of the Art

There are many factors involved in the interaction between drugs and targets, including external elements that modulate and regulate the interaction. Thus, several successful approaches, based on different perspectives, have been presented and explored to solve the problem of identifying new DTIs. Computational methods for DTI prediction are divided into three main approaches [20]: ligand based, docking simulation and chemogenomic.

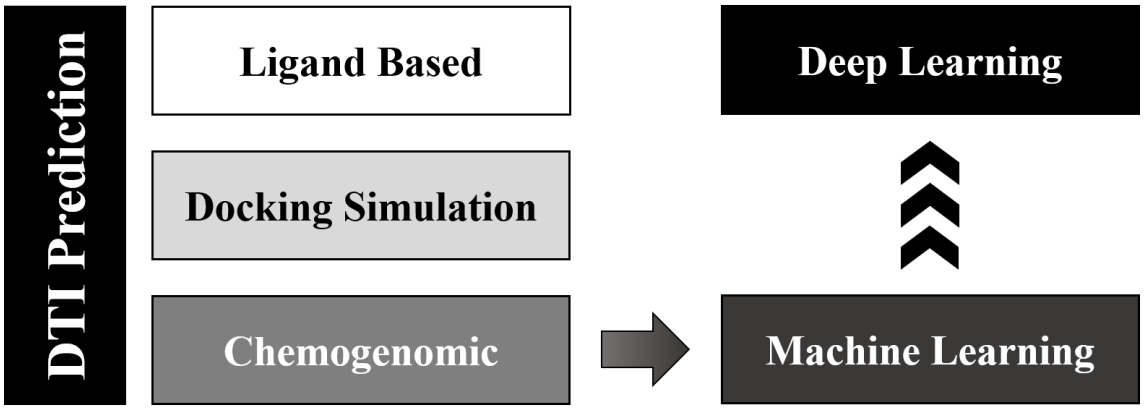

Figure 2.1: Main computational approaches for DTI prediction.

2.1 Ligand Based

Ligand based approaches are built upon the concept that similar molecules have similar properties among them and therefore should bind and interact with the same group of proteins. These kind of approaches are used to make predictions on possible potential interactions by comparing a new ligand with known proteins ligands [21]. Therefore, they are limited to the amount of known and available ligands, performing insufficiently when this number is scarce or there is no knowledge available about known interactions.

Most of the ligand based approaches use Quantitative Structure-Activity Relationships, which are essentially statistical based methods used to correlate the chemical structures with the biological activity. QSAR models have the fundamental basis that variations in the biological activity associated with a group of ligands are related to variations in their structural, physical and chemical properties, hence similar ligands should interact with similar proteins [22]. The choice of relevant descriptors, which can be related to molecular (2D QSAR) or 3D structural geometric (3D QSAR) properties, capable of encoding important structural information associated with the biological activity is fundamental for a good performance, as they are the basis of association. Additionally, the statistical method used to correlate the chemical structure with the biological activity, which can be a linear or non linear relationship, needs to be convenient and fitting for the problem.

Keiser et al. (2008) [21] developed a method, Similarity Ensemble Approach (SEA), where receptors (proteins) were quantitatively related based on the chemical similarity among their ligands. The similarity was calculated using ligand topology, expressed as a Tanimoto Coefficient, which is considered as a distance measure between two points, and ranked statistically. This approach enabled the discovery of new and unexpected associations, as well as, the discovery of potential related proteins.

Humberto et al. (2011) [23] proposed a Multi-target QSAR Web Server to make large scale predictions, derived from chemical structures and 3D structures of target proteins. This approach is combined with a Markov Chain Model, MARCHINSIDE, to calculate structural parameters of drugs, based on different physicochemical molecular properties, and proteins, derived from 3D potentials, e.g., average value of electrostatic potential, for different types of interactions. Linear Discriminant Analysis (LDA) was used to select the best model.

Cheng et al. (2012) [24] established multi-target quantitative structure–activity relationships, mt-QSAR, and chemogenomics methods based on substructure patterns (MACCS Keys) and protein sequences descriptors to predict chemical–protein interactions. The mt-QSAR method was decomposed as a multiple binary classification problem using an SVM model to classify the associations. The multiple binary models were combined in the end to make DTI predictions.

2.2 Docking Simulation

Docking simulation approaches are used to predict the 3D structure of the receptor-ligand complex, based on the 3D structures of the receptor and the ligand [25]. These kind of approaches are adopted essentially for structure based drug design, where the interaction is simulated and scored, according essentially to the intermolecular interaction energy. This process can be seen on figure 2.2.

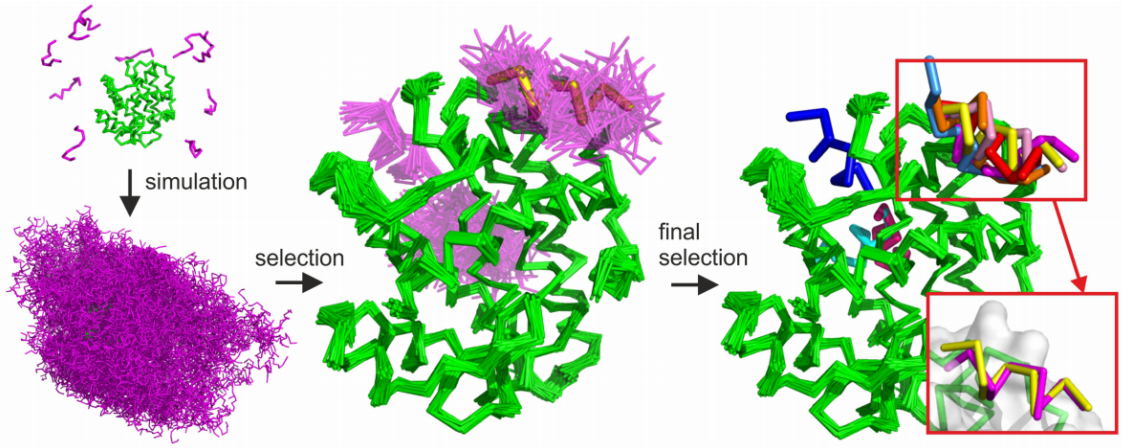

Figure 2.2: Docking simulation process. Image from "CABS-dock web server for the flexible docking of peptides to proteins without prior knowledge of the binding site" [26].

Most of the docking algorithms uses the receptor and ligand's coordinates to predict the coordinates of the resulting complex, based on given potential ligand-binding locations or, in the case there is not any knowledge available of potential binding locations or any 3D structures of a complex of the receptor, blindly docking the ligands onto the receptor structure [27]. To predict the coordinates of the resulting complex, there are different degrees of molecular flexibility that can be considered, specifically both rigid, receptor rigid and ligand flexible or both flexible [28]. Proteins are in constant motion between different conformation states with similar energies and change their conformation to promote the interaction, therefore the receptor's flexibility needs to be considered in most cases, which is a limitation in most docking simulation approaches due to all the different possible conformations.

The scoring function used in a docking simulation is essential to correctly predict the 3D structure of the resulting receptor-ligand complex, as it is used to evaluate each search result. Given the huge computational cost to use a very accurate scoring function, most of scoring functions apply assumptions and simplifications. These functions are essentially based on the intermolecular interaction energy and on the

individual contributions of the receptor and the ligand [29].

The use of 3D structures is a realistic approach to model the interaction between proteins and drugs, yet, the lack of information, the complexity of 3D structures and the amount of time it takes to simulate, makes this kind of approaches inapplicable, inefficient and unreliable in most cases. Besides, when the receptor's 3D structure is not available and there are proteins with similar sequence and structure known, it is possible to use homology modeling [30] to predict the 3D structure. However, most of the resulting structures are unreliable due to the fact that the resulting folding of two proteins with similar sequences might be completely different.

Li et al. (2006) [31] developed an useful tool for target identification, Target Fishing Dock (TarFisDock), which combines a database of potential drug targets with a reverse ligand-protein docking approach to seek and identify possible protein targets for a given small molecule. TarFisDock generates a protein target list, docks a small molecule into the possible binding sites of the proteins and calculates the interaction energy, based on Van der Waals and electrostatic interaction terms, to score the resulting complex. The database contains proteins, with known 3D structures, that are identified as targets in different therapeutic areas. This approach only considers the ligand's flexibility, not taking into account the receptor's flexibility. Additionally, it was able to correctly identify targets for vitamin E and 4H-tamoxifen.

Cheng et al. (2007) [32] designed a binding free energy model combined with parameters for drug like properties to predict the maximal affinity by a drug-like molecule (drugability) using the crystal structure and physiochemical properties of the target binding site. Computational geometry methods were used to represent the binding site and calculate the necessary parameters of curvature and surface area from 3D crystal structures. The affinity was calculated based on molecular driving forces for binding.

Yang et al. (2011) [33] established a docking based method, Antithesis Chemical-Protein Interactome, to mimic the differences in the drug-protein interactions across a set of human proteins. The docking method gives a score array containing information about the binding conformation and the binding strength. The docking scores were normalized by drug and protein, resulting in a z-score capable of representing essentially chemical and chemical-protein interactive effects. This approach was able to identify an important biomarker, HSPA1A, an off-target of clozapine.

2.3 Chemogenomic

The growth of available biological and chemical data useful for prediction resulted in a higher usage of chemogenomic methods over the traditional methods. Chemogenomic approaches are based on the chemical space of compounds, genomic space of target proteins and/or the pharmacological space (interactions between proteins and drugs) to predict new potential interactions [34]. Although these kind of approaches can be divided into four major classes, including graph-based, network-based, machine learning and deep learning, the last two are mainly pursued due to the their improved performance, to their capacity to fully use all the available information and data to learn and discover new relevant interactions and also to the achievable time-reward trade-off.

Yamanishi et al. (2008) [35] proposed a supervised method to infer DTIs, related to four classes of important drug–target interactions in human involving enzymes, ion channels, GPCRs and nuclear receptors, by integrating the chemical space and genomic space into an unified space defined as the pharmacological space. The chemical space is represented by a similarity matrix based on the similarity score between chemical structures, the genomic space by a similarity matrix based on Smith-Waterman's normalized scores, which gives information about the similarity between two protein sequences, and the pharmacological space by a bi-partite graph projected into an Euclidean space, which represents the interactions between proteins and chemicals. The proposed work uses a bi-partite graph learning method to learn the correlation (similarity or closeness) between the chemical/genomic space and the interaction space to infer new possible interactions (high scoring compound-protein pairs). Although this method was not validated experimentally, the major four datasets used in this work are still the base of many DTI studies. The bi-partite graph learning method can be seen on figure 2.3.

Cheng et al. (2012) [36] proposed a network based inference (NBI) approach using FDA approved drug-target binary links to infer new predictions. This method only uses known drug-target bipartite network topology similarity to calculate predictive scores for each drug and unlinked target. Unlike the work of Yamanishi et al. (2008) [35], this approach makes new predictions solely based on the network topology similarity, discarding the genome and chemical space similarity and therefore not relying on any structural (3D) or sequential information. Besides, it only uses known DTI information, thus for each new drug without any target information, this approach can not make any target prediction. Some of the predictions were

validated experimentally by *in vitro* assays, validating the prediction capacity of this approach.

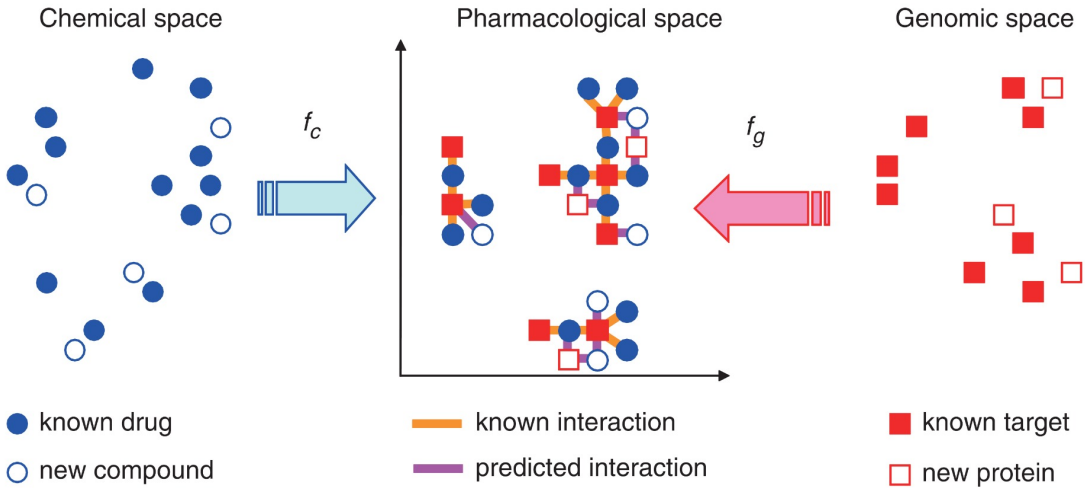

Figure 2.3: Yamanishi et al. (2008) bi-partite graph learning method. Image from "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces" [35].

2.3.1 Machine Learning

The ability to make computers automatically learn how to perform particular tasks based on a given sample of information led to the rise of machine learning methods. These can be divided into 3 major groups, specifically, unsupervised, semi-supervised and supervised [37]. Unsupervised methods are usually related to clustering and associations problems, where the only information available is the input and there is no knowledge about the corresponding output and therefore the goal is to learn the intrinsic properties and relationships within the data, in order to organize it into groups or to define rules. Supervised methods are used to predict the output based on the input data, where the corresponding output is known for all the data. Semi-supervised are a mix of these two, where unsupervised and supervised methods are both applied and only a portion of the data has a known corresponding output.

In the problem context, most of the machine learning approaches pursued are supervised or semi-supervised due to the considerable amount of available data, their ability to learn relationships and patterns among the data related to proteins and drugs and also because they can make new predictions based on the already known and validated interactions. Each receptor and ligand is characterized by a set of attributes (features), that combined describe a particular interaction. The machine learning model can learn from these features and a target vector, which identifies

the type of interaction (classification) or, on the contrary, gives information about a specific continuous or discrete binding metric value (regression).

Given the bountiful variety of information available, it is possible to extract many features related to receptors and ligands. Even though the amount of information is a plus for machine learning approaches, most of these features are usually redundant or not discriminating for a predictive model. Besides, if the number of features is too great for the amount of samples, the performance of the model is usually inferior in new data due to the curse of dimensionality. Therefore, nearly all machine learning models require some kind of preprocessing over the data, which is usually characterized as an effective search to assess and extract significant features. There are many methods to evaluate the quality of the data and they are usually divided into filters, wrappers or embedded, depending if they are independent or dependent of the classifier or integrated, respectively [38].

Cobanoglu et al. (2013) [39] presented a method using probabilistic matrix factorization (PMF) combined with active learning, without reliance on chemical/target similarity or external data. This approach decomposes the connectivity matrix, related to the DTI network, as a product of two matrices of latent variables that express each drug/target, which objective is to determine the missing interactions that are likely to exist. The active learning strategy used maximizes the discovery of unknown predictions by updating the model based on new unknown predictions discovered.

There are several machine learning approaches used in the prediction of DTIs, yet RF and SVM are the most popular methods due to the performance achieved in several studies. RF is an ensemble learning method that generates a chosen number of decision trees and returns the class that is the mode of the classes across the output of each individual decision tree [40]. SVM defines a hyperplane that maximizes the separation margin between different classes and in the case of non linear separable problems, it usually uses kernel tricks to map the data into high dimensional spaces where it is possible to classify with linear decision surfaces [41].

Nagamine et al. (2007) [42] used SVM as the predictive model to infer new interactions. Instead of using the conventional chemical and genomic descriptors, protein sequences, chemical structures and mass spectrometry, which generates information about the structure and physico-chemical properties, were encoded into numerical values, based on the existence or frequency, and concatenated into features vectors. Plus, this work evidenced the advantage of integrating mass spectrometry information.

Yamanishi et al. (2009) [43] proposed a supervised prediction method using bipartite local models, one based on chemical structure similarity and another one based on sequence similarity between proteins. The prediction is done using two SVMs to predict target proteins and drugs for a given drug or protein, respectively. The results are combined to give a definitive prediction for each interaction.

Yu et al. (2012) [44] proposed a machine learning method, using RF and SVM as the predictive models, to infer new interactions. In this work, chemical and protein descriptors were combined to create the feature vectors. Additionally, the model was validated using four different datasets associated with targets with pharmacology relevance, specifically involving human enzymes, ion channels, GPCRs and nuclear receptors.

Cao et al. (2014) [45] combined chemical data, MACCS fingerprints and/or substructure fingerprints, biological data, protein descriptors, and network properties, presence or absence of association, into feature vectors to be used in a predictive RF model, to identify new DTIs. Four independent datasets related to interactions associated with human enzymes, ion channels, GPCRs and nuclear receptors were used to evaluate the performance. Additionally, this work demonstrated the usefulness of using network topology data to predict DTIs.

2.3.2 Deep Learning

Biological learning systems are based on complex networks of interconnected neurons, which concedes the ability to transfer information from several locations to the place of action, assimilating knowledge and performing actions. The information flows from one neuron to another across a synapse, which depends on action potentials and chemical neurotransmitters. Neural computation is inspired by neurons and their adaptive connections. An ANN is a computational model based on the architecture of the brain, composed by several artificial neurons, which are identified as processing elements. Neurons are interlinked with other neurons in multiple layers, defined as hidden layers [46]. An ANN architecture is usually composed by an input layer, multiple hidden layers and an output layer. The input layer is associated with the independent values that will be fed to the neurons that constitute the hidden layers. The hidden layers are the middle layers between the input and output, responsible to transform the input value into something capable of being used by the output layer, optimizing the data according to the expected result. The output layer is the result of the weighted sum of all the outputs given by the previous layer, to which is applied an activation function. Therefore, each artificial neuron

(Figure 2.4) is organized into 5 building elements:

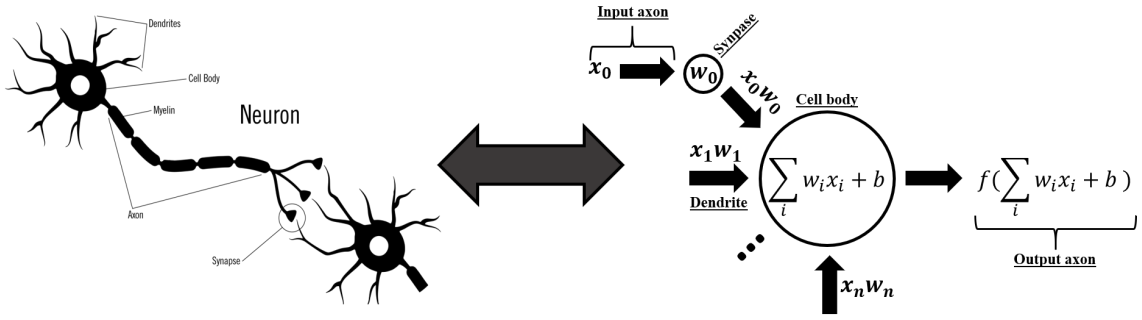

Figure 2.4: Comparison of a biological neuron with an artificial neuron. Image adapted from "`www.alive.com/health/can-rewire-brains`".

1. **Input**: independent value that is fed to the neuron.
2. **Weight**: determines the influence/importance of that input/neuron.
3. **Bias**: "extra neuron" that moves the activation function to grant a better representation of the data.
4. **Activation Function**: responsible for the activation or not of that neuron, by applying a limit/transformation to the resulting value of the weighted sum.
5. **Output**: result of the weighted sum of all the outputs given by the previous connected neurons, to which is applied the activation function. The output of the i^{th} neuron can be given by:

$$f(a_{\mathrm{i}}) = f(\sum_{j} W_{\mathrm{ij}} X_{\mathrm{j}} + b_{\mathrm{i}}) \tag{2.1}$$

, where W is the weight, X the input value and b the bias.

Additionally, neural networks, during the training phase, compare the output with the expected result in order to update the network weights and reduce the error between these two. This process, done in each iteration, is defined as backpropagation and depends on the optimizer, which defines the type of update, and loss function, that measures the inconsistency between predicted and real values. The choice of the optimizer and loss function is usually related to the type of data and problem context, e.g., prediction or regression.

Deep learning can be defined as neural network architectures that have several hidden layers, usually three or more [47]. Even thought deep learning might be a subset of machine learning, it is considered as the evolution of machine learning and there-

fore standing on its own ground. These type of algorithms are capable of exploiting the unknown structures in the data and discover hidden patterns that are capable of representing high level features in terms of lower level features. Besides, conversely to machine learning, they automatically discover the features that are going to be considered as discriminating and important, without the need of any kind of feature engineering. Deep learning methods grow potentially with the amount of data given to train, capable of further boosting the performance as more and more data is fed to the network. They are considered as the state of the art in several areas of interest, including image recognition and natural language processing [48].

Traditional machine learning approaches usually result in good performance, although with the increased computational power and the vast amount of available data, deep learning approaches are being used more often in DTI prediction, resulting in even higher performance in most cases, due to their ability to identify hidden and complex patterns (representations) of the data without using any kind of feature engineering.

Tian et al. (2016) [49] proposed a deep neural network approach, based on a feedforward architecture, DL-CPI, to predict compound-protein interactions, where chemical fingerprints and protein domains, which are binary vectors where "1" and "0" indicate the presence or the absence of certain features, respectively, were used as features. A deep feedforward architecture is constituted by several hidden layers where the information flows in one direction, from the input layer, going through the hidden layers, to the output layer. The proposed approach can be seen on Figure 2.5.

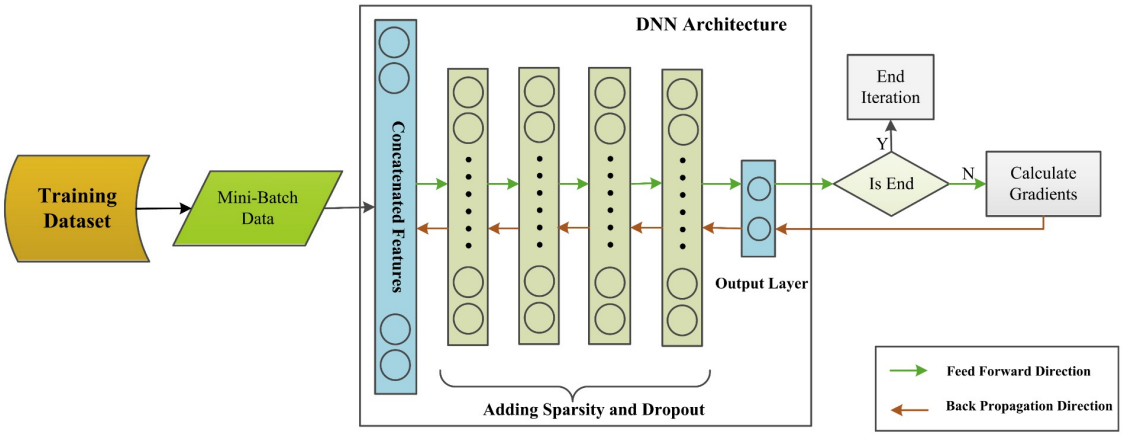

Figure 2.5: Deep feedforward neural network architecture. Image from "Boosting compound-protein interaction prediction by deep learning" [49].

Peng Wei et al. (2016) [50] developed an approach known as multi-scale features

deep representations inferring interactions (MFDR), where a certain type of deep neural network architecture, autoenconder, is used to extract low dimensional representations from chemical structure and protein sequence descriptors to be used as features in an SVM model. Autoencoders use an unsupervised learning process to compress and uncompress data into something that closely matches the original data. The main purpose of this architecture is to extract a smaller set of features that are able to represent the input data, performing reduction of the dimensionality.

Wen et al. (2017) [51] proposed a deep learning method, DeepDTIs, based on DBN. This type of neural network architecture is made by stacking restricted Boltzmann machines (RBMs), which is a graphical model that can learn a probability distribution from input data. Therefore, DBN is a graphical model that learns how to extract a deep hierarchical feature by modeling the distribution between the training sample and the hidden layers. The features used were extracted from chemical substructures and sequence order information (descriptors). The deep belief network architecture can be seen on Figure 2.6.

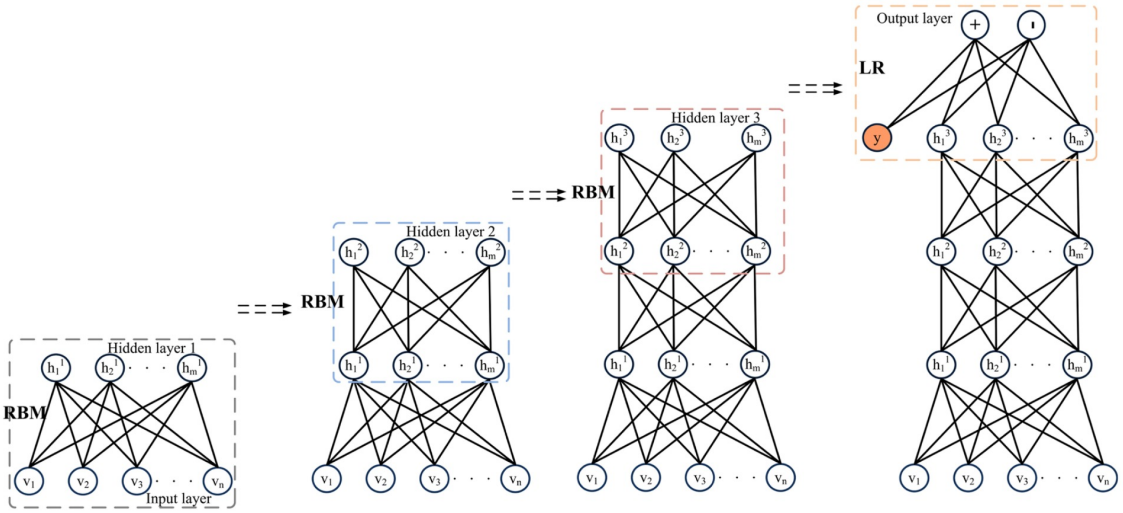

Figure 2.6: Deep belief neural network architecture. Image from "Deep Learning Based Drug Target Interaction Prediction" [51].

Xie et al. (2018) [52] used transcriptome data, z-score of genome wide gene expressions, in a deep feedforward neural network to predict new drug-target interactions. Besides, the approach had a built-in ability to adapt parameters according to the changes of the surrounding environment. The results achieved, proved the effectiveness of using transcriptome data in the prediction of DTIs.

3

Data Preparation

3.1 Processing

Conversely to several studies in Section 2.3, where a DTI pair is represented by a set of global and conventional descriptors related to different properties, we use protein sequences and SMILES strings, which represent the chemical structure of compounds, directly. Therefore, each amino acid of the sequence and each character of the SMILE string is considered as a feature.

Proteins are constituted by an unique amino acid sequence, hence different proteins have different sequence's lengths. Identically to the proteins, each SMILES string is composed by a unique set of characters that represent the chemical structure. Thus, it was necessary to define a threshold for the length of the proteins sequences and SMILES strings in order to guarantee that each protein and drug is characterized by the same amount and "type" (order) of features, respectively.

In order to define the threshold, the distributions of the lengths of the proteins and SMILES, respectively, of the datset are evaluated and an information threshold based on a certain percentage, e.g., 95 %, is applied. Every protein and SMILES string with a length superior or inferior to the threshold are removed or padded, respectively. Although there were two possible methods to evade the elimination of entries, one based on adding zeros to the maximum length of the distribution (padding) and the other one based on identifying the binding regions and removing only the non essential regions, it would lead to a lot of training noise or, in the case of the second method, most of the times the locations of the binding regions are not known for that specific interaction.

The processing methodology applied for the protein sequences and SMILES strings is illustrated on Figure 3.1.

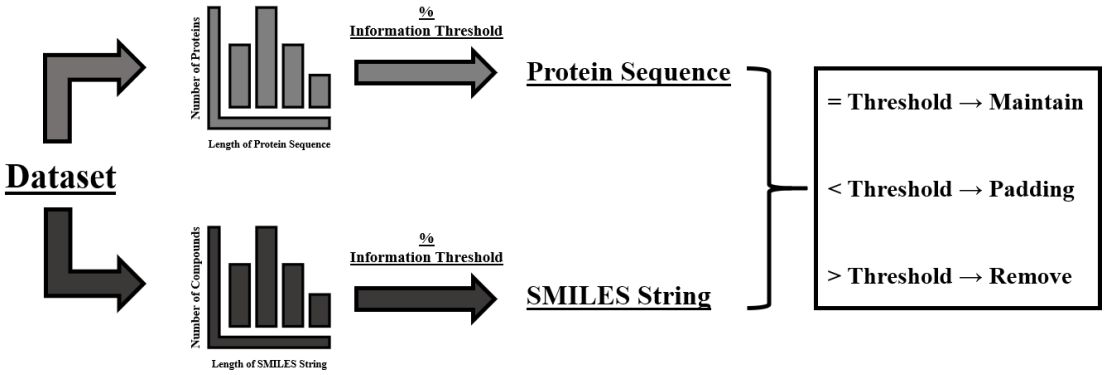

Figure 3.1: Processing methodology applied for the protein sequences and SMILES strings based on a length threshold derived from the lengths distributions and a chosen percentage of information.

3.2 Representation

Due to the fact that we are using protein amino acid sequences and SMILES strings directly, it was necessary to perform some kind of encoding to transform each sequential or structural character, respectively, into a numerical value, capable of being used by the model. There are many types of encoding, all with the objective of transforming the initial input, usually categorical, into something useful and usable by the model.

Fraction or frequency based encoding (Figure 3.2) are associated with representing each character as the fraction or frequency, respectively, of each character type.

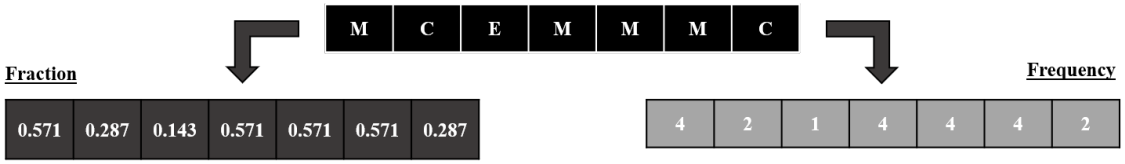

Figure 3.2: Fraction and frequency based encoding.

Integer based encoding (Figure 3.3) is a simpler kind of encoding that transforms each character into an integer, based on the number of different characters.

Figure 3.3: Integer based encoding.

Relationships based encoding (Figure 3.4), known as word embedding or embedding, transforms each character into a continuous numeric vector, mapping semantic

meaning into a geometric space. A numeric vector is associated with each character, where the distance between two vectors are capable of capturing the semantic relationship between the two characters associated. This type of encoding allows to find similarity between objects and represent the dependency of one character on the other characters [53].

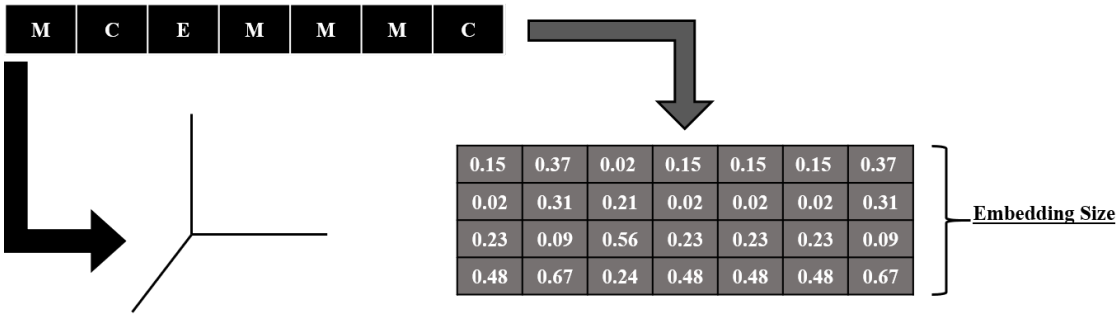

Figure 3.4: Word/Character embedding.

There is no golden rule to choose which encoding to use, however given the problem context, it is important to maintain the raw sequential and structural information (type or character and order) and also to establish a good representation-dimensionality trade-off.

3.2.1 Protein Sequence Encoding

We used Yu et al. (2010) [54] protein substitution table (Table 3.1), which organizes amino acids into 7 groups according to their physicochemical properties. Each amino acid was encoded into an integer based on the corresponding group. This representation allows to directly use protein amino acid sequences, preserve the sequential information (type and order) and also to reduce the amount of categories from 20, associated with the number of possible amino acids, to 7.

Table 3.1: Yu et al. (2010) [54] protein substitution.

Groups	Amino Acids
1	Ala, Gly, Val
2	Ile, Leu, Phe, Pro
3	Tyr, Met, Thr, Ser
4	His, Asn, Gln, Trp
5	Arg, Lys
6	Asp, Glu
7	Cys

3.2.2 SMILES String Encoding

A simple integer encoding, based on the number of different characters, was used to transform each character of the SMILES strings into a integer. A dictionary containing 32 categories (number of different characters) was established (Table 3.2). This representation preserves the structural information, character and order, and has a low computational cost given the amount of different characters.

Table 3.2: SMILES char-integer dictionary.

Integer	Character
1	I
...	...
7	[
...	...
25	P
...	...
32	g

4

Model

4.1 One-Hot Encoding Layer

Each amino acid and character of the proteins sequences and SMILES strings, respectively, were encoded into integers according to the corresponding type of encoding. However, these integers are recognized as categorical variables, representing the amino acid group or type of character, therefore it was necessary to use an one-hot encoding layer. This encoding scheme was applied to normalize the importance of each categorical value, since higher categorical values would have more influence than the others in the training process, leading to possible errors and misclassifications by the model. One-Hot Layer was used to assign a binary variable for each unique integer value, converting every integer into a binary vector, which sets the corresponding integer to "1" and "0" to the rest. In particular, this is illustrated in Figure 4.1 with respect to myocyte-specific enhancer factor 2B (protein).

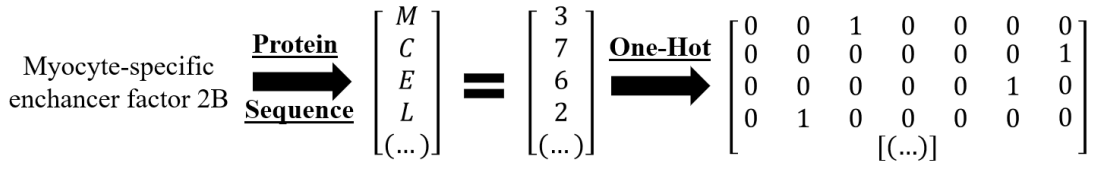

Figure 4.1: One-Hot encoding applied to myocyte-specific enhancer factor 2B.

4.2 Convolutional Neural Network

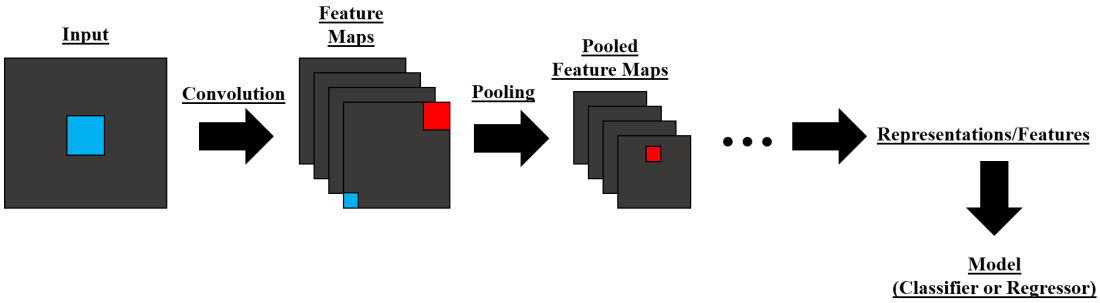

Figure 4.2: Convolutional neural network architecture.

CNNs (Figure 4.2) are a specialized kind of neural networks, based on the visual cortex where some neurons are only activated in the presence of edges in certain orientations. Neurons in the convolutional layer look for specific features, producing a stronger activation when they find them. These neural networks are known as motif detectors and features extractors, capable of identifying deep patterns from the data by moving from low level features to abstract concepts using learnable feature maps. Although they are mostly a feature engineering tool, they are capable of extracting representations not easily identified in the data and therefore known as feature extractors. There are mainly three types of CNNs: 1D, 2D and 3D, depending on the depth of the input.

They perform scattered interactions, conversely to the traditional neural networks where a matrix multiplication, identified as the weighted sum of all the outputs given by the previous connected neurons and to which is applied the activation function, is performed and the neurons are all interlinked. These scattered interactions limit the number of connections for each input, however, they are capable of describing complex interactions between many variables using a lower amount of interactions.

The convolution layer is composed by filters, which are the basic units and identified as arrays of weights that slide over the entire input. These filters work as feature identifiers and convolute at each particular location, originating activation maps, which are learnable feature maps used as the input in the next layer. The weights of the filters are usually randomized at the start and then updated according to the objective. The output volume depth (number of feature maps) is equal to the number of filters in the layer and the depth of the filter has to be the same as the depth of the input.

Convolution (Figure 4.3) is a specialized kind of linear operation, described as an

element-by-element multiplication between a particular location of the input (matrix) and the filter, followed by the sum of the results. Similar to the traditional neural networks, an activation function is applied to every value of the feature maps.

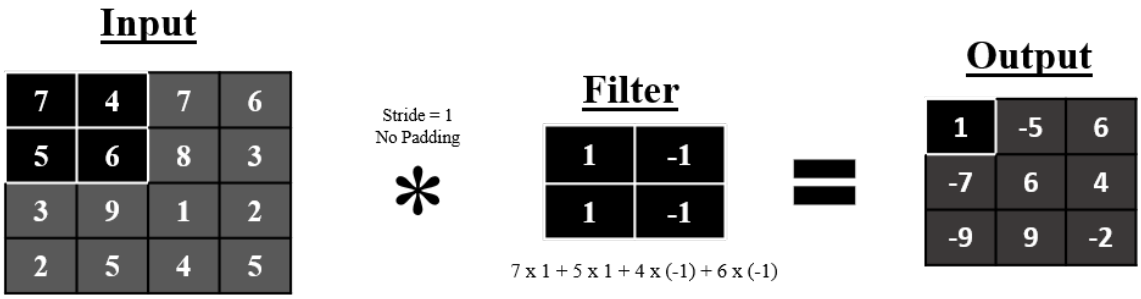

Figure 4.3: Convolution operation.

The output can be given by:

$$Output = \frac{Input\_Size - Filter\_Size + 2 * Padding}{Stride} + 1 \tag{4.1}$$

Each filter can only extract one kind of features due to the parameter sharing, as contrarily to the traditional neural networks, where each element of the weight matrix is used once when computing the output of the layer and therefore the weights and the bias of neurons are independent, the weights associated with the filter are used in every position of the input where it slides over. Thus, in order to learn more kinds of features, it is necessary to use more filters in parallel. However, the parameter sharing allows to learn only one set of weights for every location for each filter, reducing the number of weights necessary to be learned.

The number of steps that a filter moves along the input matrix, known as sliding size, is defined as stride. This parameter, that usually has the value 1 in both directions, can be modified to reduce the computational cost at the price of not extracting important features.

Convolution layers are the main layers in a CNN, however depending on the problem context, pooling layers and padding can also be applied. Pooling reduces the spatial size, width and height, by replacing the output (feature maps) at a specific location based on the nearby values and a specific function, e.g., max pooling extract the maximum value within a selected neighborhood area of the output. This method is usually used for dimensionality reduction, reducing the number of features and thus useful to lower the number of parameters to be learned in the following layers, but

also to preserve only the information about the presence of a certain feature rather than its exact location, promoting the invariance of the input to translations. The output of a pooling layer can be given by:

$$Output\_Pooling = \frac{Input\_Size - Pool\_Size}{Stride} + 1 \tag{4.2}$$

On the other hand, given the fact that after each convolution, the output reduces in size, it is possible to use padding, which is characterized as adding "zeros", to preserve and control the size of the output.

These type of neural networks are known as the state of the art for image classification [55, 56, 57, 58], outperforming several traditional approaches due to their capacity of being invariant to translations of the input, needing less parameters (reduced computational power) and for not depending on several pre-processing methodologies. Plus, some of the most recent studies apply this specific type of deep neural architectures to learn deep hidden patterns from sequences or strings [59, 60, 61, 62].

Two series of 1D convolutional layers were used, one for the protein sequences and another for the SMILES strings, to uncover deep patterns (representations or local dependencies) instead of the conventional physicochemical and/or structural descriptors, as they are general descriptors of the whole sequence or chemical structure and therefore being non relevant, in most cases, to a possible real interaction, or 3D structures, as the amount of available known structures is limited or highly complex.

A global max pooling layer (Figure 4.4) was applied, after each series of convolutional layers, to reduce the spatial size of each feature map to its maximum representative feature. The obtained deep representations were concatenated into a single feature vector, characterizing a DTI pair. The resulting features vectors were then used as the input of a FCNN architecture.

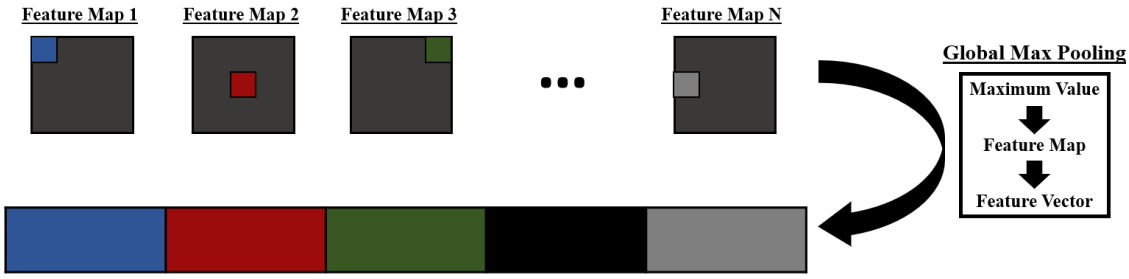

Figure 4.4: Global max pooling.

This whole process is illustrated in Figure 4.5. The two series of 1D convolutional layers are applied to the protein sequences and SMILES strings, respectively, resulting in several feature maps, depending on the number of layers and filters. Then, to each feature map, a global max pooling is applied, extracting the maximum representative feature. The resulting features, obtained from the feature maps associated with the proteins and SMILES, respectively, are concatenated into a single feature vector, characterizing a DTI pair.

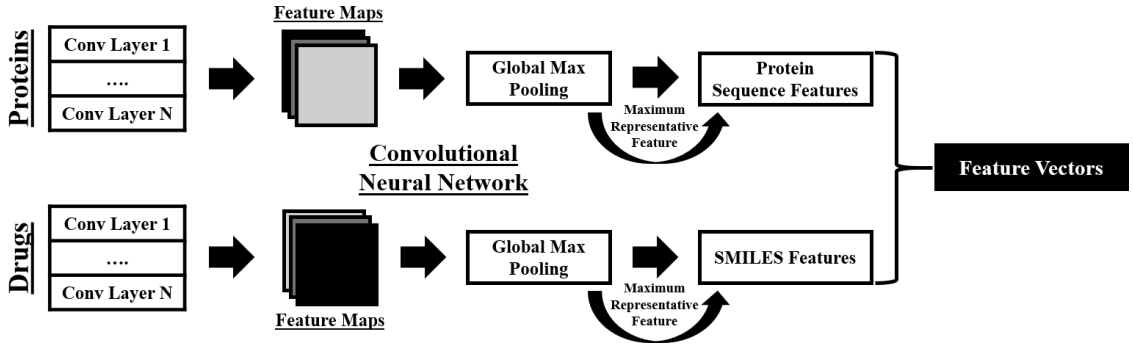

Figure 4.5: Two parallel CNNs, followed by a global max pooling, are applied to protein sequences and SMILES strings, resulting in deep representations that are concatenated into feature vectors.

4.3 Fully Connected Neural Network

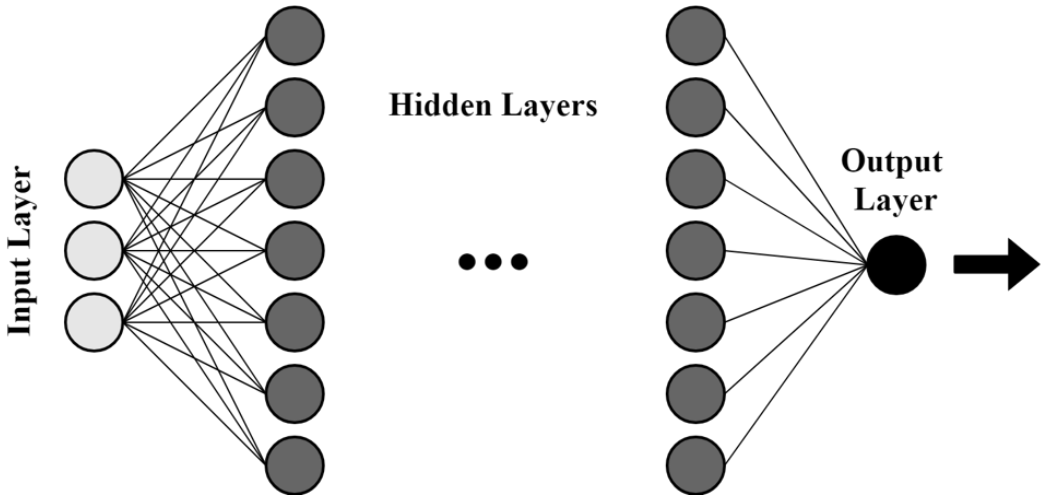

Figure 4.6: Fully Connected Neural Network Architecture.

A FCNN (Figure 4.6) was used as a binary classifier to predict DTIs as positive or negative. This type of neural networks are similar to the traditional neural networks, where all the neurons are interlinked and the output is the result of the

weighted sum of all the outputs given by the previous connected neurons and to which an activation function, which determines the activation or not of the neuron, is applied, and the path is only forward, as there are not feedback connections between the neurons. The input of this architecture is the resulting representations vectors obtained from the CNN architectures and the output a binary value, 0 or 1, representing a negative or positive interaction, respectively. This architecture differs from a traditional neural network by having more hidden layers (three or more).

Additionally, dropout (Figure 4.7) was applied between each fully connected layer, known as a dense layer composed by several artificial neurons, to reduce the overfitting [63]. Deep neural network architectures have many non-linear hidden layers, therefore there are many complex relationships to be learned between inputs and outputs, which can lead to training noise. Dropout is seen as a regularization strategy, which helps reducing learning inter-dependency and improve the generalization of the model. It works by deactivating a given percentage of neuron which develop co-dependency amongst each other during training. Although there are several regularization strategies, dropout is an inexpensive but powerful method of regularization.

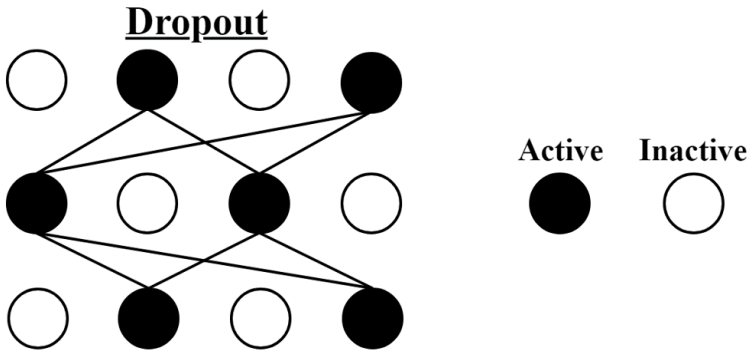

Figure 4.7: Dropout technique applied to a portion of a FCNN.

This architecture was followed by an output layer, which is essentially composed by one neuron that returns the type of interaction, 0 or 1, as it is a binary classification problem, classifying the interaction as negative or positive, respectively.

This process is demonstrated in Figure 4.8. The resulting feature vectors, which characterize DTI pairs, obtained from the two parallel CNNs model, are used as the input of a FCNN architecture. Between each dense layer of the FCNN, a dropout layer is applied, deactivating a given percentage of neurons. The final layer, identified as the output layer and constituted only by one neuron, gives a binary output, predicting the DTI pair as positive or negative.

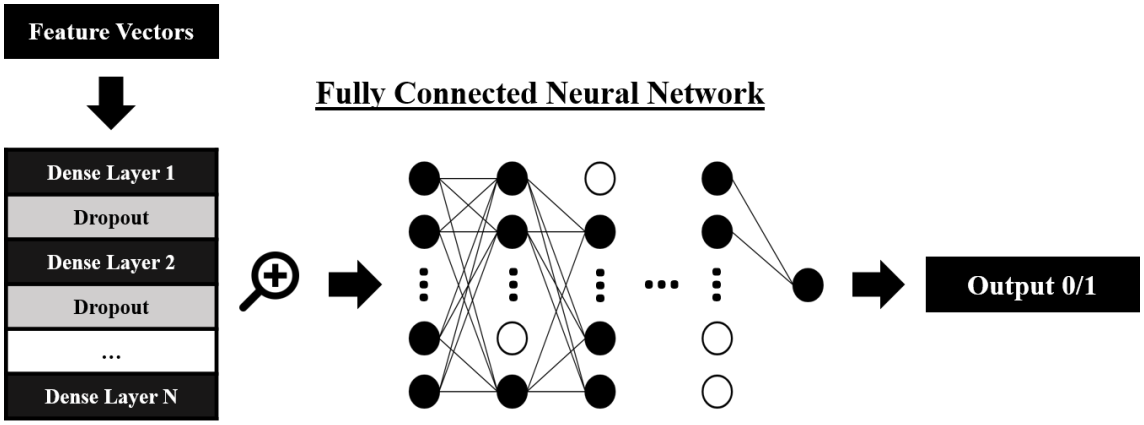

Figure 4.8: Feature vectors, obtained from the two parallel CNNs model, are used as the input of a FCNN, which is followed by a binary output layer.

4.4 Model Overview

The proposed approach is based on the combination of two deep neural network architectures, CNN and FCNN, constituting a deep learning model to predict the interactions between targets (proteins) and compounds (drugs) directly using 1D raw data, protein amino acid sequences and SMILES strings.

Protein sequences and SMILES strings are initially processed based on the length, as mentioned in Section 3.1, and then encoded into integer values according to the encoding scheme, Section 3.2.1 and 3.2.2, respectively.

These integer values are still considered as categorical values, therefore an one-hot encoding layer is applied to normalize the importance and assign a binary vector to each value.

Two parallel CNNs are used to extract deep representations from the protein sequences and SMILES, respectively. These deep representations are identified as deep patterns or local dependencies that express relevant sequential and structural regions for the prediction of DTIs.

The representations obtained are concatenated into feature vectors, characterizing DTIs, and used as the input of a FCNN. This architecture acts as a binary classifier, predicting the type of interaction as positive or negative.

The proposed end-to-end deep learning approach to predict DTIs is illustrated in Figure 4.9.

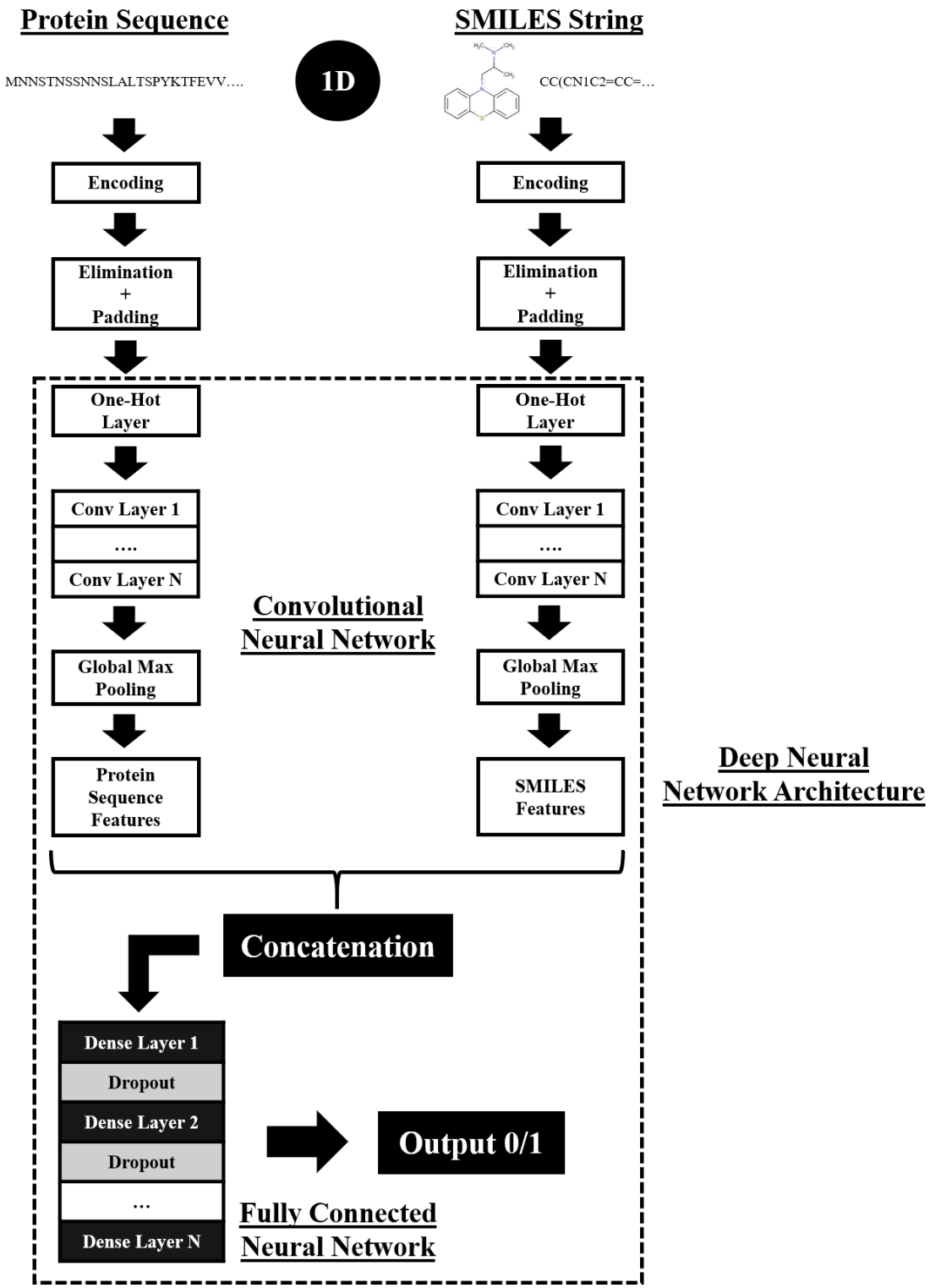

Figure 4.9: Drug-Target Interaction model architecture.

4.5 Hyperparameter Optimization Approach

The most common approach to determine the best model architecture and set of parameters is grid search with cross-validation, where the dataset is divided into training to train the model, validation to evaluate the model architecture and parameters and testing to evaluate the performance and generalization of the model. However, another strategy was applied for hyperparameter optimization (Figure 4.10) due to the fact that dividing the training set into training and validation led to high scores for every model architecture and set of parameters in both training and validation. Therefore, it was not possible to select the best model using this approach, as every model was supposedly good in the validation set but the results were inconsistent when applied to the testing set.

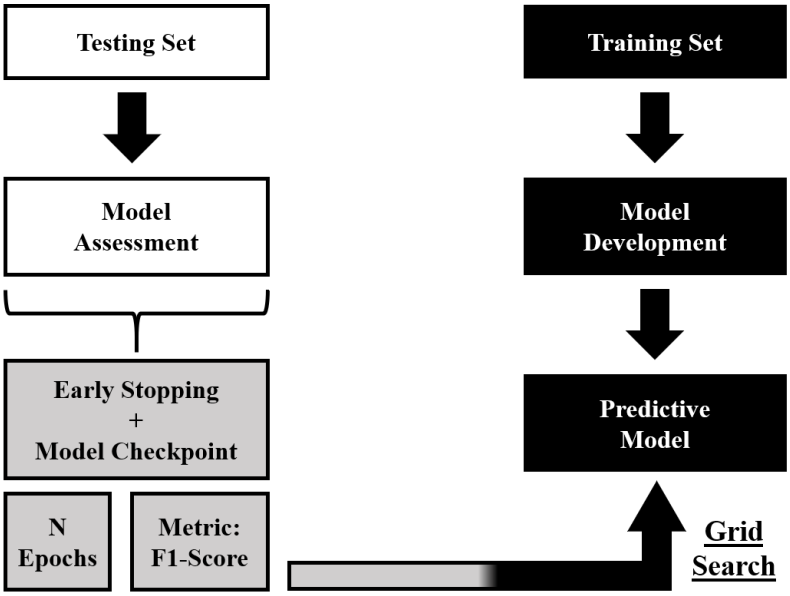

Figure 4.10: Hyperparameter optimization model based on grid search.

Two simultaneous methods, combined with grid search were used to determine the best model, early stopping and model checkpoint. Early stopping allows to interrupt the training process if, after a chosen number of epochs, there is no improvement of the evaluation metric. On the other hand, model checkpoint saves the best model, including the parameters, for that training run, independently of the finishing epoch.

Considering that splitting the training set into training and validation was not relevant for the discovery of the best model, we used the whole training set for training and the testing set to evaluate the model performance at each epoch. Since the testing set is highly imbalanced, F1-score, which is an harmonic mean that considers both the precision and recall and therefore an overall goodness of the classification,

was used for this evaluation.

The resulting methodology for hyperparameter optimization can be considered as a valid choice due to the fact that both datasets are independent and with a low similarity between the drug pairs that composed them. Besides, the main goal is to evaluate the capacity of CNNs to obtain useful representations of sequential and structural data to predict DTIs and how they perform against using global and conventional descriptors.

5

Experimental Setup

5.1 Datasets

5.1.1 Protein Sequences & SMILES Strings Dataset

DTI Pairs

Coelho et al. (2016) [64] DTI dataset, mainly the interactions pairs, was used as benchmark to evaluate and validate the proposed model. Positive interaction dataset was obtained from DrugBank [65] and Yamanishi et al. (2008) [35], where all entries related to specific classes of protein targets and proteins with unreviewed status were removed. On the other hand, the negative interaction dataset was collected from BioLiP [66] and BindingDB [67], where a bioactivity threshold of 10 μM was used to identify weak binding interactions. Interactions with a Kd, which is related to how tightly the compound is bound to a protein and therefore known as a binding affinity metric, superior of 10 μM are considered as true negative [68]. A ratio of 1.5 negative to positive was adopted, resulting in 7206 positive and 10,912 negative DTI pairs for training and 3,530 positive and 5,297 negative DTI pairs for testing. Additionally, only Yamanishi et al. (2008) [35] and DrugBank [65] positive entries were used for training and testing, respectively.

The original work [64] ensured the discriminating power by evaluating the sequence similarity within each dataset and across all datasets and guaranteeing that less than 1% of all possible drug pairs had a sequence similarity score greater than 0.85, excluding any possibility of redundancy between the two datatsets, training and testing.

Table 5.1 summarizes the amount of unique drug, targets and drug-target interactions extracted from the databases and used to create the training and testing datasets, respectively.

Table 5.1: Unique drugs, targets and DTIs.

	Positive		Negative	
	DrugBank [65]	**Yamanishi et al. (2008) [35]**	**BioLip [66]**	**BindingDB [67]**
Drugs	1328	790	894	12454
Targets	706	1371	636	404
DTI	3530	7206	1223	14985

Protein Data

The protein sequences were all extracted from UniProt [69] using their identifiers, e.g., P00489. Since we are using proteins sequences directly and not global descriptors of the sequence, each amino acid that constitutes the sequence is considered as a feature. Therefore, it was necessary to define a threshold based on their length, as mentioned in Section 3.1. An information threshold of 95 % was used, resulting in a maximum length of 1205 for the protein sequence. Every protein sequence with a length superior or inferior to the threshold was removed or padded, respectively. Figures 5.1a and 5.1b show the protein sequence length distribution for the training and testing set, respectively.

Chemical Data

The SMILES strings were collected exclusively from PubChem [70], in their canonical format, to guarantee a consistent notation to represent the chemical structures of all drugs across both datasets. Each character of the SMILES string is considered as a feature, therefore if different notations were to be used to represent the chemical structures, equal segments of the compounds would be seen as different components by the model, resulting in eventual errors.

The dataset contains IDs from multiple databases, including PDB [71], KEGG [72], ZINC [73, 74] and Drugbank [65], thus it was necessary to convert them to PubChem [70] compounds IDs first in order to extract the SMILES strings. Python packages, PyPDB [75], BioServices [76] and PubChemPy [77], were used for conversion and extraction. Identical to the protein sequences, a threshold based on their length was also applied, resulting in a maximum length of 90 for the SMILES strings, and all compounds with a length superior or inferior to the threshold were removed or padded, respectively. Figures 5.1c and 5.1d show the SMILES string length distribution for the training and testing set, respectively.

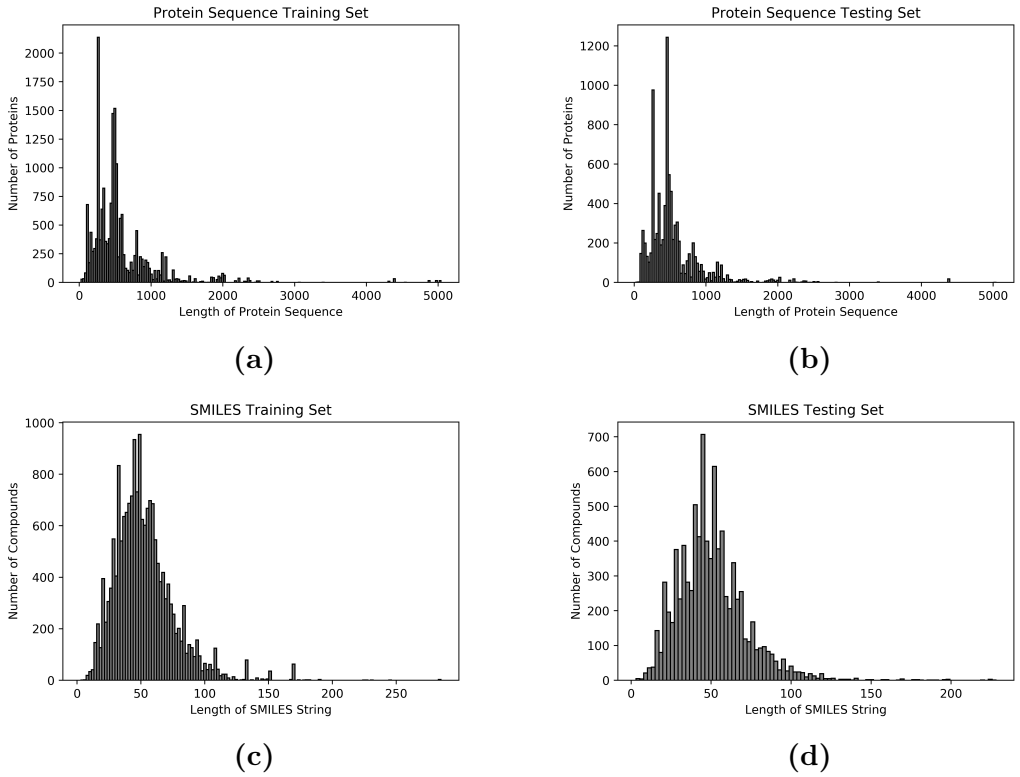

Figure 5.1: Distribution of the proteins and SMILES lengths of training and testing datasets. (a) Training protein sequences. (b) Testing protein sequences. (c) Training SMILES. (d) Testing SMILES.

Training & Testing Synopsis

All entries duplicated or containing missing characters in one of the datasets, training and testing, were removed. Table 5.2 summarizes the result of elimination and Table 5.3 the amount of unique targets, drugs and number of targets for the training and testing datasets, respectively, after elimination.

Table 5.2: Training and testing datasets after elimination.

	Positive	**Negative**	**Total**
Training	5839	10172	16011
Testing	3012	4914	7926

Table 5.3: Unique targets, drugs and number of targets for the training and testing datasets.

	Unique		Number of Targets	
	Targets	**Drugs**	**1**	**>1**
Training	1790	9583	8026	1557
Testing	1068	5718	4884	834

5.1.2 Coelho et al. (2016) Descriptors Dataset

The effectiveness of using representations over descriptors was evaluated by using the original work descriptors [64]. This dataset contains a total of 432 protein descriptors and 323 drug descriptors, all collected using PyDPI package [78]. The protein descriptors are divided into amino acid composition, Moran autocorrelation and CTD descriptors. On the other hand, drug descriptors are divided into molecular constitutional, molecular connectivity, molecular property, kappa shape and charge descriptors, MACCS keys and E-state fingerprints.

There was no feature selection performed as the main purpose was to compare the performance of using the CNNs to obtain important sequential and structural representations with already performed work using global descriptors. This dataset was used in the Section 5.3, 5.4 and 5.5 models.

5.1.3 Specific Descriptors Dataset

The model of Section 5.6 evaluates the influence of a specific group of descriptors, namely CTD descriptors for proteins and charge, molecular property and molecular connectivity descriptors for compounds. The CTD descriptors represent several structural and physicochemical properties, specifically hydrophobicity, polarity, charge, polarizability, normalized Van der Waals volume, secondary structures and solvent accessibility. On the other hand, charge descriptors express electronic features and molecular property and connectivity descriptors represent a handful of physicochemical properties. The main reason behind the choice of these descriptors was that they represent specific and intrinsic properties of the proteins and compounds.

The Python package PyDPI [78] was used to extract all the descriptors, resulting in a total of 147 CTD (21 Composition, 21 Transition and 105 Distribution) descriptors, 44 molecular connectivity descriptors, 25 charge descriptors and 6 molecular

property descriptors.

5.2 Main Model

A deep neural network architecture is used to predict the interaction, as positive or negative, between drugs and targets based on 1D raw data, protein amino acid sequences and SMILES strings. Conversely to the traditional hyperparameter optimization approaches, we used the testing set to evaluate the model performance, based on F1-score, at each epoch, as it was explained in Section 4.5, to find the best model and set of parameters. The model has several parameters possible to hyperoptimize, however we only selected six: number of filters for proteins and compounds, filter length for proteins, filter length for compounds, number of neurons for each dense layer, dropout rate and optimizer learning rate. A wide range of possible values was given for each hyperparameter and the number of convolutional layers and dense layers was fixed at three. There is currently no golden rule to determine the number of layers for each neural network architecture (CNN and FCNN), however an exponential increase of the number of layers usually does not result in better performance due to the higher computational power and tendency to overfitting. The goal of deep neural networks is to learn representations from the lower layers that can be used by the higher layers and each individual neural network architecture with an optimized choice of parameters should be able to fulfill they purpose without needing an excessive number of layers.

The training/learning process of a deep learning neural network architecture highly depends on the activation function, gradient descent optimizer and loss function.

The activation function is responsible for the activation or not of each neuron, by applying a limit/transformation to the result of the weighted sum. There are many different types of activation functions, including linear, sigmoid and ReLU. Linear is the simplest activation function, where the value does not suffer any transformation. It is easier to train with but can not learn complex relationships in the data.

$$L(x) = x \tag{5.1}$$

Sigmoid is a non linear activation function, also known as the logistic function, where the input is transformed into a value between 0 and 1. In the case that the input is lower or larger than 0 or 1, they are transformed to 0 or 1, respectively. The drawbacks of non linear activation functions is that they easily saturate, making it difficult for the learning algorithm to continue to adapt the weights, leading to

the gradient vanishing problem, which makes it difficult to know which direction the parameters should move to improve the loss function, and are only sensitive to changes around their mid-point. Nonetheless, this function was used as the output activation function of the FCNN, given the fact that this layer's only purpose is to classify the interaction as positive or negative, if the previous value is higher or lower than 0.5, respectively.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{5.2}$$

ReLU is a nearly linear activation function that preserves the main properties of the linear and non linear activation functions. It returns zero if it receives any negative input (non linear) or the value itself if any positive input (linear). This activation function allows to learn complex relationships in the data, provides sensitivity to the activation sum input, avoids saturation and easier to optimize with gradient-based methods. Besides, it is simple to compute and used in most deep learning architectures.

$$R(x) = max(0,x) \tag{5.3}$$

Gradient descent is used to train the deep neural network architecture and is considered as an optimization algorithm. It tries to minimize an objective function based on the model's parameters by updating the parameters in the opposite direction of the gradient of the objective function. This type of algorithms are associated with a learning rate that determines the size of the steps it takes to reach a local minimum. In other words, a prediction error, determined by the chosen loss function, is calculated and used to estimate a gradient that is propagated backwards (backpropagation) through the network from the output layer to the input layer and updates the weights. There are two types of problems that may happen during the gradient descent optimization, specifically the vanishing gradient and exploding gradient. The vanishing gradient problem happens when the error is so small that when it reaches the input layer, the update it performs has very little effect. On the other hand, the exploding gradient problem occurs when the gradient exponentially increases as it is propagated backwards. There are many different types of gradient descent based methods, e.g., SGD, RMSprop and Adam. However, given the problem context and the usually good performance obtained from using Adam, we decided to use only Adam in the CNNs and FCNN.

Adam [79], known as a combination of RMSprop and SGD with momentum, is an adaptive learning rate optimization algorithm that computes individual learning rates for each parameter. It uses estimators of the first and second moments of

gradient, mean and uncentered variance, respectively, to adapt the learning rate for each weight. Additionally, it was combined with mini-batch, which instead of using the entire dataset to compute the gradient, only a certain number of samples are used in every iteration, reducing the variance of the parameter updates and being less prone to overfitting.

$$W_t = W_{t\_1} - \eta \frac{\hat{m_t}}{\sqrt{\hat{v_t} + \epsilon}} \tag{5.4}$$

, where W is the weight, η the step size and m and v the moving averages.

Loss functions are used to measure the inconsistency between predicted and real values and therefore play an important role in the learning/training process of the deep neural network architecture. Binary cross entropy was selected as the loss function and measures the divergence between two probability distributions, in which y is the label and p(y) is the predicted probability:

$$L(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(p_i) + (1 - y_i) log(1 - p_i)] \tag{5.5}$$

Additionally, taking in account the existing class imbalance of the training set (64% and 36 % for the negative and positive class, respectively) we decided to switch class weights, giving special attention to the positive class, as the primary focus is around positive interactions.

Table 5.4 summarizes the hyper-parameters obtained from grid search.

The proposed model performance was compared with an RF approach, an FCNN architecture, an SVM approach and also an CNN, autoencoder and FCNN combined model. Random forest and Support vector machine are the most used traditional machine learning approaches and considered as the state of the art in several studies mentioned in Section 2.3.1, therefore it was important to compare the performance of the proposed setup (deep learning architecture) with these two methods as well as the effectiveness of using representations over descriptors. Besides, RF was the method used in the referenced work [64], which is based on descriptors, thus important to validate and evaluate the capacity of the proposed setup.

Additionally, it was crucial to compare the differences of using deep learning using both representations and descriptors, in order to evaluate the quality and discriminatory power of the representations obtained from the CNNs and also the disparity of using deep and machine learning with the same data as input.

Table 5.4: Parameter settings for the proposed model.* Initial number of epochs to allow convergence of the model, however early stopping and model checkpoint were used.

Parameters	Value
Number of Convolutional Layers	3
Number of Dense Layers (FC)	3
Number of Filters	[128, 256, 384]
Filter Length (Proteins)	[3,4,5]
Filter Length (Compounds)	[3,4,5]
Epochs*	500
Hidden Neurons	[128,128,128]
Batch Size	256
Dropout Rate	0.5
Optimizer	Adam
Learning Rate	0.0001
Loss Function	Binary Cross Entropy
Activation Function (CNN)	ReLU
Activation Function (FC)	ReLU
Activation Function Output)	Sigmoid
Class Weights (imbalanced classes)	{0: 0.36, 1: 0.64}

Even thought the novelty of the proposed work is to use 1D sequential and structural data combined, discarding completely the use of conventional and global descriptors of the proteins and drugs, to predict DTIs, we decided to evaluate the influence of characteristics considered as intrinsic properties of the proteins and drugs in the correct prediction of DTIs.

Python 3.6.6 and Keras [80] with TensorFlow [81] back-end were used to develop the proposed model. The experiments were run on 2.20GHz Intel i7-8750H and GeForce GTX 1060 6GB.

5.3 Random Forest

RF is an ensemble learning method that generates a chosen number of decision trees and returns the class that is the mode of the classes across the output of each individual decision tree. Decision trees are the building blocks of the forest and they can be defined as series of if-then-else rules that divide the dataset into smaller subsets until the predicted class or value is achieved or when the impurity can no longer be reduced. The rules (nodes) are based on a single feature and a specific threshold according to the combination that generates the less impurity, e.g., entropy, for the tree. Each decision tree, at each node, in the RF approach, only considers a subset of features that are randomly chosen. Additionally, not all

samples are used to build each tree, only a random selected portion of the dataset is used to build the tree, where the other one is used to estimate the generalization accuracy (out-of-bag). The randomness of the whole process increases the diversity among the trees, making them grow dissimilar and uncorrelated. This method is capable of describing the relationship between independent and dependent variables with high flexibility and sufficient accuracy due to being highly adaptive to data.

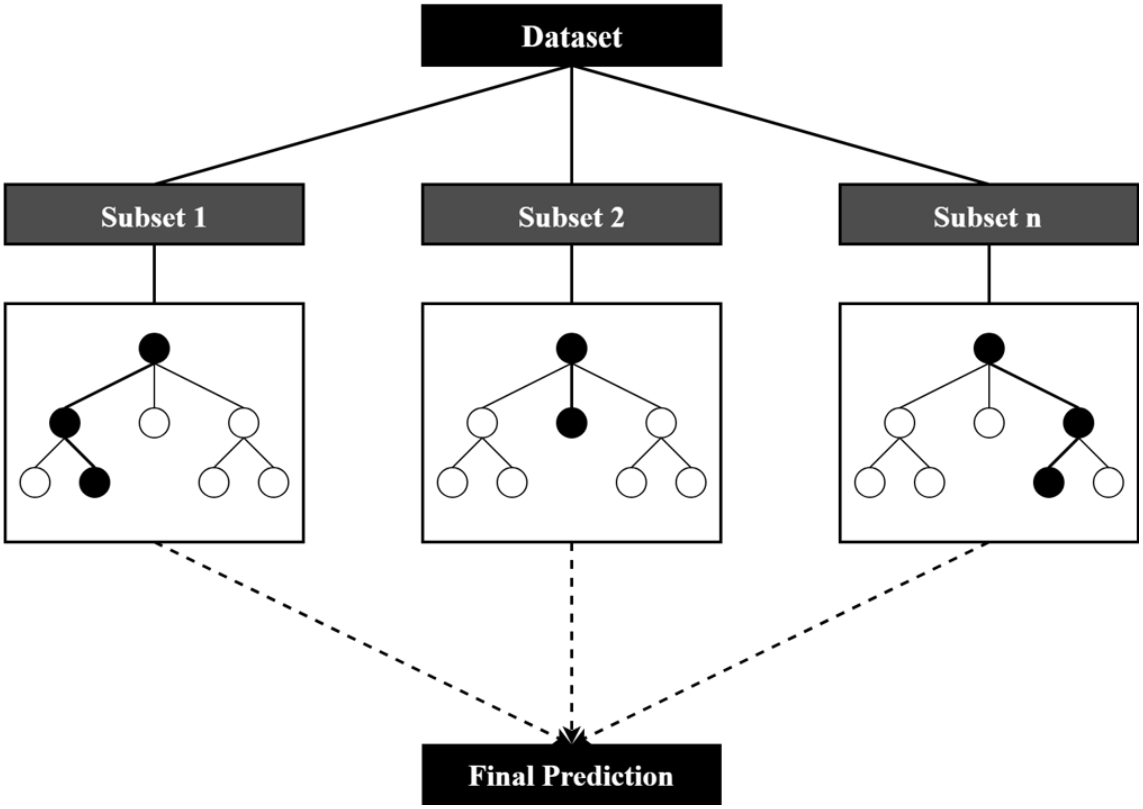

Figure 5.2: Random Forest.

RF was the ensemble learning method used by the original work [64] to make predictions on drug-target interactions. The hyperparameters setting for this method was the same as the original work and obtained using 5-fold cross-validation. This method, denominated of K-fold cross-validation, splits the training set into K subsets and then uses 1 for testing/evaluation and the rest for training, repeating the process K-1 times. It is usually used to evaluate the current model settings and also to investigate if it is overfitting.

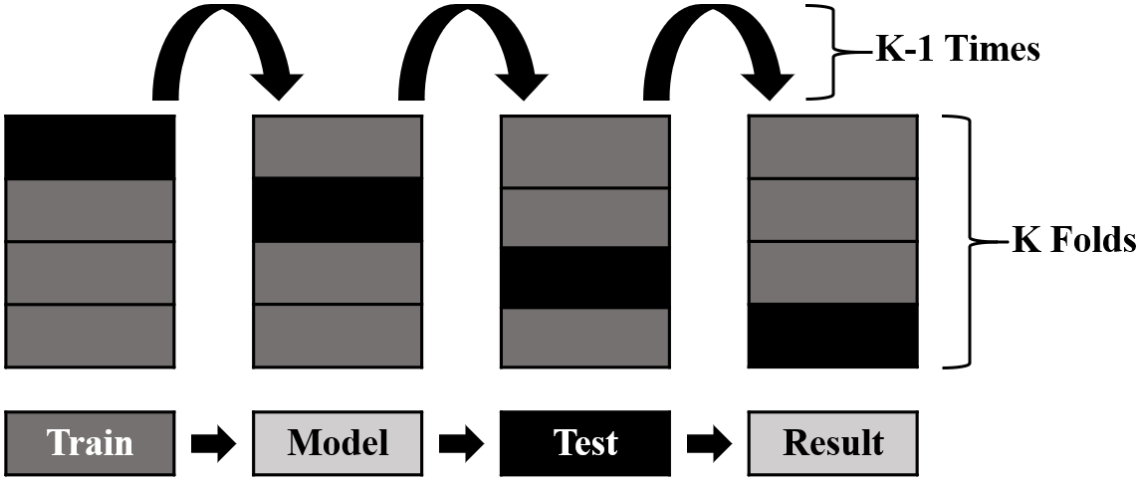

Figure 5.3: K-fold cross-validation.

Table 5.5 summarizes the parameter settings for the RF model.

Table 5.5: Parameter settings for the RF model.

Parameters	Value
n_estimators	150
max_features	100
criterion	Gini
max_depth	None
min_samples_split	2
min_samples_leaf	1
max_leaf_nodes	None

The n_estimators refers to the number of trees in the forest, max_features to the number of features to consider when looking for the best split, criterion to the measure of impurity, which Gini was selected and measures how often a randomly chosen element would be mislabeled if it was randomly labeled according to the distribution of labels in the respectively subset, max_depth to the maximum depth of the tree, min_samples_split to the minimum number of samples to split a node, min_samples_leaf to the minimum number of samples to form a node and max_leaf_nodes to the limit of nodes.

Coelho et al. (2016) [64] descriptors and protein and SMILES representations extracted from the pre-trained proposed setup were used to evaluate the performance of this approach. Scikit-learn [82] was used to implement the RF method.

5.4 Fully Connected Neural Network

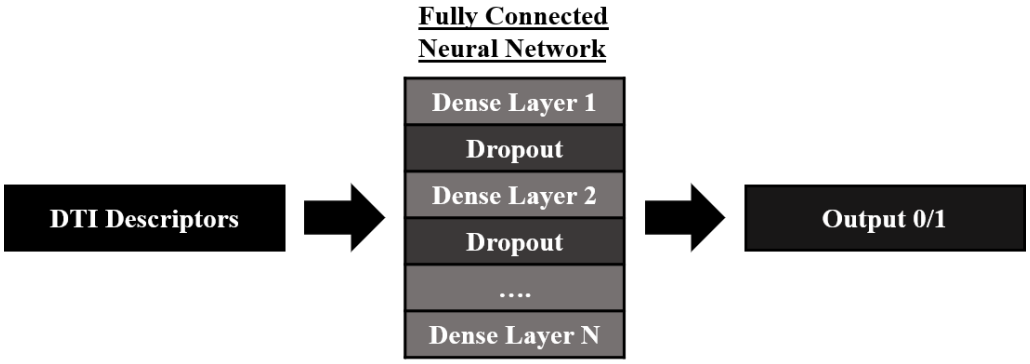

Figure 5.4: FCNN model using descriptors as input.

The proposed approach already uses a FCNN as a binary classifier, however the performance of this method was evaluated using descriptors, in order to compare the performance of using representations over descriptors and also the performance of using deep learning over traditional machine learning.

The parameter settings for the architecture were obtained by grid search using the hyperparameter optimization approach mentioned in Section 4.5. Table 5.6 summarizes the parameter settings for this architecture.

Table 5.6: Parameter settings for the FCNN model using descriptors as input. *Initial number of epochs to allow convergence of the model, however early stopping and model checkpoint were used.

Parameters	Value
Number of Dense Layers (FC)	3
Epochs*	500
Hidden Neurons	[128,1024,256]
Batch Size	256
Dropout Rate	0.2
Optimizer	Adam
Learning Rate	0.001
Loss Function	Binary Cross Entropy
Activation Function (FC)	ReLU
Activation Function (Output)	Sigmoid
Class Weights (imbalanced classes)	{0: 0.36, 1: 0.64}

5.5 Support Vector Machine

SVM defines a hyperplane that maximizes the separation margin between different classes. In the case of problems that are not linearly separable, SVM uses two

different approaches: soft-margin and kernel tricks. Soft-margin tolerates violations of the margins and gives a penalty term for misclassifications. The tolerance given when finding the decision boundary is represented by the penalty term C, which is responsible for the number of violations allowed. Given the amount of features and that most of the problems are not linearly separable, kernels, which are identified as functions capable of transforming the data, are used to map data to high dimensional spaces where it is possible to classify with linear decision surfaces. Some of the kernels used are Linear, Gaussian Radial Basis Function, Polynomial and Sigmoid.

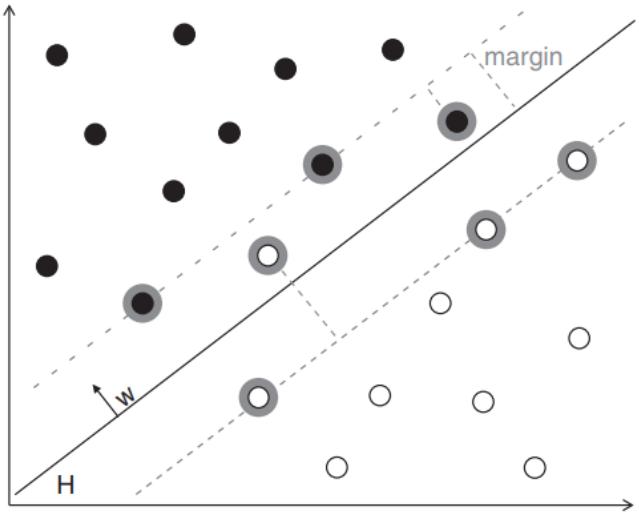

Figure 5.5: Support Vector Machine. Image from "Support vector machines for drug discovery" [83].

The parameter settings were obtained using 5-fold stratified cross-validation, which contrarily to K-fold cross-validation, ensures that each fold contains roughly the same proportion of each class. Table 5.7 summarizes the parameter settings for the SVM model.

Table 5.7: Parameters Setting for the SVM Model.

Parameters	**Value**
C	1.0
kernel	rbf
gamma	scale
tol	0.001

The parameter C refers to the penalty term and is responsible for the number of violations allowed, tol to the tolerance for the stopping criterion, kernel to the function used to transform and map the data and gamma to a coefficient of the

kernels and associated with the "spread" or decision region of the kernel. RBF uses the following equation to transform the data:

$$K(x,x') = exp(-\frac{|| x - x' ||^2}{2\sigma^2}) = exp(-\gamma || x - x' ||^2) \tag{5.6}$$

, where x and x' are two data points, $|| x - x' ||^2$ the euclidean distance between the two data points and σ a free parameter that is usually represented in the form of γ, where $\gamma = \frac{1}{2\sigma^2}$. The parameter γ set to "scale" uses for it's value $\frac{1}{num\_features * variance(x)}$

Similar to the RF approach, both descriptors and protein and SMILES deep representations were used to evaluate the performance. Scikit-learn [82] was used to implement this classifier.

5.6 CNN, Autoencoder and FCNN Combined Model

In order to determine the influence of specific descriptors to the overall prediction of drug-target interaction, a model based on CNNs, Autoenconders and FCNNs was used to evaluate this influence and compare it with the proposed setup (Figure 5.6).

Identical to the proposed model, two parallel CNNs were used to extract deep representations from protein sequences and SMILES strings, where the pre-trained proposed setup was applied for this purpose.

Autoencoders (Figure 5.7) are a specific type of neural network architecture, where the learning process is done in an unsupervised manner. The main purpose of this architecture is to perform a reduction of the feature input space by compressing the data and then uncompress it into something that closely matches the original data. Hence, it allows to extract a smaller set of features that represent the input data, by performing dimensionality reduction with some data "denoising". The select layer, known as "the bottleneck" or code layer, is where the smaller set is extracted from and in which maximum reduction is achieved through encoding. The goal behind using this architecture is to represent a specific group of descriptors in a lower dimension space, in the form of deep representations.

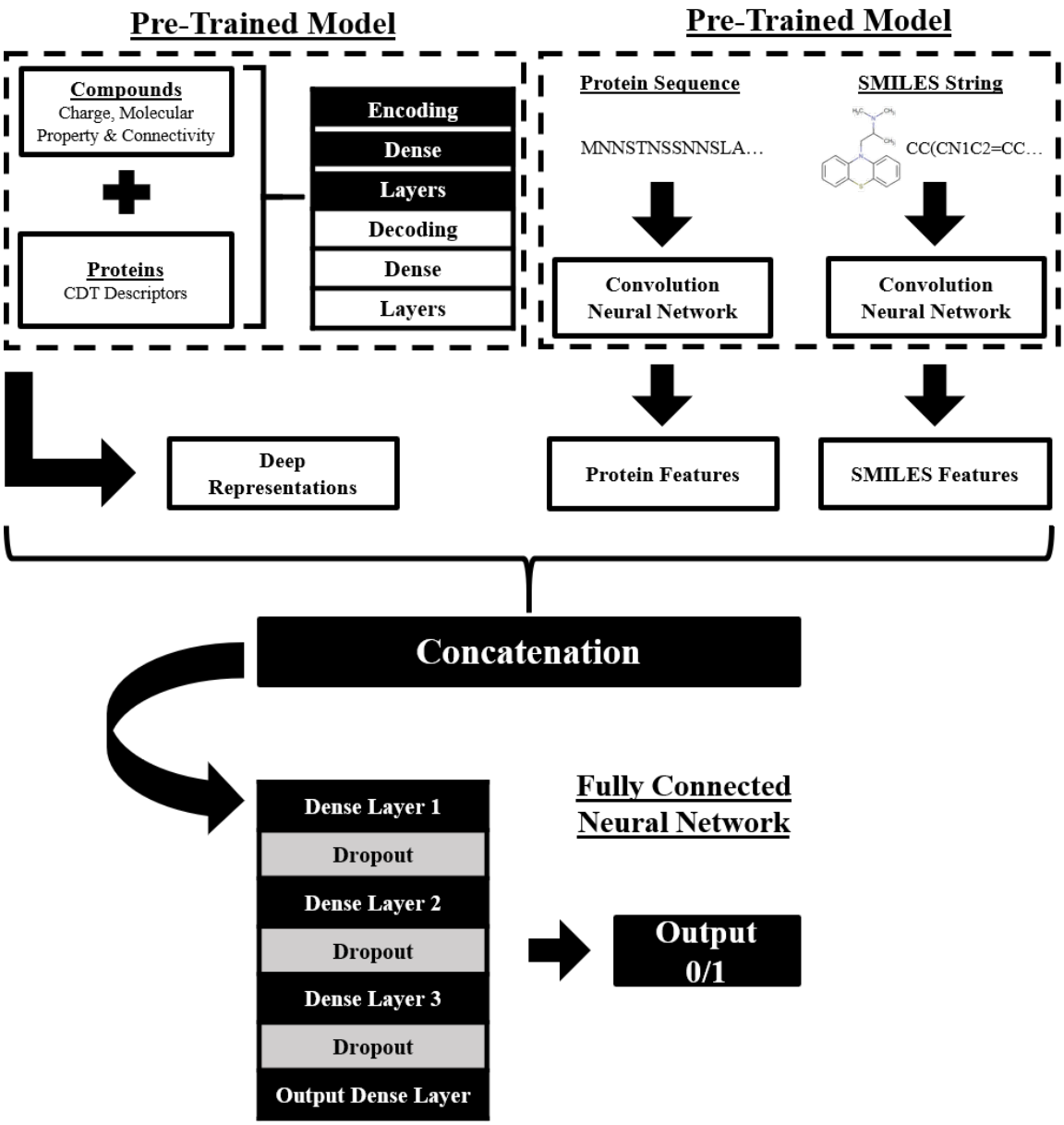

Figure 5.6: CNN, Autoencoder and FCNN combined model.

An autoencoder was applied on the particular group of descriptors mentioned in Section 5.1.3. The model uses a stack of dense layers, three for encoding and decoding, respectively. Early stopping and model checkpoint based on the loss value were applied to find the best set of weights for the network. The difference between the output and input is the goal to minimize, therefore MSE was used as the loss function:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_{\mathrm{i}} - \hat{Y}_{\mathrm{i}})^2 \tag{5.7}$$

, where n is the number of values, Y_{i} the real values and $\hat{Y}_{\mathrm{i}}$ the predicted values.

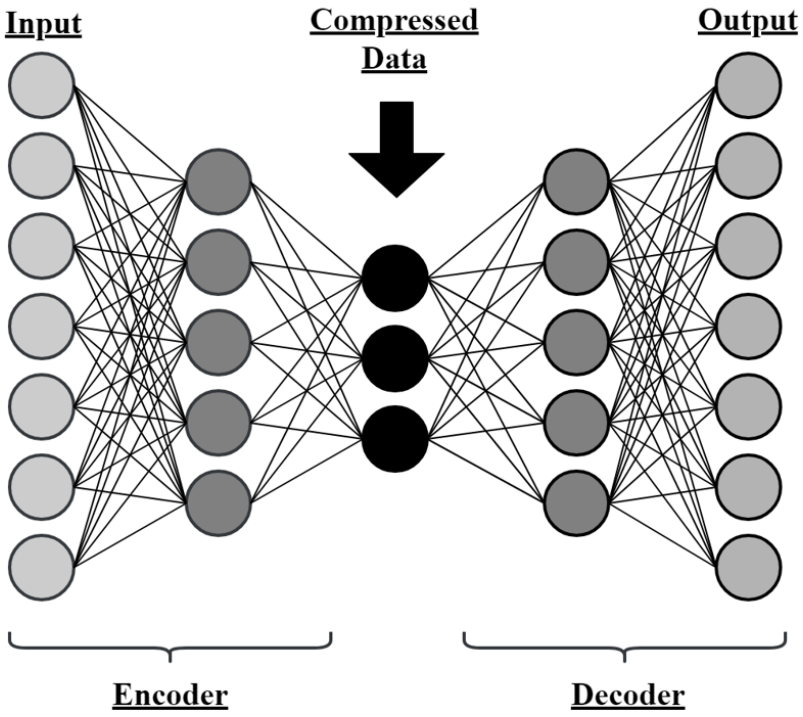

Figure 5.7: Autoencoder architecture.

This resulted in a dimensionality reduction of 222 descriptors to 32 deep representations. Keras [80] with TensorFlow [81] back-end was used to build this architecture. Table 5.8 summarizes the parameter settings for the autoencoder model.

Table 5.8: Parameter settings for the autoencoder model. * Initial number of epochs, however early stopping and model checkpoint were applied.

Parameters	Value
Number of Encoding Dense Layers	3
Number of Decoding Dense Layers	3
Encoding Hidden Neurons	[128, 64, 32]
Decoding Hidden Neurons	[64,128,222]
Epochs*	500
Batch Size	256
Optimizer	Adam
Learning Rate	0.0001
Loss Function	Mean Squared Error
Activation Function	ReLU
Activation Function (Output)	Sigmoid

The obtained features from the two pre-trained models were concatenated into a single feature vector and used as the input of a FCNN. Grid search based on the hyperparameter optimization approach of Section 4.5 was performed. Table 5.9 summarizes the parameter settings for the FCNN architecture.

Table 5.9: Parameter settings for the FCNN related to the combined model. * Initial number of epochs, however early stopping and model checkpoint were applied.

Parameters	Value
Number of Dense Layers (FC)	3
Epochs*	500
Hidden Neurons	[512,256,1024]
Batch Size	256
Dropout Rate	0.2
Optimizer	Adam
Learning Rate	0.001
Loss Function	Binary Cross Entropy
Activation Function (FC)	ReLU
Activation Function Output)	Sigmoid
Class Weights (imbalanced classes)	{0: 0.36, 1: 0.64}

5.7 Evaluation Metrics

There are many metrics used to evaluate the performance and the capacity of the models as predictors. However, the choice of which ones to use, highly depends on the problem context and the distribution of the labels of the testing set. Even though the basis of them all is to compare the predict labels with the true labels, each evaluation metric evaluates particular things and is influenced differently by the distribution of the labels and results.

For performance comparison, the following evaluation metrics were used:

1. **Accuracy**: rate of predictions correctly classified.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.8}$$

2. **Sensitivity**: rate of positives correctly classified.

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.9}$$

3. **Specificity**: rate of negatives correctly classified.

$$Specificity = \frac{TN}{TN + FP} \tag{5.10}$$

4. **F1-Score**: harmonic mean between precision and recall.

$$Precision = \frac{TP}{TP + FP} \tag{5.11}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.12}$$

$$F1 - Score = 2 * \frac{precision * recall}{precision + recall} \tag{5.13}$$

5. **Confusion Matrix**: two-dimensional table that allows visualization of the performance of the algorithm.

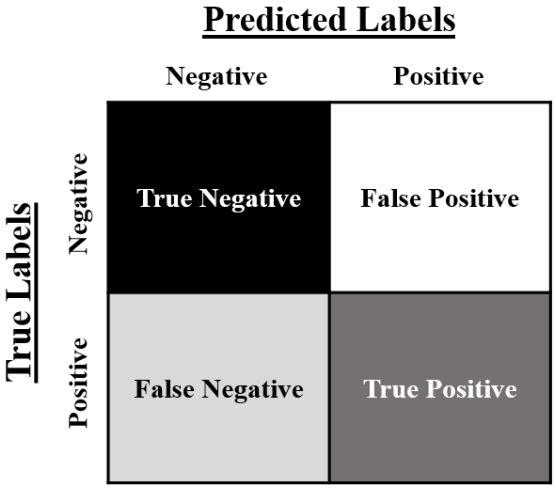

Figure 5.8: Confusion matrix.

TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative

Accuracy is one of the most used metrics to evaluate the performance of a predictive model, as it is a direct comparison of the results with the expected labels. However, it can be misleading if the testing set is highly imbalanced, resulting in good values even when the classifier is performing poorly, e.g., it correctly classifies only the dominant class/classes. Sensitivity and Specificity are usually used in binary classification problems to evaluate the capacity of the model to predict both classes. F1-score is highly used, specifically in imbalanced problems, due to the fact that it gives an overall idea of the performance of the model.

6

Results and Discussion

In the context of drug repositioning and finding new leads, identifying correctly positive interactions should be the central focus, as negative interactions are not normally registered and therefore based on possible hypotheses or absence of information. Nonetheless, the model needs to be able to accurately distinguish both types of interactions, positive and negative, in order to validate its effectiveness and also to guarantee that possible new findings, that is, the discover of new positive interactions associated with a certain drug, are potentially credible and therefore legitimate the identification of that drug as a possible lead.

Table 6.1 and Table 6.2 show the overall experimental results for the deep learning and machine learning approaches, respectively, in terms of the metrics mentioned in Section 5.7. The confusion matrix for the proposed setup is shown in Figure 6.1.

Table 6.1: Prediction results of testing set for the deep learning approaches.

		Model		
		CNN+FCNN	**CNN+Autoencoder+FCNN**	**FCNN**
		CNN Representations	**CNN+Descriptors Representations**	**Descriptors**
Metric	**Sensitivity**	0.861	0.880	0.827
	Specificity	0.961	0.948	0.963
	F1-Score	0.895	0.896	0.876
	Accuracy	0.923	0.922	0.911

Table 6.2: Prediction results of testing set for the machine learning approaches.

		Model			
		Random Forest		**SVM RBF**	
		Descriptors	**CNN Representations**	**Descriptors**	**CNN Representations**
Metric	**Sensitivity**	0.809	0.821	0.739	0.769
	Specificity	0.989	0.992	0.989	0.993
	F1-Score	0.886	0.896	0.842	0.864
	Accuracy	0.921	0.927	0.894	0.908

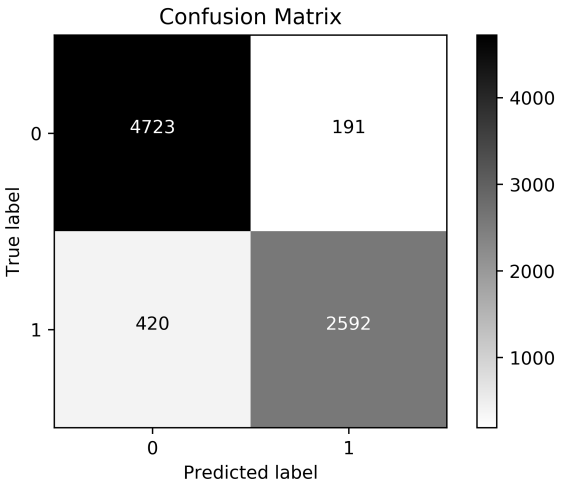

Figure 6.1: Confusion matrix of testing set classification for the proposed model.

The differences in performance between all models can be interpreted as a result of the difference between using deep representations, obtained using protein sequences and SMILES strings, and global descriptors. Besides, it is also possible to highlight the difference between applying traditional machine learning and deep learning approaches. Thus, there are essentially four research questions that need to be answered:

1. **How does the proposed model performs in the correct prediction of both positive and negative interactions?**

The proposed approach is based on the concept of using an end-to-end deep learning process, capable of extracting deep representations from data and then use them as input of another deep learning architecture. Two parallel CNNs were used to extract deep representations from protein sequences and SMILES strings and then used as the input of a FCNN.

The end-to-end deep learning proposed method resulted in a high sensitivity (0.861) and specificity (0.961) when compared to the other models, which obtained a high specificity and a low sensitivity, with the exception of the CNN, autoencoder and FCNN combined model that resulted in a high sensitivity (0.880) and a low specificity (0.948).

The testing set is imbalanced, 62% negatives and 38% positives, thus our approach exceeds other models in its capability to correctly classify both positive and negative drug-target interactions, achieving better results overall.

2. **How discriminating are the representations in comparison to the global descriptors?**

The results obtained by the proposed model validate the effectiveness of CNNs as feature extractors and their capacity to automatically surmise and identify important sequential and structural regions for drug-target interactions, as they outperform completely the results achieved using a FCNN with global descriptors.

The RF method evidences that using deep representations outperforms conventional global descriptors in every evaluation metric, which is also manifested when using a SVM. Besides, RF surpasses SVM in both configurations, which is in agreement with the notion that this method usually runs adequately on large datasets and is less susceptible to overfitting.

Protein and compound representations learned from sequential and structural information with CNNs are more discriminating for classification than global descriptors. Furthermore, these representations are extracted from sequential and structural raw data, hence the CNNs are automatically learning which sequential and structural regions are relevant for a drug-target interaction. Conversely, conventional descriptors are general information about the whole sequence or structure and not specific to the binding regions.

3. **Does deep learning completely outperforms traditional machine learning in every situation?**

The model based on a FCNN architecture with conventional descriptors as input, shows that deep learning in its essence is not enough to completely outperform traditional machine learning approaches. This is illustrated when comparing the evaluation metrics, which are higher, specifically the sensitivity (0.827), F1-score (0.876) and accuracy (0.911), than the SVM approach but lower, with the exception of the sensitivity (0.827), than RF method in both configurations, respectively. Moreover, it highlights the inefficiency of using global descriptors over deep representations extracted from CNNs. Inevitably, the quality and discriminatory power of the input data have a great influence in the performance achieved.

4. **How useful are specific descriptors in the correct prediction of interactions?**

Although the efficiency of using CNNs to extract deep representations over global descriptors is verified, we decided to evaluate the influence of specific descriptors encoded as deep representations by an autoencoder combined with proteins and SMILES deep representations. The results demonstrate that using additional information may be useful to correctly identify positive interactions, which is verified by achieving the highest sensitivity (0.880). However, it has the lowest specificity

of all models (0.948), meaning it has more difficulty to accurately classify negative (non existing) interactions. Nonetheless, the majority of the input is obtained from the CNN model, 768 protein and SMILES deep representations and 32 descriptors deep representations, which proves again the capability of this deep feature extractor model. Moreover, it reinforces the fact that using end-to-end deep learning approaches result in better performance overall.

Shortcomings

The results achieved with the proposed model, when compared to traditional machine learning methods and the conventional use of global descriptors, indicate a notably better performance, leading to significant confidence in its capacity and efficiency in the prediction of DTIs. Nevertheless, there are some points that can be further improved and in which the model struggles to achieve maximum effectiveness.

The main purpose of this work was to evaluate the capacity of CNNs to automatically identify important sequential and structural regions and extract useful representations from 1D raw data for DTI prediction and also to compare the performance against using global and conventional descriptors. Therefore, it was essentially to select a valid dataset that was used in a DTI prediction study based on descriptors and machine learning and also that the drug pairs of the training and testing set had a low similarity between them, to ensure the discriminatory power of the model. However, deep learning architectures substantially improve with the amount of data given to train, enabling the discovery of more and potential hidden relationships and patterns. On that account, the size of the dataset used to train could be improved. Nonetheless, the results are surprisingly good given the size and the class disparity of the dataset.

Usually, data is divided into three datasets, training, validation and testing sets. However, we had to use the hyperparameter optimization approach of Section 4.5 due to the fact that dividing the training set into training and validation led to high scores for every model architecture and set of parameters in both training and validation and the results were inconsistent when applied to the testing set. Even though the training and testing are completely independent and with low similarity between the drug pairs that composed them, there is no external validation set, which can lead to the idea that model is "overfitting" the testing data. Still, we believe that in order to validate and evaluate this approach and its capacity to

predict DTIs, the method used for hyperparameter optimization was a valid choice, given the facts considered.

Lastly, the proposed approach is a binary classification model, which only classifies the interactions as positive or negative. Thus, it does not give any information about the binding affinity or binding strength of the interactions and hence not being specific to the chemical nature of the interaction. However, the main goal was to validate the effectiveness of using sequential and structural data combined and also the capacity of CNNs to automatically extract meaningful deep representations of these type of data in the prediction of DTIs. On that account, given the results obtained, we believe that the goal was fulfilled.

7

Conclusion

We proposed an end-to-end deep learning approach for drug-target interaction prediction, capable of automatically feature (deep representations) extraction from 1D sequential and structural raw data, protein sequences and SMILES strings, using two parallel CNNs. We compared the performance of this model with traditional machine learning methods, RF and SVM, using both descriptors and deep representations, a deep learning approach based on a FCNN with global descriptors as the input and also an CNN, autoencoder and FCNN combined model, that uses as input a combination of deep sequential representations obtained from the CNNs and deep descriptors representations from the autoencoder. Our approach yielded better results in the correct classification of both positive and negative interactions, demonstrating its viability for practical use.

Deep learning has shown an overwhelming success in many classification studies for its capacity to learn deep hidden patterns from the data. Additionally, our model illustrates the remarkable ability of applying these approaches, specifically CNNs, to automatically extract deep representations, identified as local patterns or dependencies, and use them to describe drug-target interactions. The results obtained showed that using these representations outperformed completely global descriptors in every model applied, demonstrating the importance and relevance of the features extracted and also the capacity to identify and learn particular sequential and structural regions meaningful to the interaction. Nonetheless, deep learning does not always surpass traditional machine learning approaches, as demonstrated when comparing the FCNN model and RF.

In addition, we also evaluated the influence of particular descriptors, encoded into deep representations, combined with sequential and structural deep representations extracted from protein sequences and SMILES strings. The results demonstrated that additional information may prove to be useful to correctly identify positive interactions, as this model obtained the highest sensitivity (0.880).

The preeminent contribution of this master thesis and work is the proposal of a novel end-to-end deep learning approach for drug-target interaction prediction, solely based on the use of sequential and structural information to represent the proteins and drugs, respectively, without depending on several methodologies of feature engineering and extraction.

Future Work

The ability of the models to learn and identify potential DTIs and leads is considerably based on the datasets given as input. On that account, as future work, would be interesting to validate the whole proposed setup on bigger and more diverse and representative datasets, and explore the potential identified leads. On top of that, having more data would increase the capacity of the CNNs to learn and identify more hidden relationships in the data meaningful to the interaction between a protein and a drug, promoting the identification of more and new relationships.

The findings associated with the influence of particular descriptors demonstrated that integrating more information might be useful for the correct classification of positive interactions. Hence, building an effective ensemble of meaningful information for interaction, to be further integrated in the proposed end-to-end deep learning model, could yield appealing results.

The proposed end-to-end deep learning setup is identified as a binary classification model, thus transforming it into a multi-class classification model could be interesting to evaluate its capacity on identifying several types of specific interactions. Plus, modifying it into a regression model, with Kd as the binding affinity metric, could provide exciting results and heighten the whole effectiveness of the model.

The representation of the data when using sequential and structural data plays an important role in the whole performance of the model. Thence, exploring the use of the binding sites, which are identified as the regions of a protein that actively interact, instead of the whole sequences, could further validate the usefulness of the proposed model as well as be integrated on it.

Ultimately, the future approach that could be linked and integrated with this setup and have the greatest impact on the whole process, would be the ability to move backwards, resulting in the capacity to verify which regions of the protein sequences and SMILES strings had more influence and henceforth use that information to exploit possible modifications on the compounds to achieve certain properties.

Finally, this setup could be integrated in an ensemble pipeline, capable of using only 1D sequential and structural data to identify and validate DTIs.

Bibliography

[1] E. A. Ponomarenko, E. V. Poverennaya, E. V. Ilgisonis, M. A. Pyatnitskiy, A. T. Kopylov, V. G. Zgoda, A. V. Lisitsa, and A. I. Archakov, "The size of the human proteome: The width and depth," *International journal of analytical chemistry*, vol. 2016, pp. 7436849–7436849, 2016. 27298622[pmid].

[2] J. M. Jez, "Revisiting protein structure, function, and evolution in the genomic era," *Journal of Invertebrate Pathology*, vol. 142, pp. 11–15, 2017.

[3] R. Bhattacharya, P. W. Rose, S. K. Burley, and A. Prlic, "Impact of genetic variation on three dimensional structure and function of proteins," *PLOS ONE*, vol. 12, p. e0171355, Mar 2017.

[4] J. Drews, "Drug discovery: A historical perspective," *Science*, vol. 287, p. 1960, Mar 2000.

[5] D. G. Lambert, "Drugs and receptors," *BJA Education*, vol. 4, pp. 181–184, Dec 2004.

[6] S. Sengupta, M. Chattopadhyay, and H.-P. Grossart, "The multifaceted roles of antibiotics and antibiotic resistance in nature," *Frontiers in Microbiology*, vol. 4, p. 47, 2013.

[7] R. Gaynes, "The discovery of penicillin–new insights after more than 75 years of clinical use," *Emerging Infectious Diseases*, vol. 23, pp. 849–853, May 2017. PMC5403050[pmcid].

[8] S. S. Kadam, K. R. Mahadik, and K. G. Bothara, *Principles of Medicinal Chemistry*, vol. II. Nirali Prakashan, 2008.

[9] R. C. Mohs and N. H. Greig, "Drug discovery and development: Role of basic biological research," *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, vol. 3, no. 4, pp. 651–657, 2017.

[10] J. Krasner, "Drug-protein interaction," *Pediatric Clinics of North America*, vol. 19, no. 1, pp. 51–63, 1972.

[11] F. Guo and L. Wang, "Computing the protein binding sites," *BMC bioinformatics*, vol. 13 Suppl 10, pp. S2–S2, Jun 2012. 22759425[pmid].

[12] D. S. Hage, A. Jackson, M. R. Sobansky, J. E. Schiel, M. J. Yoo, and K. S. Joseph, "Characterization of drug-protein interactions in blood using high-performance affinity chromatography," *Journal of Separation Science*, vol. 32, pp. 835–853, Mar 2009.

[13] I. M. Gould and A. M. Bal, "New antibiotic agents in the pipeline and how they can help overcome microbial resistance," *Virulence*, vol. 4, pp. 185–191, Feb 2013.

[14] C. L. Ventola, "The antibiotic resistance crisis: part 1: causes and threats," *P & T : a peer-reviewed journal for formulary management*, vol. 40, pp. 277–283, Apr 2015. 25859123[pmid].

[15] B. Aslam, W. Wang, M. I. Arshad, M. Khurshid, S. Muzammil, M. H. Rasool, M. A. Nisar, R. F. Alvi, M. A. Aslam, M. U. Qamar, M. K. F. Salamat, and Z. Baloch, "Antibiotic resistance: a rundown of a global crisis," *Infect Drug Resist*, vol. 11, pp. 1645–1658, Oct 2018. 30349322[pmid].

[16] S. M. Paul, D. S. Mytelka, C. T. Dunwiddie, C. C. Persinger, B. H. Munos, S. R. Lindborg, and A. L. Schacht, "How to improve R&D productivity: the pharmaceutical industry's grand challenge," *Nature Reviews Drug Discovery*, vol. 9, pp. 203 EP –, Feb 2010.

[17] T. T. Ashburn and K. B. Thor, "Drug repositioning: identifying and developing new uses for existing drugs," *Nature Reviews Drug Discovery*, vol. 3, pp. 673 EP –, Aug 2004. Review Article.

[18] J. T. Dudley, T. Deshpande, and A. J. Butte, "Exploiting drug-disease relationships for computational drug repositioning," *Briefings in Bioinformatics*, vol. 12, pp. 303–311, Jun 2011.

[19] H. Zhou, M. Gao, and J. Skolnick, "Comprehensive prediction of drug-protein interactions and side effects for the human proteome," *Scientific Reports*, vol. 5, pp. 11090 EP –, Jun 2015. Article.

[20] R. Chen, X. Liu, S. Jin, J. Lin, and J. Liu, "Machine learning for drug-target interaction prediction," *Molecules*, vol. 23, no. 9, 2018.

[21] M. J. Keiser, B. L. Roth, B. N. Armbruster, P. Ernsberger, J. J. Irwin, and B. K. Shoichet, "Relating protein pharmacology by ligand chemistry," *Nature Biotechnology*, vol. 25, pp. 197 EP –, Feb 2007.

[22] R. Perkins, H. Fang, W. Tong, and W. Welsh, *Quantitative structure-activity relationship methods: Perspectives on drug discovery and toxicology*, vol. 22 of *Environmental toxicology and chemistry / SETAC*. Sep 2003.

[23] H. González-Díaz, F. Prado-Prado, X. García-Mera, N. Alonso, P. Abeijón, O. Caamaño, M. Yáñez, C. R. Munteanu, A. Pazos, M. A. Dea-Ayuela, M. T. Gómez-Muñoz, M. M. Garijo, J. Sansano, and F. M. Ubeira, "MIND-BEST: Web server for drugs and target discovery; design, synthesis, and assay of MAO-B inhibitors and theoretical-experimental study of G3PDH protein from trichomonas gallinae," *Journal of Proteome Research*, vol. 10, pp. 1698–1718, Apr 2011.

[24] F. Cheng, Y. Zhou, J. Li, W. Li, G. Liu, and Y. Tang, "Prediction of chemical-protein interactions: multitarget-QSAR versus computational chemogenomic methods," *Molecular BioSystems*, vol. 8, no. 9, pp. 2373–2384, 2012.

[25] S. Forli, R. Huey, M. E. Pique, M. F. Sanner, D. S. Goodsell, and A. J. Olson, "Computational protein-ligand docking and virtual drug screening with the autodock suite," *Nature Protocols*, vol. 11, pp. 905 EP –, Apr 2016.

[26] M. Kurcinski, M. Jamroz, M. Blaszczyk, A. Kolinski, and S. Kmiecik, "Cabs-dock web server for the flexible docking of peptides to proteins without prior knowledge of the binding site," *Nucleic Acids Research*, vol. 43, pp. W419–W424, May 2015.

[27] G. Pujadas, M. Vaqué, A. Ardèvol, C. Bladé, M.-J. Salvadó, M. Blay, J.-B. Fernandez-Larrea, and L. Arola, *Protein-ligand Docking: A Review of Recent Advances and Future Perspectives*, vol. 4 of *Current Pharmaceutical Analysis*. Feb 2008.

[28] N. S. Pagadala, K. Syed, and J. Tuszynski, "Software for molecular docking: a review," *Biophysical reviews*, vol. 9, pp. 91–102, Jan 2017. 28510083[pmid].

[29] M. L. Teodoro, G. Phillips, and L. Kavraki, *Molecular Docking: A Problem With Thousands Of Degrees Of Freedom*, vol. 1 of *Proceedings - IEEE International Conference on Robotics and Automation*. Sep 2002.

[30] A. Waterhouse, M. Bertoni, S. Bienert, G. Studer, G. Tauriello, R. Gumienny, F. T. Heer, T. A. P. de Beer, C. Rempfer, L. Bordoli, R. Lepore, and T. Schwede,

"Swiss-model: homology modelling of protein structures and complexes," *Nucleic acids research*, vol. 46, pp. W296–W303, Jul 2018. 29788355[pmid].

[31] H. Zhang, H. Li, H. Jiang, J. Shen, K. Chen, K. Yang, K. Yu, L. Kang, W. Zhu, X. Luo, X. Wang, and Z. Gao, "TarFisDock: a web server for identifying drug targets with docking approach," *Nucleic Acids Research*, vol. 34, pp. W219–W224, Jul 2006.

[32] A. C. Cheng, R. G. Coleman, K. T. Smyth, Q. Cao, P. Soulard, D. R. Caffrey, A. C. Salzberg, and E. S. Huang, "Structure-based maximal affinity model predicts small-molecule druggability," *Nature Biotechnology*, vol. 25, pp. 71 EP –, Jan 2007.

[33] L. Yang, K. Wang, J. Chen, A. G. Jegga, H. Luo, L. Shi, C. Wan, X. Guo, S. Qin, G. He, G. Feng, and L. He, "Exploring off-targets and off-systems for adverse drug reactions via chemical-protein interactome – clozapine-induced agranulocytosis as a case study," *PLOS Computational Biology*, vol. 7, p. e1002016, Mar 2011.

[34] A. Ezzat, M. Wu, X.-L. Li, and C.-K. Kwoh, "Computational prediction of drug-target interactions using chemogenomic approaches: an empirical survey," *Briefings in Bioinformatics*, Jan 2018.

[35] A. Gutteridge, M. Araki, M. Kanehisa, W. Honda, and Y. Yamanishi, "Prediction of drug-target interaction networks from the integration of chemical and genomic spaces," *Bioinformatics*, vol. 24, pp. i232–i240, Jul 2008.

[36] F. Cheng, C. Liu, J. Jiang, W. Lu, W. Li, G. Liu, W. Zhou, J. Huang, and Y. Tang, "Prediction of drug-target interactions and drug repositioning via network-based inference," *PLOS Computational Biology*, vol. 8, p. e1002503, May 2012.

[37] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, p. 255, Jul 2015.

[38] Z. Hira and D. F. Gillies, *A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data*, vol. 2015 of *Advances in Bioinformatics*. Jul 2015.

[39] M. C. Cobanoglu, C. Liu, F. Hu, Z. N. Oltvai, and I. Bahar, "Predicting drug-target interactions using probabilistic matrix factorization," *Journal of Chemical Information and Modeling*, vol. 53, pp. 3399–3409, Dec 2013.

[40] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[41] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[42] N. Nagamine and Y. Sakakibara, "Statistical prediction of protein-chemical interactions based on chemical structure and mass spectrometry data," *Bioinformatics*, vol. 23, pp. 2004–2012, May 2007.

[43] K. Bleakley and Y. Yamanishi, "Supervised prediction of drug-target interactions using bipartite local models," *Bioinformatics*, vol. 25, pp. 2397–2403, Jul 2009.

[44] H. Yu, J. Chen, X. Xu, Y. Li, H. Zhao, Y. Fang, X. Li, W. Zhou, W. Wang, and Y. Wang, "A systematic prediction of multiple drug-target interactions from chemical, genomic, and pharmacological data," *PLOS ONE*, vol. 7, p. e37608, May 2012.

[45] D.-S. Cao, L.-X. Zhang, G.-S. Tan, Z. Xiang, W. Zeng, Q. Xu, and A. Chen, *Computational Prediction of DrugTarget Interactions Using Chemical, Biological, and Network Features*, vol. 33 of *Molecular Informatics*. Oct 2014.

[46] R. Nayak, L. C. Jain, and B. K. H. Ting, *Computational Mechanics-New Frontiers for the New Millennium*, ch. Artificial Neural Networks in Biomedical Engineering: A Review, pp. 887–892. Oxford: Elsevier, 2001.

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436 EP –, May 2015.

[48] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, "Applications of deep learning in biomedicine," *Molecular Pharmaceutics*, vol. 13, pp. 1445–1454, May 2016.

[49] K. Tian, M. Shao, Y. Wang, J. Guan, and S. Zhou, "Boosting compound-protein interaction prediction by deep learning," *Methods*, vol. 110, pp. 64–72, 2016.

[50] Peng-Wei, K. Chan, and Z.-H. You, *Large-scale prediction of drug-target interactions from deep representations*. Jul 2016.

[51] M. Wen, Z. Zhang, S. Niu, H. Sha, R. Yang, Y. Yun, and H. Lu, "Deep-learning-based drug-target interaction prediction," *Journal of Proteome Research*, vol. 16, pp. 1401–1409, Apr 2017.

[52] L. Xie, S. He, X. Song, X. Bo, and Z. Zhang, "Deep learning-based transcriptome data classification for drug-target interaction prediction," *BMC Genomics*, vol. 19, no. 7, p. 667, 2018.

[53] T. B. Hashimoto, D. Alvarez-Melis, and T. S. Jaakkola, *Word Embeddings as Metric Recovery in Semantic Spaces*, vol. 4 of *Transactions of the Association for Computational Linguistics*. Dec 2016.

[54] C.-Y. Yu, L.-C. Chou, and D. T.-H. Chang, "Predicting protein-protein interactions in unbalanced data using the primary structure of proteins," *BMC Bioinformatics*, vol. 11, no. 1, p. 167, 2010.

[55] Y. Lecun, K. Kavukcuoglu, and C. Farabet, *Convolutional Networks and Applications in Vision*. ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems, May 2010.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, vol. 25 of *Neural Information Processing Systems*. Jan 2012.

[57] L. Kang, P. Ye, Y. Li, and D. Doermann, *Convolutional Neural Networks for No-Reference Image Quality Assessment*. Jun 2014.

[58] K. Üreten, H. Erbay, and H. H. Maras, "Detection of rheumatoid arthritis from hand radiographs using a convolutional neural network," *Clinical Rheumatology*, 2019.

[59] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford, "Convolutional neural network architectures for predicting dna-protein binding," *Bioinformatics*, vol. 32, pp. i121–i127, Jun 2016. 27307608[pmid].

[60] H. Öztürk, A. Özgür, and E. Ozkirimli, "DeepDTA: deep drug-target binding affinity prediction," *Bioinformatics*, vol. 34, pp. i821–i829, Sep 2018.

[61] S. Budach and A. Marsico, "pysster: classification of biological sequences by learning sequence and structure motifs with convolutional neural networks," *Bioinformatics*, vol. 34, pp. 3035–3037, Apr 2018.

[62] S. Kwon and S. Yoon, "End-to-end representation learning for chemical-chemical interaction prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1, 2018.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[64] E. D. Coelho, J. P. Arrais, and J. L. Oliveira, "Computational discovery of putative leads for drug repositioning through drug-target interaction prediction," *PLOS Computational Biology*, vol. 12, pp. 1–17, 11 2016.

[65] A. C. Guo, B. Gautam, C. Knox, D. Tzur, D. Cheng, M. Hassanali, S. Shrivastava, and D. S. Wishart, "DrugBank: a knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Research*, vol. 36, pp. D901–D906, Nov 2007.

[66] J. Yang, A. Roy, and Y. Zhang, "BioLiP: a semi-manually curated database for biologically relevant ligand-protein interactions," *Nucleic Acids Res*, vol. 41, pp. D1096–D1103, Jan 2013. 23087378[pmid].

[67] T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson, "BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities," *Nucleic Acids Res*, vol. 35, pp. D198–D201, Jan 2007. 17145705[pmid].

[68] E. C. Hulme and M. A. Trevethick, "Ligand binding assays at equilibrium: validation and interpretation," *British Journal of Pharmacology*, vol. 161, pp. 1219–1237, Nov 2010.

[69] T. U. Consortium, "UniProt: the universal protein knowledgebase," *Nucleic Acids Research*, vol. 45, pp. D158–D169, Nov 2016.

[70] S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, J. Wang, B. Yu, J. Zhang, and S. H. Bryant, "PubChem substance and compound databases," *Nucleic Acids Res*, vol. 44, pp. D1202–D1213, Jan 2016. 26400175[pmid].

[71] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic Acids Res*, vol. 28, pp. 235–242, Jan 2000. 10592235[pmid].

[72] M. Kanehisa and S. Goto, "KEGG: kyoto encyclopedia of genes and genomes," *Nucleic Acids Res*, vol. 28, pp. 27–30, Jan 2000. 10592173[pmid].

[73] J. J. Irwin and B. K. Shoichet, "ZINC–a free database of commercially available compounds for virtual screening," *J Chem Inf Model*, vol. 45, no. 1, pp. 177–182, 2005. 15667143[pmid].

[74] T. Sterling and J. J. Irwin, "ZINC 15–ligand discovery for everyone," *J Chem Inf Model*, vol. 55, pp. 2324–2337, Nov 2015. 26479676[pmid].

[75] W. Gilpin, "PyPDB: a Python API for the protein data bank," *Bioinformatics*, vol. 32, pp. 159–160, Sep 2015.

[76] D. Pultz, J. Serra-Musach, J. Saez-Rodriguez, L. M. Harder, and T. Cokelaer, "BioServices: a common Python package to access biological web services programmatically," *Bioinformatics*, vol. 29, pp. 3241–3242, Sep 2013.

[77] M. Swain *et al.*, "PubChemPy." `https://github.com/mcs07/PubChemPy`, 2014.

[78] D.-S. Cao, Y.-Z. Liang, J. Yan, G.-S. Tan, Q.-S. Xu, and S. Liu, "PyDPI: Freely available Python package for chemoinformatics, bioinformatics, and chemogenomics studies," *Journal of Chemical Information and Modeling*, vol. 53, pp. 3086–3096, Nov 2013.

[79] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[80] F. Chollet *et al.*, "Keras." `https://github.com/fchollet/keras`, 2015.

[81] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, and X. Zhang, *TensorFlow: A system for large-scale machine learning*. May 2016.

[82] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, and G. Louppe, *Scikit-learn: Machine Learning in Python*, vol. 12 of *Journal of Machine Learning Research*. Jan 2012.

[83] K. Heikamp and J. Bajorath, *Support vector machines for drug discovery*, vol. 9 of *Expert opinion on drug discovery*. Dec 2013.