People's Democratic Republic of Algeria
الجمهورية الجزائرية الديمقراطية الشعبية
Ministry of Higher Education and Scientific Research
وزارة التعليم العالي و البحث العلمي



جامعة يحي فارس بالمدية
University Yahia Fares of Medea

Faculty of Science
Department of Mathematics and Computer Science (MI)

Graduation Project

For The Master's Degree In Computer Science

Specialty: System Engineering And Web Technology

Ligand-Based Virtual Screening Using Deep Learning Attention Mechanism and Learning Representation

Realized by:
Mr. AMRI Younes
Mr. AKKACHE Lounis

Supervised by:
Dr. HENTABLI Hemza
Dr. MESSAOUD Rahim

Promotion: 2022/2023

ACKNOWLEDGMENTS

In the introduction of this thesis, we offer Allah, the Most Gracious, the Most Merciful, our sincere gratitude and thanks for guiding us and giving us the courage and patience we needed during these protracted years of study. We would like to express our profound gratitude to all of our instructors, especially our advisers from Yahia Fares University, Dr.Hentabli Hamza and Dr.Messaoud Rahim. Our understanding and academic development have been nurtured along this difficult journey by their great expertise, advice, and dedication.

We want to convey our sincere gratitude to each and every jury member for their readiness to examine and assess our graduation project. Their intelligent recommendations and proposals have significantly improved this thesis, increasing its value and quality.

Our families have been our pillars of strength throughout this protracted academic journey, and we are incredibly grateful for their consistent encouragement, tolerance, and inspiration. They have continuously encouraged us and served as our haven when faced with difficulties and uncertainties.

We want to express our sincere gratitude to everyone who helped finish this effort, whether they were directly involved or not. May Allah reward them with success in everything they do.

We also want to take this opportunity to thank our instructors, who have kindly shared their wisdom with us over these formative years. Their instruction has been essential in guiding our intellectual growth and preparing us for the challenges ahead.

We would like to close by expressing our sincere gratitude to everyone who has assisted us in our academic endeavors. Their advice, assistance, and contributions were crucial in the development of this thesis

Dedication

In the name of Allah, the Most Gracious, the Most Merciful.

I begin by expressing my deepest gratitude to Allah, the Lord of all worlds, for granting me strength, guidance, and blessings throughout this academic journey. Without His infinite mercy and unwavering support, this thesis would not have been possible.

I would like to dedicate this thesis to my loving mother and father, whose unconditional love, unwavering support, and endless sacrifices have been the cornerstone of my life. Your belief in my abilities, constant encouragement, and wise counsel have propelled me forward, filling my heart with gratitude. I am forever indebted to you for instilling in me the values of perseverance, determination, and resilience.

To my dear brothers and sisters, your presence and unwavering support have been a constant source of strength and inspiration. Your words of encouragement, late-night discussions, and shared laughter have made this journey all the more meaningful. I am grateful for your unwavering belief in my potential and your constant prayers.

I extend my sincere appreciation, to my instructors at the University, for their exceptional guidance, knowledge, and expertise. Their dedication to teaching, insightful feedback, and continuous support have played a crucial role in shaping my academic and professional development. I am grateful for their mentorship, which has had a profound impact on my intellectual growth and the successful completion of this thesis.

Furthermore, I extend my appreciation to my colleague and friend AMRI Younes, and all my friends, classmates, and family members who have supported me in countless ways. Your words of encouragement, understanding, and motivation have been invaluable throughout this challenging journey.

May Allah bless my parents, my family, and all those mentioned above, as well as anyone else who has played a part in my academic and personal growth. May He continue to guide me and grant me success in all my future endeavors.

All praise be to Allah, the Lord of all worlds.

Dedication

I would like to express my utmost gratitude and sincere appreciation to Allah for granting me strength, guidance, and inspiration throughout my academic journey. Without His blessings, this achievement would not have been possible.

I am deeply indebted to my parents for their unwavering support, love, and encouragement. Their sacrifices and belief in my abilities have been the driving force behind my pursuit of knowledge. I am forever grateful for their guidance and for instilling in me the values of perseverance and dedication.

My sister has been an incredible source of inspiration and motivation. Her unwavering faith in my abilities and her constant encouragement has been instrumental in my success. I am truly fortunate to have her unwavering support.

I would like to express my heartfelt appreciation to my dear friend, AKKACHE Lounis. His friendship, intellectual discussions, and continuous support have played a significant role in shaping my research ideas and enhancing my understanding of the subject matter.

Furthermore, I extend my gratitude to all my friends who have stood by me throughout this journey. Your friendship, encouragement, and positive energy have created a conducive environment for personal and academic growth. Your unwavering belief in my abilities has been a constant source of motivation.

I would also like to express my sincere appreciation to my academic advisors and professors for their guidance, knowledge, and expertise. Their valuable insights, constructive feedback, and intellectual mentorship have been pivotal in shaping the trajectory of my research.

Lastly, I extend my thanks to all those who have directly or indirectly contributed to the completion of this thesis. Your support, whether through discussions, resources, or encouragement, has played a significant role in this achievement.

I am truly humbled and grateful for the support and encouragement I have received from Allah, my parents, my sister, my friend AKKACHE Lounis, and all my friends. Their unwavering belief in my abilities has fueled my dedication and commitment to producing a quality thesis.

Abstract

Virtual screening techniques have emerged as computational methods for efficient chemical compound screening in the traditional drug production process. Ligand-based virtual screening (LBVS) has been successful in capturing biological similarity between test compounds and known target ligands, but existing methods often fall short when dealing with structurally heterogeneous molecules. Deep learning models and Siamese networks have been employed in similarity models to address this challenge.

This study aims to enhance the performance of similarity searching, specifically in the presence of structurally heterogeneous molecules, by integrating an attention mechanism into the Siamese network architecture. The inclusion of attention enhances the network's ability to capture important relationships and dependencies within the data. The proposed architecture incorporates multiple similarities distance layers to improve the measurement of similarity between molecules. Moreover, additional layers are introduced after the fusion layer in select models to enhance retrieval recall. To enhance their effectiveness the proposed architecture utilizes deep learning techniques including LSTM, CNN1D, and CNN2D. Through thorough testing on real-world datasets, we evaluated our approach's performance against existing methods.

The results of the experiments demonstrate the superiority of the proposed Siamese network architecture with the attention mechanism over existing methods in terms of similarity searching, particularly in the context of structurally heterogeneous molecules.

Keywords : Traditional drug production, virtual screening, Siamese network, attention mechanism, deep learning, structurally heterogeneous molecules, similarity searching.

Résumé

Les techniques de criblage virtuel sont apparues comme des méthodes informatiques permettant un criblage efficace des composés chimiques dans le cadre du processus traditionnel de production de médicaments. Le criblage virtuel basé sur les ligands (LBVS) a réussi à saisir la similarité biologique entre les composés testés et les ligands cibles connus, mais les méthodes existantes sont souvent insuffisantes lorsqu'il s'agit de molécules structurellement hétérogènes. Les modèles d'apprentissage profond et les réseaux Siamese ont été utilisés dans les modèles de similarité pour relever ce défi.

Cette étude vise à améliorer les performances de la recherche de similarité, en particulier en présence de molécules structurellement hétérogènes, en intégrant un mécanisme d'attention dans l'architecture du réseau Siamese. L'intégration d'un mécanisme d'attention améliore la capacité du réseau à capturer les relations et dépendances importantes dans les données. L'architecture proposée intègre plusieurs couches de distance de similarité afin d'améliorer la mesure de la similarité entre les molécules. En outre, des couches supplémentaires sont introduites après la couche de fusion dans certains modèles afin d'améliorer le rappel de la recherche. Pour améliorer son efficacité, l'architecture proposée utilise des techniques d'apprentissage en profondeur, notamment LSTM, CNN1D et CNN2D. Grâce à des tests approfondis sur des ensembles de données réels, nous avons évalué les performances de notre approche par rapport aux méthodes existantes.

Les résultats des expériences démontrent la supériorité de l'architecture de réseau Siamese proposée avec le mécanisme d'attention par rapport aux méthodes existantes en termes de recherche de similarité, en particulier dans le contexte de molécules structurellement hétérogènes.

Mots clés : Production de médicaments traditionnels, criblage virtuel, réseau siamese, mécanisme d'attention, apprentissage profond,molécules structurellement hétérogène, recherche de similarités.

ملخص

ظهرت تقنيات الفحص الافتراضية كطرق حسابية لفحص المركبات الكيميائية الفعالة في عملية إنتاج الأدوية التقليدية. نجح الفحص الافتراضي القائم على Ligand (LBVS) في التقاط التشابه البيولوجي بين مركبات الاختبار والرابطات المستهدفة المعروفة ، ولكن الأساليب الحالية غالبًا ما تقصر عند التعامل مع الجزيئات غير المتجانسة من الناحية الهيكلية. تم استخدام نماذج التعلم العميق والشبكات السيامية في نماذج التشابه لمواجهة هذا التحدي.

تهدف هذه الدراسة إلى تعزيز أداء البحث عن التشابه ، وتحديداً في وجود جزيئات غير متجانسة هيكلياً ، من خلال دمج آلية الانتباه الذاتي في بنية الشبكة السيامية. يعزز إدراج الاهتمام الذاتي من قدرة الشبكة على التقاط العلاقات والتبعيات المهمة داخل البيانات. تتضمن البنية المقترحة العديد من طبقات المسافات المتشابهة لتحسين قياس التشابه بين الجزيئات. علاوة على ذلك ، يتم إدخال طبقات إضافية بعد طبقة الاندماج في نماذج مختارة لتحسين استدعاء الاسترجاع. لتعزيز فعاليتها ، تستخدم البنية المقترحة تقنيات التعلم العميق بما في ذلك LSTM و CNN1D و .CNN2D من خلال الاختبار الشامل على مجموعات البيانات الواقعية ، قمنا بتقييم أداء نهجنا مقابل الأساليب الحالية.

تظهر نتائج التجارب تفوق بنية الشبكة السيامية المقترحة مع آلية الانتباه على الطرق الحالية من حيث البحث عن التشابه ، لا سيما في سياق الجزيئات غير المتجانسة هيكليًا.

كلمات مفتاحية :

إنتاج الأدوية التقليدي ، الفحص الافتراضي ، الشبكة السيامية ، آلية الانتباه ، التعلم العميق ، الجزيئات غير المتجانسة هيكليًا ، البحث عن التشابه.

Contents

Acknowledgement . . . . . **I**

Dedication . . . . . **II**

Dedication . . . . . **III**

Abstract . . . . . **IV**

Résumé . . . . . **V**

VI . . . . . **ملخص**

1 Introduction **1**
- 1.1 Overview 1
- 1.2 Problem Background 2
- 1.3 Problem Statement 3
- 1.4 Research Questions 3
- 1.5 Research Objectives 5
- 1.6 Research Scoop 5
- 1.7 Document Organization 6

2 Computer-Aided Drug Design **8**
- 2.1 Introduction 8
- 2.2 Definition 8
- 2.3 Chemoinformatics 10
- 2.4 Molecular Representation 11
  - 2.4.1 String representation 12
  - 2.4.2 Molecular Fingerprints 13
  - 2.4.3 Molecular descriptors 16
  - 2.4.4 Graph-based representations 18
  - 2.4.5 3D Molecular Representations 19
- 2.5 Molecular similarity measurement 20
  - 2.5.1 The structure representation 21
  - 2.5.2 The weighting scheme 22

2.5.3 Similarity coefficient . . . 23
2.6 Virtual screening . . . 25
2.6.1 Ligand-Based virtual screening . . . 26
2.6.2 Structure-Based virtual screening . . . 27
2.7 Related works . . . 28
2.8 Summary . . . 30

3 Deep learning . . . 31
3.1 Introduction . . . 31
3.2 Artificial Intelligence (AI) . . . 31
3.3 Machine learning (ML) . . . 33
3.4 Deep learning (DL) . . . 35
3.4.1 Feedforward neural networks (FNN) . . . 35
3.4.2 Convolutional Neural Networks (CNN) . . . 36
3.4.3 Recurrent Neural Networks (RNN) . . . 39
3.5 Attention Mechanism . . . 43
3.5.1 How Attention Mechanism Works . . . 44
3.5.2 Types of Attention Mechanism . . . 44
3.5.3 Applications of Attention Mechanism . . . 53
3.6 Siamese Networks . . . 54
3.7 Summary . . . 56

4 Research Methodology . . . 57
4.1 Introduction . . . 57
4.2 Research Framework . . . 57
4.3 Datasets . . . 58
4.4 Design And Steps . . . 59
4.4.1 Dataset Preprocessing . . . 59
4.4.2 Construction of Features Extractor model . . . 60
4.4.3 Design architecture of our Siamese attentional models . . . 61
4.5 Performance and Evaluation Measures . . . 68
4.5.1 Procedure of Similarity Search in Virtual Screening . . . 71
4.6 Benchmarking . . . 71
4.7 Summary . . . 74

5 Results and Discussion . . . 75
5.1 Development environment . . . 75
5.1.1 Python . . . 75
5.1.2 Python features . . . 76
5.1.3 Python module used . . . 77

5.1.4 Visual Studio Code 81
5.1.5 Google colaboratory 81
5.1.6 Google Drive 82
5.1.7 Jupyter notebooks 83
5.1.8 Hardware used 85
5.2 Evaluation of proposed Models 86
5.2.1 Result Model SSCNN1D 86
5.2.2 Result Model SSCNN2D 90
5.2.3 Summary Result of training and validation 92
5.2.4 Evaluation Test Set 93
5.3 Result of Ligand-Based Virtual Screening 95
5.3.1 MDDR-DS1 95
5.3.2 MDDR-DS2: 98
5.3.3 MDDR-DS3 100
5.3.4 Summary Result Ligand-Based virtual screening 103
5.4 Summary 103

6 General conclusions and perspectives 105
6.1 Conclusion 105
6.2 Contribution 106
6.3 Futur work 106

List of Figures

2.1 Computer-aided drug designing types and subtypes [32]. 9
2.2 Applications Of Cheminformatics [100]. 10
2.3 Molecular representation in cheminformatics [62]. 11
2.4 structure and string representation (SMILES) of vanillin [6]. 12
2.5 Circular patterns encoded as fingerprints [26]. 14
2.6 Representing Maccs Keys Structure [24]. 15
2.7 ECFP generation process [102]. 16
2.8 Generation of the fixed-length bit string [102]. 16
2.9 Descriptor classes [53]. 17
2.10 Caffeine graph representation [43]. 19
2.11 3D Molecular Model of Caffeine [14]. 20
2.12 Relationship between molecules A & B features [69]. 24
2.13 Virtual screening strategies. 25
2.14 Classification of ligand-based approaches. 27

3.1 Relationship between artificial intelligence machine learning neural network and deep learning [68]. 33
3.2 Diagramme d'apprentissage automatique [89]. 34
3.3 Typical architecture of a feed-forward neural network (FFN) 36
3.4 architecture of a convolutional neural networks(CNN). 37
3.5 An example of 2-D convolution operation [58]. 38
3.6 Pooling Operation using the 2×2 filters [41]. 38
3.7 A computation graph corresponding to a simple RNN. 39
3.8 Types of Rnn [29]. 40
3.9 LSTM Cell [60]. 42
3.10 BiLSTM Cell [120]. 43
3.11 attention-matrices [88]. 46
3.12 self-attention matrix calculation [25]. 47
3.13 Calculation attention weight [88]. 48
3.14 Calculation Weighted Sum [88]. 49
3.15 Overall process for Bahdanau Attention [72]. 50
3.16 Overall process for Luong Attention [72]. 52

3.17 Multihead attention mechanism [114]. 53
3.18 Architecture of Siamese Network. 55

4.1 The general research framework. 58
4.2 pairwise dataset generation. 61
4.3 balancing all The Three dataset. 63
4.4 1D Feautre extractor. 64
4.5 The SSCNN1D similarity model architecture. 66
4.6 2D feature Extractor. 67
4.7 The SSCNN2D similarity model architecture. 68
4.8 Idea of cross-validation for training and testing data. 70
4.9 Virtual Screening Procedure. 72

5.1 Python logo. 76
5.2 Pandas logo. 78
5.3 NumPy logo. 78
5.4 Tensorflow logo. 79
5.5 Keras logo. 79
5.6 RDKit logo. 80
5.7 Visual studio code logo. 81
5.8 Google Colab logo. 82
5.9 Google Drive logo. 83
5.10 Jupyter Notebook logo. 84
5.11 Variation of training and testing on MDDR-DS1 for SSCNN1D. 87
5.12 Variation of training and testing on MDDR-DS2 for SSCNN1D. 88
5.13 Variation of training and testing on MDDR-DS3 for SSCNN1D. 89
5.14 Variation value for training and testing data in DS1 data set for model SSCNN2D. 90
5.15 Variation of training and testing on MDDR-DS2 for SSCNN2D. 91
5.16 Variation value for training and testing data in DS3 data set for model SSCNN2D. 92
5.17 Boxplot for recall result distribution for each method in MDDR-DS1 at the top 1%. 97
5.18 Boxplot for recall result distribution for each method in MDDR-DS1 at the top 5%. 98
5.19 Boxplot for recall result distribution for each method in MDDR-DS2 at the top 1%. 99
5.20 Boxplot for recall result distribution for each method in MDDR-DS2 at the top 5%. 101

5.21 Boxplot for recall result distribution for each method in MDDR DS3 at the top 1%. 102
5.22 Boxplot for recall result distribution for each method in MDDR-DS3 at the top 5%. 104

List of Tables

2.1 Different Molecular Representations. 13
2.2 Common Similarity/Distance Measures in Chemical Informatics [69]. . 24

4.1 Structure of DS1 dataset. 59
4.2 Structure of DS2 dataset. 59
4.3 Structure of DS3 dataset. 60
4.4 Architecture and parameters of SSCNN1D model. 65
4.5 Architecture and parameters of SSCNN2D model. 69
4.6 Datasets Splitting. 70

5.1 R_Square Results. 93
5.2 Test set results for SSCNN1D and SSCNN2D models on various datasets. 94
5.3 Top 1% Retrieval Results for MDDR-DS1 Data Set for Descriptor ECFC-4 96
5.4 Top 5% Retrieval Results for MDDR-DS1 Data Set for Descriptor ECFC-4 96
5.5 Top 1% Retrieval Results for MDDR-DS2 Data Set for Descriptor ECFC-4 99
5.6 Top 5% Retrieval Results for MDDR-DS2 Data Set for Descriptor ECFC-4 100
5.7 Top 1% Retrieval Results for MDDR-DS3 Data Set for Descriptor ECFC-4 101
5.8 Top 5% Retrieval Results for MDDR-DS3 Data Set for Descriptor ECFC-4 103

List of Algorithms

1 Generating pairwise dataset. 62
2 Balancing pairwise dataset. 63
3 Similarity Search in Virtual Screening. 71

List of abbreviations

LBVS	*Ligand-Based Virtual Screening*
SBVS	*Structure-Based Virtual Screening*
MDDR	*MDL Drug Data Report*
AI	*Artificial intelligence*
ML	*Machine Learning*
DL	*Deep Learning*
FNN	*Feedforward Neural Network*
CNN	*Convolutional Neural Network*
RNN	*Recurrent Neural Network*
LSTM	*Long-Short-Term-Memory*
BiLSTM	*Bidirectional LSTM*
CAAD	*Computer-Aided Drug Design*
QSAR	*Quantitative Structure-Activity Relationship*
DCNN	*Deep Convolutional Neural Network*
ECFP	*Extended-Connectivity Fingerprints*
MRL	*Molecular Representation Learning*

SVM *Support Vector Machines*

RF *Random Forests*

DBN *Deep Belief Network*

GNC *Generative Network Complex*

GRU *Gated Recurrent Unit*

SSCNN1D *Self-attention Siamese Neural Network 1D*

SSCNN2D *Self-attention Siamese Neural Network 2D*

MSE *Mean Squared Error*

TAN *Tanimoto coefficient*

BIN *Bayesian Inference Network*

SQB *Quantum similarity search*

SDBN *Stack of Deep Belief Networks*

DCNNLB *Deep Convolutional Neural Network for Ligand-Based Virtual screening*

Introduction

Chapter 1

Introduction

1.1 Overview

Drug development is a complex process that results in the production of new drugs. In the course of research and development of the drug, a biomolecular target is chosen, and high-performance screening studies are run to find the bioactive chemicals needed to achieve the desired outcomes. The creation of high-performing research testing is both costly and time-consuming. In this procedure, specialized labs with chemical and biological libraries are used [4]. In truth, the chances of success are slim; approximately 1 out of every 5000 drug candidates is expected to be accepted and widely used at some point. On the other hand, the improved computing power made it possible to screen several millions of chemical compounds affordably and quickly. Virtual screening (VS) is a computer-based method used in the discovery of pharmaceuticals to identify the structures most likely to bind to a therapeutic target, frequently a protein receptor or enzyme. This technique was used to accelerate the development of drugs.

Structure-based virtual screening (SBVS) and ligand-based virtual screening are the two main divisions of virtual screening (LBVS). The SBVS entails docking a tiny molecule into a protein's active site and assessing the resultant complex to predict the binding affinity of the chemical to the protein. On the other hand, the LBVS entails defining the salient characteristics of known active ligands, such as their size, shape, and electrostatic characteristics, and utilizing them to search a database of chemicals for substances with comparable qualities that are likely to bind to the target protein.

Modern deep learning (DL) approaches have recently been introduced in various domains, and they have advanced in recent years, providing researchers with new opportunities. The fast development of DL algorithms and the improvement of high-performance computing methodologies both contribute to the success of DL approaches. Moreover, the generalization errors of DL approaches are lower, enabling them to ob-

tain satisfactory results on specific benchmarks or competitive tests and to predict molecular attributes with greater accuracy. Deep learning methods may also be used to automatically find features from input data. In addition, Siamese networks have several uses, including text matching, face recognition, and picture identification. When there is a dearth of data or a small number of classes, they are very helpful. Siamese networks are often used for signature verification because they can be trained to tell if two signatures belong to the same person or not. Another instance is picture recognition, where the network may be trained to determine if two photographs of the same item are the same despite variations in lighting, angle, and other elements.

Many databases include data on various facets of biologically active substances. While some databases contain classes of molecules with structurally homogeneous active elements, such as the MDL Drug Data Report (MDDR DS2) data set, and other databases contain classes of molecules with structural heterogeneity, such as MDDR DS3 and Maximum Unbiased Validation (MUV), it is challenging to use simple statistical tools due to the enormous amount of information stored and complex data samples of molecules with structural heterogeneity.

In this study, the Siamese deep learning model will be enhanced by the use of attention layers for feature extraction and transmission to distance layers, which will then be combined by a fusion layer. Combining these layers makes sense in order to improve molecular similarity assessments, especially when employing several descriptors.

1.2 Problem Background

The discovery and optimization of bioactive compounds that can interact with a particular biomolecular target to elicit therapeutic effects are key steps in the difficult and expensive process of drug development. Finding compounds with comparable structural properties that can bind to the same target and have biological properties to those of known active ligands is a crucial component of drug creation. This is necessary to increase the number of prospective medication candidates and the likelihood that a drug development project will be successful.

However, due to the enormous quantity of chemicals in chemical databases, the complexity of molecule structure, and the limits of current drug design methodologies, it might be difficult to find these related molecules. Common techniques for drug design, such as those based on ligands and structures, have their own set of drawbacks and might not necessarily result in the identification of the best therapeutic candidates.

In order to overcome these challenges, there is a growing interest in developing novel computational tools that can accelerate medication development and improve the

accuracy of detecting related chemicals. Particularly deep learning technologies have the potential to completely change drug design by allowing scientists to study huge and complex datasets and make more accurate predictions about chemical interactions and characteristics. Deep learning algorithms have shown promising outcomes in a variety of domains.

The necessity for a novel approach that can use deep learning techniques to increase the accuracy of identifying structurally related compounds and quickening the drug development process is thus the issue that my capstone project intends to solve. In particular, we will create a Siamese deep learning model with attention and distance layers to improve molecular similarity measures and make it easier to find novel medication candidates.

1.3 Problem Statement

The process of identifying related compounds in drug design encounters notable challenges, as current methods exhibit limitations in accurately and efficiently recognizing compounds with potential therapeutic value. This deficiency hampers the drug discovery process, resulting in significant time and resource inefficiencies, as well as a heightened risk of failures during drug development. Addressing this critical issue is of paramount importance to reduce time, and enhance the precision, and efficacy of compound identification. Novel strategies are urgently required to overcome these challenges, enabling more accurate, efficient, and time-effective identification of related compounds in drug design. By tackling this problem, researchers can potentially optimize the drug discovery process, reduce costs, and improve the success rate of developing new drugs.

1.4 Research Questions

1. Is there evidence of the utilization of Siamese neural networks in molecular similarity assessment, and what is the performance and evaluation of this model in the context of molecular similarity? Furthermore, how does the Siamese neural network model?

2. How does the Siamese neural network attention-based deep learning model enhance the accuracy and efficiency of identifying similar molecules in ligand-based virtual screening compared to traditional methods?

3. What is the impact of incorporating Siamese neural network attention mecha-

nisms on the duration and cost of drug development in ligand-based virtual screening? How does it contribute to streamlining the drug discovery process?

4. How does the performance of the Siamese attention model compare to other state-of-the-art deep learning models in identifying related compounds in drug discovery? What are the strengths and limitations of the Siamese attention model in this context?

5. How can the Siamese attention model be adapted and modified to handle molecules with structural heterogeneity in drug design? What approaches can be employed to improve its effectiveness in dealing with diverse molecular structures?

6. What are the practical implications of implementing the Siamese attention model in real-world drug design scenarios? How does the model function when applied to complex and challenging drug discovery problems, and what are the potential benefits and challenges?

1.5 Research Objectives

The research objectives of this study are focused on advancing the field of drug design through the application of ligand-based virtual screening and deep learning techniques. The primary objective is to develop a novel Siamese network attention-based deep learning model specifically designed to identify similar molecules. By incorporating attention mechanisms, this model aims to enhance the accuracy and efficiency of virtual screening in drug design.

Another important objective is to evaluate the performance of the proposed Siamese attention-based model against existing ligand-based virtual screening methods using benchmark datasets. This comparative analysis will provide insights into the effectiveness and potential advantages of the proposed model, highlighting its contributions to the field.

In addition, the research aims to compare the efficiency and scalability of the proposed model with other state-of-the-art methods for large-scale virtual screening. This evaluation will consider computational efficiency and scalability factors, providing valuable insights for practical implementation and utilization of the proposed model in real-world scenarios.

Overall, this study aims to make significant contributions to the field of drug design by advancing the capabilities of ligand-based virtual screening through the development of a novel Siamese network attention-based deep learning model. Through comprehensive evaluations, investigations into molecular descriptors and attention mechanisms, and comparisons with existing methods, this research strives to enhance the accuracy, efficiency, and scalability of virtual screening approaches, ultimately facilitating the discovery of novel drugs.

1.6 Research Scoop

The primary scope of this research is to develop and evaluate a pioneering Siamese neural network attention-based deep learning model for the identification of similar molecules in the domain of drug design through ligand-based virtual screening. By leveraging the power of deep learning and attention mechanisms, the model aims to enhance the accuracy and efficiency of identifying potential lead compounds with desirable molecular properties. To accomplish this objective, the study will rely on a comprehensive set of data sources, primarily drawn from the MDDR (Molecular Drug Data Report) datasets. These datasets, which contain a diverse range of chemical compounds and their associated biological activities, provide a robust foundation for training and

evaluating the proposed model for identifying similarities of molecular in the domain of drug design using DL including data preprocessing model development and performance evaluation.

The core focus of this study lies in the development of the Siamese neural network attention-based deep learning model. This involves designing and training a neural network architecture that effectively captures the inherent similarities and patterns within the molecular structures, where The attention mechanism plays a pivotal role in guiding the model focus on the most relevant features and interactions, leading to more accurate predictions.

The final phase of the research entails the rigorous evaluation of the proposed model's performance. This evaluation will involve conducting extensive experiments and comparing the results against existing ligand-based virtual screening methods, benchmark datasets, and relevant performance metrics. The objective is to provide a comprehensive assessment of the model's effectiveness, highlighting its potential advantages and limitations in identifying similar molecules for drug design.

1.7 Document Organization

The thesis is organized into several chapters, each focusing on specific aspects of the application of deep learning in drug design.

Chapter 1 serves as the introduction, providing an overview of the research topic and presenting the problem background that motivates the study. The problem statement is clearly defined, highlighting the need for new strategies to enhance the precision of detecting related compounds in drug design. Additionally, research questions are posed to guide the investigation, and research objectives are outlined to direct the study's focus.

Chapter 2 explores deep learning. It begins with an introduction to deep learning, including the concepts of artificial intelligence (AI), machine learning (ML), and deep learning (DL). Attention mechanisms, an essential component of deep learning, are introduced, discussing their functionality, types, and applications. The chapter also explores the utilization of Siamese neural networks.

Moving forward, Chapter 3 delves into computer-aided drug design, offering a comprehensive exploration of its definition and the field of chemoinformatics. Various molecular representation techniques are discussed, including string representation, molecular fingerprints, molecular descriptors, graph-based representations, and 3D molecular representations. The chapter also covers the measurement of molecular similarity, detailing the structure representation, weighting schemes, and similarity co-

efficients, and provides an overview of virtual screening, focusing on both ligand-based and structure-based approaches.

Chapter 4 presents the research methodology employed in the study. It introduces the research framework and describes the dataset used for experimentation. The chapter further elaborates on the design and steps involved, including dataset processing, input compound representation, construction of deep convolutional neural network (DCNN) models, and the design architecture of the proposed Siamese attention and fusion layers models. Performance evaluation measures and the procedure for similarity search in ligand-based virtual screening are also discussed, ensuring a robust evaluation process.

Finally, Chapter 5 presents the results and discussions. It provides details about the development environment, including the programming language (Python), Python features, and relevant modules used for implementation. The evaluation of the proposed models is presented, followed by the results of ligand-based virtual screening using different datasets. The chapter aims to analyze and interpret the outcomes of the experiments, shedding light on the effectiveness and performance of the developed models.

By following this organizational structure, the thesis progresses systematically, covering key aspects of deep learning in drug design. Each chapter contributes to the overall understanding of the topic, addressing the research questions and objectives set forth in Chapter 1.

Computer-Aided Drug Design

Chapter 2

Computer-Aided Drug Design

2.1 Introduction

Computer-aided drug design has revolutionized the landscape of pharmaceutical research, offering a systematic and efficient approach to the discovery and optimization of therapeutic agents. By leveraging computational algorithms and informatics tools, this discipline enables the exploration of vast chemical space, accelerating the identification of promising drug candidates. In this chapter, we provide a concise overview of the fundamental principles and methodologies employed in computer-aided drug design. We discuss the role of chemoinformatics in molecular representation and analysis, highlight the significance of molecular similarity measurements, and explore the applications of virtual screening. Through this academic exploration, we aim to provide a solid foundation for understanding the integral role of computer-aided drug design in modern drug discovery endeavors.

2.2 Definition

Computer-aided drug design (CADD) is a technique that relies on computational methods to create and find novel drugs. This strategy utilizes diverse software tools and algorithms to examine and simulate the actions of molecules and their interactions with particular proteins or receptors. Drugs that are in use or through clinical trials today have benefited greatly from the adoption of CADD methods. Predicting whether and how strongly a given chemical will attach to a target is the objective of drug design. The two kinds of CADD techniques are ligand-based and structure-based drug discovery (see Figure 2.1).

Ligand-based drug discovery involves the use of known ligands to design new com-

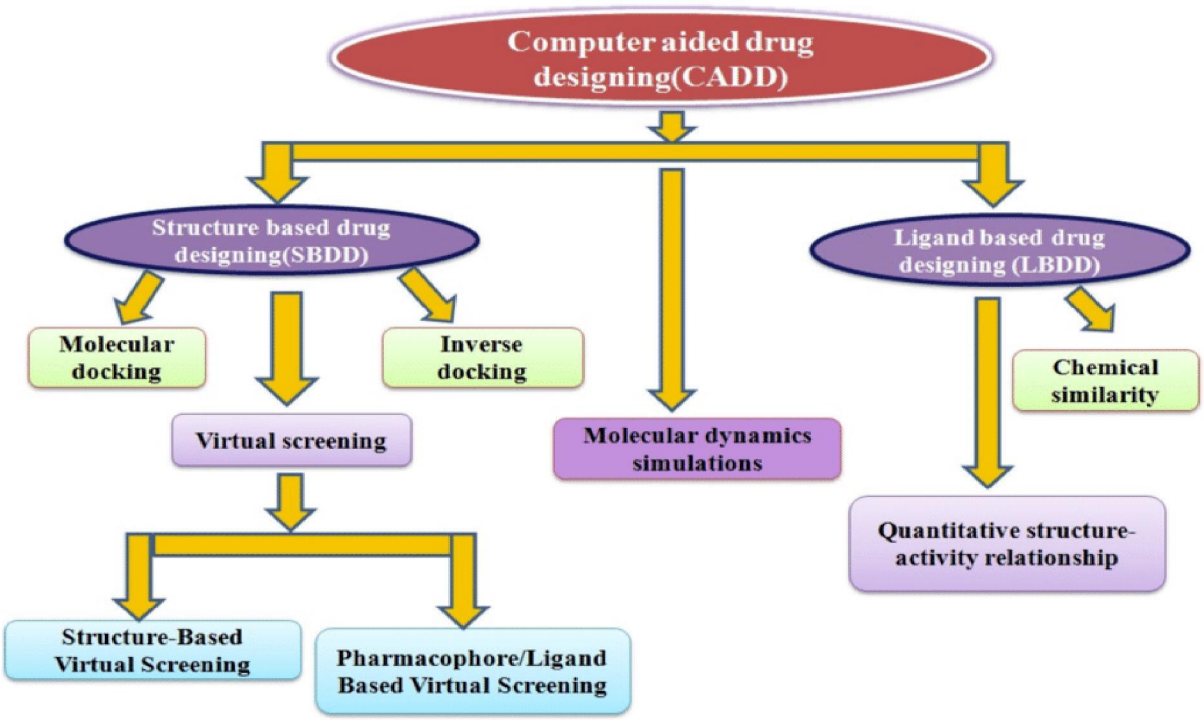


Figure 2.1: Computer-aided drug designing types and subtypes [32].

pounds with similar structures and properties. This approach is useful when the structure of the target protein is unknown[55]. Ligand-based methods include pharmacophore modeling, quantitative structure-activity relationship (QSAR), and artificial intelligence (AI) [55]. Structure-based drug discovery involves the use of the three-dimensional structure of the target protein to design new compounds that bind to it. This approach is useful when the structure of the target protein is known. Structure-based methods include molecular docking and molecular dynamic simulation[55].

Various procedures are utilized in CADD, such as molecular docking, molecular dynamics simulations, QSAR modeling, and virtual screening [8]. Molecular docking forecasts the binding manner of a small molecule drug candidate to a protein target, while molecular dynamics simulations use molecular mechanics calculations to simulate the behavior of the protein-ligand complex over time. QSAR modeling predicts a compound's activity based on its chemical configuration, and virtual screening is utilized to screen enormous databases of compounds to identify promising drug candidates [96].

CADD has become an increasingly significant tool in drug discovery and has led to the discovery of numerous successful drugs including drugs for COVID-19[55]. Furthermore, it has reduced the time and expenses connected with conventional drug discovery methods. Nonetheless, CADD cannot replace experimental methods, and the development of a new drug still demands extensive experimental validation.

2.3 Chemoinformatics

Cheminformatics is the use of computational approaches to chemical issues, with a focus on structural information manipulation. Is a relatively young field of information technology that focuses on chemical data gathering, storage, analysis, and manipulation. Chemical data of interest often comprise formulas, structures, characteristics, spectra, and activities for tiny molecules (biological or industrial). this field began as a tool to aid in drug discovery and development; nevertheless, it is now playing an increasingly essential role in many fields of biology, chemistry, and biochemistry. Cheminformatics can also be used to analyze data for industries such as paper & pulp, dyes, and other similar applications (See Figure 2.2) [85].

In practice, the search for a novel molecule begins with a list of current molecules. Cheminformatics approaches are utilized to filter these lists, resulting in a subset of compounds that are tested experimentally against the biological target via high-throughput screening. Hit molecules are molecules that bind to the target. The filtering process identifies leads and, from these, a candidate for preclinical development from the list of hits. Cheminformatics is occasionally confused with bioinformatics, possibly due to its close relationship with medicinal chemistry [76].

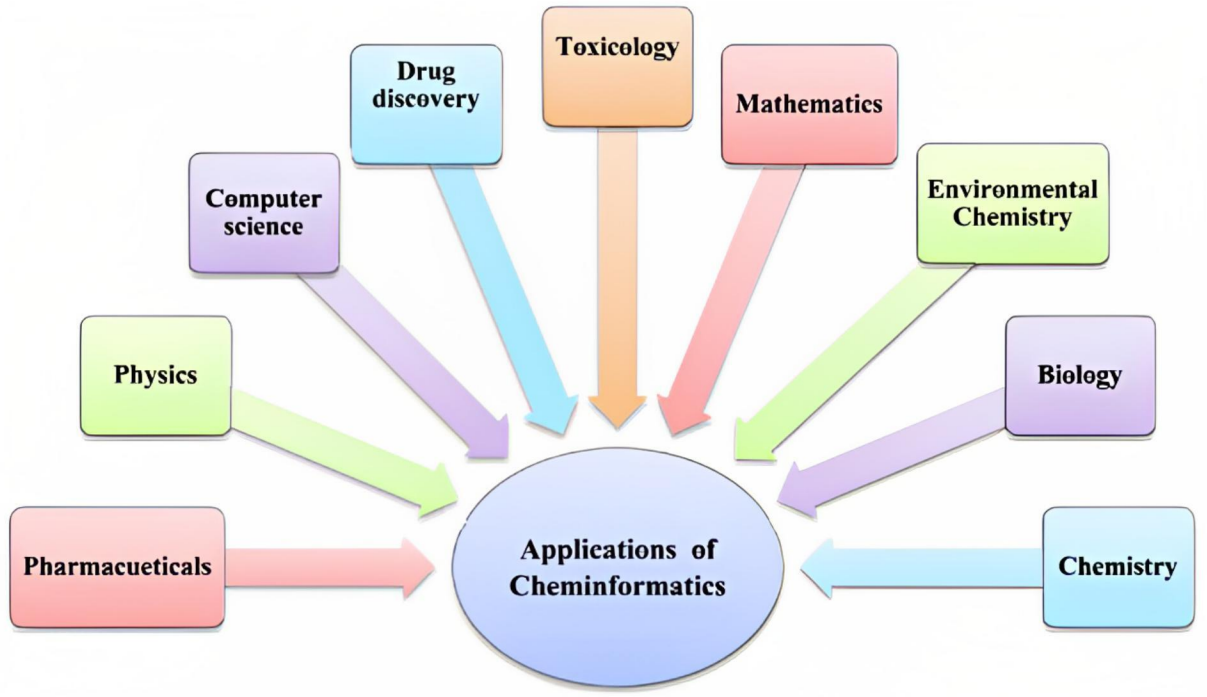


Figure 2.2: Applications Of Cheminformatics [100].

2.4 Molecular Representation

In chemoinformatics and in general, the molecule is represented as a structure diagram with bonds and atoms. Recently, computers aided humans in discovering new drugs, but the way we understand the format of molecules is different from humans. This is the reason for describing the molecule in a way that is readable by computers, so the molecular descriptors such as 1D, 2D, 3D, and fingerprints all of these descriptors make algorithms use the molecule.

Molecular representation in AI involves mapping the atoms and bonds that make up a molecule into sets of nodes and edges, which is known as the molecular graph representation [34]. There are different approaches presented in Figure 2.3 to representing molecules in a way that is machine-readable: as a string, with a connection table, as a group of features, or with a representation that a computer has learned via machine learning [107][87]. The chemical identity of a molecular entity is encoded by molecular representations, which are often referred to as descriptors or feature vectors [87]. In the context of AI and ML in chemistry, SMILES has several limitations, and SELFIES is a more robust and less limited alternative [65]. Different ways of representing the structure of a molecule include molecular formulas, structural formulas, ball-and-stick models, and perspective drawings.

In this part, we describe multiple representations of molecules, such as SMILES, fingerprints, InChI, and the Molecular Graph.

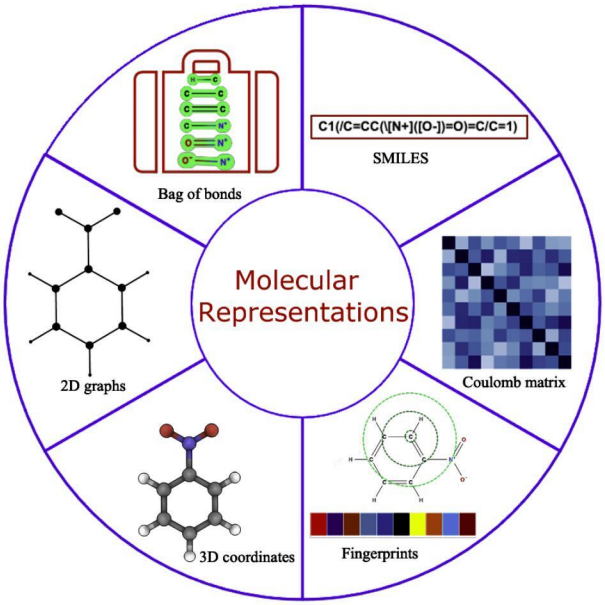


Figure 2.3: Molecular representation in cheminformatics [62].

2.4.1 String representation

String molecular representation is a common tool used to represent molecular graphs in cheminformatics and AI/ML in chemistry[65]. The most popular molecular string representation is SMILES, which has been used since the late 1980s [65].

SMILES strings are made up of ASCII letters and use a set of rules to encode a molecule's structural information, such as atom types, bond types, and stereochemistry. Water (C8H8O3), for example, has eight carbon and 3 oxygen atoms and no explicit hydrogens, hence its SMILES string is "$COC1 = C(C = CC(= C1)C = O)O$" as explained on Figure 2.4 and Table 2.1.

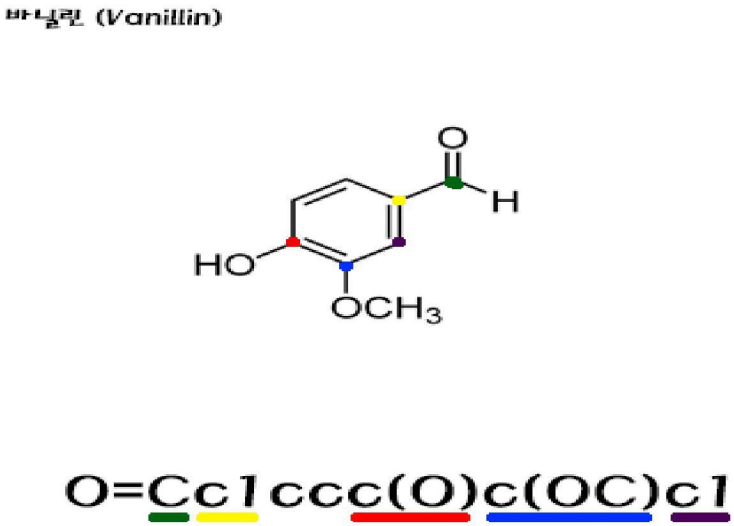


Figure 2.4: structure and string representation (SMILES) of vanillin [6].

Another string molecular representation is InChI (International Chemical Identifier), which is a unique identifier for chemical substances that encodes both structural and non-structural information about the molecule. InChI strings use a hierarchical structure to represent the different levels of structural detail. Other examples of string molecular representations include the MOL (Molecule) format, which encodes 2D and 3D coordinates of atoms and bonds, and SMARTS (SMILES Arbitrary Target Specification), which is a pattern language used for substructure searching in molecules.

Recently, a new molecular string representation called SELFIES has been developed, which is more robust and has fewer limitations than SMILES[65][27]. SELFIES is a fragment-based molecular string representation that builds on top of the chemical robustness guarantees of SMILES[27]. Other types of molecular representations include molecular fingerprints, sequence-based representations, and graph-based representations [39].

The Table 2.1 presented formulas, smiles, and structures of different molecules such as caffeine, Aspirin, Acetaminophen, Vanillin, Melatonin, and Nicotine.

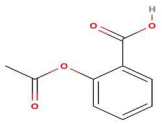
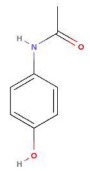
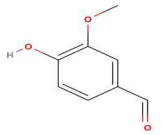
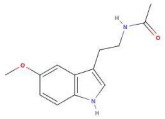
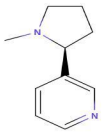
Molecule	Formula	SMILES	Structure
Caffeine	C8H10N4O2	CN1C=NC2=C1C(=O)N(C(=O)N2C)C	
Aspirin	C9H8O4	CC(=O)OC1=CC=CC=C1C(=O)O	
Acetaminophen	C8H9NO2	CC(=O)NC1=CC=C(C=C1)O	
Vanillin	C8H8O3	COC1=C(C=CC(=C1)C=O)O	
Melatonin	C13H16N2O2	CC(=O)NCCC1=CNC2=C1C=C(C=C2)OC	
Nicotine	C10H14N2	CN1CCCC1C2=CN=CC=C2	

Table 2.1: Different Molecular Representations.

2.4.2 Molecular Fingerprints

Fingerprint molecular representation is a method used in computational chemistry and cheminformatics to encode the structural and chemical properties of a molecule

into a series of binary values, also known as "bits". Fingerprint representations are commonly used to compare and search for molecules with similar properties or structures. Molecular fingerprints are classified into three types: atom-pair fingerprinting, topological fingerprints, and MACCS keys. They all, however, use the same notion of assigning a unique binary value to specific molecular properties. Some examples of frequent fingerprint representations are:

Circular fingerprints

These, also known as Morgan fingerprints, are created by running a cyclic algorithm that takes into account the atom and bond environment of each atom in the molecule. The output is a binary vector that represents the presence or absence of specific molecular substructures, which is shown in Figure 2.5.

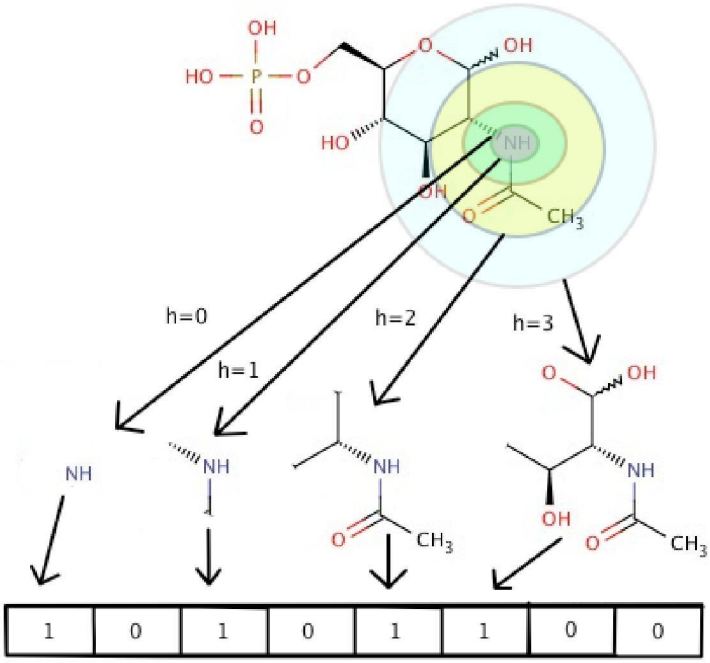


Figure 2.5: Circular patterns encoded as fingerprints [26].

MACCS keys

These are structural keys that are based on previously specified structural fragments. Figure 2.6 showed that each fragment is given a distinct binary value, and the presence or absence of these fragments in the molecule results in the generation of a binary vector.

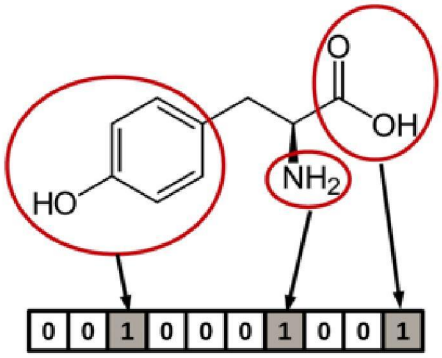


Figure 2.6: Representing Maccs Keys Structure [24].

Extended-connectivity fingerprints (ECFP)

Extended-Connectivity Fingerprints (ECFPs) are a specific form of molecular fingerprints that are meticulously crafted to encompass crucial molecular attributes pertinent to molecular activity. Widely acclaimed as one of the top choices for similarity searches in the realm of drug discovery, ECFPs find extensive utility across diverse applications with remarkable efficacy (Rogers & Hahn, 2010) [90].

ECFP Generation Process

The generation of ECFP features for a molecule involves a series of four steps. These steps are as follows:

- ***Step 1 - Initial Stage***
  In the first step, each atom in the molecule is assigned a unique integer identifier using a hashing process that combines properties such as atomic numbers and atomic mass. Initially, the atom identifier only represents the atom itself and its attached bonds.

- ***Step 2 - Update Stage***
  Moving to the next step, each atom gathers its identifier and the identifiers of its neighboring atoms into an array. A hash function described in Figure 2.7, is then applied to compress this array into a new, single-integer identifier. This step aims to capture the atom's neighborhood, as its behavior is influenced by the surrounding atom types. This updating process is iteratively repeated, allowing the identifier to incorporate information about the atom's neighbors, neighbors' neighbors, and so on. The number of iterations can be determined by the user.

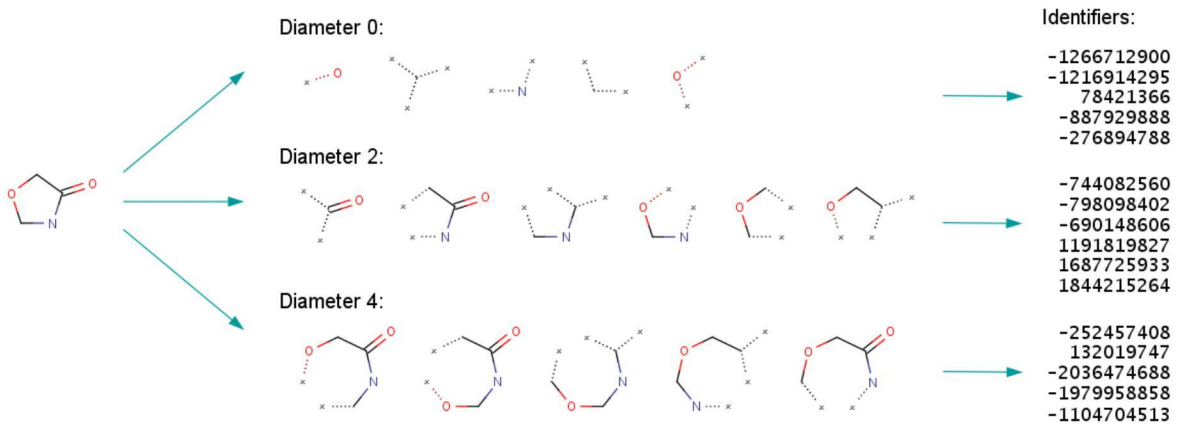


Figure 2.7: ECFP generation process [102].

- ***Step 3 - Removing duplicate structures***
  In the third stage, duplicate features are eliminated from the generated feature list. This step ensures that each unique feature is represented only once.

- ***Step 4 - Forming the bit array***
  Finally, after calculating identifiers for the specified number of iterations, the last step involves converting these identifiers into a bit array as showed on Figure 2.8, The number of elements (bits) in a fingerprint is n, where n varies depending on the type of fingerprint and is between 16 and 1024, The performance of downstream predictions may vary depending on fingerprint size [116].

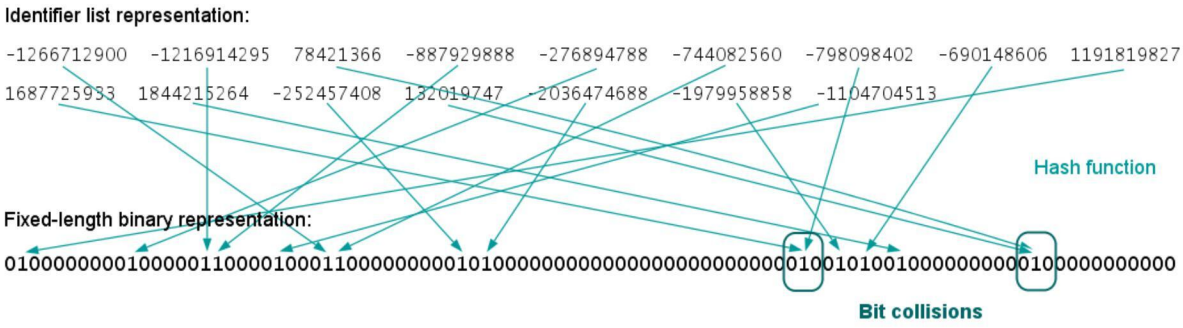


Figure 2.8: Generation of the fixed-length bit string [102].

2.4.3 Molecular descriptors

Molecular descriptors are numerical or symbolic representations of chemical compounds that capture their physicochemical, structural, and biological properties [75]. These descriptors are employed in computational chemistry, virtual screening, and

cheminformatics to forecast the characteristics of novel compounds and find prospective therapeutic candidates [75].

Molecular descriptors must meet certain criteria to be useful, such as being invariant to atom labeling and numbering, being defined by an unambiguous algorithm, and having well-defined applicability on molecular structures [75]. These descriptors can be classified into different types based on the properties they encode, such as 1D descriptors, 2D descriptors, 3D descriptors, and quantum chemical descriptors (Figure 2.9).

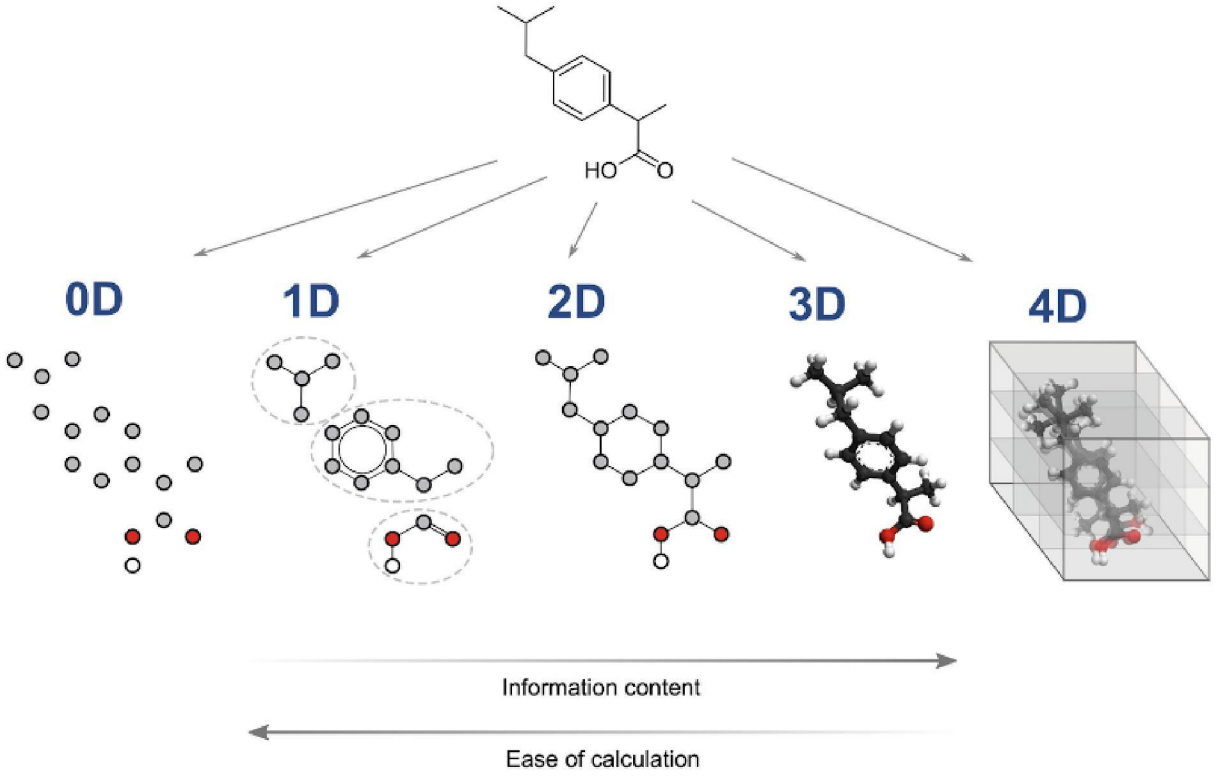


Figure 2.9: Descriptor classes [53].

1D descriptors

Which include bulk properties and physicochemical parameters such as molecular weight and polar surface area [69].

2D descriptors

Two-dimensional (2-D) descriptors are based on the chemical graph theory and provide a method for describing chemical structures, allowing for the computational processing of molecules in a data set [69].

These descriptors are calculated using the 2D structure of a molecule, which includes atom types, bond types, and connectivity. Examples of 2D descriptors include

molecular weight, number of atoms, number of bonds, and topological indices such as the Wiener index and the Randic index.

3D descriptors

Three-dimensional (3-D) descriptors are used to characterize the form, size, and orientation of molecules and are based on the geometrical features of molecules [103]. These descriptors are computed using a molecule's 3D structure, which contains information on the position and orientation of atoms in space.

4D descriptors

The 4D descriptors are also called "grid-based descriptors". These descriptions add a fourth dimension on top of the molecular geometry. The interactions of the molecule(s) with the active site(s) of a receptor or the molecule(s)' various conformational states are typically characterized by this new dimension. CoMFA and GRID are two typical 4D descriptors [96]. The 4D descriptors have the advantage of offering more information than the other descriptors and always being able to produce varied values for molecules with various structural characteristics. Due to their greater complexity, the 4D descriptors, like the 3D descriptors, are difficult to obtain.

2.4.4 Graph-based representations

A series of techniques known as graph-based molecular representations are used in computational chemistry and cheminformatics to represent the structure and characteristics of molecules as graphs. In graph-based chemical representations, atoms in a molecule are represented as nodes, and bonds between atoms are represented as edges connecting the nodes. Molecular graphs offer more detailed information for learning molecular representations than SMILES, which represent molecules as a string of lines. With the use of Molecular Representation Learning (MRL), molecules are encoded as numerical vectors while maintaining their chemical properties and structures, allowing for the execution of tasks that come later, including property prediction.

Graph-based molecular representation techniques have achieved considerable progress, especially in deep molecular graph learning-based methods. These techniques are systematically categorized into four strategies and are used in various chemical applications [54]. Overall, molecular graph-based representations are an important tool in cheminformatics for representing molecules in a way that can be understood by computers and used in machine learning models[54].

Another type of graph-based molecular representation is the fragment-based graph, as shown in the Figure 2.10 a molecule as a collection of joined pieces. The connectivity between fragments is depicted in this form by edges joining the respective subgraphs for each fragment of the molecule.

In general, graph-based molecular representations are helpful for virtual screening, lead optimization, and molecular similarity analysis, among many other chemistry and bioinformatics applications. They can also be utilized as input for machine learning algorithms for predictive modeling as well as to compare and search for compounds based on their structural and chemical characteristics.

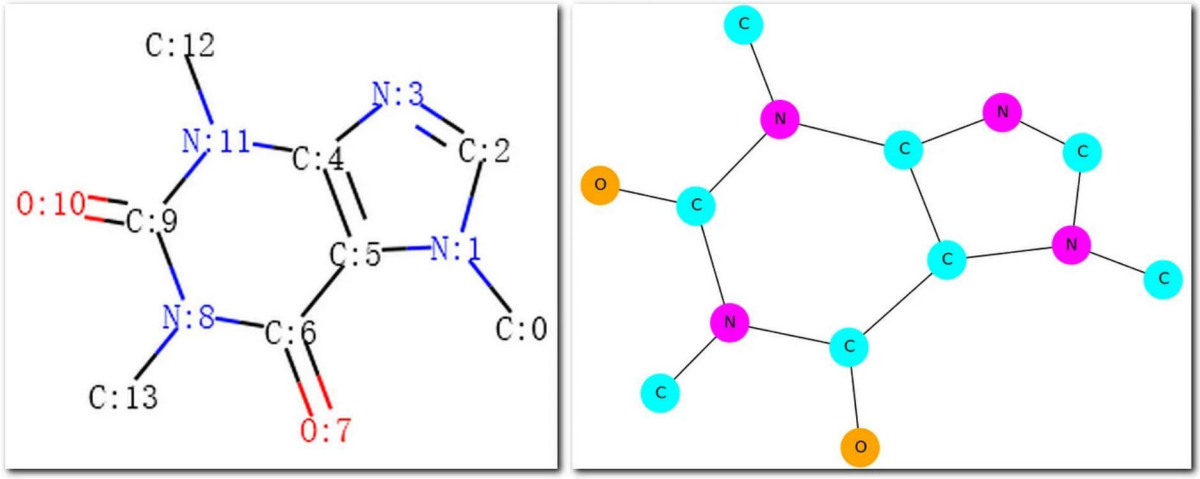


Figure 2.10: Caffeine graph representation [43].

2.4.5 3D Molecular Representations

Molecular representations are used to describe the structure and properties of molecules. A molecule can be represented in 2D or 3D. In the 2D representation, a molecule is viewed as a graph with nodes representing atoms and edges representing chemical bonds connecting the atoms. In the 3D representation, the molecule is represented by a 3D conformation with 3-dimensional coordinates of all atoms, as shown in Figure 2.11.

Deep learning methods are used for molecular representation and property prediction. Graph neural networks (GNN) are message-passing neural networks that are useful for predicting molecular properties[27]. The features of a node in a molecular graph are represented by xv, and the features of an edge (u, v) are represented by xuv[27].

Recent studies have used large-scale unlabeled molecules in a self-supervised methodology to pre-train the molecular representation [27]. In addition, new 3D representations are proving valuable for finding optimal ligands for a given chemical system [107]. Molecular chirality, a type of stereochemistry, modifies the collection of 3D conform-

ers that are available to the molecule without altering the connectivity of its 2D graph. An SE(3)-an invariant model that handles torsion angles of a 3D molecular conformer has been developed to enable representation learning on molecules with known stereochemistry [71].

To summarize, 3D molecular representations are valuable in many fields of chemistry and bioinformatics, including molecular modeling, drug design, and protein-ligand docking. They may be used to see and examine the structure and characteristics of molecules in three dimensions, providing vital insights into their biological and chemical activities.

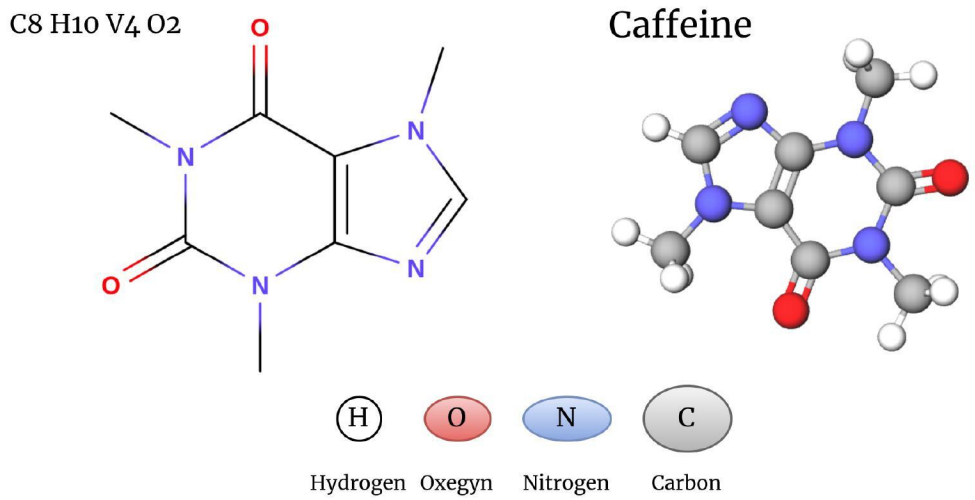


Figure 2.11: 3D Molecular Model of Caffeine [14].

2.5 Molecular similarity measurement

Molecular similarity measurement is a cheminformatics concept that involves analyzing the structural and/or functional characteristics of two or more molecules to determine their level of similarity. This technique is employed in drug discovery and design to identify potential medications that are similar in structure to existing drugs and may produce similar pharmacological effects. These metrics, which are also known as similarity coefficients or indices, transfer compatible molecular representation pairings with the same mathematical form into real values that typically, but not always, lie on the unit interval[74]. Similarity measures are used to evaluate the similarity or

dissimilarity of small molecules[74].

There are many types of molecular similarity measures, each with its strengths and limitations[74][109]. Three components make up a similarity measure: a structural representation, a weighting strategy, and a similarity coefficient[109]. The structure representation can be 1D, 2D, or 3D, and the weighting scheme can be binary, integer, or real-valued[109]. The similarity coefficient can be based on the Tanimoto coefficient, Euclidean distance, or other mathematical formulas[74][109][15].

Molecular fingerprint-based similarity includes turning the chemical structure into a set of binary bits that can be used to compare the molecules to one another to see how similar they are. This technique is commonly utilized in machine learning- and virtual screening-based approaches to drug development.

2D structural similarity includes comparing the atom types and connections of molecules in their 2D structures, whereas 3D structural similarity compares the shape, electrostatic potential, and other physical characteristics of molecules in their 3D conformations.

Pharmacophore-based similarity includes contrasting the properties of pharmacophores, the functional groups that give compounds their biological activity. This technique is very helpful for locating substances with comparable mechanisms of action.

Overall, molecular similarity measurement is a key technique in drug discovery and design, allowing researchers to identify potential drug candidates with similar structural and/or functional properties to known drugs, It can speed up the process of developing new drugs and save time.

2.5.1 The structure representation

The representation of molecular structures plays a crucial role in translating complex chemical information into numerical or mathematical forms, facilitating the measurement of molecular similarity. Standardization of molecular structures is imperative due to the inherent challenges involved in comparing and interpreting diverse chemical compositions. Various types of structure representations, such as molecular fingerprints, molecular descriptors, graph-based representations, and molecular embeddings, are employed based on the specific application and relevant properties of interest. Selecting an appropriate structural representation is pivotal for obtaining accurate and meaningful results in molecular similarity assessment. Molecular fingerprints, one widely used representation method, encode molecular structure information into binary or bit-string formats, capturing key structural features and patterns [12]. Molecular descriptors, another common representation approach, utilize numerical descrip-

tors derived from molecular properties, aiding in quantitative comparison and classification of chemical compounds [104]. Graph-based representations, on the other hand, leverage graph theory to model the connectivity and spatial arrangement of atoms and bonds within molecules, enabling efficient analysis of molecular structure [46]. Furthermore, molecular embeddings have emerged as a powerful representation technique, employing deep learning algorithms to embed molecules into low-dimensional vector spaces, enabling effective similarity measurements and downstream tasks [37].

The selection of an appropriate structural representation is a critical step that directly impacts the accuracy and reliability of molecular similarity assessment. By choosing a suitable representation method tailored to the specific application and desired properties, researchers can enhance the understanding and comparison of molecular structures, advancing the development of various chemical and pharmaceutical applications.

2.5.2 The weighting scheme

The concept of a weighting scheme plays a vital role in determining the relative importance of different molecular properties when assessing the similarity between two molecules. By assigning weights to specific chemical traits during similarity calculations, particular features can be emphasized to enhance the accuracy of molecular comparisons. Computerized representations of molecules enable the extraction of various statistical data, which can serve as the foundation for designing effective fragment weighting schemes. These schemes often assign higher weights to fragments that are frequently present within a molecule, frequently occur within small molecules, or even fragments that are relatively less common across a given dataset.

One approach to constructing a feature weighting method is to utilize statistical data derived from computational representations of molecules. Several types of statistical data can be employed, including:

(a) ***Feature Frequency*** (***ff***): This measure quantifies the frequency of occurrence of a particular feature within a compound. Features that occur more frequently are assigned higher weights compared to those that occur less frequently [36].

(b) ***Inverse Compound Frequency*** (***icf***): This measure evaluates the overall frequency of a feature across a collection of compounds. Less frequent features are given greater weight than more commonly occurring ones across the entire dataset of molecules [28].

(c) ***Compound Size or Compound Length***: Also known as the number of distinct features assigned to a compound, this property captures the size or complexity of a

molecule. Features present in smaller compounds are assigned greater weight compared to identical features present in larger compounds [80].

By incorporating these statistical measures into a weighting scheme, researchers can effectively capture the significance of different molecular properties, enabling more accurate and meaningful assessments of molecular similarity.

2.5.3 Similarity coefficient

The degree of similarity between two chemical compounds or molecules is calculated using similarity coefficients, which are mathematical formulas. They are also referred to as similarity measures, similarity indices, or distance measures. The role of similarity coefficients is vital in many areas of chemoinformatics, such as synthesis design, similarity searching, virtual screening, molecular diversity analysis, and property prediction [15].

There are different types of similarity coefficients used in chemoinformatics research, including Tanimoto, Dice, Cosine, and Euclidean distance, among others [17]. These similarity coefficients are mathematical functions that convert two compatible molecular representations of the same mathematical form into pairs of real values, often on the unit interval [74]. The Tanimoto similarity coefficient states that if two molecules are identical, their distance is zero and their similarity score is one. Yet, the similarity score decreases as the distance between two distinct molecules rises, eventually reaching zero. of course, distance and similarity coefficients are both used to measure the similarity between two chemical compounds or molecules, but they represent different concepts.

The term "distance measurements" refers to the mathematically computed physical distance between any two molecules in space. When comparing the structural differences of molecules, distance measures are used, and the lesser the distance value, the more similar the molecules are thought to be. The Euclidean distance and the Manhattan distance are two examples of distance measurements applied in chemoinformatics.

On the other hand, The term "Similarity coefficients" refers to mathematical functions that compare the similarity between two molecules based on their attributes, such as their chemical properties, molecular structure, and fingerprints. They are also known as similarity measures or similarity indices. Typically, similarity coefficients are stated as a number between 0 and 1, where a value of 1 denotes complete identity between two molecules and a value of 0 denotes complete dissimilarity. Examples of similarity coefficients used in chemoinformatics include the Tanimoto coefficient and the Dice coefficient illustrated in the following Table 2.2 and Figure 2.12.

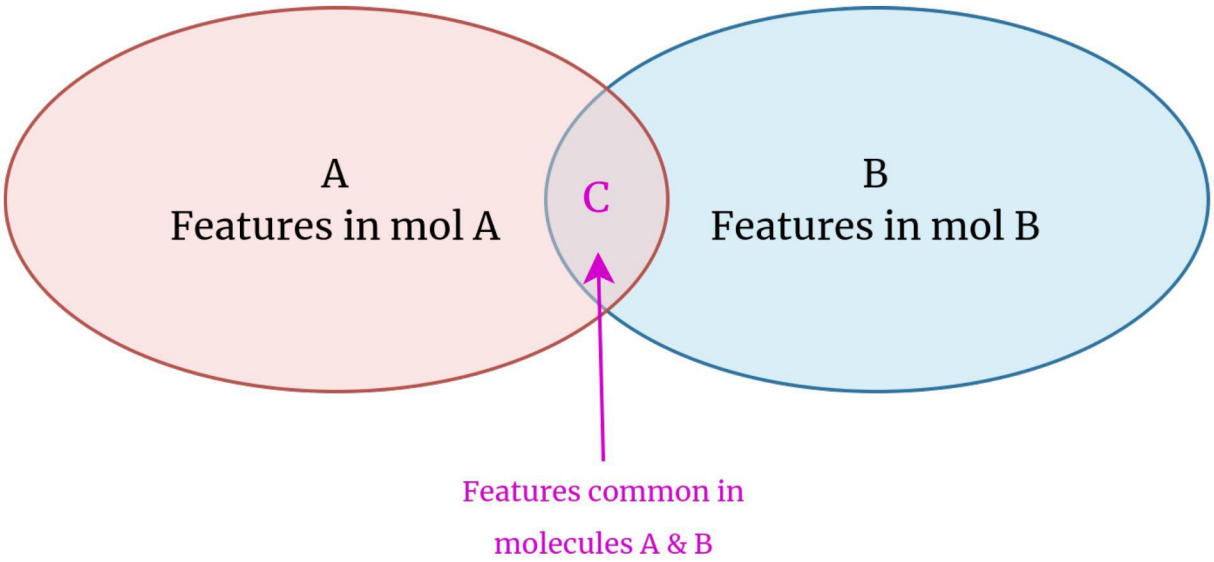


Figure 2.12: Relationship between molecules A & B features [69].

Similarity/Distance Coefficient	Formula
Tanimoto (Jaccard) Coefficient	$S = \frac{C}{A+B-C}$
Dice Coefficient	$S = \frac{2C}{A+B}$
Cosine Coefficient	$S = \frac{C}{\sqrt{A \cdot B}}$
Euclidean Distance	$D = \sqrt{A + B - 2C}$
Hamming (Manhattan or city-block) Distance	$D = A + B - 2C$

Table 2.2: Common Similarity/Distance Measures in Chemical Informatics [69].

The Table 2.2 above presents a distinction between similarity and distance metrics. The first three metrics (Tanimoto, Dice, and Cosine coefficients) are used to measure similarity (S_{AB}) and determine the similarity between two molecules. On the other hand, the remaining metrics (Euclidean, and Hamming) are utilized as distance or dissimilarity metrics (D_{AB}) to quantify the dissimilarity between molecules.
To convert dissimilarity metrics into similarity measures, a straightforward approach can be employed. For dissimilarity metrics with a range of values from 0 to 1, the similarity score (S_{AB}) between two molecules can be calculated by subtracting the dissimilarity score from unity:

$$S_{AB} = 1 - D_{AB} \tag{2.1}$$

In the case where a distance metric has a value greater than 1 the dissimilarity score

can be converted to a similarity score using the following equation:[105]:

$$S_{AB} = \frac{1}{1 + D_{AB}} \tag{2.2}$$

According to this equation, when two molecules are identical to one another, their distance (D_{AB}) is zero and their similarity score (S_{AB}) is 1. Conversely, as the (D_{AB}) value increases (indicating dissimilarity between molecules), the S_{AB} score approaches 0. This equation provides a way to transform distance metrics into similarity scores, facilitating the comparison and interpretation of molecular dissimilarities.

2.6 Virtual screening

Virtual screening is a computational technique used in drug discovery to identify potential drug candidates with desired biological activity. The procedure entails using computer tools to scan enormous chemical libraries for compounds that are likely to bind to a target protein and modify its activity. There are two main types of virtual screening methods: ligand-based virtual screening (LBVS) and structure-based virtual screening (SBVS) (See Figure 2.13). LBVS involves comparing the chemical properties of known active compounds to identify similar compounds in a chemical database, while SBVS involves docking potential ligands to the three-dimensional structure of a protein target and evaluating the binding energy of the complex.

this procedure enables researchers to quickly find prospective drug candidates from enormous chemical libraries, saving time and resources in the drug development process. It has grown to be a key tool in drug discovery. It is crucial to keep in mind that virtual screening is only the initial stage of the drug development process, and additional experimental research is required to confirm the found compounds expected activities.

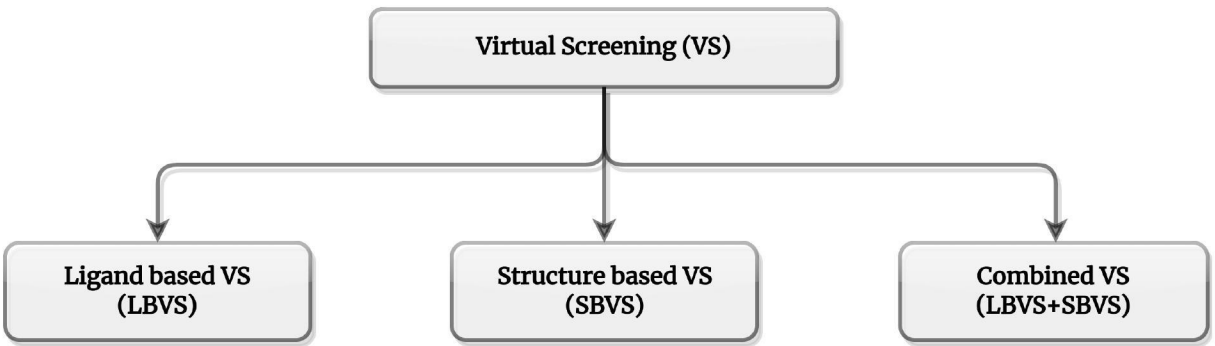


Figure 2.13: Virtual screening strategies.

2.6.1 Ligand-Based virtual screening

This method involves screening a large database of small molecules to identify those that have similar chemical structures to known ligands that bind to the target protein. This technique has been used in various studies, including identifying potential inhibitors of human hexokinase II for the development of anti-dengue therapeutics, selecting active antiviral compounds against COVID-19 disease targeting coronavirus endoribonuclease NendoU/NSP15 [62], and virtual screening strategy combined with Bayesian classification model, molecular docking for Acetyl-CoA Carboxylases inhibitors [111]. The history of LBVS dates back to the early 1970s when the concept of molecular similarity was introduced in drug discovery[95]. In the early days, LBVS was limited to the use of simple molecular descriptors such as molecular weight, log P, and hydrogen bond donors/acceptors. However, with the development of more sophisticated computational methods and the availability of high-performance computing resources, and has become more efficient and accurate. Ligand-based virtual screening (LBVS) is a computational approach used in drug discovery to identify potential ligands from large compound libraries. LBVS focuses on the chemical characteristics of ligands rather than relying on target protein structures. One of the main aspects of LBVS is the classification of compounds into active and inactive based on their similarity to known ligands or molecular descriptors (See Figure 2.14).

Similarity-based methods are commonly employed in LBVS to assess the similarity between query compounds and a database of known active ligands. Molecular fingerprints, which encode structural information, are often used to represent compounds. By applying mathematical algorithms, such as the Tanimoto coefficient or Dice coefficient, the similarity between two molecular fingerprints can be calculated. Compounds with high similarity scores to known active ligands are considered potential hits for further experimental evaluation [108]. In addition to similarity-based methods, machine learning-based approaches have gained prominence in LBVS. These methods employ statistical models trained on a dataset of known ligands and decoys (inactive compounds) to predict the activity of new compounds. Machine learning algorithms, including support vector machines (SVM), random forests (RF), and neural networks, are commonly used to learn the underlying relationships between molecular features and bioactivity [33]. Trained models can then be used to classify new compounds as active or inactive based on their features.

It is crucial to emphasize the importance of high-quality training data in LBVS classification. The selection of appropriate molecular descriptors, similarity metrics, or machine learning algorithms greatly impacts the accuracy and reliability of the classification results. Furthermore, rigorous validation of the classification models is essential to ensure their predictive power and generalizability. In summary, ligand-based virtual

screening employs both similarity-based and machine learning-based methods for the classification of compounds. These approaches enable the efficient screening of large compound libraries, aiding in the identification of potential ligands for further experimental investigation in drug discovery research.

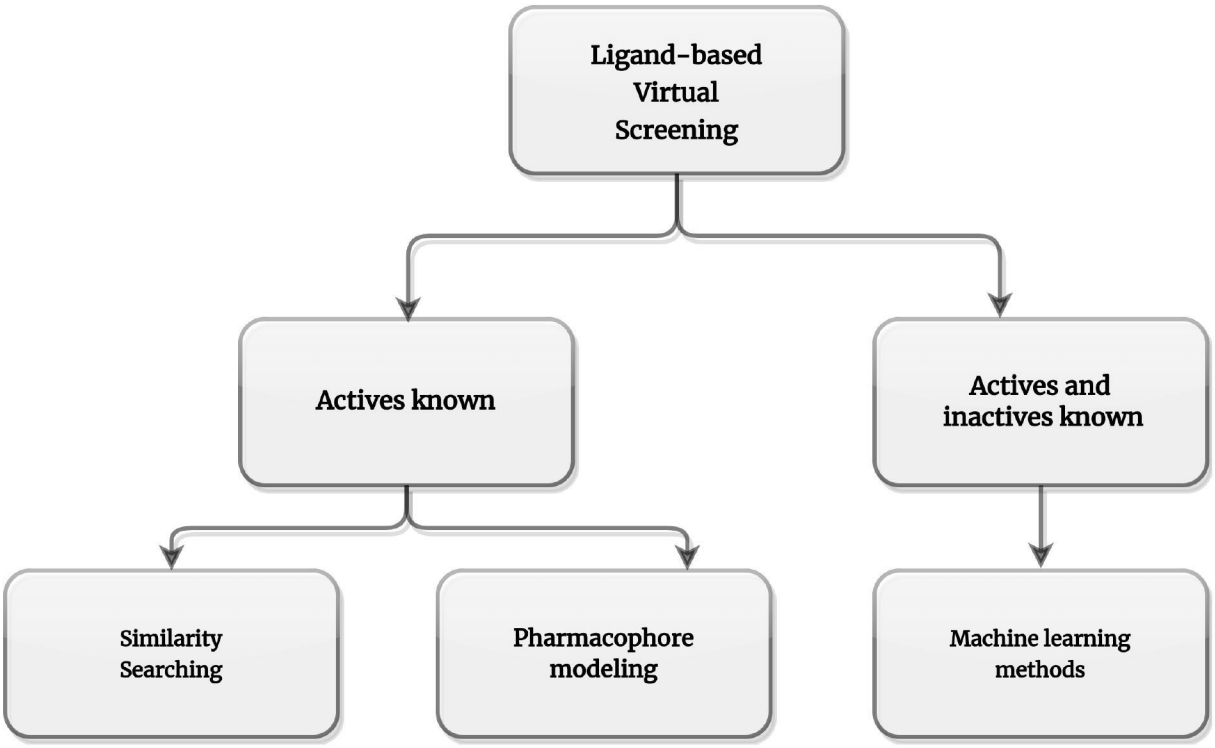


Figure 2.14: Classification of ligand-based approaches.

2.6.2 Structure-Based virtual screening

This method involves using the three-dimensional structure of the target protein to screen a large database of compounds and identify those that are most likely to bind to the protein. The method involves generating a pharmacophore model based on the structure of the target protein, screening a database of compounds against the model, and then validating the hits using molecular docking and other methods. The development of fast and accurate algorithms has allowed structure-based virtual screening to become an important tool in drug discovery and design. Other virtual screening methods include ligand-based virtual screening, which relies on the similarity of compounds to known ligands or pharmacophores [19].

The main difference between these methods is that In ligand-based virtual screening, a database of compounds is screened based on their similarity to a known ligand or pharmacophore. In contrast, structure-based virtual screening involves using the

three-dimensional structure of the target protein to screen a large database of compounds and identify those that are most likely to bind to the protein. Both methods have their strengths and limitations, and a combination of the two approaches can be used to improve the accuracy of virtual screening. Examples of studies that have used both methods include the identification of novel Hsp90 inhibitors [23] and inhibitors of human lysosomal acid lipase [9].

2.7 Related works

Similarity-based virtual screening is widely recognized as an essential aspect of drug discovery, aiming to identify molecules with structural similarity to reference compounds. Various approaches have been employed to enhance the effectiveness of similarity-searching methods. Among these, 2D similarity methods have gained widespread use due to their ability to quantify the similarity between molecules based on structural features. The Tanimoto coefficient has emerged as a commonly utilized indicator of chemical compound similarity in chemoinformatics [12, 22, 94, 99]. To further improve search methods, techniques from diverse disciplines have been explored. For instance, Bayesian inference networks proposed by (Abdo et al.) [3, 1], originally developed for text retrieval, have been successfully applied in virtual ligand screening. Additionally, reweighting techniques initially used in the text retrieval field have been adapted for cheminformatics by (Ahmed et al.) to strengthen the Bayesian network [3]. Al-Dabagh used the concepts of quantum mechanics theory concepts have also been leveraged to enhance molecular similarity searching and ranking of chemical compounds in LBVS [31]. Nasser and colleagues used deep belief networks (DBNs) to incorporate all new features from all descriptors to create a new descriptor for similarity searches when several descriptors were used, each representing different relevant elements [82].

Moreover, the advent of deep learning (DL) techniques has revolutionized drug discovery, bioinformatics, and cheminformatics. These technologies have opened up new possibilities for computational decision-making and aiding in the understanding of molecular mechanisms. Researchers have proposed various DL models to generate new molecules, predict compound activities, and optimize drug properties. Convolutional neural networks (CNN), long short-term memory (LSTM), and self-attention have been commonly used in these studies. For instance, autoencoder models proposed by Gómez-Bombarelli et al. have been utilized to convert discrete molecular representations into continuous representations, enabling the generation of new molecules [49]. Skalic et al. proposed a new model Variational autoencoders have been employed to transform seed compounds into three-dimensional representations, which are then

used to explore uncharted regions of the chemical space [97]. Gao et al. proposed a generative network complex (GNC) model using Gradient descent in the latent space of an autoencoder have been utilized for multi-property optimization in the creation of new drug like molecules [45]. CNN used by Hamza et al. to predict orphan compound activities, while both 1D and 2D CNN have been trained separately to evaluate different descriptors' performance [57]. RNN-based methods used by Goh et al. have also been explored in chemoinformatics, where SMILES representations of molecules are used to learn their coding grammar and predict molecular properties [83, 48]. Siamese architectures, such as CNN and LSTM networks, have shown promising results for similarity modeling, especially with complex and heterogeneous data samples. They have been employed in various applications, including family reunion, sentence similarity calculation, and drug interaction prediction. For example, Yu et al. used CNN Siamese architectures to determine whether two persons were related, making it possible to find lost people and reconcile them with their family members [115]. The similarity between sentences was determined by Jonas et al. using the LSTM Siamese neural network [101]. Another technique of deep learning called self-attention has been successfully employed in transformer-based models, such as the transformer model introduced by Vaswani et al. in 2017, which revolutionized machine translation [106]. Since then, self-attention has been extensively utilized in various applications, including language understanding, text summarization, image captioning, and speech recognition, due to its ability to capture long-range dependencies and improve the modeling of the context within sequences.

In the year 2022, Altalib suggested the utilization of advanced deep learning methods such as LSTM-RNN, GRU-RNN, 1D CNN, and 2D CNN. Their objective was to enhance the performance in handling molecules with diverse structures by harnessing the capabilities of deep learning and Siamese neural network architecture [5].

The Siamese architecture has demonstrated promising performance for similarity modeling, particularly with heterogeneous data samples. Inspired by these advancements, this research proposes two novel models, namely SSCNN1D and SSCNN2D, to augment the results of ligand-based virtual screening. These models leverage deep learning techniques, such as 1D and 2D convolutional neural networks, LSTM, bidirectional LSTM (BiLSTM), and attention mechanism, within the Siamese architecture. By integrating the strengths of deep learning and the Siamese architecture, SSCNN1D, and SSCNN2D aim to improve the retrieval effectiveness of the similarity model, particularly when dealing with structurally heterogeneous molecules. Through an in-depth evaluation, the performance and efficacy of these models will be assessed, contributing to the advancement of molecular similarity-based virtual screening.

2.8 Summary

This chapter delves into the field of computer-aided drug design, showcasing its profound impact on pharmaceutical research. Through the utilization of computational algorithms and informatics tools, this discipline has revolutionized the process of discovering and optimizing therapeutic agents. The chapter provides a comprehensive overview of the core principles and methodologies employed in computer-aided drug design. It explores the domain of chemoinformatics, elucidating its role in molecular representation and analysis. The significance of accurate molecular similarity measurements is emphasized, along with the various aspects of structure representation, weighting schemes, and similarity coefficients. Moreover, the chapter delves into the applications of virtual screening, encompassing both ligand-based and structure-based approaches. By presenting this academic exploration, the chapter aims to establish a strong knowledge foundation, enabling researchers to appreciate the integral role of computer-aided drug design in modern drug discovery and prepares for the next chapter which will describe our approach to detecting molecular similarity.

Deep Learning

Chapter 3

Deep learning

3.1 Introduction

The continuous development of artificial intelligence (AI) has led to remarkable advancements in the field of machine learning (ML). ML approaches have transformed data analysis by allowing computers to understand patterns and predict outcomes from massive datasets. Deep learning, a subset of ML, has emerged as a powerful paradigm that leverages artificial neural networks with multiple layers to extract complex representations from data. This chapter explores the evolution and applications of deep learning, with a specific focus on two key models: Siamese neural networks and attention mechanisms. Siamese networks employ twin subnetworks with shared weights, enabling tasks related to similarity or dissimilarity learning. Attention mechanisms, inspired by human attention processes, selectively focus on relevant elements in input data, improving model performance. By delving into the principles, methodologies, and applications of deep learning, including Siamese networks and attention mechanisms, this chapter provides a comprehensive understanding of these cutting-edge technologies and their potential for advancing AI systems across various domains.

3.2 Artificial Intelligence (AI)

Artificial intelligence (AI), a field characterized by the development of computer systems that can carry out tasks usually requiring human intelligence, has the of creating intelligent machines that interact with the world and humans in a manner similar to humans themselves. AI can be labeled into narrow AI, which focuses on precise tasks together with speech recognition, picture classification, and recommendation structures, and general AI, which objectives to attain human-degree intelligence and carry

out a wide variety of intellectual tasks [92]. AI strategies encompass diverse methods such as machine learning, deep learning, natural language processing, computer vision, and robotics as shown in Figure 3.1. Those techniques allow AI structures to examine from statistics, discover patterns, manner and apprehend language, interpret visual records, and interact with the physical international [92] In the healthcare domain, AI holds tremendous potential for revolutionizing drug development processes. By leveraging computational algorithms and machine learning techniques, AI systems can analyze vast amounts of biomedical data, such as genomic information, protein structures, and molecular interactions, to identify potential drug candidates and predict their efficacy and safety profiles [113]. Techniques like virtual screening and de novo drug design facilitated by AI enhance the efficiency and cost-effectiveness of the drug discovery process [48]. The applications of AI are extensive and span diverse fields, including healthcare, finance, education, transportation, and many others.

While AI offers significant benefits, it also raises ethical and societal concerns. One pressing concern is algorithmic bias, which occurs when machine learning models produce biased or discriminatory outcomes due to inherent biases in the training data or algorithms themselves [30]. Such bias can perpetuate social inequalities and reinforce systemic discrimination, particularly in critical domains like hiring, loans, and criminal justice. It is crucial to address and mitigate these biases by ensuring transparency, scrutiny, and ongoing evaluation throughout the life-cycle of AI technologies [13]. The widespread deployment of AI systems also raises concerns regarding privacy and data protection. The collection, storage, and analysis of personal data by AI algorithms raise issues related to consent, user control, and the risk of unauthorized access or misuse [79]. Balancing the utilization of data for AI advancements while safeguarding individual privacy rights requires robust data governance frameworks and regulatory measures [84]. Moreover, the integration of AI systems into various sectors has implications for employment and societal dynamics. While AI-driven automation can streamline processes and enhance efficiency, there are concerns about job displacement and the need for upskilling and reskilling to ensure a smooth transition in the workforce [44].

Additionally, the broader socio-economic implications of AI integration require careful examination and proactive policies to ensure equitable distribution of benefits while mitigating potential adverse consequences [38]. To navigate these complexities and challenges, it is essential to develop comprehensive governance frameworks and ethical guidelines that promote responsible development, deployment, and utilization of AI technologies. This includes fostering transparency, accountability, and explainability within AI systems, as well as establishing mechanisms for auditing and addressing biases or discriminatory outcomes [42]. Collaboration among academia, industry, policymakers, and the public is crucial to fostering a multidisciplinary and inclusive ap-

proach that embraces diverse perspectives and expertise [21]. By critically examining AI advancements, challenges, and ethical dimensions, this section aims to contribute to a comprehensive understanding and conscientious advancement of artificial intelligence technologies, fostering a human-centric and socially beneficial AI ecosystem.

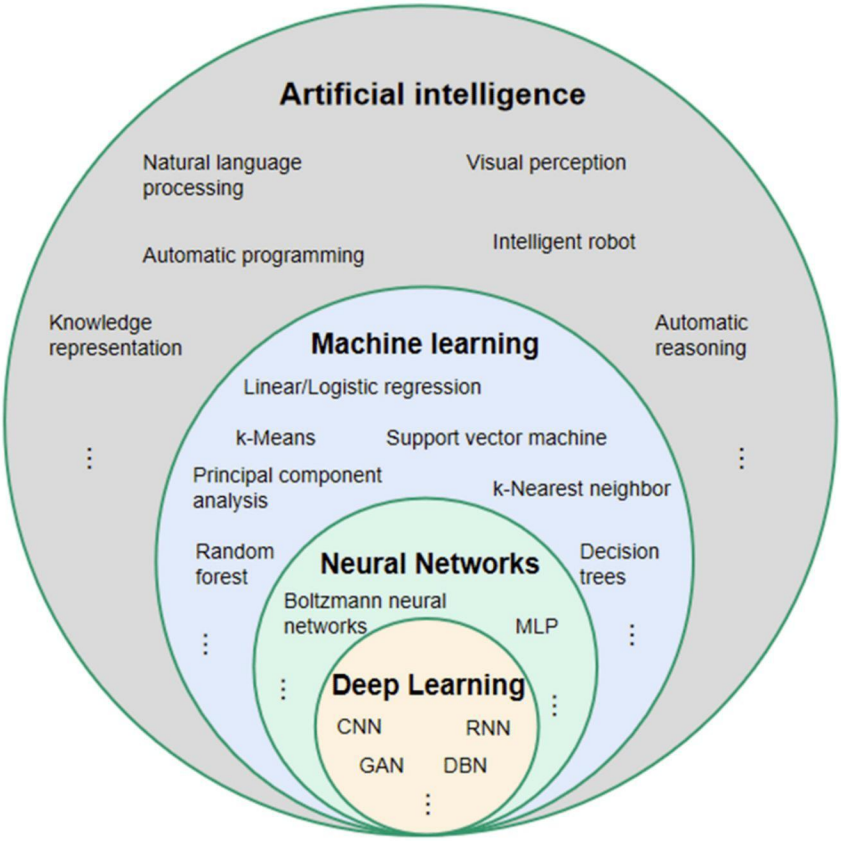


Figure 3.1: Relationship between artificial intelligence machine learning neural network and deep learning [68].

3.3 Machine learning (ML)

Machine learning focuses on creating algorithms and computer programs that can learn from data and enhance their performance without explicit programming [78] as a subset of artificial intelligence (AI). It seeks to make it possible for systems to automatically identify and use data patterns for prediction and decision-making activities. Machine learning encompasses various approaches (See Figure 3.2), with supervised learning, unsupervised learning, and reinforcement learning being prominent categories [20]. Supervised learning entails training models on labeled datasets, with each data point associated with a specified desired output [20]. The models learn to connect input features to the correct output labels, allowing them to make accurate predictions on unseen data. This method has been widely used in applications like

classification, regression, and time series forecasting. Unsupervised learning, on the other hand, deals with unlabeled datasets, seeking to discover inherent patterns and structures in the data [20]. Clustering algorithms group similar data points together based on their intrinsic characteristics, while dimensionality reduction techniques aim to reduce the complexity of the data while preserving its essential properties. Unsupervised learning has been utilized in applications like customer segmentation, anomaly detection, and data visualization.

Reinforcement learning involves training agents to make optimal decisions in an environment by interacting with it and receiving feedback in the form of rewards or punishments [20]. The agents learn through trial and error, seeking to maximize cumulative rewards over time. This approach has shown remarkable success in areas such as game-playing, robotics, and autonomous systems.

Machine learning techniques are used in a variety of fields, including medicine, email filtering, speech recognition, and image categorization [20]. In complex and data-rich contexts, these algorithms have the ability to find significant insights and make accurate predictions. Machine learning continues to evolve and has the potential for solving difficult real-world issues as technology, algorithms, and data availability improve.

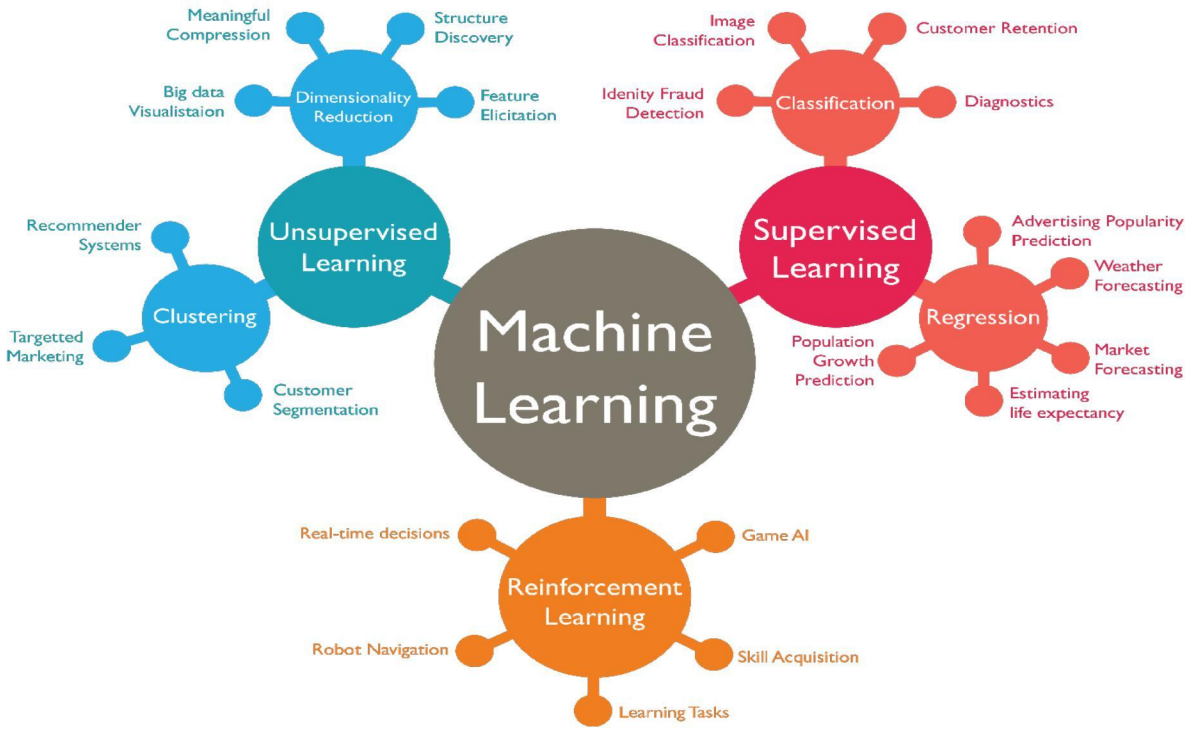


Figure 3.2: Diagramme d'apprentissage automatique [89].

3.4 Deep learning (DL)

Deep learning, a subfield of machine learning, enables computers to learn and make intelligent decisions by automatically extracting complex patterns from vast amounts of data. At its core, deep learning utilizes deep neural networks, composed of interconnected artificial neurons organized into multiple layers, to progressively learn hierarchical representations of input data [50]. The roots of deep learning can be traced back to the 1940s with the introduction of artificial neural networks [91]. However, it was not until the early 2000s that significant progress was made, thanks to breakthroughs in training algorithms and advancements in computational power [59]. One pivotal milestone occurred in 2012 with the introduction of the AlexNet architecture by Krizhevsky et al., which showcased exceptional performance in image classification tasks [66]. Since then, deep learning has rapidly evolved, benefiting from hardware advancements, the availability of large datasets, and innovative network architectures [66].

The underlying mechanism of deep learning involves training neural networks with labeled examples, using optimization algorithms such as stochastic gradient descent to minimize prediction errors [50]. This iterative learning process allows the network to automatically discover and extract intricate patterns, leading to accurate predictions and decision-making. Deep learning has made significant contributions across various domains. In computer vision, it has revolutionized tasks such as image recognition, object detection, and image generation. In natural language processing, deep learning models have achieved impressive results in language translation, sentiment analysis, and speech recognition. Furthermore, deep learning has found successful applications in healthcare, finance, autonomous vehicles, and other fields [67].

Deep learning's success can be ascribed to its capacity to manage massive datasets, recognize intricate correlations and patterns, and adapt to many fields. Deep learning has enormous promise for resolving complicated issues and propelling further developments in artificial intelligence as processing power and dataset availability increase. there are three most important architectures used in deep learning Feedforward neural networks, convolutional neural networks (CNN), and recurrent neural networks (RNN).

3.4.1 Feedforward neural networks (FNN)

Feedforward neural networks (FNN) are a type of artificial neural network where the information flows in one direction, from input to output. FNNs are widely used for function approximation, classification, and prediction tasks. They consist of multi-

ple layers of interconnected nodes, or "neurons," where each neuron in a layer receives input from the previous layer and produces output for the next layer. The activation function of each neuron determines the output of the neuron based on its input. FNN can have different architectures and activation functions, and their performance depends on the choice of these parameters [118][7]. FNN can be trained using various optimization algorithms to adjust the weights and biases of the connections between neurons to minimize the error between the predicted and actual outputs [7].

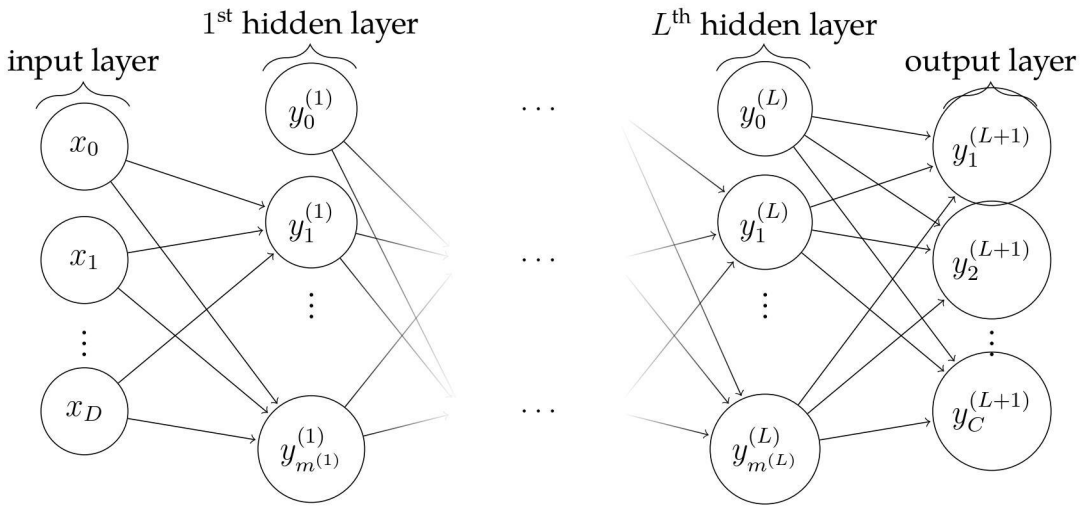


Figure 3.3: Typical architecture of a feed-forward neural network (FFN)

3.4.2 Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) are a type of artificial neural network that is commonly used for image and video recognition tasks. as shown in Figure 3.4 a CNN consists of various layers, including input, output, and hidden layers. These hidden layers can take the form of convolutional layers, pooling layers,followed by one or more fully connected layers that perform classification or regression tasks [66]. CNN are trained using back-propagation and stochastic gradient descent to adjust the weights and biases of the filters to minimize the error between the predicted and actual outputs.

- ***Convolutional layers :***
  CNN are designed to automatically learn hierarchical representations of input data with dimensions H x W x D, where H represents the height, W represents the width, and D represents the number of input channels , by applying a series of convolutional filters defined by the size of the filter to the input image or video frames Figure 3.5 performing a dot product between the filter and each local region of the input image , and the output of each filter is passed through a

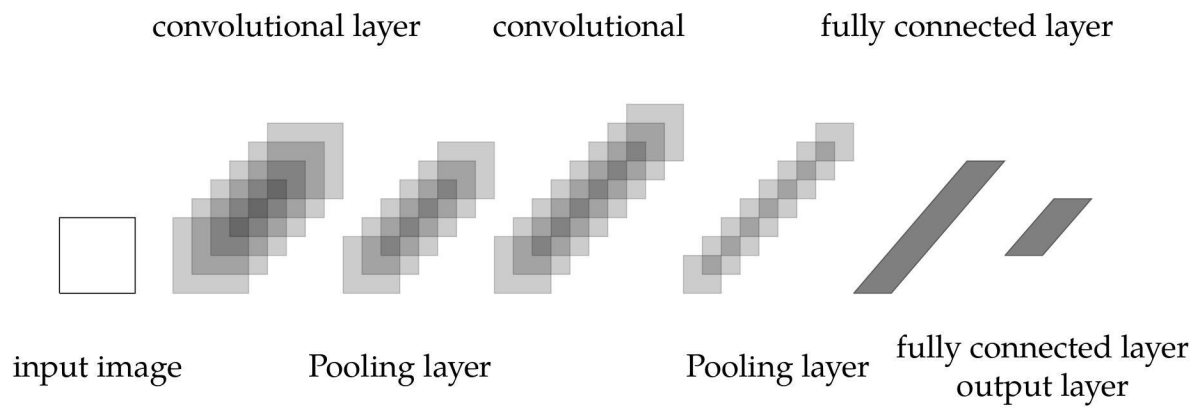


Figure 3.4: architecture of a convolutional neural networks(CNN).

non-linear activation function such as ReLU (Rectified Linear Unit), which introduces non-linearity to the network, to produce a feature map [18].
The convolutional layer produces a set of feature maps, each of which represents a different filter F. The stride S and padding P employed during the convolution operation are two parameters that affect how big the feature maps should be and the size of the output feature which can be calculated using the following formula:

$$\text{output_size} = \frac{\text{input_size} - \text{F} + 2 \times \text{P}}{\text{S}} + 1 \tag{3.1}$$

- ***Pooling layers :***
  Next The feature maps are then down-sampled using pooling layers to reduce the spatial dimensionality of the input and increase the computational efficiency of the network [18],This reduces the amount of calculation and weights needed and controlling the over-fitting. There are several pooling functions such as Max pooling which is the most widely used pooling technique. It partitions the input into non-overlapping rectangular regions and takes the maximum value within each region Figure 3.6,the size of output volume can be determined by the following formula [77]:

$$\text{W_out} = \frac{\text{W} - \text{F}}{\text{S}} + 1 \tag{3.2}$$

  were the W represent the spatial size of the input volume ,and the S indicates the stride value, which outlines the size of the step at which the pooling kernel F traverses the input volume.

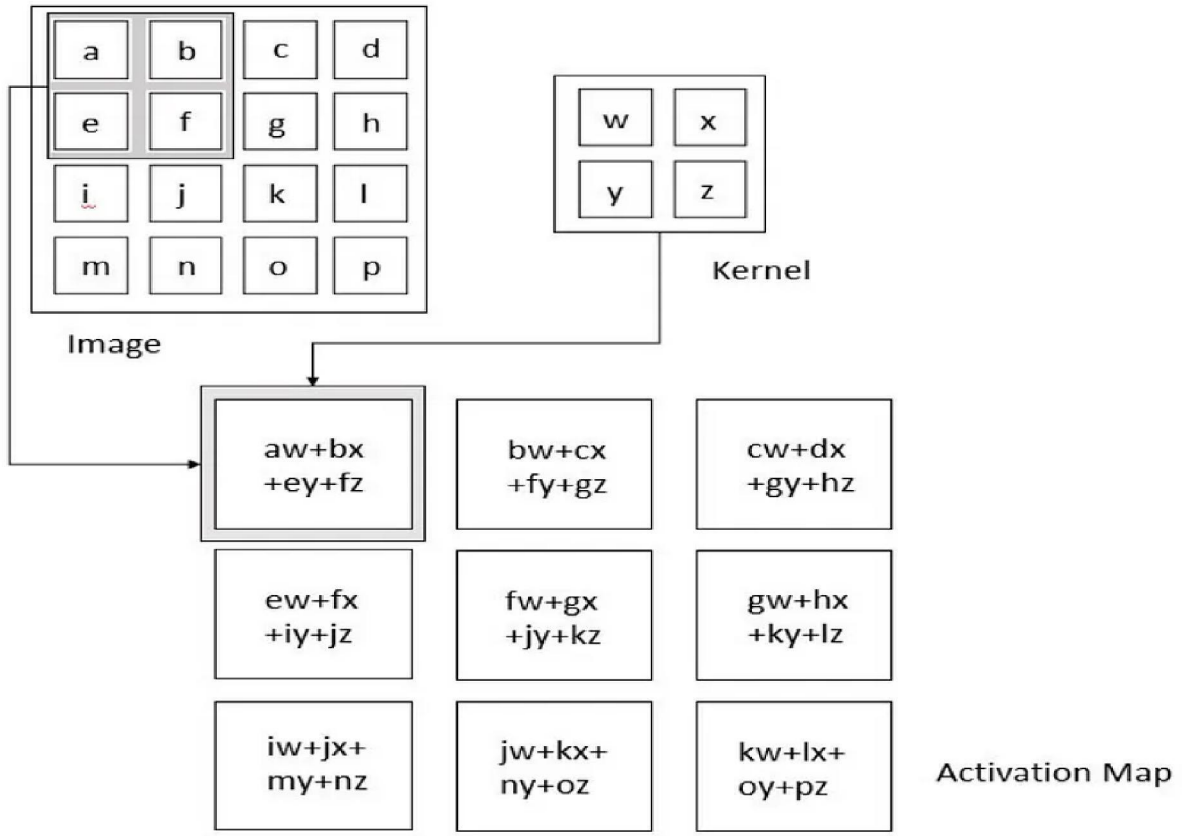


Figure 3.5: An example of 2-D convolution operation [58].

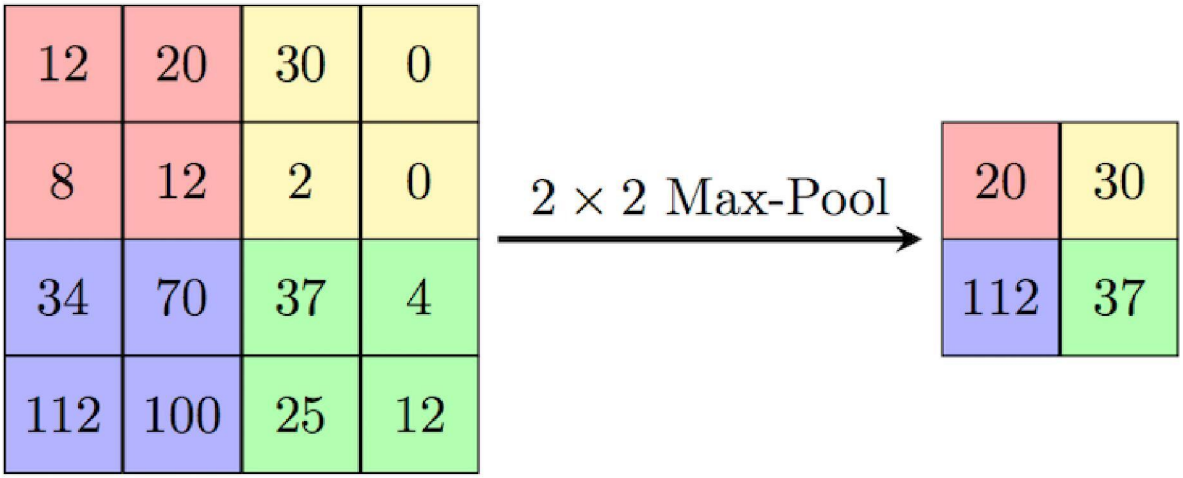


Figure 3.6: Pooling Operation using the 2 × 2 filters [41].

- ***Fully-connected layers:***
  One or more fully connected layers are frequently used as part of the convolutional neural network's (CNN) final stage of feature extraction which helps to map the representation between the input and the output Figure 3.4, preceding the classification or regression task. fully connected layers require inputs in the form of 1D vectors . Therefore, the feature maps are flattened into a single vector

by reshaping them.

3.4.3 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNN) are a type of artificial neural network that are designed to process sequential data, such as time series, speech, and text [118]. Unlike feed-forward neural networks, RNN have loops in their architecture that allow them to maintain a "memory" of previous inputs and use it to inform the processing of current inputs [118] it consists of a set of interconnected cells organized into layers, Every cell in the layer receives input from other cells in the same layer as well as from cells in the previous layer, the output of each cell is then fed back into the network as input for the next time step. RNN can have different types of cells which are described later in this section, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), that are designed to address the vanishing gradient problem and improve the memory capacity of the network[118]. Figure 3.7 shows that any length of sequence can be processed by RNN, which typically maps a sequence of inputs x_1,...,x_T to a sequence of hidden states h_1,...,h_T through the equation:

$$h_t = \tanh(W_{hh} \cdot h_{t-1} + W_{xh} \cdot x_t + b_h) \tag{3.3}$$

were W_{hh} is the weight matrix that connects the previous hidden state h_{t-1} to the cur-

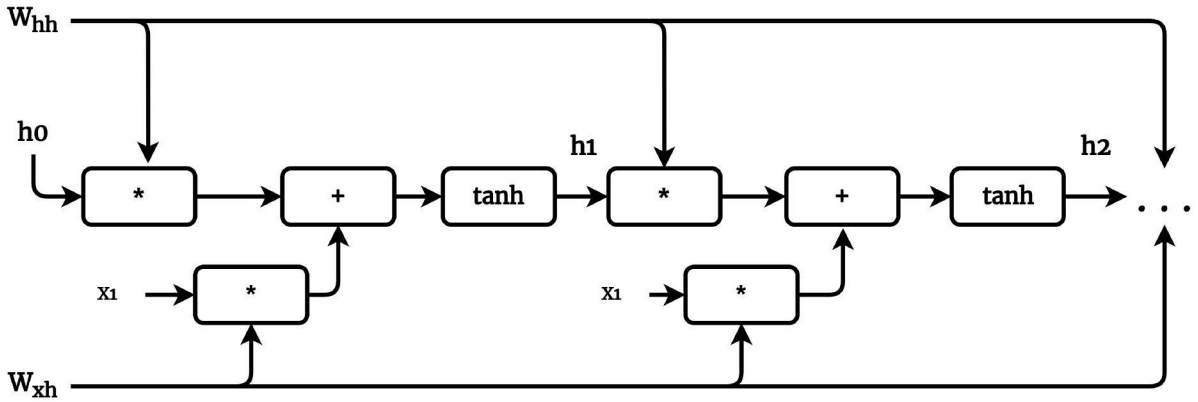


Figure 3.7: A computation graph corresponding to a simple RNN.

rent hidden state h_t using matrix multiplication, This matrix defines the influence of the previous hidden state on the current hidden state added the weight matrix W_{xh} that connects the current input x_t to the current hidden state h_t , next the bias vector, which was added to the weighted sum of the previous hidden state and the current input. The bias allows the network to shift the activation function's output [35].

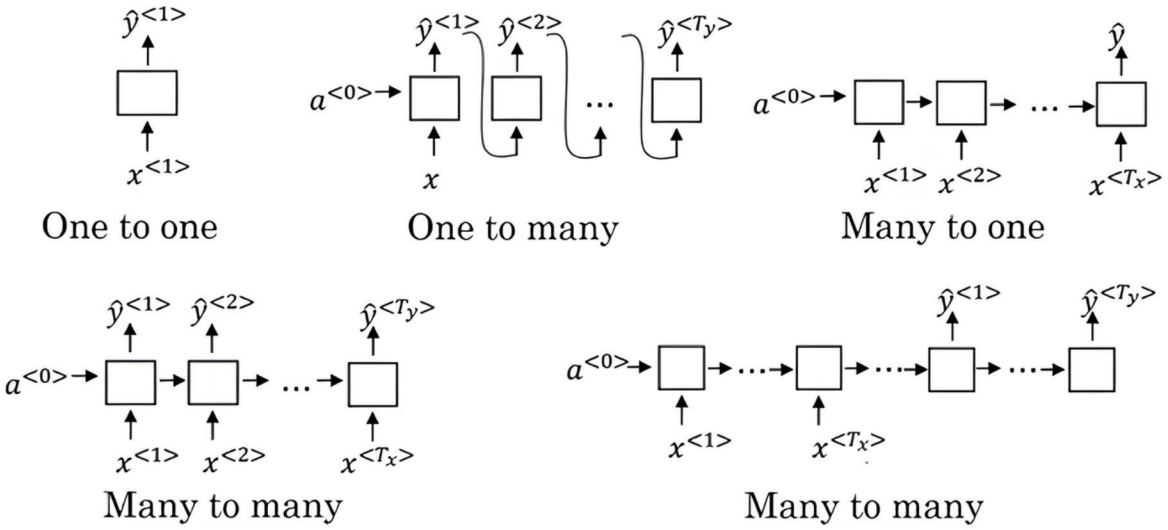


Figure 3.8: Types of Rnn [29].

RNN have been used for a variety of tasks, such as speech recognition, machine translation, and sentiment analysis[118], each task used a specific type of RNN such us (One-to-One, One-to-Many, Many-to-One, and Many-to-Many)[29] as explained on Figure 3.8.

RNN cell architectures

Several RNN cell architectures have been developed to address challenges such as capturing long-term dependencies and avoiding the vanishing gradient problem that is caused by RNN .therefore, LSTM (Long Short-Term Memory) was specifically designed to address the vanishing gradient problem and allow RNNs to effectively capture long-term dependencies. Here's how LSTM solves the issues faced by traditional RNN.

- ***Long Short-Term Memory*** **(*LSTM*)** LSTM is a type of recurrent neural network (RNN) that is designed to address the vanishing gradient problem and improve the memory capacity of the network [61]. LSTM cells have three gates (input, forget, and output) and a memory cell that stores the information, which allows them to selectively forget or remember information from previous time steps and maintain a long-term memory of the input sequence. LSTM is well-suited for processing long sequences of data and has been used for a variety of tasks, such as part-of-speech tagging for the Tamil language, cryptocurrency price prediction, and machine translation of the Holy Quran from Arabic to Italian [61] [56]. LSTM has also been used in combination with other deep learning models, such as attention mechanisms and stacked ensembles, to improve the performance of the network [56]. The computations within LSTM are represented by the following

equations [35].

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \tag{3.4}$$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \tag{3.5}$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \tag{3.6}$$

$$\tilde{C}_t = \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \tag{3.7}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{3.8}$$

$$h_t = o_t \odot \tanh(C_t) \tag{3.9}$$

Here The forget gate shown on $(Eq2.4)$ determines the amount of information from the previous memory cell state (C_{t-1}) that should be forgotten. It takes as input the previous hidden state (h_{t-1}) and the current input (x_t) and passes them through a linear transformation (W_{fh} and W_{fx}) followed by a sigmoid activation function (σ), The resulting value between 0 and 1 for each number in the cell state (C_{t-1}),(ft) represents the degree of forgetting.
the next step is to choose what the new data will be stored in the cell state (C_t), This step represented by $(Eq2.5)$ and $(Eq2.7)$ can be divided into two parts. First, a sigmoid function (σ) called the input gate (i_t) decides which values will be updated, The output of this gate (i_t) represents the amount of new information to be added to the memory cell. Next, a hyperbolic tangent $(tanh)$ function which ranges from -1 to 1, creates a vector of new candidate values $(\tilde{C}_t)$, Then The memory cell (C_t) is updated by combining the previous memory cell state (C_{t-1}) and the candidate activation $(\tilde{C}_t)$.In the $(Eq2.7)$ the element-wise multiplications is used between the old state (C_{t-1}) and The forget gate (f_t) combined with candidate update element's $(\tilde{C}_t)$ and the input gate (i_t) ,this resulting value (C_t) represents the updated memory cell state $(Eq2.8)$.
The output gate (o_t) like both (forget gate and input gate) uses the old hidden state combined with the current input (x_t) and passes the result through a sigmoid activation function (σ) . Finally, The hidden state (h_t) is computed by applying $tanh$ to the memory cell state (C_t), The resulting value multiply with the output gate (o_t).

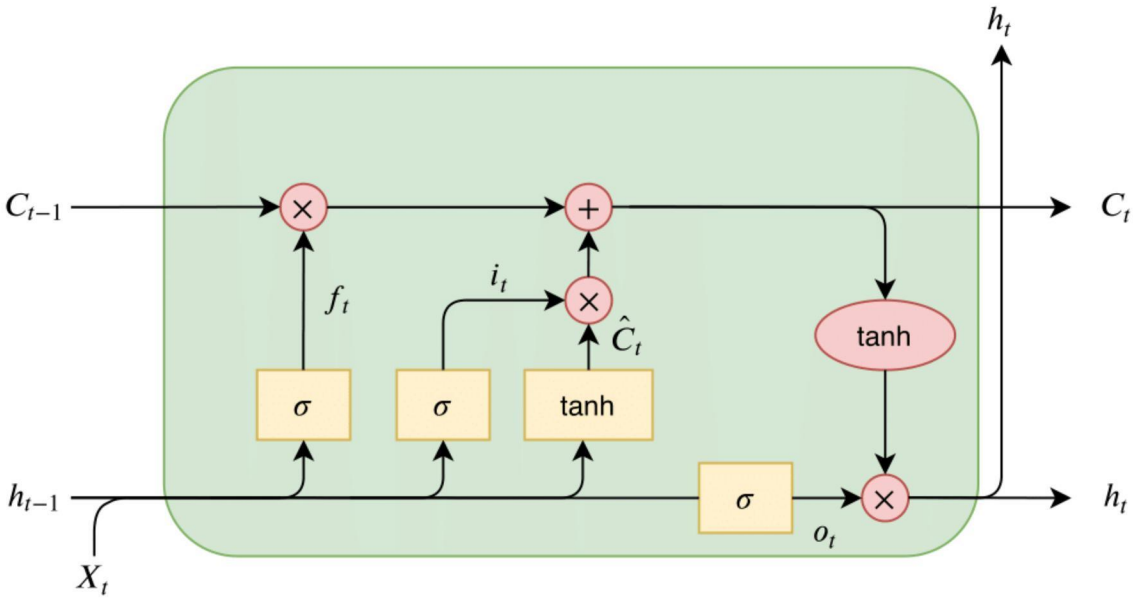


Figure 3.9: LSTM Cell [60].

the Figure 3.9 shows a clear similarity between the updates for these three different gates (forget gate, input gate, and output gate) in their structure; they take the previous hidden state (h_{t-1}) and current input (x_t) , the only difference between those gates that they have distinct weight matrices $(W_{fh}, W_{fx}, W_{ih}, W_{ix}, W_{oh}, W_{ox})$ respectively, and biases can be seen in equations that are connected to each gate allow for flexibility in learning and adapting to different tasks [60]. In particular, all three gates very much resemble simple RNNs, but with the key difference being the sigmoid activation function, rather than the tanh activation function, This choice of activation function allows the gates to regulate the information flow by controlling the amount of forgetting, storing, and outputting of information in a more effective manner.[35].

- ***Bidirectional LSTM***
  BiLSTM is a recurrent neural network used primarily on natural language processing such as handwriting recognition [52], and speech recognition [51]. Unlike standard LSTM As the name suggests The input has the ability to flow bidirectionally and leverage information from both directions, making it a potent technique for representing the sequential relationships of words and phrases in a sequence[52, 51]. as illustrated in Figure 3.10, The BiLSTM incorporates an extra LSTM layer that introduces a reverse direction of information flow in the input sequence. This is achieved by processing the sequence in reverse order in the additional LSTM layer. Various methods, including averaging, summing, multiplying, or concatenating, are used to merge the outputs from both LSTM layers[120].

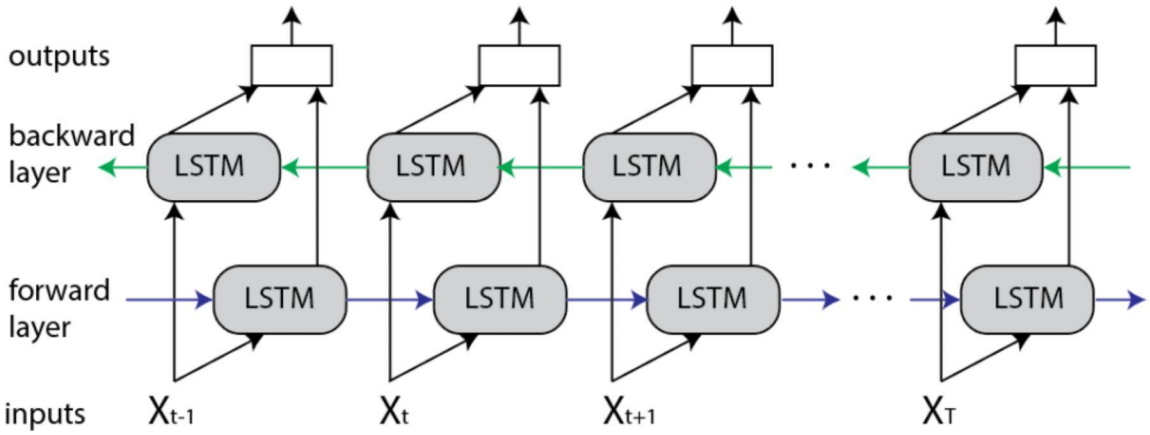


Figure 3.10: BiLSTM Cell [120].

3.5 Attention Mechanism

Attention mechanisms have emerged as a critical component in modern deep learning models, revolutionizing various domains such as natural language processing, computer vision, and speech recognition [10, 106]. In 2014, was designed to address the challenge posed by fixed-length encoding vectors, which restrict the decoder's access to information from the input. this technique is employed to address the limitations of traditional models that treat all input elements equally, regardless of their relevance or importance. By selectively attending to specific parts of the input, attention mechanisms allow models to focus on the most informative elements, improving their ability to understand and process complex data [112]. The notion behind such adaptability is that the network should allocate greater attention to the subtle yet pivotal portions of the input data, as this dynamic adjustment may enhance the representation of some elements while diminishing the relevance of others. The introduction of attention mechanisms in the context of machine translation can be credited to the work of Bahdanau et al. (2014), who proposed a neural machine translation model that learns to align and translate by attending to relevant parts of the source sentence. This pioneering approach laid the foundation for subsequent research and has since been widely adopted in various deep-learning architectures. Attention mechanisms, by enabling the flexible modeling of dependencies and capturing intricate patterns in the data, have significantly enhanced the performance of these models on a wide range of challenging tasks.

The development of the attention mechanism was primarily motivated by the challenges posed by time-varying data, particularly sequences. This led to the increasing adoption of a holistic approach, treating sequences as a whole rather than fixed-length entities. Within the attention framework, two prominent types of attention have emerged: additive attention and dot-product attention. Additive attention can be con-

sidered as the original attention mechanism, while dot-product attention represents a more recent innovation [73].

Today, the attention mechanism has found extensive applications in various domains, including image captioning and other related tasks. Its ability to selectively attend to relevant information based on contextual cues, trained using gradient descent algorithms, makes it a powerful tool in understanding and processing complex data [112].

3.5.1 How Attention Mechanism Works

In traditional neural network architectures, the model processes the complete input sequence as a whole, which can be problematic for lengthy sequences as it may lead to information overload and the loss of crucial data. However, attention mechanisms address this issue by enabling the model to selectively focus on specific segments of the input sequence, allowing it to prioritize the most relevant parts. The attention mechanism employs attention weights to determine the importance of each element in the input sequence. Subsequently, these weights are used to compute a weighted sum of the input sequence, where the weights represent the significance of each component [106]. By incorporating attention mechanisms, models can effectively capture long-range dependencies and enhance their ability to process sequential data. One widely used attention mechanism is the scaled dot-product attention, introduced in the article "Attention Is All You Need [106]" by Vaswani et al.

In the ensuing section, a comprehensive analysis will be presented, elucidating the distinct types of attention mechanisms along with a detailed exposition of their individual operational principles.

3.5.2 Types of Attention Mechanism

According to the method used to calculate the attention weights, there are many kinds of attention mechanisms. The most popular varieties are listed below:

1. **Dot-product attention :** Dot product attention is a fundamental mechanism used in neural network architectures for capturing relevant information within input sequences. The attention weights in dot product attention are computed by taking the dot product between the query vector and the key vectors, followed by a soft-max operation. Mathematically, the attention weights can be expressed as an equation :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(QK^T\right)V \tag{3.10}$$

where Q represents the query vector and K represents the key vectors. The calculation involves multiplying the query vector with the transpose of the key vectors and applying the softmax function to normalize the resulting scores. This mechanism enables the model to assign higher weights to the key vectors that have higher relevance or similarity to the query. The dot product attention mechanism has been widely employed in various tasks such as machine translation, sentiment analysis, and question answering, showcasing its effectiveness in capturing important contextual information and improving overall model performance [106, 10].

2. **Scaled Dot-Product Attention (Self-attention) :** Self-attention, also known as intra-attention, is a specific type of attention mechanism where the input sequence attends to itself. It has gained significant popularity, especially with the introduction of the transformer model [106]. In self-attention, each element in the input sequence, such as a word or a token, is treated as both a query and a key. This is an extension of dot product attention where the dot product is divided by the square root of the dimension of the query vector. Scaling helps prevent the dot products from becoming too large or too small, which can affect the gradient during training. The aim is to capture the relationships between different elements within the same sequence. By attending to itself, self-attention allows the model to weigh the importance of each element based on its relationships with other elements, enabling the model to focus on relevant information and effectively model dependencies. The self-attention mechanism typically involves the following steps:

   - **Query, Key, and Value:** The input sequence is transformed into three representations: queries (Q), keys (K), and values (V). These transformations are learned through linear transformations as illustrated in Figure 3.11

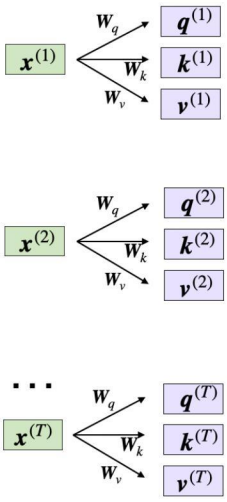


Figure 3.11: attention-matrices [88].

where weight matrices (Wq, Wk, and Wv) are applied to the input sequence. The respective query, key, and value sequences are obtained via matrix multiplication between the weight matrices **W** and the embedded inputs **x**:

$$\begin{aligned} \text{Query sequence:} \quad & \mathbf{q}^{(i)} = \mathbf{W}_q \mathbf{x}^{(i)} \quad \text{for } i \in [1, T] \\ \text{Key sequence:} \quad & \mathbf{k}^{(i)} = \mathbf{W}_k \mathbf{x}^{(i)} \quad \text{for } i \in [1, T] \\ \text{Value sequence:} \quad & \mathbf{v}^{(i)} = \mathbf{W}_v \mathbf{x}^{(i)} \quad \text{for } i \in [1, T] \end{aligned}$$

The index i refers to the token index position in the input sequence, which has length T. for more comprehension, the Figure 3.12 presents the Input transformation used in a Transformer, Query, Key, and value

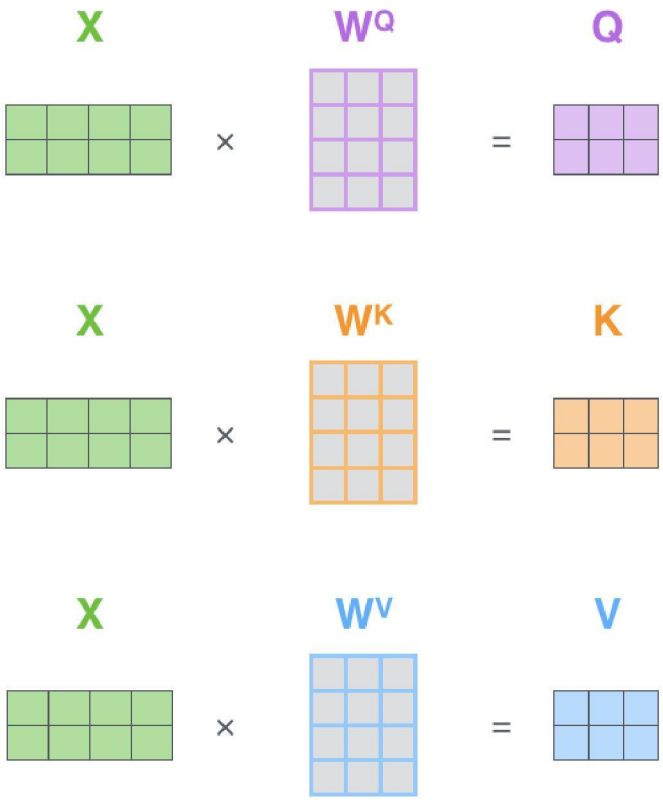


Figure 3.12: self-attention matrix calculation [25].

- **Calculating Attention Scores:** Attention scores are computed by taking the dot product between the query vector of each element and the key vector of all other elements in the sequence. This calculation captures the similarity or relevance between different elements in the sequence. Mathematically, the attention score between elements i and j can be expressed as:

$$\text{Score}(i, j) = \frac{Q_i \cdot K_j^T}{\sqrt{d_k}}$$

where Q_i represents the query vector for element i, K_j represents the key vector for element j, and d_k represents the dimension of the query and key vectors.

- **Attention Weights:** The attention scores are scaled by the square root of the dimension of the query and key vectors to ensure stability. Then, the scaled scores are passed through a softmax function, which normalizes the scores to obtain attention weights (SeeFigure 3.13). The attention weights reflect the importance or contribution of each element to the others. Mathematically,

the attention weight for element j given a query i can be calculated as:

$$\text{Weight}(i,j) = \text{softmax}(\text{Score}(i,j)) = \frac{\exp(\text{Score}(i,j))}{\sum_{j=1}^{T} \exp(\text{Score}(i,j))} \tag{3.11}$$

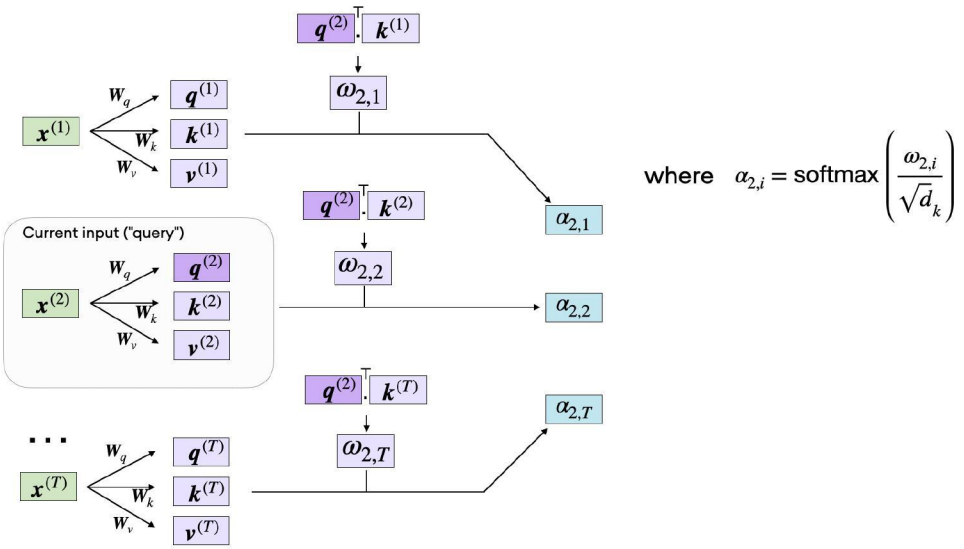


Figure 3.13: Calculation attention weight [88].

- **Weighted Sum:** The attention weights are used to calculate a weighted sum of the values (V). Each value is multiplied by its corresponding attention weight, and the weighted values are summed up to obtain the final attended representation for each element. the following equation presented the calculation of the weighted Sum:

$$\text{Weighted Sum}(i) = \sum_{j=1}^{T} \text{Weight}(i,j) \cdot \text{Value}(j) \tag{3.12}$$

the Figure 3.14 explain how to compute the context vector $z^{(2)}$, which is an attention-weighted version of our original query input $x^{(2)}$, including all the other input elements as its context via the attention weights

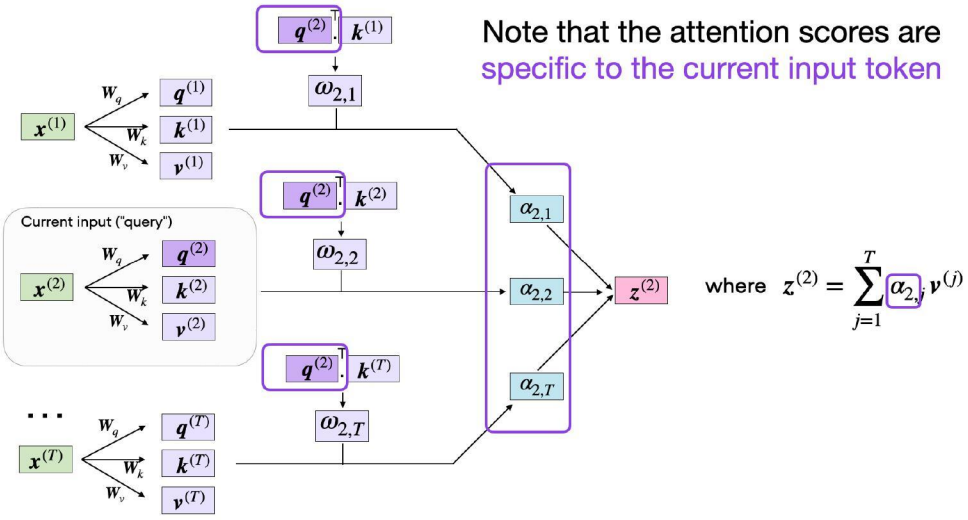


Figure 3.14: Calculation Weighted Sum [88].

The self-attention mechanism allows the model to capture long-range dependencies and consider the contextual information of each element within the input sequence. It has been proven effective in various natural language processing tasks, such as machine translation, sentiment analysis, and question-answering [106]. this technique has been used in various types of models, including vision transformers[81], speech enhancement networks, and video question-answering models [110]. It has also been used to replace recurrent neural networks in video question-answering tasks [110]. this type of attention mechanism has been shown to improve the performance of these models and reduce the computational cost of training them.

The self-attention mechanism, combined with the transformer architecture, has demonstrated state-of-the-art performance in several domains and has become a widely used technique in sequence modeling. Its ability to capture complex relationships within a sequence has led to advancements in tasks involving sequential data.

3. **Additive attention :** Additive attention, also known as Bahdanau attention, is a prominent attention mechanism widely used in sequence-to-sequence models, particularly in machine translation tasks. It was introduced by Dzmitry Bahdanau in a seminal paper [10] with the goal of improving alignment between the decoder and relevant input sentences as illustrated in Figure 3.15. It introduces an alignment model, which is a small neural network that learns to align the source and target sequences during the decoding process. The alignment model takes the decoder's previous hidden state (query) and the encoder's hidden states (keys) as inputs. Through a feed-forward neural network with shared

parameterization, the alignment model computes a score for each key, indicating its relevance to the query. Mathematically, the score can be computed using the equation:

$$\text{Score}(Q, K) = \mathbf{v}^\top \cdot \tanh(\mathbf{W}_1 \cdot Q + \mathbf{W}_2 \cdot K) \tag{3.13}$$

where Q represents the query vector, K represents the key vectors $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices, and v is a weight vector. The scores are then passed through a softmax function to obtain the attention weights. The weighted sum of the encoder's hidden states, multiplied by their corresponding attention weights, is used as context information for the decoder. The Bahdanau attention mechanism enables the model to dynamically attend to different parts of the input sequence as needed, enhancing its ability to capture dependencies and align information during generation. This attention mechanism has been successfully applied in various tasks, such as machine translation and text summarization [10]. The attention weights obtained through the Bahdanau mechanism provide interpretable insights into which parts of the input sequence the model is focusing on, aiding in understanding the model's decision-making process.

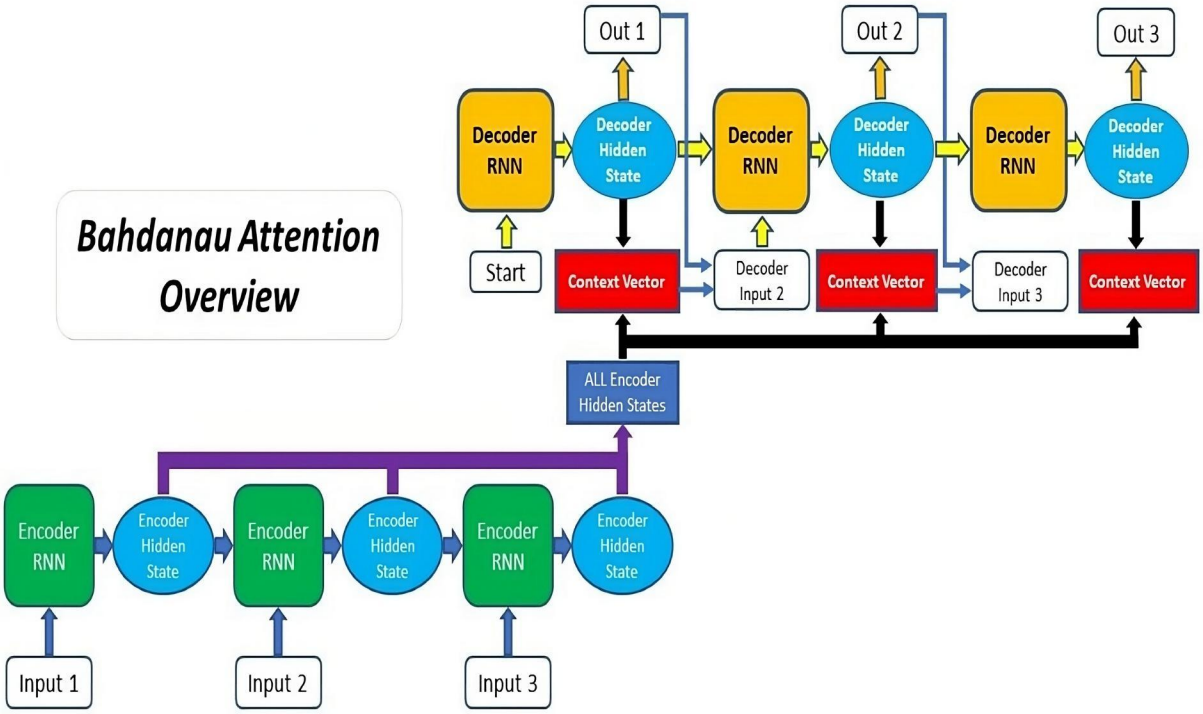


Figure 3.15: Overall process for Bahdanau Attention [72].

4. **Multiplicative attention :** Multiplicative attention, also known as Luong attention, is a variant of the attention mechanism proposed by Thang Luong [73] as illustrated in Figure 3.16. Luong attention is similar to Bahdanau attention, but instead of using a feedforward neural network to calculate the alignment score, it

uses a dot product between the decoder's current hidden state and the encoder's hidden states[11]. There are different ways to define the score function, depending on the specific variant of Luong attention being used. The three commonly used variants are:

- ***Dot product attention:***

$$\text{score}(h_i, \bar{h}_j) = h_i^\top \bar{h}_j \tag{3.14}$$

- ***General attention:***

$$\text{score}(h_i, \bar{h}_j) = h_i^\top W_a \bar{h}_j \tag{3.15}$$

  Here, W_a is a learnable weight matrix.

- ***Concatenation attention:***

$$\text{score}(h_i, \bar{h}_j) = v_a^\top \tanh(W_a[h_i; \bar{h}_j]) \tag{3.16}$$

  In this case,W_a and v_a are learnable weight matrices, and $[h_i; \bar{h}_j]$ represents the concatenation of h_i and $\bar{h}_j$

in the above score equations (2.14, 2.15, 2.16), h_i represents the hidden state of the decoder at time step i, and $\bar{h}_j$ represents the hidden state of the encoder at time step j.

Once the attention scores are computed, they are normalized by the softmax function to obtain the attention weights. These weights determine the importance of each encoder's hidden state with respect to the decoder's hidden state at a given time step. The attention weights are used to weight the encoder's hidden states to produce the context vector, which is further combined with the decoder's hidden state to make predictions or generate outputs. The Luong attention mechanism enables the model to selectively focus on different parts of the input sequence, allowing it to capture relevant information for the task at hand. By attending to specific encoder hidden states, the model can effectively align the source and target sequences and make more accurate predictions or generate coherent outputs [73].

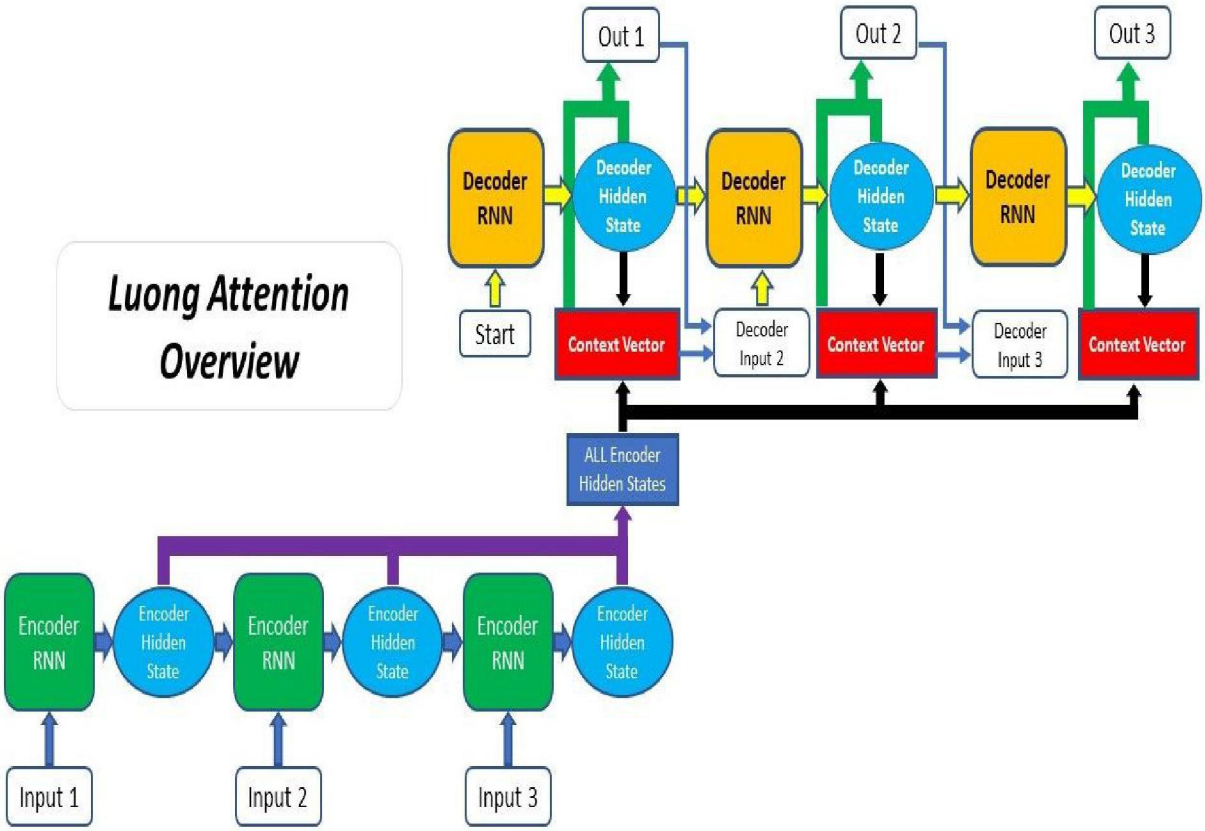


Figure 3.16: Overall process for Luong Attention [72].

5. **Multi-head attention :** Multi-head attention is a mechanism that allows the model to attend to different parts of the input sequence simultaneously, as presented in Figure 3.17. By dividing the queries, keys, and values and projecting them into multiple subspaces using different learned linear transformations. Each transformation corresponds to a different attention "head" as illustrated in Figure 3.17 in FC (Fully Connected Layer), and computing the attention weights for each head individually is done by the following equation:

$$\text{Head}_i = \text{Attention}(\text{Query}_i, \text{Key}_i, \text{Value}_i) \tag{3.17}$$

where $Query_i$,Key_i, and $Value_i$ are the query, key, and value vectors projected for the attention ith attention head, respectively. this step enables the model to attend to various portions of the input sequence at once.

The outputs from all the attention heads are then concatenated and linearly transformed to obtain the final multi-head attention output:

$$\text{MultiheadAttention} = \text{Concatenate}(\text{Head}_1, \text{Head}_2, ..., \text{Head}_N) \cdot \mathbf{W}_O \tag{3.18}$$

where W_o is a learnable weight matrix.

The multi-head attention mechanism provides the model with the ability to capture various patterns and dependencies in the input sequence, leading to improved performance in tasks such as machine translation, language modeling, and sequence generation. Its parallel nature allows for efficient computation and enables the model to learn complex interactions within the sequence, leading to enhanced representation learning [106].

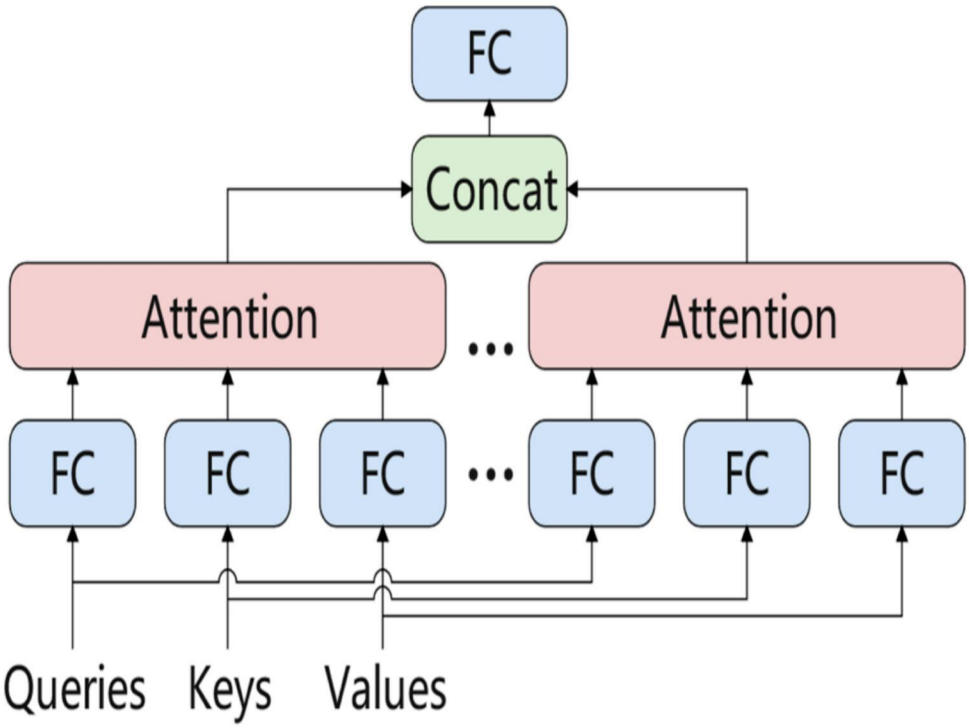


Figure 3.17: Multihead attention mechanism [114].

3.5.3 Applications of Attention Mechanism

The attention mechanism is a technique used in deep learning to improve the performance of neural networks in various applications. One of the applications of attention mechanisms is remote sensing image processing. this technique has been used to improve the performance of deep learning methods in remote sensing image processing [47]. Researchers have developed new architectural designs of networks and techniques to integrate attention mechanisms with different deep learning neural network architectures[47]. also used to improve the accuracy of synthetic aperture radar automatic target recognition (SAR ATR) [117]. The proposed attention mechanism CNN (AM-CNN) model for SAR ATR achieved a superior recognition performance, with an average recognition accuracy of 99.35% on the classification of 10 class targets [117] On the other hand has been applied in speech-emotion recognition (SER) modules, which play an important role in the development of human-computer inter-

action (HCI) applications[70]. Attention-based deep neural networks (DNNs) have been shown to be suitable tools for mining information that is unevenly time distributed in multimedia content[70]. The attention mechanism has been incorporated into DNN architectures to emphasize emotional salient information, improving the performance of SER systems[70]. Another application of the attention mechanism is in drug discovery. Attentive FP is a new graph neural network architecture that uses a graph attention mechanism to learn from relevant drug discovery datasets [111]. Attentive FP achieved state-of-the-art predictive performances on a variety of datasets and what it learns is interpretable[111]. The feature visualization for Attentive FP suggests that it automatically learns non-local intramolecular interactions from specified tasks, which can help gain chemical insights directly from data beyond human perception[111]. In summary, the attention mechanism is a powerful technique used in deep learning to improve the performance of neural networks in various applications, including remote sensing image processing, SAR ATR, SER modules, and drug discovery.

3.6 Siamese Networks

The term "Siamese" alludes to the famous conjoined twin's Chang and Eng Bunker, often known as Siamese twins, who had identical features[86]. Siamese neural networks try to determine how similar two inputs are by comparing their feature representations [86]. This method uses a specific kind of neural network design that comprises two or more subnetworks that are exactly alike, have the same weights, and are trained to do the same task. The parallel subnetworks are in charge of generating vector representations of the inputs. The output vectors are then compared using a distance function to determine their similarity or dissimilarity. One of the output vectors can be precomputed to form a baseline against which the other output vector is compared. Siamese neural networks are commonly used for tasks that involve comparing two inputs and determining their similarity or dissimilarity, such as biomedical ontologies, object tracking in satellite videos, image manipulation detection, and music recommendation systems. Siamese neural networks are particularly useful for tasks that involve small datasets or limited labeled data, as they allow for the efficient use of available data.

In the paper by Bromley et al "Signature Verification Using a Siamese Time Delay Neural Network" published in the early 1990s. were the first to introduce the idea of Siamese neural networks. The authors suggested an architecture for a neural network that could identify signatures by comparing pairs of signature photos. The network was made up of two identical subnetworks with common weights that were trained using pairs of distinct image examples. The network consisted of two identical subnet-

works that shared the same weights and were trained on pairs of signature images. The output of the network was a similarity score that indicated how similar the two signatures were. Since then, Siamese neural networks have been applied to various fields, including biomedical ontologies [98], object tracking in satellite videos, image manipulation detection, music recommendation systems[86], and cover song detection. In biomedical ontologies, a deep Siamese neural network model is used to predict the interactions between proteins and gene-disease associations[98]. In object tracking in satellite videos, a Siamese network and a motion regression network are adopted to form a two-stream deep neural network for satellite object tracking. In image manipulation detection, a deep Siamese convolutional neural network is used to differentiate between different types of image editing operations. In music recommendation systems, Siamese neural networks are used to determine the similarity between two audio clips and to find songs that match the user's tastes[86]. In cover song detection, Siamese network configurations are used to approach the cover song identification problem.

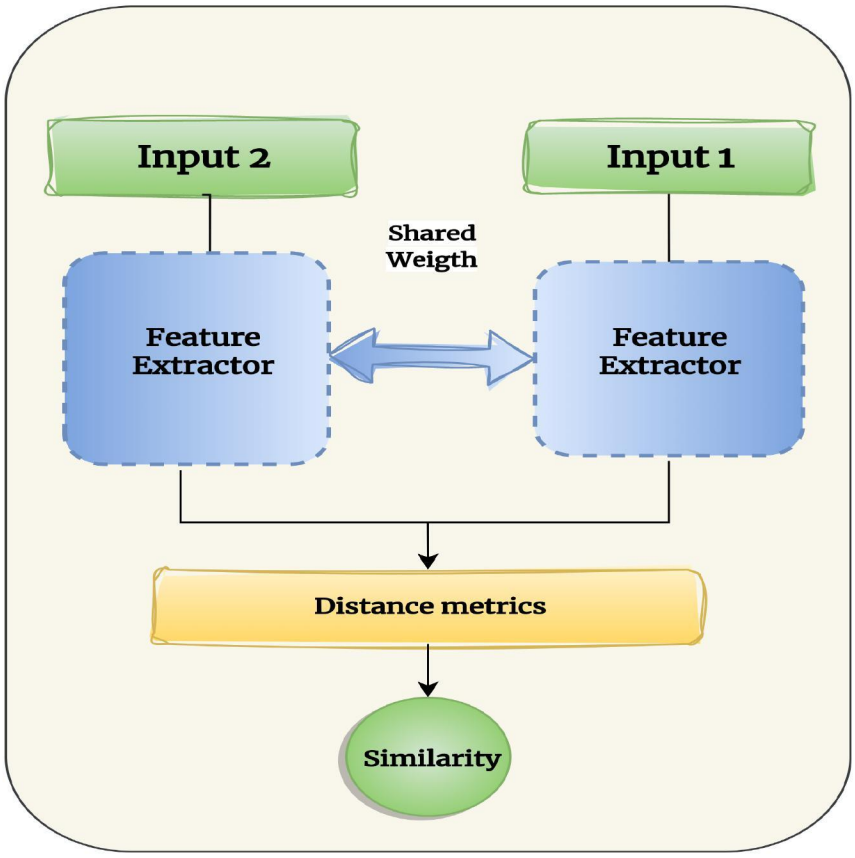


Figure 3.18: Architecture of Siamese Network.

Siamese neural networks have become increasingly popular in recent years due to their ability to perform well on tasks with limited labeled data. They have been used in a variety of applications, including biomedical ontologies, object tracking in satellite videos, image manipulation detection, and music recommendation systems. In conclu-

sion, siamese neural networks are particularly useful for tasks that involve comparing two inputs and determining their similarity or dissimilarity as shown on fig. 3.18

3.7 Summary

This chapter offers a comprehensive and rigorous examination of deep learning, a rapidly advancing field within artificial intelligence (AI). The chapter commences by introducing the fundamental concept of deep learning and its profound implications for solving intricate problems across diverse domains. Moreover, it establishes a clear and robust connection between deep learning and AI, underlining the foundational role of machine learning (ML) as the underlying framework for deep learning algorithms.

Moving forward, the chapter accurately explores the core architectures of deep learning. It starts with a thorough examination of feedforward neural networks (FNN), which constitute the groundwork for deep learning models. On the other hand, forward convolutional neural networks (CNN) are comprehensively expounded upon to showcase their exceptional proficiency in processing and identifying crucial information from visual data. Furthermore, recurrent neural networks (RNN) are extensively examined, highlighting their efficacy in handling sequential information and their relevance across diverse application domains. A noteworthy focal point of this chapter is the comprehensive investigation of attention mechanisms, an integral component of numerous deep learning models. The chapter provides an intricate and detailed exposition of the operational principles of attention mechanisms, shedding light on their capacity to selectively focus on pertinent information within a given context. Multiple types of attention mechanisms are exhaustively explored, uncovering their diverse applications and the enhancements they confer upon deep learning performance. Furthermore, the chapter introduces Siamese networks, a specialized class of neural networks that exhibit exceptional prowess in comparing and quantifying similarity between input pairs. The discussion underscores the practical implications of Siamese networks in various tasks, including image similarity assessment and facial recognition, substantiating their invaluable utility in real-world scenarios.

In essence, Chapter 3 constitutes an authoritative and erudite resource, endowing readers with a profound understanding of deep learning. Through its systematic and meticulous approach, the chapter comprehensively elucidates the foundational principles, architectures, and components of deep learning, thereby establishing a robust framework for the exploration of advanced techniques and their myriad applications. Its scrupulous analysis and articulate explanations render this chapter an indispensable asset for researchers, academics, and practitioners in the burgeoning field of AI and machine learning.

Research Methodology

Chapter 4

Research Methodology

4.1 Introduction

To assess the feasibility and dependability of the proposed models which is the prediction of molecular similarity, the present chapter provides details of our proposal and describes all the steps for constructing a DL model allowing for detecting molecular similarity, In general Molecular similarity deals with finding molecules that have a comparable amount of structural similarity [93], This is used to find structures that are similar to a molecule with less information. One of the main objectives presented in this research is to develop a method for predicting the similarity of chemical compounds and improve the performance of the virtual screening process. To accomplish this objective a new way of merging multi-deep learning techniques: convolutional neural network, recurrent neural network, and attention mechanism are proposed to form the proposed model for predicting similarity.

4.2 Research Framework

This section describes the general framework containing the development process. As depicted in Figure 4.1 the framework includes all the activities of the proposed approach for identifying the similarity of molecular compounds.

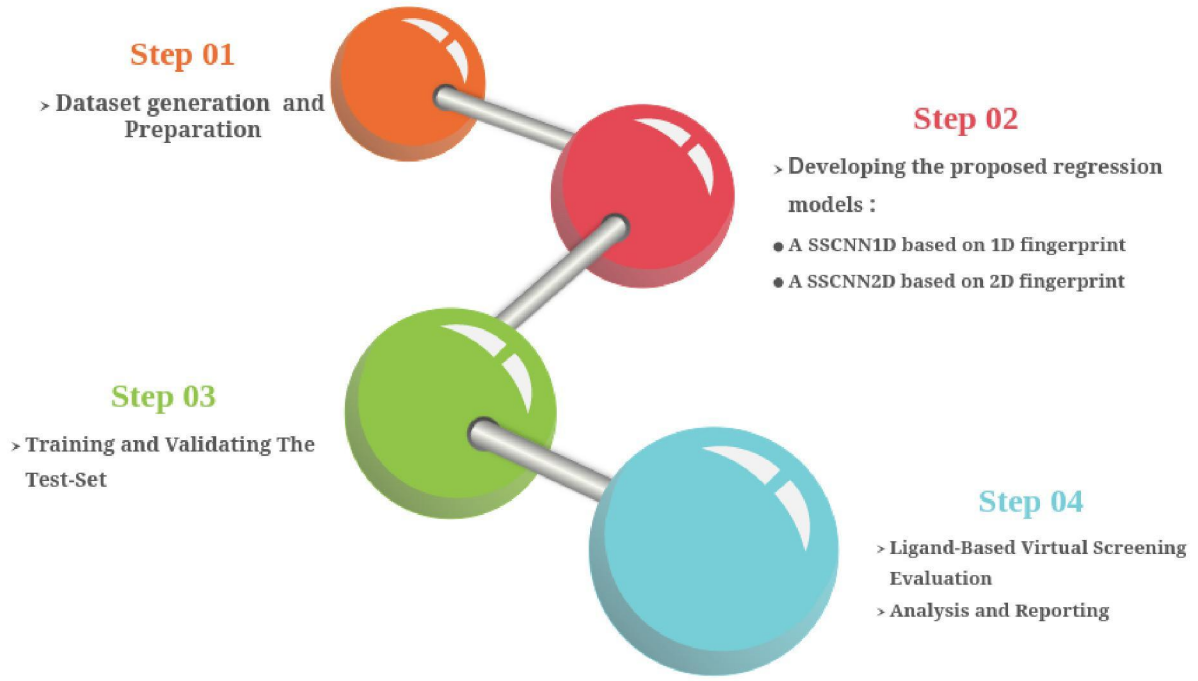


Figure 4.1: The general research framework.

4.3 Datasets

This study has estimated the quality of the proposed methods using multiple datasets that were used by several studies for validating the ligand-based virtual screening methods [16, 2]. A variety of legitimate databases assign biological activity to substances. However, since it was created in 1988, the MDDR database has remained one of the most widely used chemo-informatics databases. The 102,516 chemical compounds with various different actions are included in the MDDR database. Some of these compounds are connected to therapeutic areas like antihypertensives, while others, such as renin inhibitors, are connected to certain enzymes. Unlike other databases, the MDDR database has a small selection of activities from which to pick [16]. The MDDR database contains three data sets: DS1, DS2, and DS3. including two types of active compounds (heterogeneous and homogeneous), DS1 consists of 8294 structurally heterogeneous where they are structurally diverse ,and homogeneous bioactive molecules distributed on 11 Activity class. DS2 consists of 5083 homogeneous bioactive molecules with 10 classes and DS3 consists of 8560 heterogenous bioactive molecules with 10 classes also, The rest of the compounds in each dataset are inactive compounds(called also decoys), Full details of these datasets are given in Tables 4.1 - 4.2 -4.3. Each row in the tables contains an activity class, the number of molecules belonging to the class.

Index	Activity class	Active molecules
31 420	Renin inhibitors	1130
71 523	HIV protease	750
37 110	Thrombin inhibitors	803
31 432	Angiotensin II AT1antagonists	943
42 731	Substance P antagonists	1246
6233	Substance P antagonists	752
6245	5HT reuptake de inhibitors	359
7701	D2 antagonists	395
6235	5HT1A agonists	827
78 374	Protein kinase C inhibitors	453
78 331	Cyclooxygenase inhibitors	636

Table 4.1: Structure of DS1 dataset.

Index	Activity class	Active molecules
7707	Adenosine (AI) agonists	207
7708	Adenosine (A2) agonists	156
31420	Renin inhibitors 1	1300
42710	CCK agonists	111
64220	Carbacephems	1051
64500	Carbapenems	126
64350	Tribactams	388
75755	Vitamin D analogues	455

Table 4.2: Structure of DS2 dataset.

4.4 Design And Steps

4.4.1 Dataset Preprocessing

The data preparation phase was crucial for achieving accurate results in the training phase of the deep learning network. The deliverable of this stage included a detailed study of the current prediction of similarity search techniques and existing molecule representations in chemoinformatic research, as well as the chemical dataset preparation process, which is discussed in this section.

In the data preparation phase of this research, we aimed to generate compound pairs for training the proposed models to predict similarity from molecular fingerprints obtained from ECFC_4 fingerprint generated by Scitegics Pipeline Pilot and PaDEL descriptor software. and creating a new dataset of labeled pairs of molecular fingerprints indicating similarity scores to represent whether pairs of molecules belong to the same biological activity, The following Algorithm 1 describes the generation process for the pairs

Index	Activity class	Active molecules
9249	Muscarinic (M1) agonists	900
12 455	NMDA receptor antagonists	1400
12 464	Nitric oxide synthase inhibitors	505
31 281	Dopamine-β-hydroxyl inhibitors	106
43 210	Aldose reductase inhibitors	957
71 522	Reverse transcriptasede	700
75 721	Aromatase inhibitors	636
78 331	Cyclooxygenase inhibitors	636
78 348	Phospholipase A2 inhibitors	617
78 351	Lipoxygenase inhibitors	2111

Table 4.3: Structure of DS3 dataset.

dataset by comparing each activity class of two molecule indexes. If both molecules have the same activity class, the similarity between them is 1, but if they have different activity classes, it is 0 as illustrated on Figure 4.2

Before passing the data to the proposed models , One last additional step is performed, It consists eliminating from the training dataset a ten active compounds known as queries, from each activity class ,which they reserved for the virtual screening process on the evaluation phase,Queries represent compounds with significant importance during the screening process. To ensure accuracy when evaluating virtual screenings, we exclude queries from the training phase for an authentic assessment of their detection capabilities.
Through this methodology, we can determine how well our model generalizes and makes accurate predictions on unseen compounds similar to queries. By measuring effectiveness independently on hidden queries, we acquire dependable information for a practical application of virtual screenings [64].

The next step is to balance the datasets so that the number of samples in each activity class is roughly equal. There are more pairs of dissimilar compounds in the new dataset that was generated as shown on Figure 4.3. In this case, all the similar pairs were selected randomly. In addition to choosing a group of dissimilar pairs in a random manner, this makes the dataset balanced in terms of the number of samples (Algorithm 2). The goal is to ensure that the model receives sufficient exposure to all classes and can learn meaningful patterns from both the majority and minority classes Figure 4.3.

4.4.2 Construction of Features Extractor model

The integration of deep convolutional neural networks (DCNN) into the field of ligand-based virtual screening (LBVS) is a relatively new development. As such, there

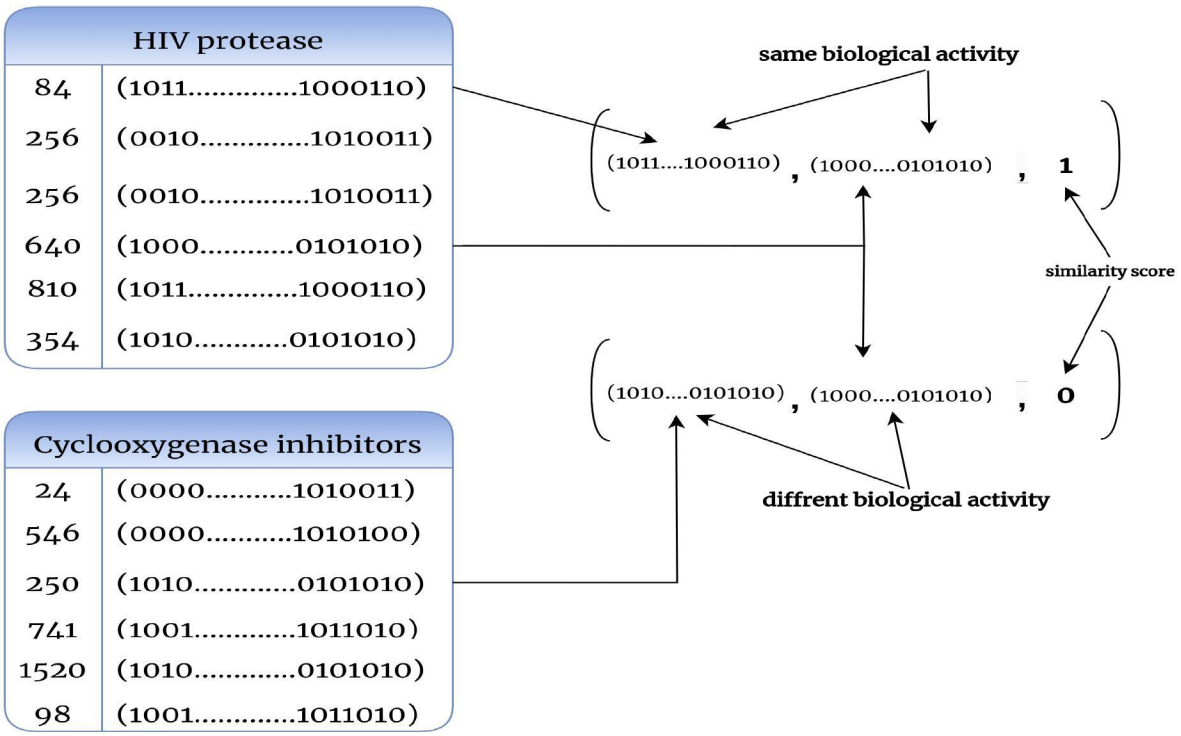


Figure 4.2: pairwise dataset generation.

exists a significant potential for further research and exploration in order to develop novel methods and techniques based on deep learning that can effectively address challenges within the LBVS process. [16].

A DCNN architecture can contain an alternation of convolution and pooling layers, using different parameters such as filter size and depth. This allows for the successful extraction of higher-level features, which is considered one of the strong points of DCNNs. In the final step, the last subsampling layer can be reduced to a single vector containing all its weights and linked to a fully connected layer. This layer is then attached to the output layer, which contains a field that provides an estimated prediction of the similarity value of the input compound in the DCNN model. One way to improve the DCNN model's ability to identify relevant or essential features from the molecular input representation is to incorporate the self-attention layer into the architecture presented in Fernández-Llaneza.D works [40]. Self-attention layer enables the model to weigh the importance of each feature in the input sequence relative to the others, allowing the model to focus on the most relevant information. By incorporating self-attention as a layer into the DCNN architecture, the model can better identify essential chemical features and improve the accuracy of the LBVS process.

4.4.3 Design architecture of our Siamese attentional models

In this section, we describe the development of the two Siamese models with sep-

Algorithm 1: Generating pairwise dataset.

Input : querie_values
Output: mol_pairs

```
mol_pairs ← { "Mol1"← [], "classMol1"← [] , "classMol2"← [] ,
  "similarity"← []}
mols_Indexes ← {"index"← [], "class"← []}
for each molecule index x in the range of active compounds do
    for each molecule index z in the range from x + 1 to the number of active
      compounds do
        mol_pairs["Mol1"].Append( mols_Indexes["index"][x])
        mol_pairs["Mol2"].Append(mols_Indexes["index"][z])
        mol_pairs["class_m1"].Append( mols_Indexes["class"][x])
        mol_pairs["class_m2"].Append( mols_Indexes["class"][z])
        if mols_Indexes["class"][x] is equal to mols_Indexes["class"][z] then
            mol_pairs["similarity"].Append(1)
        end
        else
            mol_pairs["similarity"].Append(0)
        end
    end
    pairs_filtered ← DataFrame(sim_mole)
    pairs_filtered['Mol1'] ← Excluding(pairs_filtered['Mol1'],querie_values)
    pairs_filtered['Mol2'] ← Excluding(pairs_filtered['Mol2'],querie_values)
end
return pairs_filtered
```

arate input layers for each fingerprint. The inputs are processed through the shared feature extractor network, and the output layers are connected accordingly,The feature extractor based on recurrent neural network by choosing the long short-term memory (LSTM) cells architecture with a self-attention mechanism comprises of several convolutional neural networks (CNN1D and CNN2D) layers, several distance layers (ecludiean, manhattan, exponential manhattan) topologies to find a learning model that can initially provide better performance.

The architecture of our SSCNN1D based on 1D fingerprint

The Siamese model consists of two identical feature extractor networks that take as input two molecular 1D fingerprints of length 1024 each , aims to predict the similarity between the two compounds.
The feature extractor network is a sequential model consisting of several layers. The input fingerprints, with a shape of (1024,), are reshaped to (1024, 1) to match the expected

Algorithm 2: Balancing pairwise dataset.

Input : mol_pairs
Output: balanced_pairs

Initialize an empty dictionary "balanced_pairs" with columns 'Mol1', 'Mol2', 'classMol1', 'classMol2', and 'similarity'
ones_arr ← Filter the mol_pairs to select rows where 'similarity' is equal to 1
ones_len ← length(ones_arr)
zeros_arr ← Filter the mol_pairs to select rows where 'similarity' is equal to 0
zeros_arr ← shuffle_rows_randomly(zeros_arr)
zeros_arr ← zeros_arr[:ones_len]
balanced_pairs ← Concatenate(ones_arr,zeros_arr)
balanced_pairs ← shuffle_rows_randomly(balanced_pairs)
return $balanced_pairs$

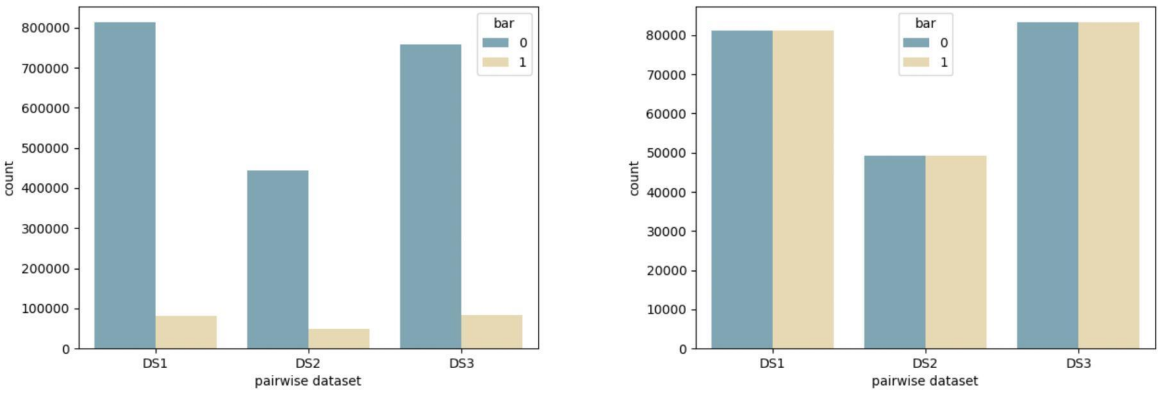


Figure 4.3: balancing all The Three dataset.

input format. The Conv1D layers apply 64 filters with a size of 3 and use the ReLU activation function to capture important features from the fingerprints. Padding is used to ensure that the output retains the same dimensions as the input. Subsequently, MaxPooling1D layers perform down-sampling by selecting the maximum value over a window of size 2, which helps in reducing the spatial dimensionality of the features. Further Conv1D layers with reduced filters are applied to extract additional features. The Flatten layer is used to transform the multidimensional feature maps into a one-dimensional vector. A Dense layer with 128 units is introduced to capture higher-level representations of the features.

After reshaping the features to have a shape of $(-1, 128)$, they are passed into an LSTM layer with 64 hidden units. The return_sequences parameter is set to True to obtain the output sequence from the LSTM layer. Dropout regularization with a rate of 0.15 is applied to prevent overfitting.

To incorporate self-attention, the output from the LSTM layer is fed into the Self-

Attention layer, which applies the tanh activation function to extract meaningful information by assigning weights to different elements of the sequence.

The output of the Self-Attention layer is flattened again, This layer serves as a fusion point for combining the features extracted from the input fingerprint,Figure 4.4 shows the different components of the feature extractor.

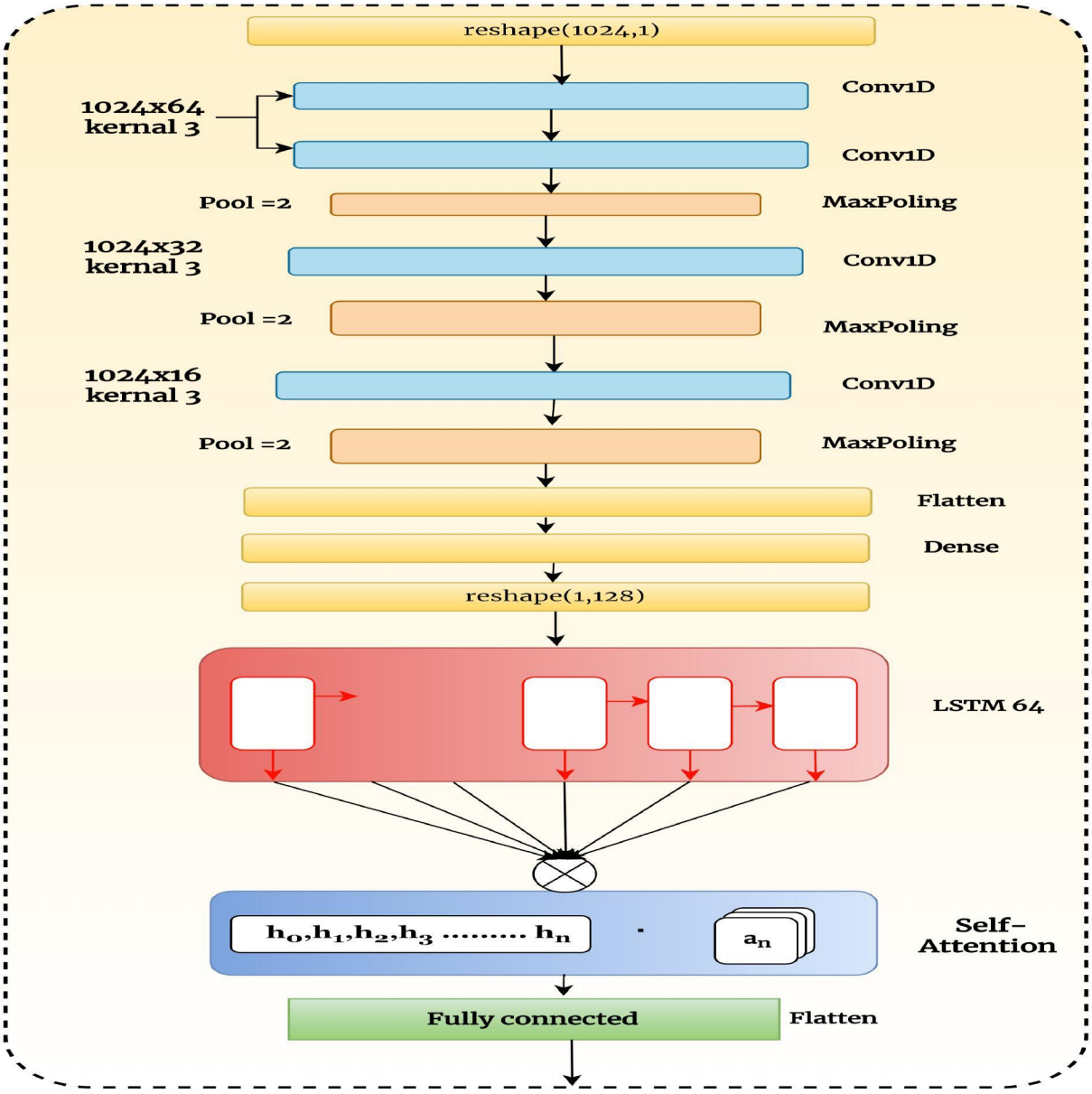


Figure 4.4: 1D Feautre extractor.

The distance between the features extracted from the two input compounds is computed using the Euclidean distance function, which can be represented as

$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4.1}$$

This distance represents the difference between the corresponding elements of the two feature vectors, then passed through a Dense layer with a ReLU activation function. Finally, a Dense layer with a single unit and a sigmoid activation function produce a probability score indicating the likelihood of the two molecules being a match. All details of the proposed SSCNN1D model for similarity searching in LBVS are shown and summarized in Figure 4.5 and Table 4.4.

Index	Layer type	Number of filters	kernel	Parameters	shape
1	Input	-	-	0	(None,1024)
2	Input	-	-	0	(None,1024)
3	Reshape	-	-	0	(None, 1024, 1)
4	Conv1D	64	3	256	(None, 1024, 64)
5	Conv1D	64	3	12352	(None, 1024, 64)
6	MaxPooling1D	-	2	0	(None, 512, 64)
7	Conv1D	32	3	1568	(None, 512, 32)
8	MaxPooling1D	-	2	0	(None, 256, 32)
9	Conv1D	16	3	6208	(None, 256, 16)
10	MaxPooling1D	-	2	0	(None, 128, 16)
11	Flatten	-	-	0	(None, 2048)
12	Dense	-	-	524352	(None, 128)
13	Reshape	-	-	0	(None, 1, 128)
14	LSTM	-	-	33024	(None, 1, 64)
15	self-Attention	-	-	4161	(None, 1, 64)
16	Flatten	-	-	0	(None, 64)
17	euclidean_distance	-	-	0	(None,1)
18	Dense	-	-	20	(None, 10)
19	output	-	-	11	(None, 1)
	Total params: 389,712 Trainable params: 389,712 Non-trainable params: 0				

Table 4.4: Architecture and parameters of SSCNN1D model.

The architecture of the SSCNN2D based on 2D fingerprint

In a second step, as mentioned previously, we have proposed several network typologies to find a learning model that can initially provide better performance for the process of predicting the similarity of compounds, we proposed a second architecture based on 2D fingerprints, the feature extractor divided into three parts which take two 2D vectors of size 32x32 as inputs representing two different molecules (see Figure 4.6). The first part of the feature extractor starts with two convolutional layers (CNN2D) with 16 filters that help to extract and combine features from different regions, both with a ReLU activation function. The max-pooling layer with a pool size of (2,2) is applied next, followed by a third CNN2D layer with 16 filters and ReLU activation, and another max-pooling layer with the same pool size as before, which helps to downsam-

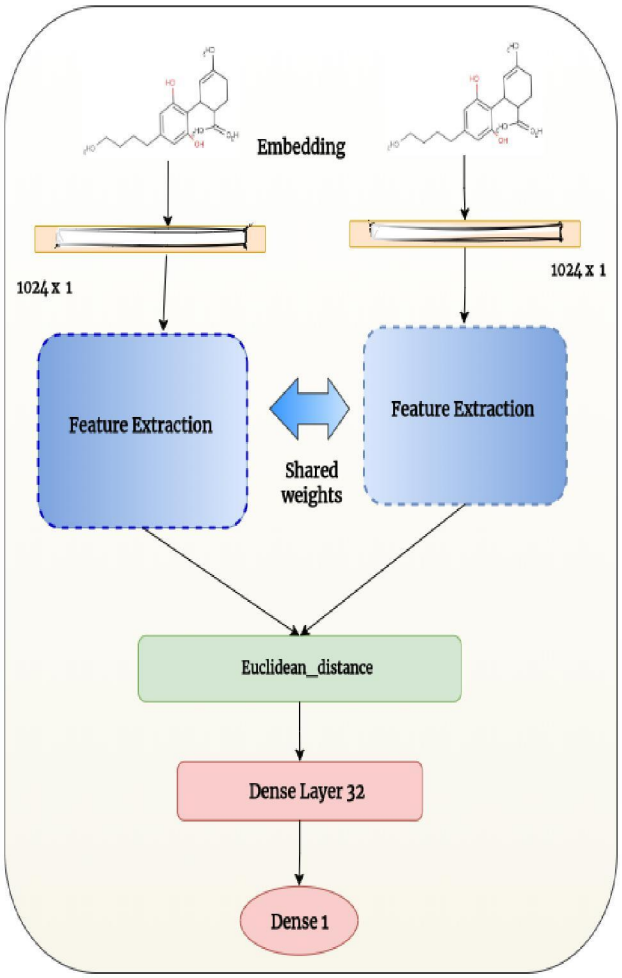


Figure 4.5: The SSCNN1D similarity model architecture.

ple the output of the convolutional layers, reducing the dimensionality of the feature maps. The output of the max-pooling layer is then reshaped into a 2D matrix with a size of (-1, 64) using the Reshape layer. The second part of the feature extractor consists of a Bidirectional LSTM layer with 64 units, followed by a Self-Attention layer with a sigmoid activation function to improve bidirectional long short-term memory [119](BiLSTM)model for high interpretability and convenient modeling as well as a considerable predictive performance This part of the network is designed to capture high-level features and temporal dependencies between the features extracted in the previous part. The third and final part of the feature extractor consists of a Flatten layer that flattens the output of the attentional vectors was extracted into a 1D tensor, followed by a Dense layer with 64 units and a ReLU activation function. This layer maps the high-level features extracted in the previous layer to a lower-dimensional space, facilitating the computation of the distance between the two input sequences in the subsequent layers.

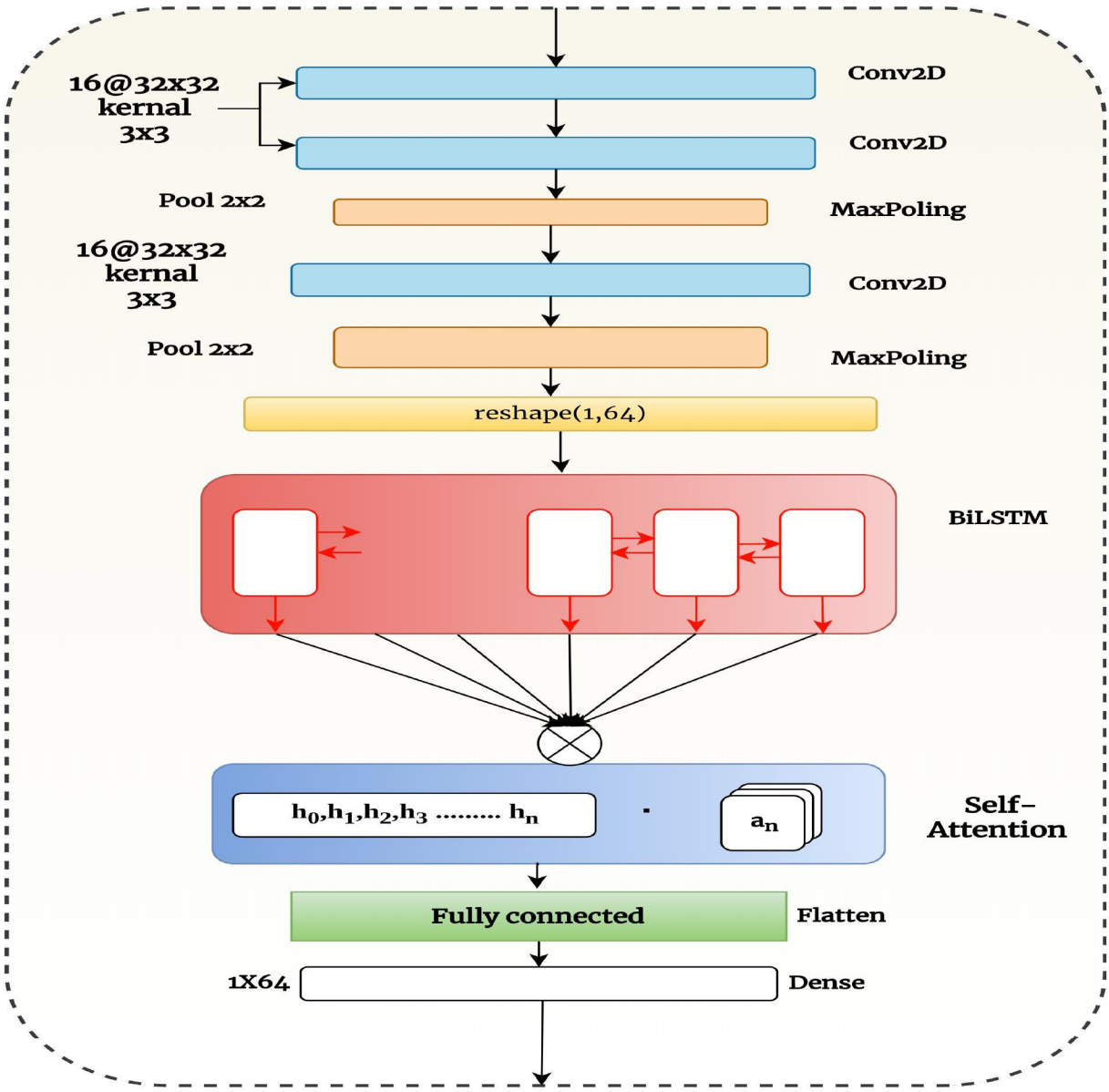


Figure 4.6: 2D feature Extractor.

After encoding the two input fingerprints using the feature extractor, Two distances inspired by Altalib's work are used to calculate the similarity[4], Manhattan, and exponential Manhattan which measure how different the two extracted features are, by Equation (4.2) and Equation (4.3) respectively:

$$\text{Manhattan} = \sum_{i=1}^{n} |x_i - y_i| \tag{4.2}$$

$$\text{Exp_Manhattan} = \exp(-|\mathbf{x} - \mathbf{y}|) \tag{4.3}$$

Then, The two distances are concatenated and passed through a couple of dense layers to generate a fused feature vector, each dense having 64, 32 cells, and 1 cell respectively, A ReLu activation applies to all dense layers except the last dense layer which

the sigmoid activation function has been used, which outputs a probability value ranging from 0 to 1 prediction of whether the two molecules are similar or dissimilar. Figure 4.7 and Table 4.5 demonstrates the architecture of the SSCNN2D similarity model.

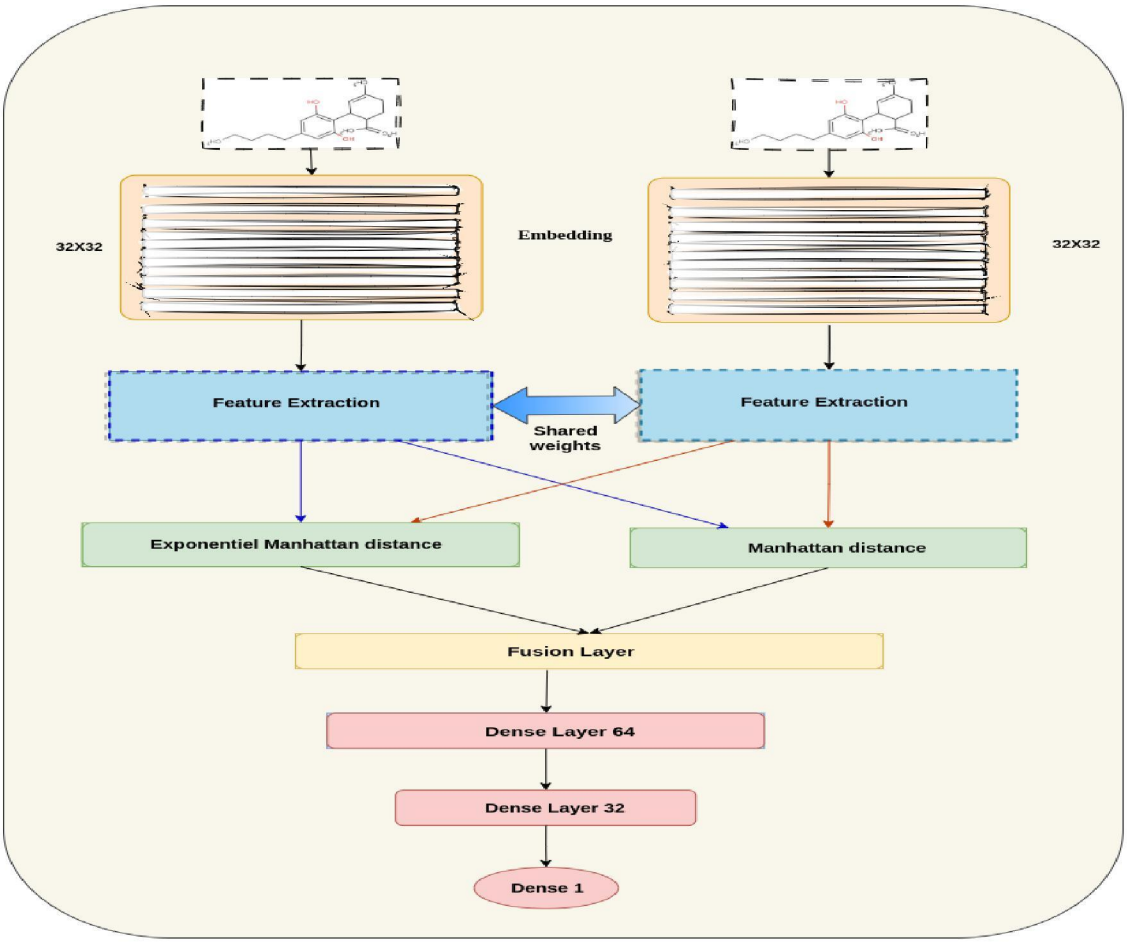


Figure 4.7: The SSCNN2D similarity model architecture.

4.5 Performance and Evaluation Measures

The Efficiency of All Proposed Techniques was Measured with the help of Adam optimizer with the default value of 0.001 with the same batch size 64, and evaluated by Mean Squared Error (MSE) as a loss function used for regression problems. It measures the average squared difference between the predicted and actual values. A lower MSE indicates better performance.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.4}$$

Index	Layer type	Number of filters	kernel	Parameters	shape
1	Input	-	-	0	(32,32,1)
2	Input	-	-	0	(32,32,1)
3	Conv2D	16	(3,3)	160	(None, 30, 30, 16)
4	Conv2D	16	(3,3)	2320	(None, 28, 28, 16)
5	MaxPooling2D	-	(2,2)	0	(None, 14, 14, 16)
6	Conv2D	16	(3,3)	2320	(None, 12, 12, 16)
7	MaxPooling2D	-	(2,2)	0	(None, 6, 6, 16)
8	Reshape	-	-	0	(None, 9, 64)
9	BiLSTM	-	-	66048	(None, 9, 128)
10	selfattention	-	-	8257	(None, 9, 128)
11	Flatten	-	-	0	(None, 1152)
12	Dense	-	-	73792	(None, 64)
13	ManhattanDistance	-	-	0	(None,64)
14	exponentialManhattanDistance	-	-	0	(None, 64)
15	Concatenate	-	-	0	(None, 128)
16	Dense	-	-	8256	(None, 64)
17	Dense	-	-	1040	(None, 16)
18	output	-	-	17	(None, 1)
	Total params: 162,338 Trainable params: 162,274 Non-trainable params: 64				

Table 4.5: Architecture and parameters of SSCNN2D model.

where n is the number of samples, y_i is the actual value, and $\hat{y}_i$ is the predicted value. The R-squared (R^2) is a statistical measure that represents the proportion of the variance in the dependent variable that is explained by the independent variables in a regression model.
It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.5}$$

where $\bar{y}$ is the mean of the actual values. R^2 takes values between 0 and 1, with higher values indicating better performance of the model. We trained both models using five-fold stratified cross-validation where The whole data is divided into five sets of equal size, In each cross-validation round, every fold was used for training and validation, Considering the rest as the test set as demonstrated in Table 4.6, Each iteration will use a different test set as shown in Figure 4.8

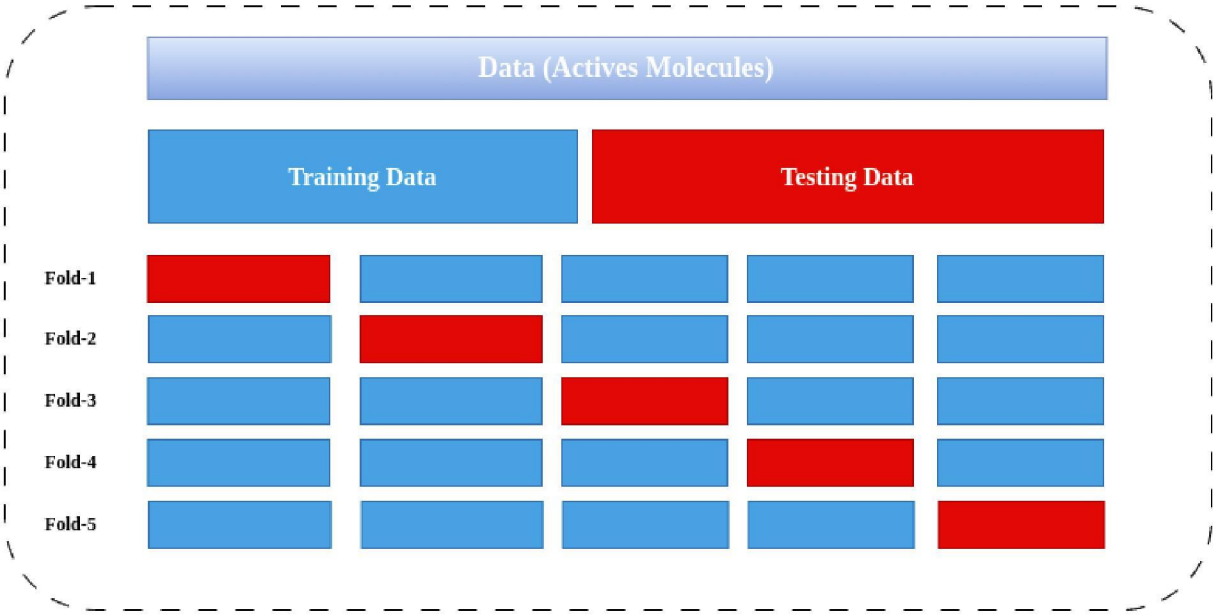


Figure 4.8: Idea of cross-validation for training and testing data.

Another two steps were implemented to improve the efficiency and performance of the training process, a periodic checkpoint was used to save the best weight of the current state of the model's parameters during training after every epoch finally an Early stopping involving monitoring the performance of the model on a validation set during training and stopping the training process early if the model's performance on the validation set stops improving after five epoch starting from the third epoch each cross-validation round, ensuring that the model is trained for the optimal number of epochs. A second way to evaluate the performance of the retrieval models is to use virtual screening which is The collection of computer-based techniques used in drug discovery to search through large libraries of chemical compounds is referred to as virtual screening (Berrhail et al., 2017)[17].

Table 4.6 shows the splitting of three datasets DS1, DS2, and DS3, into three different sets: 60% for training set ,20% for validation set , and 20% for testing set .

Datasets	Training Set	Validating Set	Testing Set
DS1	103980	25996	32494
DS2	63078	15770	19712
DS3	106539	26635	33294

Table 4.6: Datasets Splitting.

4.5.1 Procedure of Similarity Search in Virtual Screening

The ligand-based virtual screening technique involves comparing the known active molecule with every structure in the database. The degree of similarity is then calculated for each comparison, and the database structures are ranked in decreasing order of similarity (Abdo et al., 2012)[3]. Figure 4.9 and Algorithm 3 presents the workflow of all the similarity Processes for conducting similarity searches during the Ligand-Based Virtual Screening., described as follows:

* **Step 01 :** Ten active reference molecules were chosen from each activity class in the MDDR DS1 and MDDR DS2, MDDR DS3 databases and were comparable to those used in the earlier studies (Berrhail et al., 2022)[16].

* **Step 02 :** The proposed models were used to predict the similarity of the reference structures to the molecules in the database. The number of active molecules in the top 1-5% of the database, ranked in order of decreasing similarity to the reference structures, was then calculated.

* **Step 03 :** For each reference structure, active molecules of the same class were selected in the same number as the reference structure's class.

* **Step 04 :** The mean number of active molecules was determined for each class by averaging the results of 10 different experiments.

Algorithm 3: Similarity Search in Virtual Screening.

Input : Database, Reference_Molecules
Output: Top_K%_SimilarMolecules

$SimilarityScores \leftarrow []$
for *each Molecule in Reference_Molecules* **do**
    $Similarityscore \leftarrow$ Predict(Molecule, each molecule in Database)
    $SimilarityScores$.append($Similarityscore$)
end
$SimilarityScores$.sort(reverse=True)
return [Top(1%), Top(5%)] ▷ Top k molecules with the highest similarity scores

4.6 Benchmarking

In many studies based on ligand-based virtual screening (LBVS), benchmarking is a crucial technique that should be used to assess the efficiency and dependability of

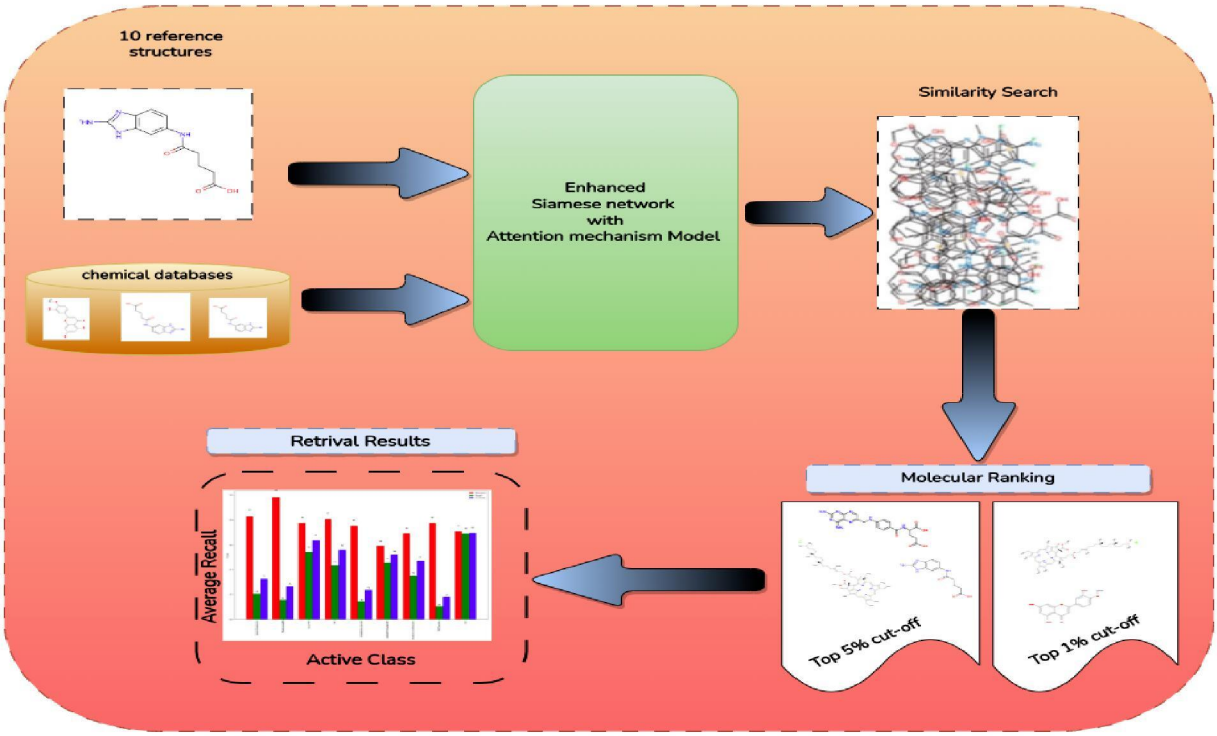


Figure 4.9: Virtual Screening Procedure.

proposed techniques compared to existing methods [82, 4, 63, 16].
In this section, we introduce different previously established methods used as a reference for assessing the effectiveness of the proposed model. These methods comprise:

* **TAN :** For many years, the Tanimoto similarity coefficient has been used as a benchmark in LBVS. The Tanimoto-based model calculates similarities using the continuous form of the Tanimoto coefficient, which is suitable for non-binary fingerprint data. Non-binary fingerprints represent the frequency or occurrence of each substructure or descriptor, whereas binary fingerprints represent the presence or absence of each substructure or descriptor (Bajusz et al., 2015)[12].

* **BIN :** By incorporating the relevance feedback information, Bayesian inference networks (BIN), a type of probabilistic graphical model, can reweight molecular fragments and assign higher weights to those fragments that are more frequently found in active reference structures, enhancing the retrieval performance of LBVS (Ahmed et al., 2012)[3].

* **SQB :** The study's third benchmarking technique, known as SQB-(Complex), uses a quantum similarity search methodology. The technique, in particular, depends on the complex pure Hilbert space of molecules to enhance the performance of the model (Nasser et al., 2020)[82].

* **SDBN :** is a multi-descriptor-based Stack of deep belief networks method for reweight-

ing the molecular features for ECFC-4, ECFP-4, and EPFP-4 descriptors in the MDDR data set (DS1, DS2, and DS3) (Nasser et al., 2020)[82].

* **CNN :** In the Enhanced Siamese CNN Similarity Model, both CNN1D and CNN2D have been used to determine the similarity between the two molecules. This model uses a Siamese CNN framework to determine molecular similarity (Altalib & Salim, 2022)[4].

* **LSTM :** Another approach used as a benchmark technique for this study and developed by (Altalib & Salim, 2022) to assess the similarity between two molecules is the Enhanced Siamese LSTM Similarity Model [4].

* **DCNNLB :** The latest method using Deep Convolutional Neural Network trained to predict the biological activity and representations generated by the DCNN model can be used for similarity search and virtual screening (Berrhail et al., 2022) [16].

4.7 Summary

This chapter gives the details of our proposal of the methods employed in the conducted study. It encompasses the evaluation of various databases, presenting a detailed exploration of molecular similarity, along with the inherent challenges associated with its measurement. Furthermore, the chapter delves into elucidating the architecture of a Siamese neural network and its application in comparing two molecules, ultimately predicting their similarity score using the proposed techniques. Essential considerations for training a Siamese network for molecule similarity prediction, such as the selection of input representations, loss functions, and evaluation metrics, are thoroughly discussed. Additionally, recent advancements in the field are highlighted, including the utilization of attention mechanisms, experimental design considerations, and statistical methods employed for evaluating the experimental results. The next chapter brings implementation and experimental details for the proposed approach and discusses the obtained results.

Results and Discussion

Chapter 5

Results and Discussion

This chapter, we present the environment used in our study and discuss the results obtained by the application of our models SSCNN1D and SSCNN2D for molecular similarity searching in LBVS. At first, we considered R_Square as an evaluation metric, and we set the error loss function (losses.categorical crossentropy of keras), which is a measure of the error taken by our model. This error is defined in the Keras API, which is a deep learning API written in Python, and executed on most of the deep learning platforms, including TensorFlow. Consequently, both metrics are calculated at each epoch during the training and testing stages. The results obtained from the evolution of the accuracy value and the loss value for each data set for the training and testing data are shown graphically. We used all the activity classes of the MDDR datasets that divide into three datasets: DS1, DS2, and DS3. After that, we tested LBVS and compared the obtained results with previous study results. The experimental results obtained by our method and other recognized similarity methods These results are calculated on the MDDR dataset and are organized as follows: Each data set has two tables for the 1% and 5% cut-offs.

5.1 Development environment

5.1.1 Python

Python was selected as the programming language for our experiment due to its flexibility, ease of use, and dynamic typing, which permits the declaration of variables without consideration for their predetermined data types. The extensive standard library contains pre-built modules for a wide range of tasks, such as scientific calculation, web development, and file processing. Simple object-oriented programming capabilities and Python's syntax make it possible to create reusable code. The language is

cross-platform compatible, and because it is interpreted, on-the-spot code execution is made possible, accelerating the development process. Python is well-liked in the scientific computing and machine learning fields thanks to its effective libraries like NumPy, Pandas, and Scikit-learn, and the large Python user and development community offers a wealth of resources and assistance. Guido van Rossum developed Python as a replacement for ABC language in the late 1980s. The language has undergone multiple updates and enhancements since then, with the most recent version being Python 3.10.0, which was released in October 2021.

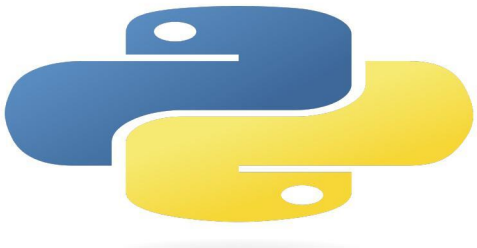


Figure 5.1: Python logo.

5.1.2 Python features

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, and procedural programming approaches and provides dynamic memory allocation. We have listed below a few essential features.

- **Free and Open Source:** Python is an open-source programming language and can download it for free from Python's official website. In order to make Python's code better, the Python user community continuously makes contributions.
- **Easy to Learn and Use :** Learning Python is quite simple. Python has a relatively easy learning curve and a very straightforward grammar. Python is incredibly simple to learn and use, and because indentation is utilized in Python code instead of curly braces, it is very simple to read.
- **Expressive Language :** Python is a powerful programming language that enables developers to create clear, readable code. With a short, clear syntax, it can convey complicated ideas.
- **Dynamic Typing :** Python is dynamically typed, which implies that variable types are decided upon while the program is running. This makes programming more flexible and can make the development process easier.

- **Object-Oriented :** An object-oriented programming language centers its architecture around data and objects rather than functions and logic. On the other hand, a programming language is procedure-oriented if it emphasizes functions (reusable code) more. Support for both object-oriented and procedure-oriented programming is a key Python feature.

- **Interpreted Language :** Python is an interpreted language, which means that its source code is translated into bytecode before being run by the Python virtual machine.

- **Extensive Library :** Python has a large standard library that provides a range of modules and functions for a variety of tasks, including data manipulation, file handling, and web development. There are also many third-party libraries and packages that can be easily installed and used to extend the functionality of the language.

- **Multi-Platform :** Python is multi-Platform and can be executed in different OS Windows, Mac, Linux hardware, and even Android and iOS.

- **High-level Language :** Python is a high-level language is sort of programming language that allows programmers to create applications that are largely independent of a certain kind of computer. Given their closeness to human languages and their distance from machine languages, these languages are referred to as high-level languages. Python is a high-level language, unlike C. Python is simpler for us to understand than middle-level languages like C and is more user-friendly. This means that as programmers, we don't need to remember the system architecture. Also, we need not manage memory. This makes it more programmer-friendly and is one of the key Python features.

- **Dynamically Scalable :** Python is dynamically scalable, which means that it can scale up or down depending on the needs of the application. This makes it ideal for use in cloud computing and other scalable applications.

5.1.3 Python module used

* **Pandas module :** The pandas module is a popular library for data manipulation and analysis in Python programming language. It offers tools for data cleansing, combining, and reshaping, as well as data structures for effectively storing and handling massive datasets. The library is based on the NumPy library and is made to integrate easily with other Python-based scientific computing packages. The management of missing data, time series capabilities, and data alignment

are a few of the pandas module's important features. The library is extensively used for data analysis and visualization in business, academia, and research. In general, the Python pandas module is a potent tool for handling data and is a necessary package for anyone working with sizable datasets.



Figure 5.2: Pandas logo.

* **Numpy module :** NumPy (Numerical Python) is an open-source Python library that's used in almost every field of science and engineering. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development. The NumPy library contains a multidimensional array and matrix data structures. It offers ways to effectively manipulate the homogeneous n-dimensional array object ndarray. Numerous mathematical operations can be carried out on arrays using NumPy. It provides a vast library of high-level mathematical functions that work on these arrays and matrices, as well as strong data structures that ensure efficient calculations with arrays and matrices. The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.



Figure 5.3: NumPy logo.

* **Matplotlib module :** When writing Python scripts, users can use the Matplotlib library to generate 2D graphs and plots. By offering features to adjust line styles, font attributes, formatting axes, etc., its pyplot module makes things simple for plotting. Histograms, bar charts, power spectra, error charts, and many other graphs and plots are supported. It works in tandem with NumPy to create a MatLab-like environment that is a powerful open-source substitute. wxPython and PyQt are a few of examples of graphics toolkits that it can be utilized with.

* **Tensorflow module :** In order to create, construct, and train deep learning models, Google uses a machine learning framework called TensorFlow. Although numerical computations may not seem very significant in and of themselves, the TensorFlow library may be used to conduct them. These computations are carried out using data flow graphs. The mathematical operations represented by these graphs' nodes are called tensors or multidimensional data arrays, and their edges are the data that is conveyed between them. The library is accessible across several platforms, including mobile, and it may be used with a range of CPUs and GPUs.1 For our hands-on work in the Google colab environment, we utilized Tensorflow 2.5.0, the most recent version.



Figure 5.4: Tensorflow logo.

* **Keras module :** In Python, Keras is a high-level neural network API that may be used with TensorFlow, CNTK, or Theano. It was created with the goal of facilitating quick experimentation and deep learning model creation. Convolutional networks, recurrent networks, and hybrids of the two are all supported by Keras' easy-to-use interface for creating and training neural networks. Additionally, it offers a vast array of tools for model evaluation, data preprocessing, and visualization.



Figure 5.5: Keras logo.

* **Python library RDkit module :** RDKit is a popular open-source cheminformatics software library written in Python. It provides a wide range of functionalities for working with chemical structures, molecules, and related data. Some of the key features of RDKit include:

1. **Chemical structure representation:** RDKit allows you to create, manipulate, and visualize molecular structures. It supports various molecular file formats, including SMILES, SDF, Mol, and PDB.
2. **Molecular fingerprinting:** RDKit provides methods for generating molecular fingerprints, which are compact binary representations of molecules. These fingerprints can be used for tasks like similarity searching, clustering, and machine learning.
3. **Substructure searching:** RDKit enables you to search for specific substructures within a molecule or a database of molecules. You can define substructure patterns using SMARTS or SMILES strings and perform searches efficiently.
4. **Chemical reactions:** RDKit supports reaction handling, allowing you to define and apply transformations to molecules. It provides functionality for reaction enumeration, reaction center identification, and reaction substructure searching.
5. **Descriptor calculations:** RDKit offers a wide range of molecular descriptors that quantify various properties of molecules, such as molecular weight, logP, and topological indices. These descriptors are useful for modeling, QSAR, and property prediction tasks.
6. **2D and 3D molecule visualization:** RDKit includes tools for visualizing molecules in 2D and 3D. You can generate 2D depictions of molecules, highlight substructures, and render molecules in 3D using popular visualization libraries like Matplotlib
7. **Cheminformatics algorithms:** RDKit implements various algorithms commonly used in cheminformatics, such as molecular alignment, similarity calculation, diversity analysis, and pharmacophore searching.

RDKit is extensively used in the field of cheminformatics and computational chemistry. It is available as a Python package that can be installed using the Python package manager pip. The official documentation of RDKit provides comprehensive information about its usage and available functionalities.



Figure 5.6: RDKit logo.

5.1.4 Visual Studio Code

VS Code is an open source code editor that can be used in different fields, runs on desktops, and is available for Windows, macOS, and Linux.It comes with built-in support for JavaScript, Typescript, and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, C#, Java, Python, PHP, Go, .NET). For our hands-on work, we chose this editor, installed Jupyter Notebook for executing code in the form of cells, and installed the Python module used.

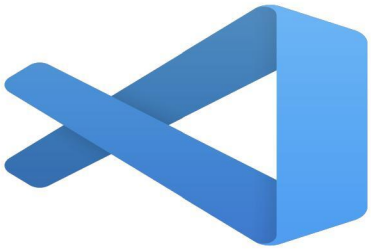


Figure 5.7: Visual studio code logo.

5.1.5 Google colaboratory

Google colaboratory definition

One of Google Research's products is called Colaboratory, or "Colab" for short. Colab is extremely useful for machine learning, data analysis, and education because it enables anyone to write and run arbitrary Python code through the browser. Technically speaking, Colab is a hosted Jupyter Notebook service that can be accessed instantly and offers free access to computer resources, including GPUs. Colab is an open-source service for anyone with a Gmail account. In a single runtime, the Google Colab service offers 358.27 GB of hard drive space and 12.72 GB of RAM. Each runtime lasts for 12 hours before being reset, and the user has to re-establish their connection. This is done to prevent people from using the GPU service for unlawful activities like cryptocurrency mining. The list below includes various run instances. The user must choose a runtime type after opening a Google Colab file. For the same, there are three options available:

1. None (that will use the computer's CPU the user is using)
2. CPU
3. TPU (especially for tensor processing)



Figure 5.8: Google Colab logo.

Benefits of colab

For data scientists and machine learning engineers, Colab is a great platform since it offers a free, simple-to-use environment for collaboration along with access to strong computing resources. Some of the benefits of using Colab are:

* **Free to use:** Colab offers access to a potent GPU that can be utilized for deep learning and machine learning activities, and it is absolutely free to use.
* **Easy to use:** Users can quickly build and launch Jupyter notebooks using Colab's user-friendly interface. Coding is greatly facilitated by the interface's additional capabilities, such as code completion and syntax highlighting.
* **Collaboration:** Colab is a fantastic platform for teamwork because it lets users share their notebooks with others. The same notebook can be used by multiple people at once, and updates are synced instantly.
* **Google Drive integration:** Because Colab and Google Drive are fully integrated, users can store and retrieve their notebooks and data files from any location.
* **High-performance computing:** For high-performance computing operations like training deep neural networks, Colab offers access to a potent GPU.
* **Easy access to popular libraries:** Many well-known data science and machine learning libraries are pre-installed in Colab, making it simple for users to begin working on their projects.
* **Availability of pre-built:** Colab gives customers access to pre-built settings that can be utilized for certain tasks like Natural Language Processing or Computer Vision, saving them a ton of time and effort from having to set up their own environments.environments

5.1.6 Google Drive

Google Drive is a free online cloud storage option for Google files. It was introduced

in 2012 to take the place of Google Docs and enables users to store, share, and work together on files and folders from any location with an internet connection.

Among its main features are:

* **File Storage:** 15 GB of free storage space, which may be increased to 200 TB at various rates. It enables us to host a variety of items, including text, pictures, videos, and photos. These files can also be seamlessly integrated with other free Google products like Gmail, Google Photos, or Chromebook.
* **Document creation:** We can make text files, slideshows, and spreadsheets using Google Docs, which is integrated with Drive. It also has a tool for designing web forms and another for drawing graphs and flowcharts.
* **Connectivity:** With Google Drive, we can effortlessly share our files with anyone we choose, collaborate on documents with other users and view them from any internet-connected device, including computers, smartphones, and tablets.
* **Integrated apps:** Drive includes hundreds of diverse suitable programs that we can install from the Chrome Web Store in addition to the tools indicated above for creating documents.
* **Secure storage:** Drive files are encrypted using either AES256 or AES1283, the same security mechanism employed by other Google services.



Figure 5.9: Google Drive logo.

5.1.7 Jupyter notebooks

The most recent web-based interactive development environment for code, data, and notebooks is JupyterLab. Users can configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning using the

interface's flexibility. A modular structure encourages expansions to increase and improve functionality.

Jupyter notebooks have a wide range of applications. A notebook is an interactive computational environment where users can execute a specific piece of code, inspect the results, and alter the code to get the desired results or carry out additional research. Due to the numerous iterations involved in data exploration, Jupyter notebooks are frequently employed. It is also utilized in other data science procedures, including machine learning modeling and experimentation. Code samples may also be documented using it. Users can execute code cells independently and in any order in a Jupyter Notebook. Alternating between code and markdown cells can be used for documentation. Among its main features are:

* **Live code execution:** Users can create and run code in real-time using Jupyter Notebook, which makes it simple to test and debug programs.
* **Interactive visualization:** It is simple to create and share data visualizations using Jupyter Notebook because it supports a large number of interactive visualization libraries.
* **Markdown support:** The lightweight markup language Markdown, which is supported by Jupyter Notebook, enables users to format text, make headings, lists, and links, as well as incorporate images and videos.



Figure 5.10: Jupyter Notebook logo.

5.1.8 Hardware used

1. *Dell XPS15 :*
    - **CPU intel core i7-7700HQ.**
    - **RAM with 24 GB.**
    - **Disk with 512 GB of storage.**
    - **Graphics Cards Ge-Force GTX 1050 4GB**
2. *ASUS ROG STRIX :*
    - **CPU intel core i7-7700HQ.**
    - **RAM with 16 GB.**
    - **Disk with 1000 GB of storage.**
    - **Graphics Cards Ge-Force GTX 1050 4GB**

5.2 Evaluation of proposed Models

The correct prediction of molecular similarity is of paramount importance in various domains, including drug discovery, chemical informatics, and material science. Machine learning models have emerged as powerful tools for predicting molecular similarity, providing the capacity to expedite the method of figuring out compounds with similar properties. We assess and examine the performance of two proposed models, namely SSCNN1D and SSCNN2D. with the mission of predicting molecular similarity. Through a comprehensive assessment of multiple datasets, We conduct a thorough analysis of the model's predictive competencies and gain insights into their applicability in real-world situations. By examining key overall performance metrics consisting of loss, coefficient of determination (R-squared), and generalization capacity, this evaluation will contribute to a deeper understanding of the strengths and boundaries of these models and aid in informed decision-making concerning their adoption in practical molecular similarity prediction.

5.2.1 Result Model SSCNN1D

The trained model was subjected to a robust cross-validation scheme, employing a 5-fold split on the MDDR-DS1, MDDR-DS2, and MDDR-DS3 datasets. Each fold consisted of 10 epochs, ensuring a comprehensive assessment of the model's performance across different subsets of the data. This approach allowed for a more rigorous evaluation, as the model's performance was assessed on multiple non-overlapping partitions of the dataset. the obtained results are presented below :

- **MDDR-DS1**

  The evaluation of the trained model using the metrics 'loss', 'r-square', 'val_loss', and 'val_r-square' using the MDDR-DS1 dataset subset produced interesting results. Throughout the training process, the model consistently improved its performance, as evidenced by the decreasing 'loss' values presented in Figure 5.11. Starting from an initial 'loss' of 0.0763, the model progressively reduced this measure to a significantly lower value of 0.0007. This consistent decrease indicates the model's ability to effectively minimize the disparities between the predicted and actual values. In terms of generalization, the 'val_loss' metric, which gauges the model's performance on unseen data, demonstrated a similar trend. The initial 'val_loss' value of 0.0588 consistently decreased with each epoch, stabilizing at 0.0044. This compelling result suggests that the model consistently achieved accurate and reliable predictions when evaluated on previously unseen data points.

Moreover, the 'r-square' metric presented in Figure 5.11, employed to assess the model's capacity to explain the variance in the dataset, exhibited significant improvement during the training process. Commencing at 0.0796, the 'r-square' value progressively increased to an impressive 0.9912, indicating the model's capability to capture underlying patterns and relationships in the data, thereby yielding a high degree of predictive power. The 'val_r-square' metric, which measures the model's performance on the validation set in terms of explained variance, followed a similar trend. With an initial value of 0.2906, the 'val_r-square' consistently improved with each epoch and reached a final value of 0.9388. This outcome underscores the model's ability to generalize well to unseen data, providing accurate and consistent predictions.

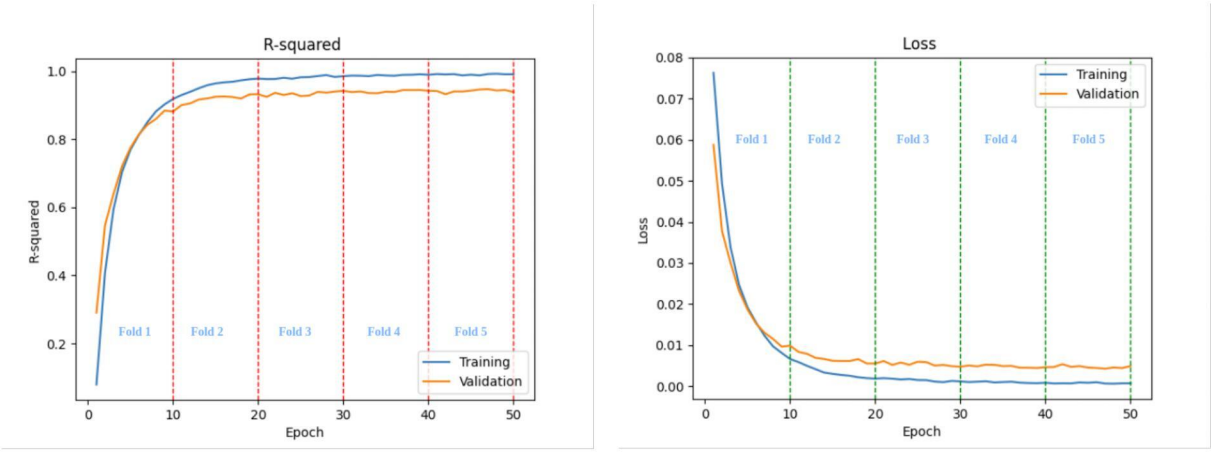


Figure 5.11: Variation of training and testing on MDDR-DS1 for SSCNN1D.

- **MDDR-DS2**

The obtained results presented in Figure 5.12 for the MDDR-DS2 dataset demonstrate the performance of our proposed model, SSCCNN1D, in a rigorous cross-validation scheme.

During the training process, the model achieved a significant reduction in loss, starting from 0.0204 and converging to 0.0002 over the 5 folds. This progressive decrease indicates that the model effectively learned from the dataset, improving its predictions with each iteration. Similarly, the validation loss followed a decreasing pattern, starting at 0.0039 and stabilizing around 0.0001. This suggests that the model's performance on unseen data was consistently strong, further validating its generalization capabilities. The R-Square metric, which measures the proportion of variance explained by the model, showed consistent improvement over the training epochs. Starting at 0.7683, the R-Square values reached a maximum of 0.9974, indicating a high level of accuracy in capturing the underlying

patterns and variability in the target variable. The validation R-Square values also exhibited a positive trend, with a maximum value of 0.9984, showcasing the model's ability to generalize well to unseen data. The decreasing loss and increasing R-Square values demonstrate the model's capacity to learn and make precise predictions. Furthermore, the low validation loss and high validation R-Square values emphasize the model's robustness and its ability to generalize effectively.

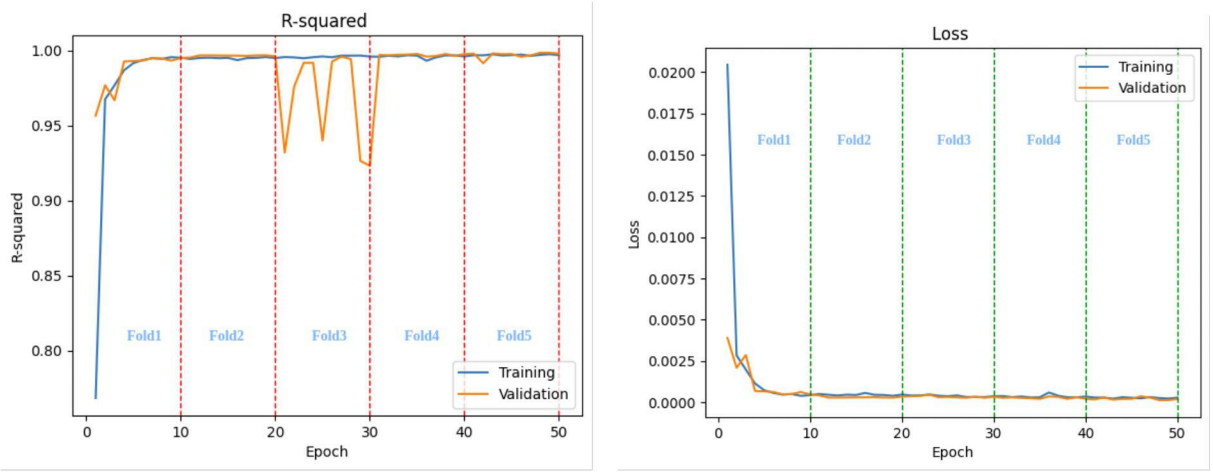


Figure 5.12: Variation of training and testing on MDDR-DS2 for SSCNN1D.

- **MDDR-DS3**

  The obtained results for the MDDR-DS3 dataset using the SSCNN1D model exhibit promising performance (see Figure 5.13). The training loss consistently decreases throughout the epochs, starting from an initial value of 0.087 and converging to an impressive 0.0018. This signifies the model's ability to effectively learn from the provided data and minimize errors during the training process.

  The validation loss closely follows the same decreasing trend as the training loss. It starts at 0.083 and steadily declines to 0.0019, indicating that the model generalizes well to unseen data and avoids overfitting.

  The R-Square metric, which measures the degree of fit between the model and the data, exhibits notable improvement throughout the training. It begins at 0.01 and progressively increases to an impressive 0.979. These results suggest that the model successfully captures a substantial portion of the variance in the target variable, showcasing its ability to make accurate predictions.

  Validation R-Square values also demonstrate an increasing trend, starting at 0.067 and reaching a high of 0.978. This implies that the model performs well not only on the training data but also on previously unseen validation data, indicating its generalization capability.

In conclusion, the results obtained from the SSCNN1D model on the MDDR-DS3 dataset present a highly promising outcome. The consistently decreasing loss and increasing R-Square values indicate the model's effectiveness in capturing the underlying patterns in the data and accurately predicting the target variable.

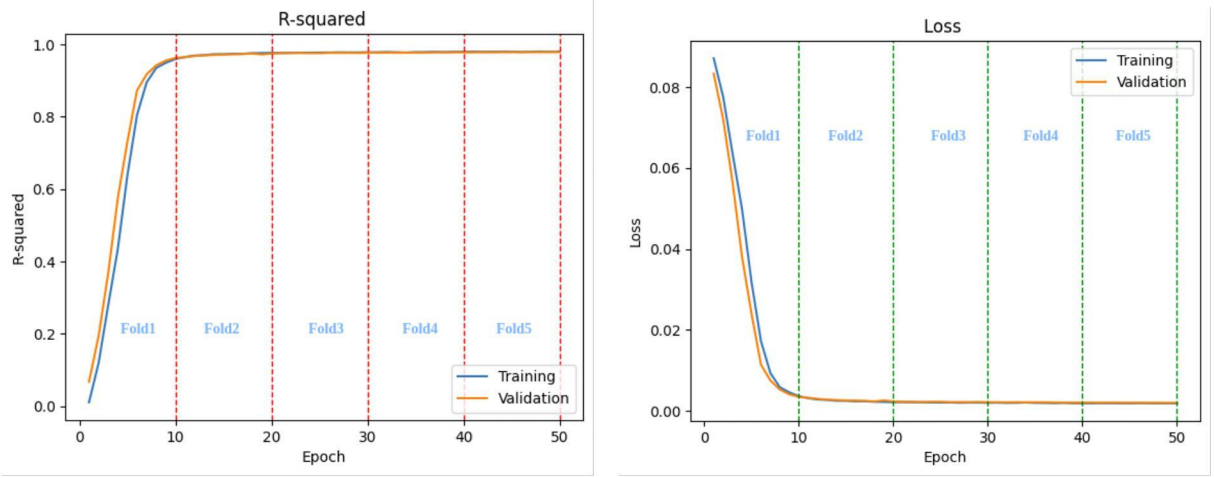


Figure 5.13: Variation of training and testing on MDDR-DS3 for SSCNN1D.

5.2.2 Result Model SSCNN2D

The performance evaluation of the SSCNN2D model involved a rigorous cross-validation scheme utilizing a 5-fold split on the MDDR-DS1, MDDR-DS2, and MDDR-DS3 datasets. Each fold consisted of training the model for three epochs, ensuring a meticulous assessment of its performance across multiple non-overlapping partitions of the dataset. This meticulous approach allowed for a comprehensive evaluation of the SSCNN2D model's capabilities, providing valuable insights into its performance and generalizability across diverse subsets of the dataset. The results obtained from this rigorous evaluation are presented below, highlighting the model's effectiveness and robustness.

- **MDDR-DS1**

  Our proposed model trained on the DS1 dataset demonstrated remarkable performance presented in Figure 5.14. Throughout the training process, the model consistently reduced the loss, indicating its ability to effectively minimize the discrepancies between the predicted and actual values. The final loss value of 0.0017 suggests that the model achieved a high level of accuracy. Additionally, the R-square values steadily increased, signifying the model's improved ability to explain the variance in the target variable. The final R-square value of 0.9466 indicates that the model can account for approximately 94.66% of the variability in the dataset. The validation results further reinforced the model's proficiency, with the validation loss decreasing and the validation R-square increasing consistently. These findings demonstrate that the model generalizes well to unseen data and has the potential for robust performance in real-world applications.

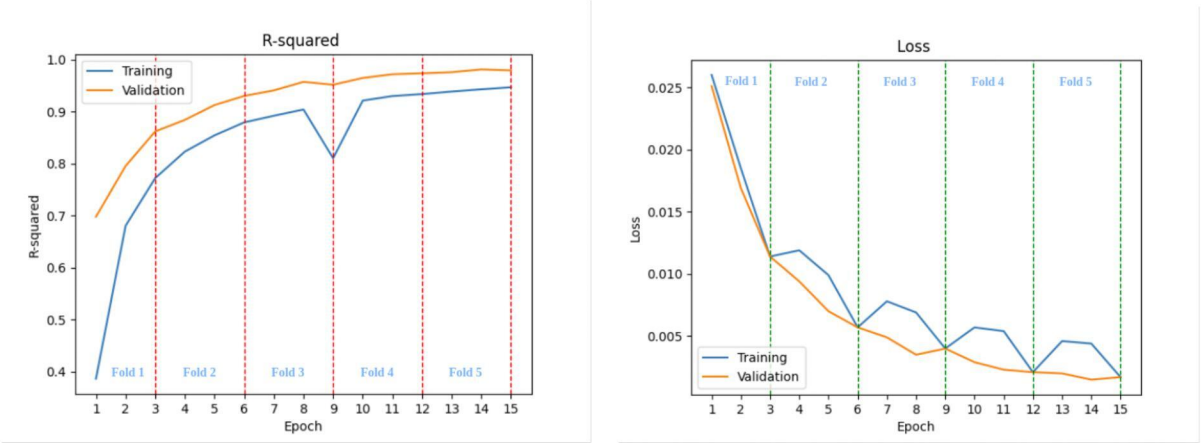


Figure 5.14: Variation value for training and testing data in DS1 data set for model SSCNN2D.

- **MDDR-DS2**

  The evaluation of the proposed model on the DS2 dataset revealed exceptional performance (see Figure 5.15). The model consistently reduced the loss during training, reaching a final value of 0.0023. This indicates that the model effectively minimized the errors between the predicted and actual values, resulting in accurate predictions. The R-square values demonstrated a consistent improvement, culminating in a high value of 0.9741. This suggests that the model can explain approximately 97.41% of the variance in the target variable, indicating a strong correlation between the predicted and actual values. The validation results further validated the model's robustness, as indicated by the low validation loss and high validation R-square values. The model's ability to generalize well to unseen data makes it highly reliable for making accurate predictions in practical scenarios.

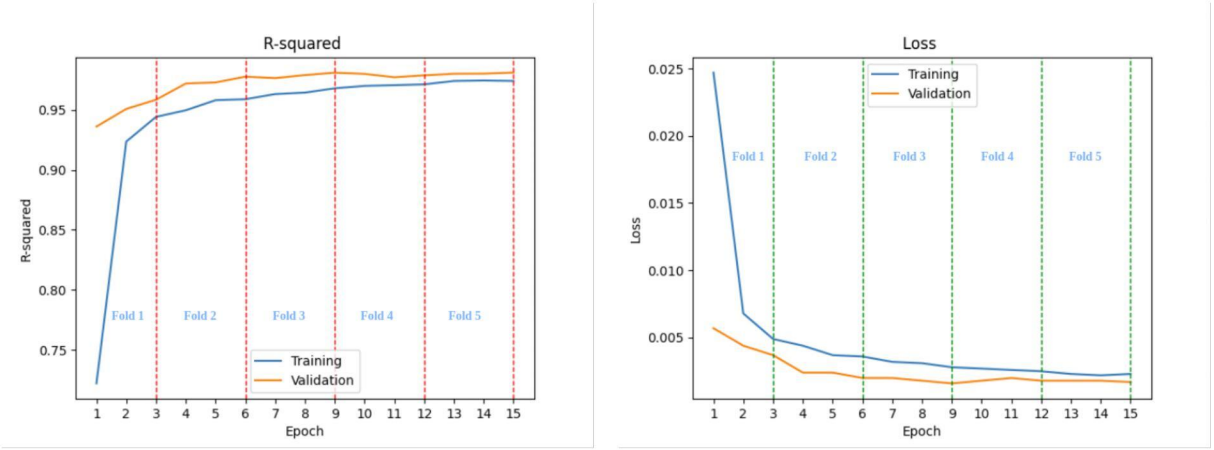


Figure 5.15: Variation of training and testing on MDDR-DS2 for SSCNN2D.

- **MDDR-DS3**

  The performance of model on the DS3 dataset was satisfactory. The model exhibited a gradual decrease in loss during training, reaching a final value of 0.0097. This implies that the model effectively minimized the discrepancies between the predicted and actual values, resulting in precise predictions. The R-square values steadily increased throughout the training process, reaching a value of 0.8769. This indicates that the model can explain approximately 87.69% of the variance in the target variable, illustrating a reasonable level of predictive power. The validation results further confirmed the model's competence, with the validation loss decreasing and the validation R-square increasing consistently. These outcomes signify that the model can generalize well to unseen data, making it suitable for making reliable predictions in practical applications.

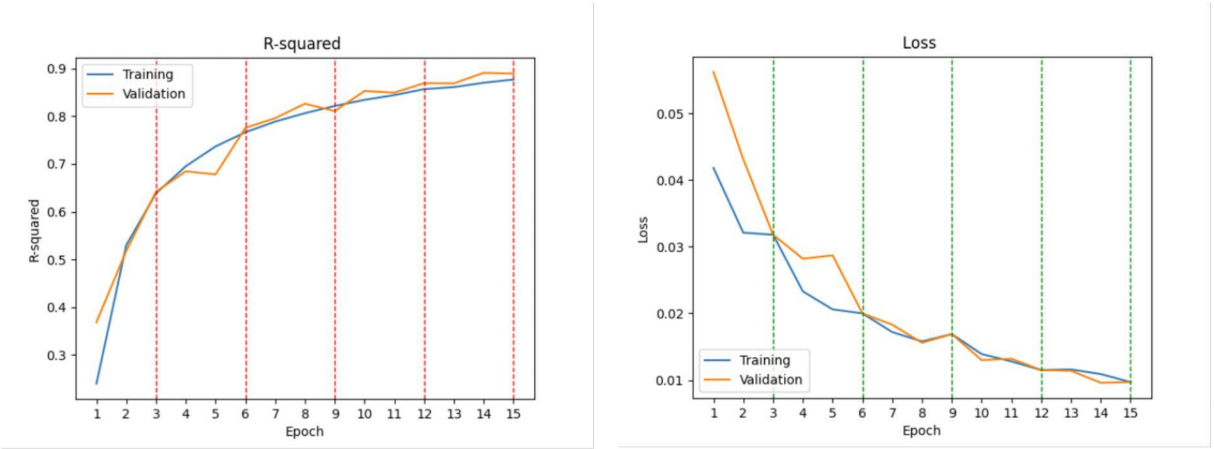


Figure 5.16: Variation value for training and testing data in DS3 data set for model SSCNN2D.

5.2.3 Summary Result of training and validation

The evaluation of the SSCNN1D and SSCNN2D models on the DS1, DS2, and DS3 datasets provides valuable insights into their performance in predicting molecular similarity between two compounds. The analysis of loss reveals consistent downward trends during training for both models across all datasets, indicating their ability to minimize the disparity between predicted and actual similarity values. However, the SSCNN1D model consistently achieves lower loss values compared to the SSCNN2D model, suggesting its superior capacity to effectively optimize and reduce the discrepancy.

Furthermore, the assessment of the coefficient of determination (R-squared) enables a comprehensive evaluation of the model's ability to capture the variance in the predicted similarity values. Notably, as the training progresses, the R-squared values demonstrate incremental improvements for both models, signifying their enhanced capacity to elucidate the variability in molecular similarity. The Table 5.1 presents the R-square results for two proposed models, SSCNN1D and SSCNN2D, trained for each dataset DS1, DS2, and DS3 ,The R-square values serve as a measure of the model's ability to explain the variance in the data. SSCNN1D consistently exhibits higher R-square values across all datasets, ranging from 97.80% to 99.74%, indicating a strong capacity to capture and model the underlying patterns within the data. In contrast, SSCNN2D shows comparatively lower R-square values, ranging from 87.69% to 97.41%. These findings suggest that the design and architecture of SSCNN1D are more effective in representing the relationships and predicting the outcomes of the datasets. Thus, SSCNN1D demonstrates superior predictive performance compared to SSCNN2D in this particular context. Nevertheless, the SSCNN1D model consistently outperforms the

SSCNN2D model in terms of R-squared across all datasets, indicating its superior overall fit to the observed molecular similarity values.

By examining the validation loss and validation R-squared values, it becomes evident that both models exhibit superior performance on the training set compared to the validation set, as expected. However, the SSCNN1D model exhibits a smaller performance gap between the training and validation sets, suggesting its superior generalization ability and enhanced adaptability to unseen data. This implies that the SSCNN1D model possesses a greater propensity to generalize well and is less susceptible to overfitting.

Models	SSCNN1D	SSCNN2D
R-square for DS1	99.12	94.66
R-square for DS2	99.74	97.41
R-square for DS3	97.80	87.69
Mean	**98.89**	**93.25**

Table 5.1: R_Square Results.

Overall, the SSCNN1D model displays superior predictive performance and improved generalization. The selection between the two models would depend on the specific requirements of the application and the relative importance of loss minimization versus achieving a more comprehensive fit to the observed molecular similarity values.

5.2.4 Evaluation Test Set

Table 5.2 presents the prediction results of the test set for the SSCNN1D and SSCNN2D models on different datasets. The table provides information on the prediction outcomes based on similarity percentages. The models are evaluated on datasets DS1, DS2, and DS3, and the similarity percentages are categorized into three intervals: [0-10%],]10-90%[, and [90-100%].

The SSCNN1D model succeeds in 16123 correct predictions out of 16246 samples in ds1, 9780 right predictions out of 9821 samples in ds2, and 16499 correct predictions out of 16696 samples in ds3 in the [0-10%] similarity interval. Similar results were obtained with the SSCNN2D model, which produced 16513 correct predictions out of 16762 samples in ds3, 9769 correct predictions out of 9780 samples in ds2, and 16233 correct predictions out of 16246 samples in ds1.

Models	SSCNN1D						SSCNN2D					
dataset	ds1		ds2		ds3		ds1		ds2		ds3	
Similarity	0%	100%	0%	100%	0%	100%	0%	100%	0%	100%	0%	100%
[0-10%]	16123	88	9780	31	16499	9	16223	16	9769	9	16513	30
]10%-90%[	108	52	37	6	87	40	131	107	6	9	225	180
[90%-100%]	15	16108	5	9853	12	16647	21	15996	5	9914	24	16322

Table 5.2: Test set results for SSCNN1D and SSCNN2D models on various datasets.

For the]10-90%[interval, which may represent uncertain or ambiguous predictions.

In the [90-100%] similarity interval, the SSCNN1D model achieves 16108 correct predictions out of 16320 samples in ds1, 7145 correct predictions out of 7182 samples in ds2, and 16647 correct predictions out of 16696 samples in ds3. Similarly, the SSCNN2D model achieves 15970 correct predictions out of 16093 samples in ds1, 9903 correct predictions out of 9921 samples in ds2, and 16322 correct predictions out of 16532 samples in ds3.

Overall, both SSCNN1D and SSCNN2D models demonstrate promising performance in predicting molecular similarities. The results indicate that these models are effective in capturing the complex relationships between molecular structures and can accurately classify molecular pairs into similarity categories. These findings highlight the potential of using deep learning approaches for molecular similarity prediction, which can have significant implications in drug discovery, chemical informatics, and other fields where molecular similarity plays a crucial role.

5.3 Result of Ligand-Based Virtual Screening

In this part, we evaluate the performance of various ligand-based virtual screening methods for drug activity prediction. Our evaluation includes a comparison of established methods, such as TAN, BIN, SQB, SDBN, LSTM, CNN1D, CNN2D, and DCNNLB, with our proposed methods, SSCNN1D and SSCNN2D. By analyzing both retrieval percentages 1% and 5% achieved by these methods across different activity classes, we aim to assess the effectiveness and potential of our proposed methods in improving the accuracy and reliability of ligand-based virtual screening. The results of this evaluation contribute to the advancement of computational methods in drug discovery and have implications for the efficient identification of promising drug candidates.

5.3.1 MDDR-DS1

The obtained results for the ligand-based retrieval of the MDDR-DS1 data set using the ECFC-4 descriptor are presented in Tables 5.3 and 5.4. These tables compare the performance of various previous studies with the proposed methods. The activity indices of the classes are ranked, and the percentages indicate the retrieval rate within the top 1% and 5% of the prediction results with the molecules in index with all datasets.

To better understand the obtained results presented in Tables 5.3 and 5.4, let's consider an example. Looking at Table 5.3, let's focus on the first row under the "Activity Index" column, which corresponds to the activity class with index 3142. This row represents the retrieval results for the top 1% of the prediction results with the MDDR-DS1 dataset using the ECFC-4 descriptor for compounds belonging to the activity class with index 3142. In the previous studies columns, each value represents the retrieval performance achieved by a specific method for the given activity class. For example, under the "TAN" column, the value 69.69 indicates the retrieval percentage achieved by the TAN method for compounds in the activity class with the index 3142. Similarly, the values in the "BIN," "SQB," "SDBN," "LSTM," "CNN1D," "CNN2D," "DCNNLB," "SSCNN1D," and "SSCNN2D" columns represent the retrieval percentages achieved by the corresponding methods for the activity class with index 3142. Comparing these values with the proposed methods column, we can observe that the proposed method using SSCNN1D achieved the highest retrieval percentage of 87.32% for compounds in the activity class with index 3142. This value is highlighted in the table using a shaded cell. Similarly, in Table 5.4, the process is the same. For example, considering the activity class with index 3142 again, we can see that the proposed method using SSCNN1D achieved a retrieval percentage of 100% for the top 5% of the MDDR-DS1 dataset using

Table 5.3: Top 1% Retrieval Results for MDDR-DS1 Data Set for Descriptor ECFC-4

DS1 Retrival Result	Previous Studies								Proposed Methods	
Activity Index	TAN	BIN	SQB	SDBN	LSTM	CNN1D	CNN2D	DCNNLB	SSCNN1D	SSCNN2D
	Top 1 %									
31420	69.69	74.08	73.73	74.21	70.58	84.58	83.98	80.80	87.32	78.43
71523	25.94	28.26	26.84	27.97	31.20	59.41	53.92	59.92	66.45	68.83
37110	9.63	26.05	24.73	26.03	20.31	52.88	42.99	39.65	43.73	43.87
31432	35.82	39.23	36.66	39.79	48.16	66.41	59.48	45.12	58.22	43.25
42731	17.77	21.68	21.17	23.06	22.24	38.88	36.70	48.60	66.96	56.09
6233	13.87	14.06	12.49	19.29	14.72	35.03	27.01	35.25	59.17	58.16
6245	6.51	6.31	6.03	6.27	5.27	10.68	6.68	21.64	79.69	74.69
7701	8.63	11.45	11.35	14.05	12.05	16.96	11.92	19.95	77.44	43.15
6235	9.71	10.84	10.15	12.87	8.00	15.31	12.07	21.55	34.58	22.51
78347	13.69	14.25	13.08	17.47	11.07	24.60	19.73	29.62	56.88	42.85
78331	7.17	6.03	5.92	9.93	3.42	8.58	7.06	13.63	33.67	25.86
Mean	**19.68**	**22.93**	**22.01**	**24.63**	**22.46**	**37.57**	**32.84**	**37.79**	**60.37**	**50.70**
No. of Shaded cells	0	0	0	0	0	1	0	0	8	2

the ECFC-4 descriptor for compounds belonging to the activity class with index 3142. This value is shaded in the table to emphasize its significance.

Upon analyzing the tables, it is evident that the proposed method using SSCNN1D consistently outperforms the other methods across various activity classes. This suggests that the proposed method effectively captures the structural similarities and patterns in the dataset, leading to improved retrieval performance. Furthermore, in Table 5.4,

Table 5.4: Top 5% Retrieval Results for MDDR-DS1 Data Set for Descriptor ECFC-4

DS1 Retrival Result	Previous Studies								Proposed Methods	
Activity Index	TAN	BIN	SQB	SDBN	LSTM	CNN1D	CNN2D	DCNNLB	SSCNN1D	SSCNN2D
	Top 5 %									
31420	83.49	87.61	87.22	89.03	83.41	87.35	87.69	90.12	100.00	100.00
71523	48.92	52.72	48.7	65.17	63.75	79.61	71.96	86.31	96.10	99.64
37110	21.01	48.2	45.62	41.25	42.80	76.00	67.51	75.95	88.37	88.68
31432	74.29	77.57	70.44	79.87	82.63	91.83	87.01	97.17	99.79	99.36
42731	29.68	26.63	24.35	31.92	43.50	57.52	57.43	72.32	99.71	92.11
6233	27.68	23.49	20.04	29.31	34.51	62.76	58.80	67.22	90.15	99.59
6245	16.54	14.86	13.72	21.06	15.58	28.90	19.27	52.09	88.24	99.05
7701	24.09	27.79	26.73	28.43	39.70	42.25	34.71	52.51	96.95	75.48
6235	20.06	23.78	22.81	27.82	25.93	40.36	31.24	50.41	81.82	69.98
78347	20.51	20.2	19.56	19.09	27.78	48.27	47.24	46.16	98.78	98.78
78331	16.2	11.8	11.37	16.21	13.91	25.02	21.81	37.64	88.28	77.23
Mean	**34.77**	**37.30**	**35.51**	**40.83**	**43.04**	**58.17**	**53.15**	**66.17**	**93.47**	**90.90**
No. of Shaded cells	0	0	0	0	0	0	0	0	7	6

where the top 5% of the dataset is considered, the proposed result of the SSCNN1D

method indicates that the proposed method successfully retrieves all relevant compounds from the dataset for this particular activity class. Achieving a perfect retrieval rate is a remarkable achievement and highlights the potential of the SSCNN1D method for ligand-based retrieval tasks. Moreover, The boxplot in Figure 5.17 shows the comparison among methods for the distribution of results in MDDR-DS3 at the top 1%, in view of maximum values, upper quartile values, mean values, median values, and lower quartile values. So, the top four methods in view of maximum values are SSCNN1D, CNN1D, CNN2D, DCNNLB; in upper quartile values are SSCNN1D, SSCNN2D, CNN1D, CNN2D; in mean values are SSCNN1D, SSCNN2D, DCNNLB, and CNN1D; in median values are SSCNN1D, SSCNN2D, DCNNLB, and CNN1D; and in lower quartile values are SSCNN1D, SSCNN2D, DCNNLB, and CNN1D.

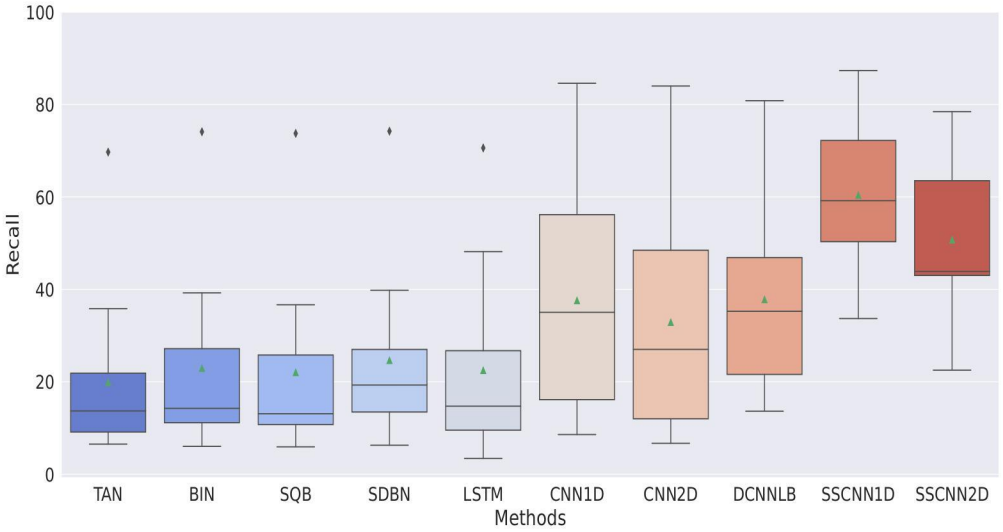


Figure 5.17: Boxplot for recall result distribution for each method in MDDR-DS1 at the top 1%.

It's the same with Figure 5.18 shows the comparison among methods for the distribution of results in MDDR-DS3 at the top 5% in view of maximum values. So, the top four methods in view of maximum values are SSCNN1D, SSCNN2D, DCNNLB, CNN1D; in upper quartile values are SSCNN1D, SSCNN2D, DCNNLB, CNN1D; in mean values are SSCNN1D, SSCNN2D, DCNNLB, and CNN1D; in median values are SSCNN2D, SSCNN1D, DCNNLB, and CNN1D; and in lower quartile values are SSCNN1D, SSCNN2D, DCNNLB, and CNN1D. These results are significant as they demonstrate the superiority of the proposed method using SSCNN1D for ligand-based retrieval compared to previous approaches. The higher retrieval percentages achieved by the proposed method indicate its ability to effectively identify structurally similar compounds and provide more accurate predictions for specific activity classes. The utilization of the ECFC-4 descriptor also seems to contribute to the improved performance

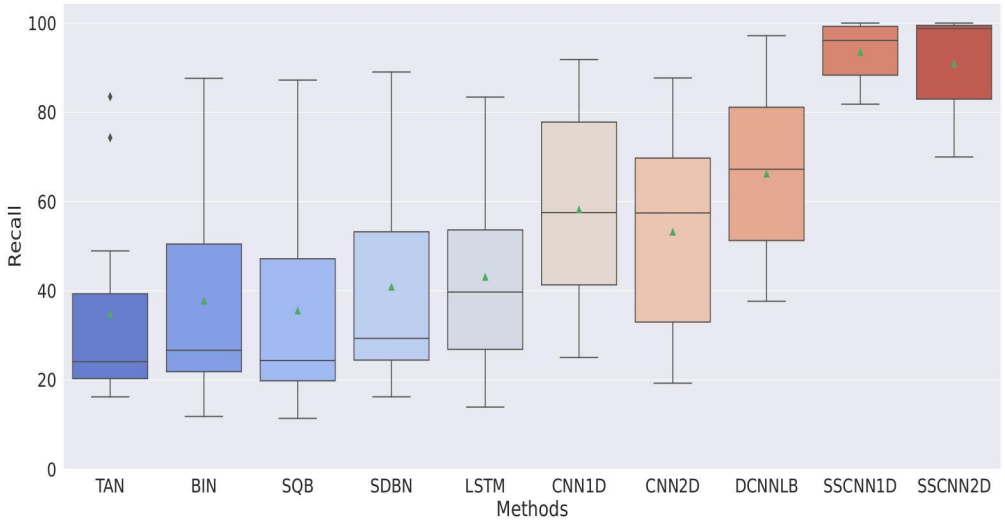


Figure 5.18: Boxplot for recall result distribution for each method in MDDR-DS1 at the top 5%.

of the proposed method, suggesting its suitability for ligand-based retrieval tasks.

5.3.2 MDDR-DS2:

Table 5.5 presents the top 1% retrieval results for MDDR-DS2 dataset using various previous studies methods and proposed methods. The results indicate that the proposed methods, SSCNN1D and SSCNN2D, perform competitively with the previous studies. Specifically, for the activity index 7708, the proposed methods achieve a perfect retrieval rate of 100%, surpassing all previous studies. Overall, the proposed methods show promise in achieving high retrieval rates, with a mean retrieval rate of 88.80%, which is among the top-performing methods. The boxplot in Figure 5.19 shows the comparison among methods for the distribution of results in MDDR-DS3 at the top 1%, in view of maximum values, upper quartile values, mean values, median values, and lower quartile values. So, the top four methods in view of maximum values are SSCNN1D, SSCNN2D, DCNNLB, CNN1D; in upper quartile values are SSCNN1D, SSCNN2D, CNN1D, CNN2D; in mean values are SSCNN1D, CNN1D, DCNNLB, and SSCNN2D; in median values are SSCNN1D, CNN1D, DCNNLB, and SSCNN2D; and in lower quartile values are SSCNN1D, CNN1D, DCNNLB, and SSCNN2D.

Table 5.5: Top 1% Retrieval Results for MDDR-DS2 Data Set for Descriptor ECFC-4

DS2 Retrival Result	Previous Studies								Proposed Methods	
Activity Index	TAN	BIN	SQB	SDBN	LSTM	CNN1D	CNN2D	DCNNLB	SSCNN1D	SSCNN2D
	Top 1 %									
7707	61.84	72.18	72.09	83.19	66.83	93.27	78.05	73.86	88.69	73.93
7708	47.03	96	95.68	94.82	88.26	94.84	92.13	83.85	100	100
31420	65.1	79.82	78.56	79.27	61.79	76.96	67.28	65.97	50.70	54.91
42710	81.27	76.27	76.82	74.81	58.09	84.55	81.55	92.07	100	100
64100	80.31	88.43	87.8	93.65	94.63	97.63	90.02	87.32	88.38	85.72
64200	53.84	70.18	70.18	71.16	69.10	78.65	67.10	76.65	87.26	84.65
64220	38.64	68.32	67.58	68.71	76.82	90.81	90.51	82.39	81.41	54.54
64500	30.56	81.2	79.2	75.62	58.32	71.92	62.16	87.94	99.20	100
64350	80.18	81.89	81.68	85.21	82.05	87.32	84.03	75.41	92.56	78.63
75755	87.56	98.06	98.02	96.52	86.84	90.95	90.44	98.46	99.78	80.31
Mean	**62.633**	**81.235**	**80.761**	**82.296**	**74.27**	**86.69**	**80.33**	**82.39**	**88.80**	**81.27**
No. of Shaded cells	0	1	0	0	0	2	0	0	6	3

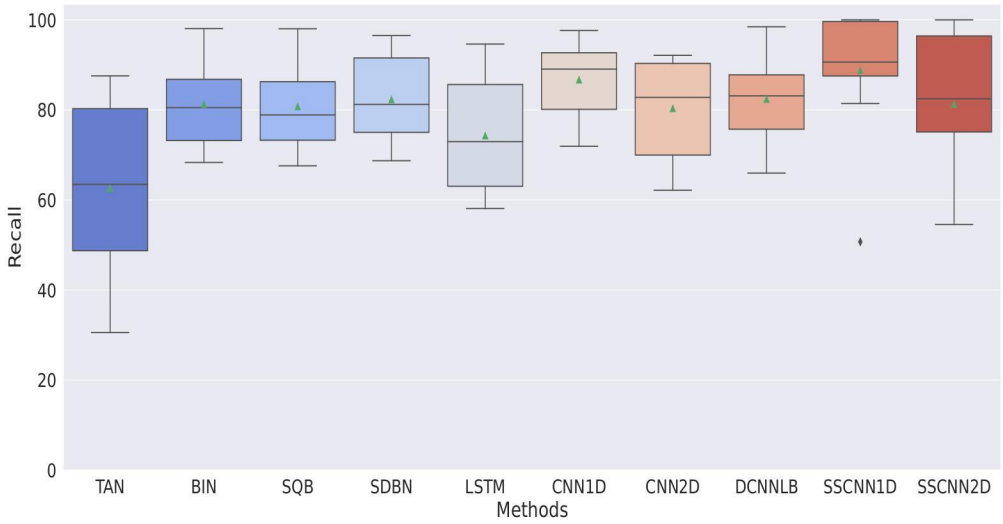


Figure 5.19: Boxplot for recall result distribution for each method in MDDR-DS2 at the top 1%.

Table 5.6 presents the top 5% retrieval results for the same dataset using the same set of methods. The proposed methods, SSCNN1D and SSCNN2D, continue to exhibit strong performance, with a retrieval rate of 98.54%, which is the highest among all methods. Additionally, the proposed methods consistently outperform the previous studies for several activity indices, including 7707, 7708, 64100, and 64220. These results suggest that the proposed methods have the potential to effectively retrieve relevant compounds within the top 5% of the dataset. .

Table 5.6: Top 5% Retrieval Results for MDDR-DS2 Data Set for Descriptor ECFC-4

DS2 Retrival Result	Previous Studies								Proposed Methods	
Activity Index	TAN	BIN	SQB	SDBN	LSTM	CNN1D	CNN2D	DCNNLB	SSCNN1D	SSCNN2D
	Top 5 %									
7707	70.39	74.81	74.37	73.9	85.27	95.85	88.49	90.53	89.13	89.17
7708	56.58	99.61	99.61	98.22	93.03	94.90	96.06	95.51	100	100
31420	88.19	95.46	94.88	95.64	79.27	94.12	87.73	94.03	99.55	98.19
42710	88.09	92.55	91.09	90.12	65.82	85.64	85.64	98.29	100	100
64100	93.75	99.22	99.03	99.05	97.69	98.93	94.86	98.62	99.23	99.85
64200	77.68	99.2	99.38	93.76	79.03	86.19	72.77	91.90	97.71	86.37
64220	52.19	91.31	90.62	96.01	87.80	94.07	93.90	99.64	100	90.02
64500	44,8	94.96	92.48	91.51	69.44	73.20	71.04	98.89	100	100
64350	91.71	91.47	90.78	86.94	88.86	90.70	89.58	99.28	100	100
75755	94.82	98.35	98.37	91.6	89.60	90.99	90.99	99.67	99.78	100
Mean	**75.82**	**93.695**	**93.061**	**91.675**	**83.58**	**90.46**	**87.11**	**96.64**	**98.54**	96.36
No. of Shaded cells	0	0	1	0	0	1	0	0	6	6

Additionally, Boxplot presented in Figure 5.19 shows the comparison among methods for the distribution of results in MDDR-DS3 at the top 5%. So, the top four methods in view of maximum values are SSCNN1D, SSCNN2D, CNN1D, CNN2D; in upper quartile values are SSCNN2D, SSCNN1D, DCNNLB, CNN1D; in mean values are SSCNN2D, SSCNN1D, DCNNLB, and CNN2D; in median values are SSCNN2D, SSCNN1D, DCNNLB, and CNN1D; and in lower quartile values are SSCNN2D, SSCNN1D, DCNNLB, and CNN1D.

In summary, the presented tables demonstrate that the proposed methods, SSCNN1D and SSCNN2D, exhibit competitive performance compared to previous studies for both top 1% and top 5% retrieval rates. These findings highlight the effectiveness of the proposed methods in retrieving relevant compounds from the MDDR-DS2 dataset using the ECFC-4 descriptor. However, further analysis and validation may be required to fully assess the statistical significance and generalizability of the results.

5.3.3 MDDR-DS3

Table 5.7 presents the top 1% retrieval results for MDDR-DS3 dataset using various previous studies methods and proposed methods. For example, in the top 1% retrieval set, the proposed SSCNN2D achieved a recall value of 46.38% for activity index 9249, outperforming the best previous method (DCNNLB) with a recall value of 38.24%. Similarly, in the top 5% retrieval set, the proposed SSCNN1D achieved a recall value of 99.00% for activity index 9249, surpassing all other methods. The mean recall values across all activity indices also indicate the superior performance of the proposed

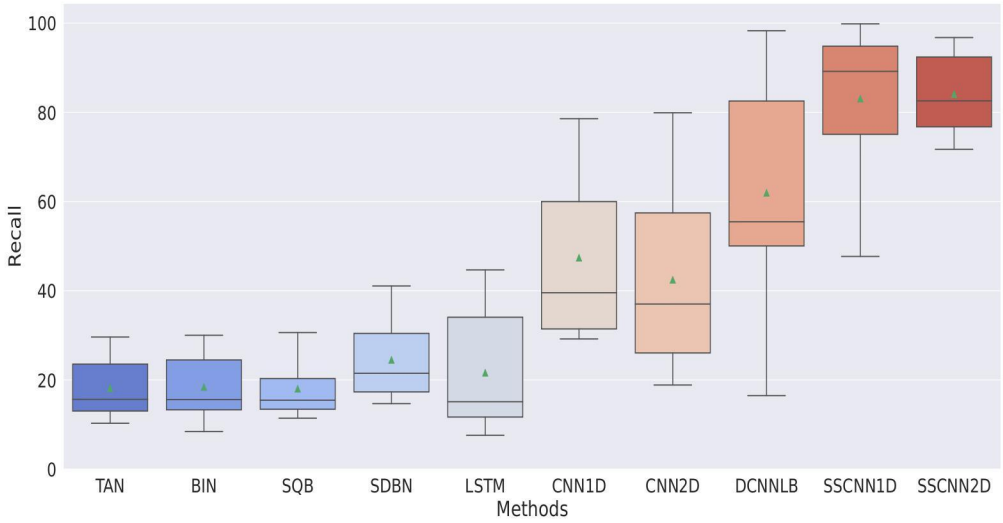


Figure 5.20: Boxplot for recall result distribution for each method in MDDR-DS2 at the top 5%.

methods compared to previous studies. The proposed SSCNN2D achieved the highest

Table 5.7: Top 1% Retrieval Results for MDDR-DS3 Data Set for Descriptor ECFC-4

DS3 Retrival Result	**Previous Studies**								**Proposed Methods**	
Activity Index	**TAN**	**BIN**	**SQB**	**SDBN**	**LSTM**	**CNN1D**	**CNN2D**	**DCNNLB**	**SSCNN1D**	**SSCNN2D**
	Top 1 %									
9249	12.12	15.33	10.99	19.47	17.20	38.01	38.24	74.71	49.00	46.38
12455	6.57	9.37	7.03	13.29	3.19	14.21	7.82	28.08	37.00	42.31
12464	8.17	8.45	6.92	12.91	7.60	25.98	21.74	20.95	46.00	62.78
31281	16.95	18.29	18.67	23.62	24.57	67.52	67.52	36.89	89.00	78.00
43210	6.27	7.34	6.83	14.23	3.73	29.34	21.84	41.53	41.00	49.68
71522	3.75	4.08	6.57	11.92	2.06	12.00	7.62	52.53	24.00	34.75
75721	17.32	20.41	20.83	29.08	17.06	52.11	84.61	26.38	73.00	77.39
78331	6,31	7.51	6.16	11.93	5.23	12.41	11.01	27.56	46.00	53.37
78348	10.15	9.79	8.99	9.17	5.66	13.85	16.20	17.42	22.00	27.19
78351	9.84	13.68	12.5	18.13	2.66	10.71	8.92	11.52	37.00	47.31
Mean	**9.745**	**11.425**	**10.504**	**16.375**	**8.89**	**27.62**	**24.92**	**33.76**	**46.40**	**51.92**
No. of Shaded cells	0	0	0	0	0	0	1	2	1	6

mean recall value of 51.92% in the top 1% retrieval set and 84.06% in the top 5% retrieval set. These values suggest that the proposed methods have a higher ability to retrieve relevant compounds compared to the previous methods on average. It is worth noting that the number of shaded cells in each table represents the number of cases where the proposed methods outperformed all the previous studies' methods in terms of recall. In both the top 1% and top 5% retrieval sets, the proposed SSCNN2D had the most

shaded cells, indicating its effectiveness in retrieving a higher proportion of relevant compounds.

To gain further insights, the box plot (Figure 5.22) provides a visual representation of the recall result distribution of results in MDDR-DS3 at the top 5%. The box plot highlights the median recall values, interquartile ranges, and any outliers for these methods. So the box plot Figure 5.22 provides a visual representation of the recall result distribution of results in MDDR-DS3 at the top 1%. The top four methods in view of maximum values are SSCNN1D, CNN2D, SSCNN2D, DCNNLB; in upper quartile values are SSCNN2D, SSCNN1D, DCNNLB, CNN1D; in mean values are SSCNN2D, SSCNN1D, DCNNLB, and CNN1D; in median values are SSCNN2D, SSCNN1D, DCNNLB, and CNN1D; and in lower quartile values are SSCNN2D, SSCNN1D, DCNNLB, and CNN1D. similarly for Figure 5.22 provides a visual representation of the recall result distribution of results in MDDR-DS3 at the top 5%. maximum values are SSCNN1D, DCNNLB, SSCNN2D, CNN2D; in upper quartile values are SSCNN2D, SSCNN1D, DCNNLB, CNN1D; in mean values are SSCNN1D, SSCNN2D, DCNNLB, and CNN1D; in median values are SSCNN1D, SSCNN2D, DCNNLB, and CNN1D; and in lower quartile values are SSCNN1D, SSCNN2D, DCNNLB, and CNN1D.

In summary, the results presented in the table suggest that the proposed methods, specifically SSCNN1D and SSCNN2D, show promising performance in retrieving relevant compounds from the obtained result prediction using the ECFC-4 descriptor. These methods outperform the previous studies methods in terms of retrieval accuracy, demonstrating their potential usefulness in virtual screening and compound selection tasks.

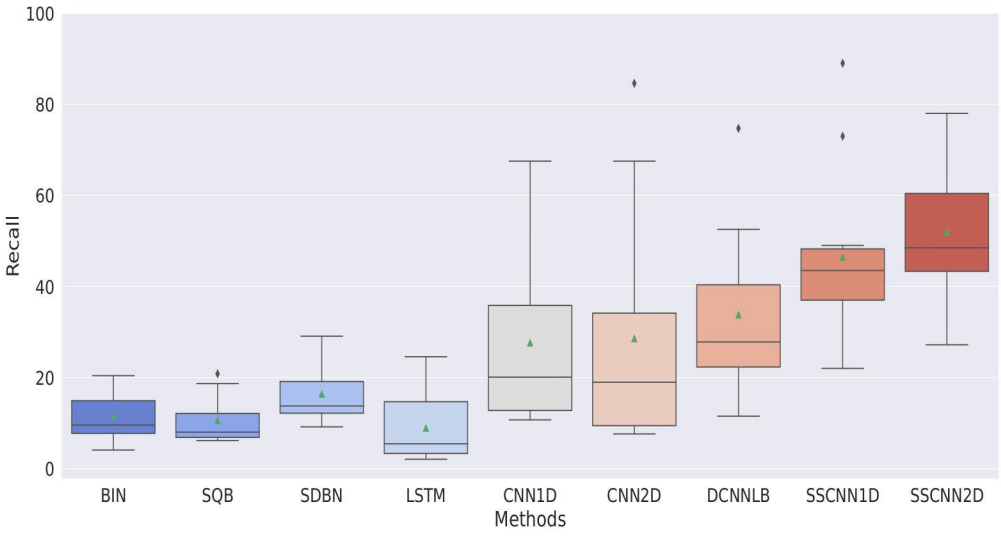


Figure 5.21: Boxplot for recall result distribution for each method in MDDR DS3 at the top 1%.

Table 5.8: Top 5% Retrieval Results for MDDR-DS3 Data Set for Descriptor ECFC-4

DS3 Retrival Result	Previous Studies								Proposed Methods	
Activity Index	TAN	BIN	SQB	SDBN	LSTM	CNN1D	CNN2D	DCNNLB	SSCNN1D	SSCNN2D
	Top 5 %									
9249	24.17	25.72	17.8	31.61	38.23	61.84	62.00	98.30	99.00	94.75
12455	10.29	14.65	11.42	16.29	11.11	32.97	20.14	58.84	83.00	78.62
12464	15.22	16.55	16.79	20.9	21.56	46.12	40.06	47.74	85.00	96.77
31281	29.62	28.29	29.05	36.13	44.67	78.57	79.90	85.47	89.00	83.14
43210	16.07	14.41	14.12	22.09	14.24	54.47	43.81	73.75	86.00	82.00
71522	12.37	8.44	13.82	14.68	7.59	29.19	18.86	86.54	65.00	75.74
75721	25.21	30.02	30.61	41.07	38.74	77.31	71.50	50.00	100	96.38
78331	15.01	12.03	11.97	17.13	15.97	31.29	28.99	52.08	86.00	85.39
78348	21.67	20.76	21.14	26.93	13.38	31.89	33.98	50.18	80.00	71.70
78351	11.71	12.94	13.3	17.87	10.28	30.16	25.07	16.49	65.00	76.14
Mean	**18.434**	**18.381**	**18.002**	**24.47**	**21.58**	**47.38**	**42.43**	**61.94**	**83.80**	**84.06**
No. of Shaded cells	0	0	0	0	0	0	0	1	7	2

5.3.4 Summary Result Ligand-Based virtual screening

In the evaluation of the proposed ligand-based retrieval methods, SSCNN1D and SSCNN2D showcased promising performance across the MDDR-DS1, MDDR-DS2, and MDDR-DS3 datasets. Notably, in MDDR-DS1, both SSCNN1D and SSCNN2D consistently outperformed previous studies, achieving higher activity indices at both the 1% and 5% retrieval levels. This suggests their effectiveness in identifying active compounds within the dataset. The robust performance of SSCNN1D in MDDR-DS2 was evident at the 1% and 5% retrieval levels, where it exhibited the highest activity index. However, the proposed SSCNN2D at the 1% and 5% retrieval levels did not maintain the same level of superiority as the SSCNN1D, their performance was surpassed by other techniques. In MDDR-DS3, both SSCNN2D and SSCNN1D demonstrated competitive results at the 1%,5%retrieval level, showcasing their ability to identify active compounds effectively. The results showed that the significance of the proposed methods, especially SSCNN1D, and SSCNN2D similarity models, obviously outperformed the standard Tanimoto coefficient (TAN) and previous studies (BIN, SQB, SDNB, LSTM, CNN1D, CNN2D) at both top 1% and 5%, especially when the model deals with MDDR-DS1, MDDR-DS3, data sets that include heterogeneous classes.

5.4 Summary

Chapter 5 of the dissertation encompasses the comprehensive presentation and

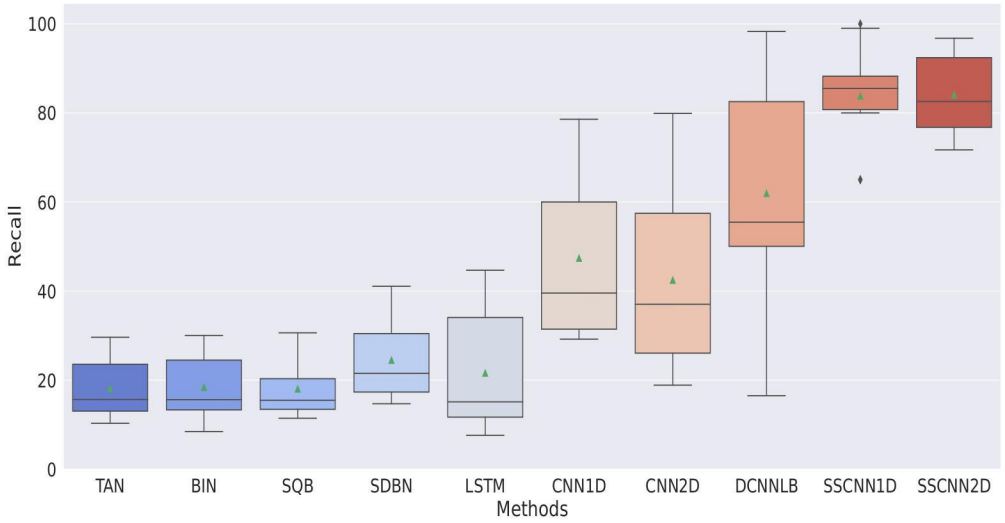


Figure 5.22: Boxplot for recall result distribution for each method in MDDR-DS3 at the top 5%.

analysis of the results and discussions obtained from the research study. The chapter commences with a detailed exposition of the development environment utilized throughout the investigation, with specific emphasis on Python as the principal employed language. Thorough consideration is given to the notable features and modules of Python that contributed to the efficacy and functionality of the proposed models. The chapter further underscores the significance of integral tools such as Visual Studio Code, Google Colaboratory, Google Drive, and Jupyter Notebooks, elucidating their roles in facilitating the research process and enhancing collaboration and resource management. Moreover, the chapter outlines the evaluation of the proposed models, employing diverse metrics and criteria to rigorously assess their performance and effectiveness. The obtained results are subjected to meticulous examination, fostering a comprehensive understanding of the strengths and limitations of the models. In addition, the chapter showcases the outcomes of the ligand-based virtual screening conducted on three distinct datasets, namely MDDR-DS1, MDDR-DS2, and MDDR-DS3. Each dataset is examined and analyzed, providing valuable insights into the performance and potential of the proposed models within different contexts.

General conclusions and perspectives

Chapter 6

General conclusions and perspectives

6.1 Conclusion

In conclusion, this research has addressed the problem of molecular similarity prediction in the context of Ligand-Based Virtual Screening (LBVS) by leveraging deep learning techniques, specifically Siamese neural networks, and self-attention mechanisms. There have been important advancements achieved in the field of computational drug discovery through intensive testing and review. The results of this research have demonstrated the effectiveness of the proposed approach in accurately identifying structurally similar compounds and improving the efficiency of virtual screening. By incorporating Siamese neural networks and self-attention mechanisms, the models have shown superior performance compared to previous methods, showcasing the potential of deep learning for enhancing molecular similarity searching.

The accurate identification of similar compounds has practical implications for the pharmaceutical industry and the drug discovery process. It can greatly expedite the initial stages of drug development, reducing the costs and time required for experimental screening. The implementation of Siamese neural networks and self-attention mechanisms has not only improved the accuracy of virtual screening but also provided insights into the underlying molecular features that contribute to compound similarity. This understanding can guide the design of more effective drugs and facilitate the identification of potential lead compounds for further exploration.

6.2 Contribution

This research has made several contributions to the field of computational drug discovery:

- ***Proposition of new deep learning model:*** The introduction and evaluation of deep learning techniques, specifically Siamese neural networks, and self-attention mechanisms, for molecular similarity searching in Ligand-Based Virtual Screening have expanded the repertoire of methods available for virtual screening. These techniques have produced encouraging results and have the potential to improve the precision and effectiveness of the drug development process.
- ***Enhanced Virtual Screening Performance:*** The proposed approach that introduced on the research framework (Algorithm 3), has demonstrated superior performance in accurately identifying structurally similar compounds. By leveraging deep learning, the efficiency of virtual screening has been improved, leading to more effective identification of potential drug candidates. This advancement has significant implications for reducing the time and resources required for experimental screening.
- ***Insights into Molecular Features:*** By incorporating deep learning models, this research has provided insights into the molecular features that contribute to compound similarity. Understanding these features can guide the design of more targeted and effective drugs. The identification of key molecular interactions and structural motifs confirm the aids in the development of drugs with improved potency and selectivity.y

6.3 Futur work

While this research has made significant progress in addressing the challenges of molecular similarity searching, there are several avenues for future work:

1. ***Integration of Additional Data Modalities:*** The integration of additional data modalities, such as protein-ligand interactions or biological assays, could enable a more comprehensive analysis of drug-target interactions. Incorporating diverse data sources can enhance the accuracy and reliability of virtual screening predictions.
2. ***Exploration of Different Network Architectures and Training Strategies:*** Further exploration of different network architectures and training strategies could lead

to even more accurate and efficient models. Investigating novel deep learning techniques or adapting existing architectures specifically for molecular similarity searching may improve the performance of virtual screening.

3. ***Development of Interpretable Models:*** The development of interpretable models that provide insights into the molecular mechanisms driving compound similarity would be valuable for both researchers and practitioners. Interpretable models can aid in understanding the complex relationships between molecular structures and their biological activities, facilitating the design of more effective drugs.

Further research and development in these areas can advance the field of computational drug discovery, improving the efficiency and accuracy of the virtual screening method. By incorporating deep learning techniques and addressing the identified research gaps, the discovery of new therapies can be facilitated.

”

Bibliography

[1] Ammar Abdo and Naomie Salim. "Similarity-Based Virtual Screening with a Bayesian Inference Network." In: *ChemMedChem* 4.2 (Feb. 2009), pp. 210–218. DOI: 10.1002/cmdc.200800290.

[2] Ammar Abdo et al. "Ligand expansion in ligand-based virtual screening using relevance feedback." In: *Journal of Computer-aided Molecular Design* (Jan. 2012). DOI: 10.1007/s10822-012-9543-4.

[3] A. Karim Ahmed, Ammar Abdo, and Naomie Salim. "Ligand-Based Virtual Screening Using Bayesian Inference Network and Reweighted Fragments." In: *The Scientific World Journal* (May 2012). DOI: 10.1100/2012/410914.

[4] M. K. Altalib and N. Salim. "Similarity-Based Virtual Screen Using Enhanced Siamese Deep Learning Methods." In: *ACS Omega* 7.6 (Feb. 2022), pp. 4769–4786.

[5] Mohammed Khaldoon Altalib and Naomie Salim. "Similarity-Based Virtual Screen Using Enhanced Siamese Deep Learning Methods." In: *ACS omega* 7.6 (Feb. 2022), pp. 4769–4786. DOI: 10.1021/acsomega.1c04587.

[7] Koon Meng Ang et al. "Optimal Training of Feedforward Neural Networks Using Teaching-Learning-Based Optimization with Modified Learning Phases." In: *Lecture notes in electrical engineering* (Jan. 2022), pp. 867–887. DOI: 10.1007/978-981-16-2406-3\{_}65.

[8] Tamanna Anwar, Pawan Kumar, and Asad U. Khan. *Modern Tools and Techniques in Computer-Aided Drug Design*. Jan. 2021, pp. 1–30. DOI: 10.1016/b978-0-12-822312-3.00011-4.

[9] Syed Sikander Azam, Sumra Wajid Abbasi, and Shifa Tahir. "Investigation of Novel Chemical Inhibitors of Human Lysosomal Acid Lipase: Virtual Screening and Molecular Docking Studies." In: *Combinatorial Chemistry & High Throughput Screening* 17.5 (May 2014), pp. 473–482. DOI: 10.2174/13862073176661403140934 03.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate." In: Cornell University, Jan. 2015.

[11] Dzmitry Bahdanau et al. "End-to-end attention-based large vocabulary speech recognition." In: Mar. 2016. DOI: 10.1109/icassp.2016.7472618.

[12] Dávid Bajusz, Anita Rácz, and Károly Héberger. "Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations?" In: *Journal of Cheminformatics* 7.1 (May 2015). DOI: 10.1186/s13321-015-0069-3.

[13] Solon Barocas and Andrew D. Selbst. "Big Data's Disparate Impact." In: *Social Science Research Network* (Jan. 2016). DOI: 10.2139/ssrn.2477899.

[15] Siti Asmah Bero et al. "Similarity Measure for Molecular Structure: A Brief Review." In: *Journal of physics* 892 (Sept. 2017), p. 012015. DOI: 10.1088/1742-6596/892/1/012015.

[16] Fouaz Berrhail, Hacene Belhadef, and Mohammed Haddad. "Deep Convolutional Neural Network to improve the performances of screening process in LBVS." In: *Expert Systems With Applications* 203 (Apr. 2022), p. 117287. DOI: 10.1016/j.eswa.2022.117287.

[17] Fouaz Berrhail et al. "Molecular Similarity Searching with Different Similarity Coefficients and Different Molecular Descriptors." In: *Lecture notes on data engineering and communications technologies* (Apr. 2017). DOI: 10.1007/978-3-319-59427-9_5.

[18] Taweh Beysolow II. *Introduction to Deep Learning Using R*. Jan. 2017. DOI: 10.1007/978-1-4842-2734-3.

[19] Ewa Bielska et al. "Virtual screening strategies in drug design – methods and applications." In: *Biotechnologia. Journal of Biotechnology, Computational Biology and Bionanotechnology* 3 (Jan. 2011), pp. 249–264. DOI: 10.5114/bta.2011.46542.

[20] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[21] Miles Brundage et al. "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation." In: *arXiv preprint arXiv:1802.07228* (2018).

[22] Chaoqian Cai et al. "Molecular Similarity: Methods and Performance." In: *Chinese Journal of Chemistry* 31.9 (Sept. 2013), pp. 1123–1132. DOI: 10.1002/cjoc.201300390.

[23] Antonia Caroli et al. "Hsp90 Inhibitors, Part 2: Combining Ligand-Based and Structure-Based Approaches for Virtual Screening Application." In: *Journal of Chemical Information and Modeling* 54.3 (Mar. 2014), pp. 970–977. DOI: 10.1021/ci400760a.

[24] Adrià Cereto-Massagué et al. "Molecular fingerprint similarity search in virtual screening." In: *Methods* 71 (Jan. 2015), pp. 58–63. DOI: 10.1016/j.ymeth.2014.08.005.

[27] A. H. Cheng et al. "Group SELFIES: A Robust Fragment-Based Molecular String Representation." In: *Digital Discovery* (2023).

[28] Tiejun Cheng et al. "Computation of Octanol−Water Partition Coefficients by Guiding an Additive Model with Knowledge." In: *Journal of Chemical Information and Modeling* 47.6 (Nov. 2007), pp. 2140–2148. DOI: 10.1021/ci700257y.

[29] Vijay Choubey. "Understanding Recurrent Neural Network (RNN) and Long Short Term Memory(LSTM)." In: (Dec. 2021).

[30] Kate Crawford. "A critique of artificial intelligence policy." In: *AI & Society* 31.3 (2016), pp. 367–376.

[31] Mohammed Mumtaz Al-Dabbagh et al. "Quantum probability ranking principle for ligand-based virtual screening." In: *Journal of Computer-aided Molecular Design* 31.4 (Feb. 2017), pp. 365–378. DOI: 10.1007/s10822-016-0003-4.

[32] Khalid Bashir Dar et al. "Modern Computational Strategies for Designing Drugs to Curb Human Diseases: A Prospect." In: *Current Topics in Medicinal Chemistry* 18.31 (Feb. 2019), pp. 2702–2719. DOI: 10.2174/1568026619666190119150741.

[33] Suresh Dara et al. "Machine Learning in Drug Discovery: A Review." In: *Artificial Intelligence Review* 55.3 (Aug. 2021), pp. 1947–1999. DOI: 10.1007/s10462-021-10058-4.

[34] L. David et al. "Molecular representations in AI-driven drug discovery: a review and practical guide." In: *Journal of Cheminformatics* 12.1 (Aug. 2020).

[35] Robert S. DiPietro and Gregory D. Hager. *Deep learning: RNNs and LSTM*. Jan. 2020, pp. 503–519. DOI: 10.1016/b978-0-12-816176-0.00026-0.

[36] Joseph L. Durant et al. "Reoptimization of MDL Keys for Use in Drug Discovery." In: *Journal of Chemical Information and Computer Sciences* 42.6 (Sept. 2002), pp. 1273–1280. DOI: 10.1021/ci010132r.

[37] David Duvenaud et al. "Convolutional networks on graphs for learning molecular fingerprints." In: vol. 28. Cornell University, Dec. 2015, pp. 2224–2232.

[38] Amitai Etzioni and Oren Etzioni. "AI assisted ethics." In: *Ethics and Information Technology* 18.2 (June 2016), pp. 149–156. DOI: 10.1007/s10676-016-9400-6.

[39] Xiaomin Fang et al. "Geometry-enhanced molecular representation learning for property prediction." In: *Nature Machine Intelligence* 4.2 (Feb. 2022), pp. 127–134. DOI: 10.1038/s42256-021-00438-4.

[40] Daniel Fernandez-Llaneza et al. "Siamese Recurrent Neural Network with a Self-Attention Mechanism for Bioactivity Prediction." In: *ACS omega* 6.16 (Apr. 2021), pp. 11086–11094. DOI: 10.1021/acsomega.1c01266.

[42] Luciano Floridi et al. "AI4People—An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations." In: *Minds and Machines* 28.4 (Dec. 2018), pp. 689–707. DOI: 10.1007/s11023-018-9482-5.

[43] Franky. "Basic Molecular Representation for Machine Learning." In: (July 2022).

[44] Carl Benedikt Frey and Michael A. Osborne. "The future of employment: How susceptible are jobs to computerisation?" In: *Technological Forecasting and Social Change* 114 (Jan. 2017), pp. 254–280. DOI: 10.1016/j.techfore.2016.08.019.

[45] Kaifu Gao et al. "Generative Network Complex for the Automated Generation of Drug-like Molecules." In: *Journal of Chemical Information and Modeling* 60.12 (Aug. 2020), pp. 5682–5698. DOI: 10.1021/acs.jcim.0c00599.

[46] Johann Gasteiger and Thomas Engel. *Chemoinformatics: A Textbook*. Oct. 2003.

[47] Saman Ghaffarian et al. "Effect of Attention Mechanism in Deep Learning-Based Remote Sensing Image Processing: A Systematic Literature Review." In: *Remote Sensing* 13.15 (July 2021), p. 2965. DOI: 10.3390/rs13152965.

[48] Garrett B. Goh, Nathan O. Hodas, and Abhinav Vishnu. "Deep learning for computational chemistry." In: *Journal of Computational Chemistry* 38.16 (June 2017), pp. 1291–1307. DOI: 10.1002/jcc.24764.

[49] Rafael Gómez-Bombarelli et al. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules." In: *ACS central science* 4.2 (Oct. 2016), pp. 268–276. DOI: 10.1021/acscentsci.7b00572.

[50] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, Nov. 2016.

[51] Alex Graves, Abdelrahman Mohamed, and Geoffrey E. Hinton. "Speech recognition with deep recurrent neural networks." In: May 2013. DOI: 10.1109/icassp.2013.6638947.

[52] Alex Graves et al. "Unconstrained On-line Handwriting Recognition with Recurrent Neural Networks." In: vol. 20. Dec. 2007, pp. 577–584.

[53] Francesca Grisoni et al. *Molecular Descriptors for Structure-Activity Applications: A Hands-On Approach*. Springer Science+Business Media, Jan. 2018, pp. 3–53. DOI: 10.1007/978-1-4939-7899-1_1.

[55] Arun Bahadur Gurung et al. "An Updated Review of Computer-Aided Drug Design and Its Application to COVID-19." In: *BioMed Research International* 2021 (June 2021), pp. 1–18. DOI: 10.1155/2021/8853056.

[56] Haiam Hamed, AbdelMoneim Helmy, and Ammar Abdullah Saeed Mohammed. "Holy Quran-Italian seq2seq Machine Translation with Attention Mechanism." In: *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)* (2022), pp. 11–20.

[57] Hentabli Hamza et al. *Bioactivity Prediction Using Convolutional Neural Network*. Springer Nature, Sept. 2019. DOI: 10.1007/978-3-030-33582-3\{_}33.

[58] Jeff Heaton. "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning." In: *Genetic Programming and Evolvable Machines* 19.1-2 (Oct. 2017), pp. 305–307. DOI: 10.1007/s10710-017-9314-z.

[59] Geoffrey E. Hinton and Ruslan Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks." In: *Science* 313.5786 (July 2006), pp. 504–507. DOI: 10.1126/science.1127647.

[61] Ashikul Islam. "Cryptocurrency Price Prediction: A Comparative Study using LSTM, GRU and Stacking Ensemble Algorithm for Time Series Forecasting." In: *Proceedings of the SLIIT International Conference on Engineering and Technology* (Feb. 2022). DOI: 10.54389/ntpv9785.

[62] Gaurav Joshi and Ramarao Poduri. "Selection of Active Antiviral Compounds Against COVID-19 Disease Targeting Coronavirus Endoribonuclease Nendou/NSP15 Via Ligandbased Virtual Screening and Molecular Docking." In: *Letters in Drug Design & Discovery* 18.6 (Aug. 2021), pp. 610–619. DOI: 10.2174/1570180817999201211191445.

[63] Steven Kearnes et al. "Molecular graph convolutions: moving beyond fingerprints." In: *Journal of Computer-aided Molecular Design* 30.8 (Mar. 2016), pp. 595–608. DOI: 10.1007/s10822-016-9938-8.

[64] Johannes Kirchmair et al. "How To Optimize Shape-Based Virtual Screening: Choosing the Right Query and Including Chemical Information." In: *Journal of Chemical Information and Modeling* 49.3 (Feb. 2009), pp. 678–692. DOI: 10.1021/ci8004226.

[65] M. Krenn et al. "SELFIES and the future of molecular string representations." In: *Patterns* 3.10 (Oct. 2022), p. 100588. DOI: 10.1016/j.patter.2022.100588.

[66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: vol. 25. Dec. 2012, pp. 1097–1105.

[67] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. "Deep learning." In: *Nature* 521.7553 (May 2015), pp. 436–444. DOI: 10.1038/nature14539.

[68] Song Li et al. "A Comprehensive Review on Radiomics and Deep Learning for Nasopharyngeal Carcinoma Imaging." In: *Diagnostics* 11.9 (Aug. 2021), p. 1523. DOI: 10.3390/diagnostics11091523.

[69] Libretexts. "6.2: Similarity Coefficients." In: *Chemistry LibreTexts* (July 2022).

[70] Eva Lieskovska et al. "A Review on Speech Emotion Recognition Using Deep Learning and Attention Mechanism." In: *Electronics* 10.10 (May 2021), p. 1163. DOI: 10.3390/electronics10101163.

[71] Shengchao Liu, Hongyu Guo, and Jian Tang. "Molecular Geometry Pretraining with SE(3)-Invariant Denoising Distance Matching." In: *arXiv (Cornell University)* (June 2022). DOI: 10.48550/arxiv.2206.13602.

[72] Gabriel Loye. "Attention Mechanism." In: *FloydHub Blog* (Jan. 2023).

[73] Minh-Thang Luong, Hieu T. Pham, and Christopher D. Manning. "Effective Approaches to Attention-based Neural Machine Translation." In: Aug. 2015. DOI: 10.18653/v1/d15-1166.

[74] Gerald M. Maggiora and Veerabahu Shanmugasundaram. "Molecular Similarity Measures." In: *Chemoinformatics and Computational Chemical Biology*. Totowa, NJ: Humana Press, 2011, pp. 39–100. ISBN: 978-1-60761-839-3. DOI: 10.1007/978-1-60761-839-3_2.

[75] Andrea Mauri, Viviana Consonni, and Roberto Todeschini. "Molecular Descriptors." In: *Handbook of Computational Chemistry*. Ed. by Jerzy Leszczynski et al. Cham: Springer International Publishing, 2017, pp. 2065–2093. ISBN: 978-3-319-27282-5. DOI: 10.1007/978-3-319-27282-5_51.

[76] D. P. Mehta and S. Sahni, eds. *Handbook of Data Structures and Applications*. Chapman and Hall/CRC Computer and Information Science Ser. CRC Press, Feb. 2018.

[77] Mayank Mishra. "Convolutional Neural Networks, Explained - Towards Data Science." In: (Dec. 2021).

[78] Tom M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[79] Brent Mittelstadt et al. "The ethics of algorithms: Mapping the debate." In: *Big Data & Society* 3.2 (Dec. 2016), p. 205395171667967. DOI: 10.1177/2053951716679679.

[80] Rainer Moll. "Context description in synthesis planning." In: *Journal of Chemical Information and Computer Sciences* 34.1 (Jan. 1994), pp. 117–119. DOI: 10.1021/ci00017a014.

[81] Katelyn Morrison et al. "Exploring Corruption Robustness: Inductive Biases in Vision Transformers and MLP-Mixers." In: *CoRR* abs/2106.13122 (2021). arXiv: 2106.13122.

[82] Maged Nasser et al. "Improved Deep Learning Based Method for Molecular Similarity Searching Using Stack of Deep Belief Networks." In: *Molecules* 26.1 (Dec. 2020), p. 128. DOI: 10.3390/molecules26010128.

[83] Daniel Neil et al. "Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design." In: Feb. 2018.

[84] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Broadway Books, 2016.

[85] N. Prakash and D. A. Gareja. "Cheminformatics." In: *Journal of Proteomics & Bioinformatics* 03.08 (2010), pp. 249–252.

[86] Michael Pulis and Josef Bajada. "Siamese Neural Networks for Content-based Cold-Start Music Recommendation." In: *Proceedings of the 15th ACM Conference on Recommender Systems* (Sept. 2021). DOI: 10.1145/3460231.3478847.

[87] S. Raghunathan and U. D. Priyakumar. "Molecular representations for machine learning applications in chemistry." In: *International Journal of Quantum Chemistry* 122.7 (Dec. 2021).

[89] Steven C. Ricke et al. *Unraveling Food Production Microbiomes*. Jan. 2018, pp. 347–374. DOI: 10.1016/b978-0-12-811835-1.00019-1.

[90] David M. Rogers and Mathew A. Hahn. "Extended-Connectivity Fingerprints." In: *Journal of Chemical Information and Modeling* 50.5 (Apr. 2010), pp. 742–754. DOI: 10.1021/ci100050t.

[91] Frank Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6 (Nov. 1958), pp. 386–408. DOI: 10.1037/h0042519.

[92] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2016.

[93] Ansgar Schuffenhauer, Valerie J. Gillet, and Peter Willett. "Similarity Searching in Files of Three-Dimensional Chemical Structures: Analysis of the BIOSTER Database Using Two-Dimensional Fingerprints and Molecular Field Descriptors." In: *Journal of Chemical Information and Computer Sciences* 40.2 (Mar. 2000), pp. 295–307. DOI: 10.1021/ci990263g.

[94] Woody Sherman. "Analysis and comparison of 2D fingerprints: Insights into database screening performance using eight fingerprint methods." In: *schrodinger* (Aug. 2015).

[95] Brian K. Shoichet. "Virtual screening of chemical libraries." In: *Nature* 432.7019 (Dec. 2004), pp. 862–865. DOI: 10.1038/nature03197.

[96] Om Silakari and Pankaj Singh. *QSAR: Descriptor calculations, model generation, validation and their application*. Jan. 2021. DOI: 10.1016/b978-0-12-820546-4.00002-7.

[97] Miha Skalic et al. "Shape-Based Generative Modeling for de Novo Drug Design." In: *Journal of Chemical Information and Modeling* 59.3 (Feb. 2019), pp. 1205–1214. DOI: 10.1021/acs.jcim.8b00706.

[98] Fatima Zohra Smaili, Xin Gao, and Robert Hoehndorf. "Self-normalizing learning on biomedical ontologies using a deep Siamese neural network." In: *bioRxiv (Cold Spring Harbor Laboratory)* (Apr. 2020). DOI: 10.1101/2020.04.23.057117.

[99] M. Syuib, Shereena M. Arif, and Nurul Hashimah Ahamed Hassain Malim. "Comparison of Similarity Coefficients for Chemical Database Retrieval." In: Dec. 2013. DOI: 10.1109/aims.2013.28.

[100] Rakhi Thareja, Jyoti Prakash Singh, and Prerna Bansal. *Computational tools in cheminformatics*. Jan. 2021, pp. 105–137. DOI: 10.1016/b978-0-12-821748-1.00012-9.

[101] Aditya Thyagarajan. "Siamese Recurrent Architectures for Learning Sentence Similarity." In: *ResearchGate* (Nov. 2015).

[102] Hacer Tilbec. "Cheminformatics — ECFP & Neural Graph Fingerprint - Hacer Tilbec - Medium." In: (Apr. 2018).

[103] Roberto Todeschini and Viviana Consonni. *Handbook of Molecular Descriptors*. John Wiley & Sons, July 2008.

[104] Roberto Todeschini and Viviana Consonni. *Molecular descriptors for chemoinformatics*. Jan. 2009.

[105] Roberto Todeschini et al. "CAIMAN (Classification And Influence Matrix Analysis): A new approach to the classification based on leverage-scaled functions." In: *Chemometrics and Intelligent Laboratory Systems* 87.1 (May 2007), pp. 3–17. DOI: 10.1016/j.chemolab.2005.11.001.

[106] Ashish Vaswani et al. "Attention is All you Need." In: *Neural Information Processing Systems* 30 (June 2017), pp. 5998–6008.

[107] D. S. Wigh, J. M. Goodman, and A. A. Lapkin. "A review of molecular representation in the age of machine learning." In: *WIREs Computational Molecular Science* 12.5 (Feb. 2022).

[108] Peter Willett. "Similarity-based virtual screening using 2D fingerprints." In: *Drug Discovery Today* 11.23-24 (Dec. 2006), pp. 1046–1053. DOI: 10.1016/j.drudis.2006.10.005.

[109] Peter Willett. "The Calculation of Molecular Structural Similarity: Principles and Practice." In: *Molecular Informatics* 33.6-7 (June 2014), pp. 403–413. DOI: 10.1002/minf.201400024.

[110] Li Xiangpeng et al. "Beyond RNNs: Positional Self-Attention with Co-Attention for Video Question Answering." In: *Proceedings of the ... AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 8658–8665. DOI: 10.1609/aaai.v33i01.33018658.

[111] Zhaoping Xiong et al. "Pushing the Boundaries of Molecular Representation for Drug Discovery with the Graph Attention Mechanism." In: *Journal of Medicinal Chemistry* 63.16 (Aug. 2020), pp. 8749–8760. DOI: 10.1021/acs.jmedchem.9b00959.

[112] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." In: vol. 3. July 2015, pp. 2048–2057.

[113] Yucheng Xu and Sheng Wang. "Drug Discovery and Artificial Intelligence." In: *Frontiers in Pharmacology* 12 (2021), p. 651120.

[114] Li-Ming Yang et al. "FusionNet: A Convolution–Transformer Fusion Network for Hyperspectral Image Classification." In: *Remote Sensing* 14.16 (Aug. 2022), p. 4066. DOI: 10.3390/rs14164066.

[115] Jun Yu et al. "Retrieval of Family Members Using Siamese Neural Network." In: Nov. 2020. DOI: 10.1109/fg47880.2020.00136.

[116] B Zagidullin et al. "Comparative analysis of molecular fingerprints in prediction of drug combination effects." In: *Briefings in Bioinformatics* 22.6 (Aug. 2021). bbab291. ISSN: 1477-4054. DOI: 10.1093/bib/bbab291. eprint: https://academic.oup.com/bib/article-pdf/22/6/bbab291/41974967/supplementary_data_bbab291.pdf.

[117] Ming Zhang et al. "Convolutional Neural Network With Attention Mechanism for SAR Automatic Target Recognition." In: *IEEE Geoscience and Remote Sensing Letters* 19 (Jan. 2022), pp. 1–5. DOI: 10.1109/lgrs.2020.3031593.

[118] Zhao Zhang, Feng Feng, and Tingting Huang. "FNNS: An Effective Feedforward Neural Network Scheme with Random Weights for Processing Large-Scale Datasets." In: *Applied sciences* 12.23 (Dec. 2022), p. 12478. DOI: 10.3390/app122312478.

[119] Shuangjia Zheng et al. "Identifying Structure–Property Relationships through SMILES Syntax Analysis with Self-Attention Mechanism." In: *Journal of Chemical Information and Modeling* 59.2 (Jan. 2019), pp. 914–923. DOI: 10.1021/acs.jcim.8b00803.

[120] Enes Zvornicanin and Enes Zvornicanin. “Differences Between Bidirectional and Unidirectional LSTM | Baeldung on Computer Science.” In: *Baeldung on Computer Science* (Nov. 2022).

Webographie

[6] Amygdala. *SMILES (Simplified Molecular Input Line Entry System)*. Jan. 2015. URL: https://lsy5518.wordpress.com/2014/11/10/1198/.

[14] Herman Bergwerf. *MolView*. URL: https://molview.org/.

[25] *Chapter 8 Attention and Self-Attention for NLP | Modern Approaches in Natural Language Processing*. Sept. 2020. URL: https://slds-lmu.github.io/seminar_nlp_ss20/attention-and-self-attention-for-nlp.html.

[26] Chembioinfo. *Revisiting Molecular Hashed Fingerprints*. Nov. 2011. URL: https://chembioinfo.wordpress.com/2011/10/30/revisiting-molecular-hashed-fingerprints/.

[41] *File:MaxpoolSample2.png - Computer Science Wiki*. URL: https://computersciencewiki.org/index.php/File:MaxpoolSample2.png.

[54] Zhichun Guo et al. *Graph-based Molecular Representation Learning*. 2022. arXiv: 2207.04869 [q-bio.QM].

[60] Thorir Mar Ingolfsson. *Insights into LSTM architecture*. Nov. 2021. URL: https://thorirmar.com/post/insight_into_lstm/.

[88] Sebastian Raschka. *Understanding and Coding the Self-Attention Mechanism of Large Language Models From Scratch*. Feb. 2023. URL: https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html.